

Spinnaker C++

2.4.0.143

Generated by Doxygen 1.8.13

Contents

1	Getting Started	1
2	Programmer's Guide	3
3	Benefits of Spinnaker	5
4	FlyCapture2 Feature Comparison with Spinnaker	7
5	Working with GenICam GenTL Devices	9
5.1	GenTL Overview	9
5.2	Installation	9
5.3	Troubleshooting	10
5.3.1	Enable FLIR GenTL Logging	10
5.3.2	USB3 Device Image Tearing	10
6	Software Licensing Information	11
7	Software Maintenance Policy	13
7.1	GenTL Overview	13
7.2	Platform Support Policy	13
7.2.1	Windows Support	13
7.2.2	Linux Desktop Support	13
7.2.3	Linux Embedded Support	13
7.2.4	MacOS Support	14
7.3	Versioning Policy	14

8	Module Index	15
8.1	Modules	15
9	Namespace Index	19
9.1	Namespace List	19
10	Hierarchical Index	21
10.1	Class Hierarchy	21
11	Class Index	27
11.1	Class List	27
12	File Index	35
12.1	File List	35
13	Module Documentation	41
13.1	Spinnaker Classes	42
13.1.1	Detailed Description	45
13.1.2	Enumeration Type Documentation	45
13.1.2.1	InferenceBoxType	45
13.2	AVI Recorder Class	46
13.2.1	Detailed Description	46
13.2.2	Function Documentation	46
13.2.2.1	DEPRECATED_CLASS()	46
13.3	BasePtr Class	49
13.3.1	Detailed Description	49
13.3.2	Function Documentation	49
13.3.2.1	operator==()	49
13.4	Camera Class	50
13.4.1	Detailed Description	50
13.5	Camera Base Class	51
13.5.1	Detailed Description	51
13.6	CameraDefs Class	52

13.6.1 Detailed Description	84
13.6.2 Enumeration Type Documentation	84
13.6.2.1 AcquisitionModeEnums	84
13.6.2.2 AcquisitionStatusSelectorEnums	84
13.6.2.3 ActionUnconditionalModeEnums	85
13.6.2.4 AdcBitDepthEnums	85
13.6.2.5 AutoAlgorithmSelectorEnums	86
13.6.2.6 AutoExposureControlPriorityEnums	86
13.6.2.7 AutoExposureLightingModeEnums	86
13.6.2.8 AutoExposureMeteringModeEnums	87
13.6.2.9 AutoExposureTargetGreyValueAutoEnums	87
13.6.2.10 BalanceRatioSelectorEnums	87
13.6.2.11 BalanceWhiteAutoEnums	88
13.6.2.12 BalanceWhiteAutoProfileEnums	88
13.6.2.13 BinningHorizontalModeEnums	88
13.6.2.14 BinningSelectorEnums	89
13.6.2.15 BinningVerticalModeEnums	89
13.6.2.16 BlackLevelAutoBalanceEnums	89
13.6.2.17 BlackLevelAutoEnums	90
13.6.2.18 BlackLevelSelectorEnums	90
13.6.2.19 ChunkBlackLevelSelectorEnums	90
13.6.2.20 ChunkCounterSelectorEnums	91
13.6.2.21 ChunkEncoderSelectorEnums	91
13.6.2.22 ChunkEncoderStatusEnums	91
13.6.2.23 ChunkExposureTimeSelectorEnums	92
13.6.2.24 ChunkGainSelectorEnums	92
13.6.2.25 ChunkImageComponentEnums	92
13.6.2.26 ChunkPixelFormatEnums	93
13.6.2.27 ChunkRegionIDEnums	93
13.6.2.28 ChunkScan3dCoordinateReferenceSelectorEnums	94

13.6.2.29 ChunkScan3dCoordinateSelectorEnums	94
13.6.2.30 ChunkScan3dCoordinateSystemEnums	94
13.6.2.31 ChunkScan3dCoordinateSystemReferenceEnums	95
13.6.2.32 ChunkScan3dCoordinateTransformSelectorEnums	95
13.6.2.33 ChunkScan3dDistanceUnitEnums	95
13.6.2.34 ChunkScan3dOutputModeEnums	96
13.6.2.35 ChunkSelectorEnums	97
13.6.2.36 ChunkSourceIDEnums	97
13.6.2.37 ChunkTimerSelectorEnums	97
13.6.2.38 ChunkTransferStreamIDEnums	98
13.6.2.39 CIConfigurationEnums	98
13.6.2.40 CITimeSlotsCountEnums	99
13.6.2.41 ColorTransformationSelectorEnums	99
13.6.2.42 ColorTransformationValueSelectorEnums	99
13.6.2.43 CompressionSaturationPriorityEnums	100
13.6.2.44 CounterEventActivationEnums	100
13.6.2.45 CounterEventSourceEnums	100
13.6.2.46 CounterResetActivationEnums	101
13.6.2.47 CounterResetSourceEnums	101
13.6.2.48 CounterSelectorEnums	102
13.6.2.49 CounterStatusEnums	102
13.6.2.50 CounterTriggerActivationEnums	103
13.6.2.51 CounterTriggerSourceEnums	103
13.6.2.52 CxpConnectionTestModeEnums	104
13.6.2.53 CxpLinkConfigurationEnums	104
13.6.2.54 CxpLinkConfigurationPreferredEnums	105
13.6.2.55 CxpLinkConfigurationStatusEnums	106
13.6.2.56 CxpPoCxpStatusEnums	107
13.6.2.57 DecimationHorizontalModeEnums	107
13.6.2.58 DecimationSelectorEnums	107

13.6.2.59 DecimationVerticalModeEnums	108
13.6.2.60 DefectCorrectionModeEnums	108
13.6.2.61 DeinterlacingEnums	108
13.6.2.62 DeviceCharacterSetEnums	109
13.6.2.63 DeviceClockSelectorEnums	109
13.6.2.64 DeviceConnectionStatusEnums	109
13.6.2.65 DeviceIndicatorModeEnums	110
13.6.2.66 DeviceLinkHeartbeatModeEnums	110
13.6.2.67 DeviceLinkThroughputLimitModeEnums	110
13.6.2.68 DevicePowerSupplySelectorEnums	111
13.6.2.69 DeviceRegistersEndiannessEnums	111
13.6.2.70 DeviceScanTypeEnums	111
13.6.2.71 DeviceSerialPortBaudRateEnums	111
13.6.2.72 DeviceSerialPortSelectorEnums	112
13.6.2.73 DeviceStreamChannelEndiannessEnums	112
13.6.2.74 DeviceStreamChannelTypeEnums	112
13.6.2.75 DeviceTapGeometryEnums	113
13.6.2.76 DeviceTemperatureSelectorEnums	114
13.6.2.77 DeviceTLTypeEnums	114
13.6.2.78 DeviceTypeEnums	115
13.6.2.79 EncoderModeEnums	115
13.6.2.80 EncoderOutputModeEnums	115
13.6.2.81 EncoderResetActivationEnums	116
13.6.2.82 EncoderResetSourceEnums	116
13.6.2.83 EncoderSelectorEnums	117
13.6.2.84 EncoderSourceAEnums	118
13.6.2.85 EncoderSourceBEnums	118
13.6.2.86 EncoderStatusEnums	118
13.6.2.87 EventNotificationEnums	119
13.6.2.88 EventSelectorEnums	119

13.6.2.89 ExposureActiveModeEnums	119
13.6.2.90 ExposureAutoEnums	120
13.6.2.91 ExposureModeEnums	120
13.6.2.92 ExposureTimeModeEnums	120
13.6.2.93 ExposureTimeSelectorEnums	121
13.6.2.94 FileOpenModeEnums	121
13.6.2.95 FileOperationSelectorEnums	121
13.6.2.96 FileOperationStatusEnums	122
13.6.2.97 FileSelectorEnums	122
13.6.2.98 GainAutoBalanceEnums	122
13.6.2.99 GainAutoEnums	124
13.6.2.100 GainSelectorEnums	124
13.6.2.101 GevCCPEnums	124
13.6.2.102 GevCurrentPhysicalLinkConfigurationEnums	125
13.6.2.103 GevGVCPExtendedStatusCodesSelectorEnums	125
13.6.2.104 GevGVSPExtendedIDModeEnums	125
13.6.2.105 GevIEEE1588ClockAccuracyEnums	126
13.6.2.106 GevIEEE1588ModeEnums	126
13.6.2.107 GevIEEE1588StatusEnums	126
13.6.2.108 GevIPConfigurationStatusEnums	127
13.6.2.109 GevPhysicalLinkConfigurationEnums	127
13.6.2.110 GevSupportedOptionSelectorEnums	128
13.6.2.111 ImageComponentSelectorEnums	128
13.6.2.112 ImageCompressionJPEGFormatOptionEnums	129
13.6.2.113 ImageCompressionModeEnums	129
13.6.2.114 ImageCompressionRateOptionEnums	130
13.6.2.115 LineFormatEnums	130
13.6.2.116 LineInputFilterSelectorEnums	130
13.6.2.117 LineModeEnums	131
13.6.2.118 LineSelectorEnums	131

13.6.2.119	LineSourceEnums	131
13.6.2.120	LogicBlockLUTInputActivationEnums	132
13.6.2.121	LogicBlockLUTInputSelectorEnums	132
13.6.2.122	LogicBlockLUTInputSourceEnums	133
13.6.2.123	LogicBlockLUTSelectorEnums	133
13.6.2.124	LogicBlockSelectorEnums	134
13.6.2.125	LUTSelectorEnums	134
13.6.2.126	PixelColorFilterEnums	134
13.6.2.127	PixelFormatEnums	135
13.6.2.128	PixelFormatInfoSelectorEnums	140
13.6.2.129	PixelSizeEnums	146
13.6.2.130	RegionDestinationEnums	147
13.6.2.131	RegionModeEnums	147
13.6.2.132	RegionSelectorEnums	147
13.6.2.133	RgbTransformLightSourceEnums	148
13.6.2.134	Scan3dCoordinateReferenceSelectorEnums	148
13.6.2.135	Scan3dCoordinateSelectorEnums	149
13.6.2.136	Scan3dCoordinateSystemEnums	149
13.6.2.137	Scan3dCoordinateSystemReferenceEnums	149
13.6.2.138	Scan3dCoordinateTransformSelectorEnums	150
13.6.2.139	Scan3dDistanceUnitEnums	150
13.6.2.140	Scan3dOutputModeEnums	150
13.6.2.141	SensorDigitizationTapsEnums	151
13.6.2.142	SensorShutterModeEnums	152
13.6.2.143	SensorTapsEnums	152
13.6.2.144	SequencerConfigurationModeEnums	152
13.6.2.145	SequencerConfigurationValidEnums	153
13.6.2.146	SequencerModeEnums	153
13.6.2.147	SequencerSetValidEnums	153
13.6.2.148	SequencerTriggerActivationEnums	153

13.6.2.149SequencerTriggerSourceEnums	154
13.6.2.150SerialPortBaudRateEnums	154
13.6.2.151SerialPortParityEnums	155
13.6.2.152SerialPortSelectorEnums	155
13.6.2.153SerialPortSourceEnums	155
13.6.2.154SerialPortStopBitsEnums	156
13.6.2.155SoftwareSignalSelectorEnums	156
13.6.2.156SourceSelectorEnums	156
13.6.2.157TestPatternEnums	157
13.6.2.158TestPatternGeneratorSelectorEnums	157
13.6.2.159TimerSelectorEnums	157
13.6.2.160TimerStatusEnums	158
13.6.2.161TimerTriggerActivationEnums	158
13.6.2.162TimerTriggerSourceEnums	158
13.6.2.163TransferComponentSelectorEnums	160
13.6.2.164TransferControlModeEnums	160
13.6.2.165TransferOperationModeEnums	160
13.6.2.166TransferQueueModeEnums	161
13.6.2.167TransferSelectorEnums	161
13.6.2.168TransferStatusSelectorEnums	161
13.6.2.169TransferTriggerActivationEnums	162
13.6.2.170TransferTriggerModeEnums	162
13.6.2.171TransferTriggerSelectorEnums	163
13.6.2.172TransferTriggerSourceEnums	163
13.6.2.173TriggerActivationEnums	164
13.6.2.174TriggerModeEnums	164
13.6.2.175TriggerOverlapEnums	165
13.6.2.176TriggerSelectorEnums	165
13.6.2.177TriggerSourceEnums	165
13.6.2.178UserOutputSelectorEnums	166

13.6.2.179	UserSetDefaultEnums	166
13.6.2.180	UserSetSelectorEnums	167
13.6.2.181	WhiteClipSelectorEnums	167
13.7	Camera List Class	168
13.7.1	Detailed Description	168
13.8	CameraPtr Class	169
13.8.1	Detailed Description	169
13.8.2	Function Documentation	169
13.8.2.1	CameraPtr() [1/4]	169
13.8.2.2	CameraPtr() [2/4]	169
13.8.2.3	CameraPtr() [3/4]	170
13.8.2.4	CameraPtr() [4/4]	170
13.9	ChunkData Class	171
13.9.1	Detailed Description	171
13.10	Chunk Data Inference Class	172
13.10.1	Detailed Description	173
13.10.2	Function Documentation	173
13.10.2.1	GetBoxAt()	173
13.10.2.2	GetBoxCount()	173
13.10.2.3	GetBoxSize()	173
13.10.2.4	GetVersion()	174
13.10.2.5	InferenceBoundingBoxResult() [1/3]	174
13.10.2.6	InferenceBoundingBoxResult() [2/3]	174
13.10.2.7	InferenceBoundingBoxResult() [3/3]	174
13.10.2.8	operator=()	174
13.10.2.9	~InferenceBoundingBoxResult()	175
13.10.3	Variable Documentation	175
13.10.3.1	bottomRightXCoord [1/2]	175
13.10.3.2	bottomRightXCoord [2/2]	175
13.10.3.3	bottomRightYCoord [1/2]	175

13.10.3.4 bottomRightYCoord [2/2]	175
13.10.3.5 boxType	175
13.10.3.6 centerXCoord	175
13.10.3.7 centerYCoord	176
13.10.3.8 circle	176
13.10.3.9 classId	176
13.10.3.10 confidence	176
13.10.3.11 radius	176
13.10.3.12 rect	176
13.10.3.13 rotatedRect	176
13.10.3.14 rotationAngle	176
13.10.3.15 topLeftXCoord [1/2]	177
13.10.3.16 topLeftXCoord [2/2]	177
13.10.3.17 topLeftYCoord [1/2]	177
13.10.3.18 topLeftYCoord [2/2]	177
13.11 Spinnaker EventHandler Classes	178
13.11.1 Detailed Description	179
13.12 DeviceArrivalEventHandler Class	180
13.12.1 Detailed Description	180
13.13 DeviceEventHandler Class	181
13.13.1 Detailed Description	181
13.14 DeviceRemovalEventHandler Class	182
13.14.1 Detailed Description	182
13.15 EventHandler Class	183
13.15.1 Detailed Description	183
13.16 Exception Class	184
13.16.1 Detailed Description	184
13.17 Image Class	185
13.17.1 Detailed Description	185
13.18 ImageEventHandler Class	186

13.18.1 Detailed Description	186
13.19ImagePtr Class	187
13.19.1 Detailed Description	187
13.20ImageStatistics Class	188
13.20.1 Detailed Description	188
13.21Image Utility Class	189
13.21.1 Detailed Description	189
13.22Image Utility CCM Class	190
13.22.1 Detailed Description	190
13.23Image Utility Heatmap Class	191
13.23.1 Detailed Description	191
13.24Image Utility Polarization Class	192
13.24.1 Detailed Description	192
13.25Interface Class	193
13.25.1 Detailed Description	193
13.26InterfaceArrivalEventHandler Class	194
13.26.1 Detailed Description	194
13.27InterfaceEventHandler Class	195
13.27.1 Detailed Description	195
13.28InterfaceList Class	196
13.28.1 Detailed Description	196
13.29InterfacePtr Class	197
13.29.1 Detailed Description	197
13.30InterfaceRemovalEventHandler Class	198
13.30.1 Detailed Description	198
13.31Logging EventHandler Class	199
13.31.1 Detailed Description	199
13.32LoggingEventDataPtr Class	200
13.32.1 Detailed Description	200
13.33LoggingEventHandler Class	201

13.33.1 Detailed Description	201
13.34 Spinnaker Headers	202
13.34.1 Detailed Description	205
13.34.2 Function Documentation	205
13.34.2.1 BooleanGetValue()	205
13.34.2.2 BooleanSetValue()	205
13.34.2.3 CommandExecute()	206
13.34.2.4 EnumerationGetEntry()	206
13.34.2.5 EnumerationGetNumEntries()	206
13.34.2.6 EnumerationSetEntry()	207
13.34.2.7 FloatGetInc()	207
13.34.2.8 FloatGetIncMode()	208
13.34.2.9 FloatGetMax()	208
13.34.2.10 FloatGetMin()	208
13.34.2.11 FloatGetValue()	209
13.34.2.12 FloatSetValue()	209
13.34.2.13 IntegerGetInc()	209
13.34.2.14 IntegerGetIncMode()	210
13.34.2.15 IntegerGetMax()	210
13.34.2.16 IntegerGetMin()	211
13.34.2.17 IntegerGetValue()	211
13.34.2.18 IntegerSetValue()	211
13.34.2.19 NodeGetDisplayName()	212
13.34.2.20 NodeGetType()	212
13.34.2.21 NodeIsAvailable()	213
13.34.2.22 NodeIsImplemented()	213
13.34.2.23 NodeIsReadable()	213
13.34.2.24 NodeIsWritable()	214
13.34.2.25 NodeMapGetNodeAtIndex()	214
13.34.2.26 NodeMapGetNumNodes()	214

13.34.2.27	NodeToString()	216
13.34.2.28	StringGetValue()	216
13.34.2.29	StringSetValue()	217
13.34.3	Variable Documentation	217
13.34.3.1	bufferSize	217
13.34.3.2	entryExist	217
13.34.3.3	entryIndex	217
13.34.3.4	enumerationEntryName	218
13.34.3.5	enumerationName	218
13.34.3.6	EVENT_TIMEOUT_INFINITE	218
13.34.3.7	EVENT_TIMEOUT_NONE	218
13.35	Spinnaker.h	219
13.36	Spinnaker Definitions	220
13.36.1	Detailed Description	224
13.36.2	Enumeration Type Documentation	224
13.36.2.1	ActionCommandStatus	224
13.36.2.2	BufferOwnership	224
13.36.2.3	CCMColorTemperature	224
13.36.2.4	CCMSensor	225
13.36.2.5	CCMType	225
13.36.2.6	ColorProcessingAlgorithm	225
13.36.2.7	Error	226
13.36.2.8	EventType	227
13.36.2.9	ImageFileFormat	227
13.36.2.10	ImageStatus	228
13.36.2.11	PayloadTypeInfoIDs	229
13.36.2.12	PixelFormatIntType	229
13.36.2.13	PixelFormatNamespaceID	230
13.36.2.14	SpinnakerLogLevel	230
13.36.2.15	StatisticsChannel	231

13.37SpinnakerDirectShow.h	232
13.37.1 Detailed Description	232
13.37.2 Function Documentation	232
13.37.2.1 BeginAcquisition()	233
13.37.2.2 EndAcquisition()	233
13.37.2.3 GetCameraInfo()	233
13.37.2.4 GetSelectedCameraIndex()	234
13.37.2.5 IsStreaming()	234
13.37.2.6 SetSelectedCameraIndex()	234
13.37.3 Variable Documentation	235
13.37.3.1 IID_ISpinnakerInterface	235
13.38Spinnaker Platform	236
13.38.1 Detailed Description	236
13.38.2 Macro Definition Documentation	236
13.38.2.1 SPINNAKER_API	236
13.38.2.2 SPINNAKER_API_ABSTRACT	236
13.38.2.3 SPINNAKER_LOCAL	236
13.39Spinnaker Video Class	237
13.39.1 Detailed Description	237
13.40Spinnaker Video Definitions	238
13.41System Class	239
13.41.1 Detailed Description	239
13.42SystemEventHandler Class	240
13.42.1 Detailed Description	240
13.43SystemPtr Class	241
13.43.1 Detailed Description	241
13.44Spinnaker QuickSpin Classes	242
13.44.1 Detailed Description	242
13.45TransportLayerDefs Class	243
13.45.1 Detailed Description	244

13.45.2 Enumeration Type Documentation	244
13.45.2.1 DeviceAccessStatusEnum	245
13.45.2.2 DeviceCurrentSpeedEnum	245
13.45.2.3 DeviceEndianessMechanismEnum	245
13.45.2.4 DeviceTypeEnum	246
13.45.2.5 FilterDriverStatusEnum	246
13.45.2.6 GenICamXMLLocationEnum	246
13.45.2.7 GevCCPEnum	247
13.45.2.8 GUIXMLLocationEnum	247
13.45.2.9 InterfaceTypeEnum	247
13.45.2.10 POEStatusEnum	248
13.45.2.11 StreamBufferCountModeEnum	248
13.45.2.12 StreamBufferHandlingModeEnum	248
13.45.2.13 StreamTypeEnum	249
13.45.2.14 TLTypeEnum	249
13.46 TransportLayerDevice Class	251
13.46.1 Detailed Description	251
13.47 TransportLayerInterface Class	252
13.47.1 Detailed Description	252
13.48 TransportLayerStream Class	253
13.48.1 Detailed Description	253
13.49 TransportLayerSystem Class	254
13.49.1 Detailed Description	254
13.50 Camera Base Interface Class	255
13.50.1 Detailed Description	255
13.51 IChunkData Class	256
13.51.1 Detailed Description	256
13.52 IImage Class	257
13.52.1 Detailed Description	257
13.53 IImageStatistics Class	258

13.53.1 Detailed Description	258
13.54 Interface Class	259
13.54.1 Detailed Description	259
13.55 InterfaceList Class	260
13.55.1 Detailed Description	260
13.56 ISystem Class	261
13.56.1 Detailed Description	261
13.57 Spinnaker GenApi Classes	262
13.57.1 Detailed Description	268
13.57.2 Typedef Documentation	268
13.57.2.1 CNodeMapRef	268
13.57.2.2 CNodeRef	268
13.57.2.3 CSelectorRef	268
13.57.3 Function Documentation	268
13.57.3.1 _ClearXMLCache()	268
13.57.3.2 _Connect() [1/2]	269
13.57.3.3 _Connect() [2/2]	269
13.57.3.4 _Destroy()	269
13.57.3.5 _GetDeviceName()	269
13.57.3.6 _GetNode()	269
13.57.3.7 _GetNodes()	269
13.57.3.8 _GetSupportedSchemaVersions()	269
13.57.3.9 _InvalidateNodes()	270
13.57.3.10 _LoadXMLFromFile()	270
13.57.3.11 _LoadXMLFromFileInject()	270
13.57.3.12 _LoadXMLFromString()	270
13.57.3.13 _LoadXMLFromStringInject()	270
13.57.3.14 _LoadXMLFromZIPData()	270
13.57.3.15 _LoadXMLFromZIPFile()	270
13.57.3.16 _Poll()	271

13.57.3.17	<code>CastToIDestroy()</code>	271
13.57.3.18	<code>CNodeMapRefT()</code> [1/3]	271
13.57.3.19	<code>CNodeMapRefT()</code> [2/3]	271
13.57.3.20	<code>CNodeMapRefT()</code> [3/3]	271
13.57.3.21	<code>EatComments()</code>	271
13.57.3.22	<code>operator<<()</code>	272
13.57.3.23	<code>operator=()</code> [1/2]	272
13.57.3.24	<code>operator=()</code> [2/2]	272
13.57.3.25	<code>operator>>()</code>	272
13.57.3.26	<code>~CNodeMapRefT()</code>	272
13.58	<code>AutoVector</code> Class	273
13.58.1	Detailed Description	273
13.59	<code>Spinnaker GenApi</code> Interfaces	274
13.59.1	Detailed Description	275
13.59.2	Typedef Documentation	275
13.59.2.1	<code>CallbackHandleType</code>	275
13.59.2.2	<code>NodeList_t</code>	275
13.60	<code>IBase</code> Interface	276
13.60.1	Detailed Description	276
13.60.2	Variable Documentation	276
13.60.2.1	<code>IBase</code>	276
13.61	<code>BooleanNode</code> Class	277
13.61.1	Detailed Description	277
13.61.2	Typedef Documentation	277
13.61.2.1	<code>CBooleanRef</code>	277
13.62	<code>CategoryNode</code> Class	278
13.62.1	Detailed Description	278
13.62.2	Typedef Documentation	278
13.62.2.1	<code>CCategoryRef</code>	278
13.63	<code>ChunkAdapter</code> Class	279

13.63.1 Detailed Description	279
13.64ChunkAdapterDcam Class	280
13.64.1 Detailed Description	280
13.65ChunkAdapterGeneric Class	281
13.65.1 Detailed Description	281
13.66ChunkAdapterGEV Class	282
13.66.1 Detailed Description	282
13.67ChunkPort Class	283
13.67.1 Detailed Description	283
13.68CommandNode Class	284
13.68.1 Detailed Description	284
13.68.2 Typedef Documentation	284
13.68.2.1 CCommandRef	284
13.69Container Class	285
13.70Counter Class	286
13.70.1 Detailed Description	286
13.71EnumClasses Class	287
13.71.1 Detailed Description	288
13.72EnumEntryNode Class	289
13.72.1 Detailed Description	289
13.72.2 Typedef Documentation	289
13.72.2.1 CEnumEntryRef	289
13.73EnumNode Class	290
13.73.1 Detailed Description	290
13.73.2 Typedef Documentation	290
13.73.2.1 CEnumerationRef	290
13.74EnumNodeT Class	291
13.74.1 Detailed Description	291
13.75EventAdapter Class	292
13.75.1 Detailed Description	292

13.76EventAdapter1394 Class	293
13.76.1 Detailed Description	293
13.77EventAdapterGeneric Class	294
13.77.1 Detailed Description	294
13.78EventAdapterGEV Class	295
13.78.1 Detailed Description	295
13.79EventAdapterU3V Class	296
13.79.1 Detailed Description	296
13.80EventPort Class	297
13.80.1 Detailed Description	297
13.81Filestream Class	298
13.81.1 Detailed Description	298
13.82FloatNode Class	299
13.82.1 Detailed Description	299
13.82.2 Typedef Documentation	299
13.82.2.1 CFloatRef	299
13.83FloatRegNode Class	300
13.83.1 Detailed Description	300
13.84GCString Class	301
13.84.1 Detailed Description	301
13.85GCSynch Class	302
13.85.1 Detailed Description	302
13.86GCTypes Class	303
13.86.1 Detailed Description	303
13.86.2 Typedef Documentation	303
13.86.2.1 float32_t	303
13.86.2.2 float64_t	303
13.87Spinnaker GenApi Utilities	304
13.87.1 Detailed Description	304
13.88GCUtilities Utility	305

13.88.1 Detailed Description	306
13.88.2 Function Documentation	306
13.88.2.1 DoesEnvironmentVariableExist()	306
13.88.2.2 GetFiles()	306
13.88.2.3 GetGenIcamCacheFolder()	306
13.88.2.4 GetGenIcamCLProtocolFolder()	307
13.88.2.5 GetGenIcamLogConfig()	307
13.88.2.6 GetModulePathFromFunction()	307
13.88.2.7 GetValueOfEnvironmentVariable() [1/2]	307
13.88.2.8 GetValueOfEnvironmentVariable() [2/2]	308
13.88.2.9 INTEGRAL_CAST()	308
13.88.2.10INTEGRAL_CAST2()	308
13.88.2.11ReplaceEnvironmentVariables()	308
13.88.2.12SetGenIcamCacheFolder()	309
13.88.2.13SetGenIcamCLProtocolFolder()	309
13.88.2.14SetGenIcamLogConfig()	309
13.88.2.15Tokenize()	309
13.88.2.16UrlDecode()	309
13.88.2.17UrlEncode()	310
13.89IBoolean Interface	311
13.89.1 Detailed Description	311
13.89.2 Function Documentation	311
13.89.2.1 GetValue()	311
13.89.2.2 operator()()	312
13.89.2.3 operator=()	312
13.89.3 Variable Documentation	312
13.89.3.1 IBoolean	312
13.89.3.2 Verify	312
13.90ICategory Interfaces	313
13.90.1 Detailed Description	313

13.90.2 Variable Documentation	313
13.90.2.1 ICategory	313
13.91 IChunkPort Interface	314
13.91.1 Detailed Description	314
13.91.2 Macro Definition Documentation	314
13.91.2.1 CHUNK_BASE_ADDRESS_REGISTER	314
13.91.2.2 CHUNK_BASE_ADDRESS_REGISTER_LEN	315
13.91.2.3 CHUNK_LENGTH_REGISTER	315
13.91.2.4 CHUNK_LENGTH_REGISTER_LEN	315
13.91.3 Function Documentation	315
13.91.3.1 CacheChunkData()	315
13.91.4 Variable Documentation	315
13.91.4.1 IChunkPort	315
13.92 ICommand Interface	316
13.92.1 Detailed Description	316
13.92.2 Function Documentation	316
13.92.2.1 IsDone()	316
13.92.3 Variable Documentation	316
13.92.3.1 ICommand	317
13.93 IDestroy Interface	318
13.93.1 Detailed Description	318
13.93.2 Variable Documentation	318
13.93.2.1 IDestroy	318
13.94 IDeviceInfo Interface	319
13.94.1 Detailed Description	319
13.94.2 Function Documentation	319
13.94.2.1 GetDeviceVersion()	319
13.94.2.2 GetGenApiVersion()	320
13.94.2.3 GetProductGuid()	320
13.94.2.4 GetSchemaVersion()	320

13.94.2.5 GetStandardNameSpace()	320
13.94.2.6 GetToolTip()	320
13.94.2.7 GetVendorName()	320
13.94.2.8 GetVersionGuid()	321
13.94.3 Variable Documentation	321
13.94.3.1 IDeviceInfo	321
13.95IEnumEntry Interface	322
13.95.1 Detailed Description	322
13.95.2 Function Documentation	322
13.95.2.1 GetNumericValue()	322
13.95.2.2 GetSymbolic()	322
13.95.2.3 IsSelfClearing()	323
13.95.3 Variable Documentation	323
13.95.3.1 IEnumEntry	323
13.96IEnumeration Interface	324
13.96.1 Detailed Description	324
13.96.2 Function Documentation	324
13.96.2.1 GetCurrentEntry()	324
13.96.2.2 GetEntries()	325
13.96.2.3 GetEntry()	325
13.96.2.4 GetEntryByName()	325
13.96.2.5 GetIntValue()	325
13.96.2.6 operator*()	325
13.96.2.7 SetIntValue()	326
13.96.3 Variable Documentation	326
13.96.3.1 IEnumeration	326
13.97IEnumerationT Interface	327
13.97.1 Detailed Description	327
13.97.2 Function Documentation	327
13.97.2.1 GetEntry()	327

13.97.2.2 operator=() [1/2]	328
13.97.2.3 operator=() [2/2]	328
13.97.3 Variable Documentation	328
13.97.3.1 IEnumerationT	328
13.97.3.2 IEnumReference	328
13.98 IFloat Interface	329
13.98.1 Detailed Description	330
13.98.2 Function Documentation	330
13.98.2.1 GetDisplayNotation()	330
13.98.2.2 GetDisplayPrecision()	330
13.98.2.3 GetInc()	330
13.98.2.4 GetIncMode()	330
13.98.2.5 GetListOfValidValues()	330
13.98.2.6 GetMax()	331
13.98.2.7 GetMin()	331
13.98.2.8 GetRepresentation()	331
13.98.2.9 GetUnit()	331
13.98.2.10 HasInc()	331
13.98.2.11 ImposeMax()	331
13.98.2.12 ImposeMin()	332
13.98.2.13 operator=()	332
13.98.3 Variable Documentation	332
13.98.3.1 IFloat	332
13.99 Integer Interface	333
13.99.1 Detailed Description	333
13.99.2 Function Documentation	333
13.99.2.1 ImposeMax()	333
13.99.2.2 ImposeMin()	333
13.99.2.3 operator=()	334
13.99.3 Variable Documentation	334

13.99.3.1 Integer	334
13.100 Node Interface	335
13.100.1 Detailed Description	337
13.100.2 Function Documentation	337
13.100.2.1 Combine() [1/3]	337
13.100.2.2 Combine() [2/3]	337
13.100.2.3 Combine() [3/3]	337
13.100.2.4 DeregisterCallback()	338
13.100.2.5 GetAlias()	338
13.100.2.6 GetCachingMode()	338
13.100.2.7 GetCastAlias()	338
13.100.2.8 GetChildren()	338
13.100.2.9 GetDescription()	339
13.100.2.10 GetDisplayName()	339
13.100.2.11 GetDocuURL()	339
13.100.2.12 GetEventID()	339
13.100.2.13 GetNameSpace()	339
13.100.2.14 GetNodeMap()	339
13.100.2.15 GetParents()	339
13.100.2.16 GetPollingTime()	340
13.100.2.17 GetPrincipalInterfaceType()	340
13.100.2.18 GetProperty()	340
13.100.2.19 GetPropertyNames()	340
13.100.2.20 GetVisibility()	340
13.100.2.21 ImposeAccessMode()	341
13.100.2.22 ImposeVisibility()	341
13.100.2.23 InvalidateNode()	341
13.100.2.24 IsAccessModeCacheable()	341
13.100.2.25 Available() [1/3]	341
13.100.2.26 Available() [2/3]	341

13.100.2.27Available() [3/3]	342
13.100.2.28Cachable()	342
13.100.2.29Cacheable()	342
13.100.2.30Deprecated()	342
13.100.2.31Feature()	342
13.100.2.32Implemented() [1/3]	342
13.100.2.33Implemented() [2/3]	343
13.100.2.34Implemented() [3/3]	343
13.100.2.35Readable() [1/3]	343
13.100.2.36Readable() [2/3]	343
13.100.2.37Readable() [3/3]	343
13.100.2.38Streamable()	343
13.100.2.39Visible()	344
13.100.2.40Writable() [1/3]	344
13.100.2.41Writable() [2/3]	344
13.100.2.42Writable() [3/3]	344
13.100.2.43operator!=(())	344
13.100.2.44operator==(())	344
13.100.2.45RegisterCallback()	345
13.100.3/variable Documentation	345
13.100.3.1Node	345
13.100.3.2Reference	345
13.101NodeMap Interface	346
13.101.1Detailed Description	346
13.101.2Function Documentation	346
13.101.2.1Connect() [1/2]	347
13.101.2.2Connect() [2/2]	347
13.101.2.3GetDeviceName()	347
13.101.2.4GetLock()	347
13.101.2.5GetNode()	347

13.101.2.6	GetNumNodes()	348
13.101.2.7	InvalidateNodes()	348
13.101.2.8	Poll()	348
13.101.3	Variable Documentation	348
13.101.3.1	NodeMap	348
13.102	NodeMapDyn Interface	349
13.102.1	Detailed Description	350
13.102.2	Function Documentation	350
13.102.2.1	ExtractIndependentSubtree()	350
13.102.2.2	GetSupportedSchemaVersions()	350
13.102.2.3	LoadXMLFromFile()	350
13.102.2.4	LoadXMLFromFileInject()	351
13.102.2.5	LoadXMLFromString()	351
13.102.2.6	LoadXMLFromStringInject()	351
13.102.2.7	LoadXMLFromZIPData()	351
13.102.2.8	LoadXMLFromZIPFile()	351
13.102.2.9	MergeXMLFiles()	351
13.102.2.10	PreprocessXMLFromFile()	352
13.102.2.11	PreprocessXMLFromZIPFile()	352
13.102.3	Variable Documentation	353
13.102.3.1	NodeMapDyn	353
13.103	IntegerNode Class	354
13.103.1	Detailed Description	354
13.103.2	Typedef Documentation	354
13.103.2.1	IntegerRef	354
13.104	IntRegNode Class	355
13.104.1	Detailed Description	355
13.105	Port Interface	356
13.105.1	Detailed Description	356
13.105.2	Function Documentation	356

13.105.2.1Write()	356
13.105.3/variable Documentation	356
13.105.3.1Address	357
13.105.3.2Port	357
13.105.3.3Length	357
13.106PortConstruct Interface	358
13.106.1Detailed Description	358
13.106.2Function Documentation	358
13.106.2.1GetSwapEndianness()	358
13.106.3/variable Documentation	358
13.106.3.1PortConstruct	358
13.107PortRecorder Interface	359
13.107.1Detailed Description	359
13.107.2Function Documentation	359
13.107.2.1GetCookie()	359
13.107.2.2Replay()	360
13.107.2.3SetCookie()	360
13.107.2.4StopRecording()	360
13.107.3/variable Documentation	360
13.107.3.1Invalidate	360
13.107.3.2PortRecorder	360
13.107.3.3PortReplay	360
13.107.3.4PortWriteList	360
13.108Register Interfaces	361
13.108.1Detailed Description	361
13.108.2Function Documentation	361
13.108.2.1Get()	361
13.108.2.2GetAddress()	362
13.108.2.3GetLength()	362
13.108.3/variable Documentation	362

13.108.3.1IRegister	362
13.109ISelector Interface	363
13.109.1Detailed Description	363
13.109.2Function Documentation	363
13.109.2.1GetSelectedFeatures()	363
13.109.2.2GetSelectingFeatures()	363
13.109.3Variable Documentation	363
13.109.3.1ISelector	363
13.110ISelectorDigit Interface	364
13.110.1Detailed Description	364
13.110.2Function Documentation	364
13.110.2.1GetSelectorList()	364
13.110.2.2Restore()	365
13.110.2.3SetNext()	365
13.110.2.4ToString()	365
13.110.3Variable Documentation	365
13.110.3.1ISelectorDigit	366
13.111IString Class	367
13.111.1Detailed Description	367
13.111.2Function Documentation	367
13.111.2.1GetMaxLength()	367
13.111.3Variable Documentation	367
13.111.3.1IString	367
13.112IValue Class	368
13.112.1Detailed Description	368
13.112.2Function Documentation	368
13.112.2.1FromString()	368
13.112.2.2sValueCacheValid()	369
13.112.2.3ToString()	369
13.112.3Variable Documentation	369

13.112.3.1Value	369
13.113Node Class	370
13.113.1Detailed Description	370
13.114NodeCallback Class	371
13.114.1Detailed Description	372
13.114.2Enumeration Type Documentation	372
13.114.2.1ECallbackType	372
13.114.3Function Documentation	372
13.114.3.1Deregister()	372
13.114.3.2make_NodeCallback() [1/2]	372
13.114.3.3make_NodeCallback() [2/2]	373
13.114.3.4Register() [1/2]	373
13.114.3.5Register() [2/2]	373
13.115NodeMap Class	374
13.115.1Detailed Description	374
13.116NodeMapFactory Class	375
13.116.1Detailed Description	375
13.116.2Enumeration Type Documentation	375
13.116.2.1ECacheUsage_t	375
13.116.2.2EContentType_t	376
13.117NodeMapRef Class	377
13.117.1Detailed Description	377
13.118Persistence Class	378
13.118.1Detailed Description	378
13.119Pointer Class	379
13.119.1Detailed Description	380
13.119.2Typedef Documentation	380
13.119.2.1CBasePtr	380
13.119.2.2CBooleanPtr	381
13.119.2.3CCategoryPtr	381

13.119.2.4CChunkPortPtr	381
13.119.2.5CCommandPtr	381
13.119.2.6CDeviceInfoPtr	381
13.119.2.7CEnumEntryPtr	381
13.119.2.8CEnumerationPtr	382
13.119.2.9CIntegerPtr	382
13.119.2.10CNodeMapDynPtr	382
13.119.2.11CNodeMapPtr	382
13.119.2.12CNodePtr	382
13.119.2.13CPortConstructPtr	382
13.119.2.14CPortPtr	383
13.119.2.15CPortRecorderPtr	383
13.119.2.16CPortReplayPtr	383
13.119.2.17CPortWriteListPtr	383
13.119.2.18CRegisterPtr	383
13.119.2.19CSelectorPtr	383
13.119.2.20CStringPtr	384
13.119.2.21CValuePtr	384
13.119.3Function Documentation	384
13.119.3.1GetInterfaceName()	384
13.119.3.2IsAvailable()	384
13.119.3.3IsImplemented()	384
13.119.3.4IsReadable()	384
13.119.3.5IsWritable()	384
13.120PortImpl Class	385
13.120.1Detailed Description	385
13.121PortNode Class	386
13.121.1Detailed Description	386
13.121.2Typedef Documentation	386
13.121.2.1CPortRef	386

13.122	PortRecorder Class	387
13.122.1	Detailed Description	387
13.122.2	Typedef Documentation	387
13.122.2.1	CPortRecorderRef	387
13.123	PortReplay Class	388
13.123.1	Detailed Description	388
13.124	PortWriteList Class	389
13.124.1	Detailed Description	389
13.125	Reference Interfaces	390
13.125.1	Detailed Description	390
13.125.2	Function Documentation	390
13.125.2.1	SetNumEnums()	390
13.126	RegisterNode Class	391
13.126.1	Detailed Description	391
13.126.2	Typedef Documentation	391
13.126.2.1	CRegisterRef	391
13.127	RegisterPortImpl Class	392
13.127.1	Detailed Description	392
13.128	SelectorSet Class	393
13.128.1	Detailed Description	393
13.129	SpinTestCamera Class	394
13.129.1	Detailed Description	394
13.130	StringNode Class	395
13.130.1	Detailed Description	395
13.130.2	Typedef Documentation	395
13.130.2.1	CStringRef	395
13.131	StringRegNode Class	396
13.131.1	Detailed Description	396
13.132	StructPort Class	397
13.132.1	Detailed Description	397

13.13	Synch Class	398
13.133	Detailed Description	398
13.13	Spinnaker GenApi Enums	399
13.134	Detailed Description	399
13.135	Types Enums	400
13.135.1	Detailed Description	402
13.135.2	Macro Definition Documentation	402
13.135.2.1	UndefinedRepresentation	402
13.135.3	Typedef Documentation	402
13.135.3.1	StringList_t	403
13.135.4	Enumeration Type Documentation	403
13.135.4.1	EAccessMode	403
13.135.4.2	ECachingMode	403
13.135.4.3	EDisplayNotation	403
13.135.4.4	EEndianess	405
13.135.4.5	GenApiSchemaVersion	405
13.135.4.6	EIncMode	405
13.135.4.7	EInputDirection	406
13.135.4.8	EInterfaceType	406
13.135.4.9	ELinkType	406
13.135.4.10	ENamespace	407
13.135.4.11	ERepresentation	407
13.135.4.12	ESign	407
13.135.4.13	ESlope	408
13.135.4.14	EStandardNameSpace	408
13.135.4.15	EVisibility	408
13.135.4.16	XMLValidation	409
13.135.4.17	EYesNo	409
13.136	ValueNode Class	410
13.136.1	Detailed Description	410
13.136.2	Typedef Documentation	410
13.136.2.1	CValueRef	410
13.137	ChunkAdapterU3V Class	411
13.137.1	Detailed Description	411

14 Namespace Documentation	413
14.1 AdapterConfig Namespace Reference	413
14.1.1 Enumeration Type Documentation	414
14.1.1.1 AdapterConfigErr	414
14.1.2 Function Documentation	414
14.1.2.1 AutoPopulateAdapterInfo()	414
14.1.2.2 AutoPopulateAdvancedProperties()	414
14.1.2.3 ConfigureAdapter()	415
14.1.2.4 GetAuto10GDesc()	415
14.1.2.5 GetAutoGigabitDesc()	415
14.1.2.6 GetAutoStartIp()	415
14.1.2.7 GetAutoSubnetMask()	415
14.1.2.8 GetAutoSubnetMaskLength()	415
14.1.2.9 GetConfigLogFileName()	415
14.1.2.10 GetEnumerationLogFileName()	416
14.1.2.11 GetMaxIpAddress()	416
14.1.2.12 GetMinIpAddress()	416
14.1.2.13 GetSubnetMaskLength()	416
14.1.2.14 IsOnSameSubnet()	416
14.1.2.15 IsValidIpAddress()	416
14.1.2.16 IsValidSubnetMask()	416
14.1.2.17 PopulateAdapterIpInfo()	417
14.1.2.18 RetrieveAllAdapters()	417
14.1.2.19 ValidateIpAddress()	417
14.2 Conversion Namespace Reference	417
14.2.1 Function Documentation	417
14.2.1.1 NumToCString() [1/3]	417
14.2.1.2 NumToCString() [2/3]	417
14.2.1.3 NumToCString() [3/3]	418
14.3 CpuUtil Namespace Reference	418

14.3.1	Function Documentation	418
14.3.1.1	GetCpuStats()	418
14.3.1.2	StartCpuTracing()	418
14.3.1.3	StopCpuTracing()	418
14.4	PerformanceCounter Namespace Reference	418
14.4.1	Function Documentation	419
14.4.1.1	GetPerformanceCounter()	419
14.4.1.2	StartPerformanceCounter()	419
14.4.2	Variable Documentation	419
14.4.2.1	CounterStart	419
14.4.2.2	PCFreq	419
14.5	SecondsCounter Namespace Reference	419
14.5.1	Function Documentation	420
14.5.1.1	GetSecondsCounter()	420
14.5.1.2	StartSecondsCounter()	420
14.5.2	Variable Documentation	420
14.5.2.1	endTime	420
14.5.2.2	startTime	420
14.5.2.3	timeDiff	420
14.6	Spinnaker Namespace Reference	421
14.7	Spinnaker::GenApi Namespace Reference	460
14.7.1	Typedef Documentation	475
14.7.1.1	IDevFileStream	475
14.7.1.2	ODevFileStream	475
14.7.2	Enumeration Type Documentation	475
14.7.2.1	GVCP_MESSAGE_TAGS	475
14.7.3	Function Documentation	476
14.7.3.1	PersistFeature()	476
14.7.3.2	SET_GUID()	476
14.7.4	Variable Documentation	476
14.7.4.1	COMMAND_MAGIC	476
14.7.4.2	GENCP_COMMAND_HEADER_SIZE	476
14.7.4.3	GENCP_EVENT_BASIC_SIZE	477
14.7.4.4	GENCP_EVENT_CMD_ID	477
14.7.4.5	IPersistScript	477
14.7.4.6	U3V_EVENT_PREFIX	477
14.8	Spinnaker::GenICam Namespace Reference	477
14.8.1	Function Documentation	478
14.8.1.1	getline() [1/2]	479
14.8.1.2	getline() [2/2]	479
14.8.1.3	ThrowBadAlloc()	479
14.9	Spinnaker::Video Namespace Reference	479

15 Class Documentation	481
15.1 ActionCommandResult Struct Reference	481
15.1.1 Detailed Description	481
15.1.2 Member Data Documentation	481
15.1.2.1 DeviceAddress	481
15.1.2.2 Status	481
15.2 AdapterConfigException Class Reference	482
15.2.1 Constructor & Destructor Documentation	482
15.2.1.1 AdapterConfigException() [1/2]	482
15.2.1.2 AdapterConfigException() [2/2]	483
15.2.2 Member Function Documentation	483
15.2.2.1 ErrCode()	483
15.2.2.2 GetParamStr()	483
15.3 AdapterInfo Struct Reference	483
15.3.1 Constructor & Destructor Documentation	484
15.3.1.1 AdapterInfo()	484
15.3.2 Member Data Documentation	484
15.3.2.1 adapterDescription	484
15.3.2.2 adapterGUID	484
15.3.2.3 adapterMACAddress	485
15.3.2.4 adapterName	485
15.3.2.5 dhcpEnabled	485
15.3.2.6 ipInfo	485
15.3.2.7 jumboPackets	485
15.3.2.8 jumboPacketsRegKey	485
15.3.2.9 jumboPacketValidValues	485
15.3.2.10 receiveBuffers	485
15.3.2.11 receiveBuffersMax	486
15.3.2.12 receiveBuffersMin	486
15.3.2.13 receiveBuffersRegKey	486

15.3.2.14 receiveBuffersStep	486
15.3.2.15 transmitBuffers	486
15.3.2.16 transmitBuffersMax	486
15.3.2.17 transmitBuffersMin	486
15.3.2.18 transmitBuffersRegKey	486
15.3.2.19 transmitBuffersStep	487
15.4 AttachStatistics_t Struct Reference	487
15.4.1 Detailed Description	487
15.4.2 Member Data Documentation	487
15.4.2.1 NumAttachedChunks	487
15.4.2.2 NumChunkPorts	487
15.4.2.3 NumChunks	488
15.5 AutoLock Class Reference	488
15.5.1 Constructor & Destructor Documentation	488
15.5.1.1 AutoLock()	488
15.5.1.2 ~AutoLock()	488
15.6 AutoLock Class Reference	488
15.6.1 Constructor & Destructor Documentation	489
15.6.1.1 AutoLock()	489
15.6.1.2 ~AutoLock()	489
15.7 AVIOption Struct Reference	489
15.7.1 Detailed Description	489
15.7.2 Constructor & Destructor Documentation	489
15.7.2.1 AVIOption()	490
15.7.3 Member Data Documentation	490
15.7.3.1 frameRate	490
15.7.3.2 reserved	490
15.8 BasePtr< T, B > Class Template Reference	490
15.8.1 Detailed Description	491
15.8.2 Constructor & Destructor Documentation	491

15.8.2.1	BasePtr() [1/2]	491
15.8.2.2	~BasePtr()	491
15.8.2.3	BasePtr() [2/2]	492
15.8.3	Member Function Documentation	492
15.8.3.1	get()	492
15.8.3.2	IsValid()	492
15.8.3.3	operator bool()	492
15.8.3.4	operator T*()	492
15.8.3.5	operator->()	493
15.8.3.6	operator=() [1/4]	493
15.8.3.7	operator=() [2/4]	493
15.8.3.8	operator=() [3/4]	493
15.8.3.9	operator=() [4/4]	493
15.8.3.10	operator==() [1/4]	493
15.8.3.11	operator==() [2/4]	494
15.8.3.12	operator==() [3/4]	494
15.8.3.13	operator==() [4/4]	494
15.8.4	Member Data Documentation	494
15.8.4.1	m_pT	494
15.9	BMPOption Struct Reference	494
15.9.1	Detailed Description	495
15.9.2	Constructor & Destructor Documentation	495
15.9.2.1	BMPOption()	495
15.9.3	Member Data Documentation	495
15.9.3.1	indexedColor_8bit	495
15.9.3.2	reserved	495
15.10	BooleanNode Class Reference	496
15.10.1	Detailed Description	497
15.10.2	Constructor & Destructor Documentation	497
15.10.2.1	BooleanNode() [1/2]	497

15.10.2.2 BooleanNode() [2/2]	497
15.10.2.3 ~BooleanNode()	497
15.10.3 Member Function Documentation	497
15.10.3.1 GetValue()	497
15.10.3.2 operator=()	498
15.10.3.3 SetReference()	498
15.10.3.4 SetValue()	498
15.11 Camera Class Reference	499
15.11.1 Detailed Description	529
15.11.2 Constructor & Destructor Documentation	529
15.11.2.1 ~Camera()	529
15.11.2.2 Camera()	529
15.11.3 Member Function Documentation	529
15.11.3.1 Init()	529
15.11.4 Member Data Documentation	529
15.11.4.1 AasRoiEnable	530
15.11.4.2 AasRoiHeight	530
15.11.4.3 AasRoiOffsetX	530
15.11.4.4 AasRoiOffsetY	530
15.11.4.5 AasRoiWidth	531
15.11.4.6 AcquisitionAbort	531
15.11.4.7 AcquisitionArm	531
15.11.4.8 AcquisitionBurstFrameCount	531
15.11.4.9 AcquisitionFrameCount	532
15.11.4.10AcquisitionFrameRate	532
15.11.4.11AcquisitionFrameRateEnable	532
15.11.4.12AcquisitionLineRate	532
15.11.4.13AcquisitionMode	532
15.11.4.14AcquisitionResultingFrameRate	532
15.11.4.15AcquisitionStart	533

15.11.4.16AcquisitionStatus	533
15.11.4.17AcquisitionStatusSelector	533
15.11.4.18AcquisitionStop	533
15.11.4.19ActionDeviceKey	533
15.11.4.20ActionGroupKey	533
15.11.4.21ActionGroupMask	534
15.11.4.22ActionQueueSize	534
15.11.4.23ActionSelector	534
15.11.4.24ActionUnconditionalMode	534
15.11.4.25AdaptiveCompressionEnable	534
15.11.4.26AdcBitDepth	535
15.11.4.27aPAUSEMACCtrlFramesReceived	535
15.11.4.28aPAUSEMACCtrlFramesTransmitted	535
15.11.4.29AutoAlgorithmSelector	535
15.11.4.30AutoExposureControlLoopDamping	535
15.11.4.31AutoExposureControlPriority	536
15.11.4.32AutoExposureEVCompensation	536
15.11.4.33AutoExposureExposureTimeLowerLimit	536
15.11.4.34AutoExposureExposureTimeUpperLimit	536
15.11.4.35AutoExposureGainLowerLimit	537
15.11.4.36AutoExposureGainUpperLimit	537
15.11.4.37AutoExposureGreyValueLowerLimit	537
15.11.4.38AutoExposureGreyValueUpperLimit	537
15.11.4.39AutoExposureLightingMode	538
15.11.4.40AutoExposureMeteringMode	538
15.11.4.41AutoExposureTargetGreyValue	538
15.11.4.42AutoExposureTargetGreyValueAuto	539
15.11.4.43BalanceRatio	539
15.11.4.44BalanceRatioSelector	539
15.11.4.45BalanceWhiteAuto	539

15.11.4.46BalanceWhiteAutoDamping	540
15.11.4.47BalanceWhiteAutoLowerLimit	540
15.11.4.48BalanceWhiteAutoProfile	540
15.11.4.49BalanceWhiteAutoUpperLimit	540
15.11.4.50BinningHorizontal	541
15.11.4.51BinningHorizontalMode	541
15.11.4.52BinningSelector	541
15.11.4.53BinningVertical	541
15.11.4.54BinningVerticalMode	542
15.11.4.55BlackLevel	542
15.11.4.56BlackLevelAuto	542
15.11.4.57BlackLevelAutoBalance	542
15.11.4.58BlackLevelClampingEnable	542
15.11.4.59BlackLevelRaw	543
15.11.4.60BlackLevelSelector	543
15.11.4.61ChunkBlackLevel	543
15.11.4.62ChunkBlackLevelSelector	543
15.11.4.63ChunkCompressionMode	543
15.11.4.64ChunkCompressionRatio	544
15.11.4.65ChunkCounterSelector	544
15.11.4.66ChunkCounterValue	544
15.11.4.67ChunkCRC	544
15.11.4.68ChunkEnable	544
15.11.4.69ChunkEncoderSelector	544
15.11.4.70ChunkEncoderStatus	545
15.11.4.71ChunkEncoderValue	545
15.11.4.72ChunkExposureEndLineStatusAll	545
15.11.4.73ChunkExposureTime	545
15.11.4.74ChunkExposureTimeSelector	545
15.11.4.75ChunkFrameID	545

15.11.4.76	ChunkGain	546
15.11.4.77	ChunkGainSelector	546
15.11.4.78	ChunkHeight	546
15.11.4.79	ChunkImage	546
15.11.4.80	ChunkImageComponent	546
15.11.4.81	ChunkInferenceBoundingBoxResult	546
15.11.4.82	ChunkInferenceConfidence	547
15.11.4.83	ChunkInferenceFrameId	547
15.11.4.84	ChunkInferenceResult	547
15.11.4.85	ChunkLinePitch	547
15.11.4.86	ChunkLineStatusAll	547
15.11.4.87	ChunkModeActive	547
15.11.4.88	ChunkOffsetX	548
15.11.4.89	ChunkOffsetY	548
15.11.4.90	ChunkPartSelector	548
15.11.4.91	ChunkPixelDynamicRangeMax	548
15.11.4.92	ChunkPixelDynamicRangeMin	548
15.11.4.93	ChunkPixelFormat	548
15.11.4.94	ChunkRegionID	549
15.11.4.95	ChunkScan3dAxisMax	549
15.11.4.96	ChunkScan3dAxisMin	549
15.11.4.97	ChunkScan3dCoordinateOffset	549
15.11.4.98	ChunkScan3dCoordinateReferenceSelector	549
15.11.4.99	ChunkScan3dCoordinateReferenceValue	549
15.11.4.100	ChunkScan3dCoordinateScale	550
15.11.4.101	ChunkScan3dCoordinateSelector	550
15.11.4.102	ChunkScan3dCoordinateSystem	550
15.11.4.103	ChunkScan3dCoordinateSystemReference	550
15.11.4.104	ChunkScan3dCoordinateTransformSelector	550
15.11.4.105	ChunkScan3dDistanceUnit	550

15.11.4.106hunkScan3dInvalidDataFlag	551
15.11.4.107hunkScan3dInvalidDataValue	551
15.11.4.108hunkScan3dOutputMode	551
15.11.4.109hunkScan3dTransformValue	551
15.11.4.110hunkScanLineSelector	551
15.11.4.111hunkSelector	551
15.11.4.112hunkSequencerSetActive	552
15.11.4.113hunkSerialData	552
15.11.4.114hunkSerialDataLength	552
15.11.4.115hunkSerialReceiveOverflow	552
15.11.4.116hunkSourceID	552
15.11.4.117hunkStreamChannelID	552
15.11.4.118hunkTimerSelector	553
15.11.4.119hunkTimerValue	553
15.11.4.120hunkTimestamp	553
15.11.4.121hunkTimestampLatchValue	553
15.11.4.122hunkTransferBlockID	553
15.11.4.123hunkTransferQueueCurrentBlockCount	553
15.11.4.124hunkTransferStreamID	554
15.11.4.125hunkWidth	554
15.11.4.126IConfiguration	554
15.11.4.127ITimeSlotsCount	554
15.11.4.128olorTransformationEnable	554
15.11.4.129olorTransformationSelector	555
15.11.4.130olorTransformationValue	555
15.11.4.131olorTransformationValueSelector	555
15.11.4.132ompressionRatio	555
15.11.4.133ompressionSaturationPriority	555
15.11.4.134ounterDelay	556
15.11.4.135ounterDuration	556

15.11.4.136	CounterEventActivation	556
15.11.4.137	CounterEventSource	556
15.11.4.138	CounterReset	556
15.11.4.139	CounterResetActivation	556
15.11.4.140	CounterResetSource	557
15.11.4.141	CounterSelector	557
15.11.4.142	CounterStatus	557
15.11.4.143	CounterTriggerActivation	557
15.11.4.144	CounterTriggerSource	557
15.11.4.145	CounterValue	557
15.11.4.146	CounterValueAtReset	558
15.11.4.147	CxpConnectionSelector	558
15.11.4.148	CxpConnectionTestErrorCount	558
15.11.4.149	CxpConnectionTestMode	558
15.11.4.150	CxpConnectionTestPacketCount	558
15.11.4.151	CxpLinkConfiguration	558
15.11.4.152	CxpLinkConfigurationPreferred	559
15.11.4.153	CxpLinkConfigurationStatus	559
15.11.4.154	CxpPoCxpAuto	559
15.11.4.155	CxpPoCxpStatus	559
15.11.4.156	CxpPoCxpTripReset	559
15.11.4.157	CxpPoCxpTurnOff	559
15.11.4.158	DecimationHorizontal	560
15.11.4.159	DecimationHorizontalMode	560
15.11.4.160	DecimationSelector	560
15.11.4.161	DecimationVertical	560
15.11.4.162	DecimationVerticalMode	561
15.11.4.163	DefectCorrectionMode	561
15.11.4.164	DefectCorrectStaticEnable	561
15.11.4.165	DefectTableApply	561

15.11.4.166	DefectTableCoordinateX	561
15.11.4.167	DefectTableCoordinateY	562
15.11.4.168	DefectTableFactoryRestore	562
15.11.4.169	DefectTableIndex	562
15.11.4.170	DefectTablePixelCount	562
15.11.4.171	DefectTableSave	563
15.11.4.172	Deinterlacing	563
15.11.4.173	DeviceCharacterSet	563
15.11.4.174	DeviceClockFrequency	563
15.11.4.175	DeviceClockSelector	563
15.11.4.176	DeviceConnectionSelector	564
15.11.4.177	DeviceConnectionSpeed	564
15.11.4.178	DeviceConnectionStatus	564
15.11.4.179	DeviceEventChannelCount	564
15.11.4.180	DeviceFamilyName	564
15.11.4.181	DeviceFeaturePersistenceEnd	564
15.11.4.182	DeviceFeaturePersistenceStart	565
15.11.4.183	DeviceFirmwareVersion	565
15.11.4.184	DeviceGenCPVersionMajor	565
15.11.4.185	DeviceGenCPVersionMinor	565
15.11.4.186	DeviceID	565
15.11.4.187	DeviceIndicatorMode	565
15.11.4.188	DeviceLinkBandwidthReserve	566
15.11.4.189	DeviceLinkCommandTimeout	566
15.11.4.190	DeviceLinkConnectionCount	566
15.11.4.191	DeviceLinkCurrentThroughput	566
15.11.4.192	DeviceLinkHeartbeatMode	566
15.11.4.193	DeviceLinkHeartbeatTimeout	566
15.11.4.194	DeviceLinkSelector	567
15.11.4.195	DeviceLinkSpeed	567

15.11.4.196	DeviceLinkThroughputLimit	567
15.11.4.197	DeviceLinkThroughputLimitMode	567
15.11.4.198	DeviceManifestEntrySelector	568
15.11.4.199	DeviceManifestPrimaryURL	568
15.11.4.200	DeviceManifestSchemaMajorVersion	568
15.11.4.201	DeviceManifestSchemaMinorVersion	568
15.11.4.202	DeviceManifestSecondaryURL	568
15.11.4.203	DeviceManifestXMLMajorVersion	568
15.11.4.204	DeviceManifestXMLMinorVersion	569
15.11.4.205	DeviceManifestXMLSubMinorVersion	569
15.11.4.206	DeviceManufacturerInfo	569
15.11.4.207	DeviceMaxThroughput	569
15.11.4.208	DeviceModelName	569
15.11.4.209	DevicePowerSupplySelector	570
15.11.4.210	DeviceRegistersCheck	570
15.11.4.211	DeviceRegistersEndianness	570
15.11.4.212	DeviceRegistersStreamingEnd	570
15.11.4.213	DeviceRegistersStreamingStart	570
15.11.4.214	DeviceRegistersValid	570
15.11.4.215	DeviceReset	571
15.11.4.216	DeviceScanType	571
15.11.4.217	DeviceSerialNumber	571
15.11.4.218	DeviceSerialPortBaudRate	571
15.11.4.219	DeviceSerialPortSelector	571
15.11.4.220	DeviceSFNCVersionMajor	572
15.11.4.221	DeviceSFNCVersionMinor	572
15.11.4.222	DeviceSFNCVersionSubMinor	572
15.11.4.223	DeviceStreamChannelCount	572
15.11.4.224	DeviceStreamChannelEndianness	572
15.11.4.225	DeviceStreamChannelLink	573

15.11.4.226	DeviceStreamChannelPacketSize	573
15.11.4.227	DeviceStreamChannelSelector	573
15.11.4.228	DeviceStreamChannelType	573
15.11.4.229	DeviceTapGeometry	573
15.11.4.230	DeviceTemperature	573
15.11.4.231	DeviceTemperatureSelector	574
15.11.4.232	DeviceTLType	574
15.11.4.233	DeviceTLVersionMajor	574
15.11.4.234	DeviceTLVersionMinor	574
15.11.4.235	DeviceTLVersionSubMinor	574
15.11.4.236	DeviceType	575
15.11.4.237	DeviceUptime	575
15.11.4.238	DeviceUserID	575
15.11.4.239	DeviceVendorName	575
15.11.4.240	DeviceVersion	575
15.11.4.241	EncoderDivider	575
15.11.4.242	EncoderMode	576
15.11.4.243	EncoderOutputMode	576
15.11.4.244	EncoderReset	576
15.11.4.245	EncoderResetActivation	576
15.11.4.246	EncoderResetSource	576
15.11.4.247	EncoderSelector	576
15.11.4.248	EncoderSourceA	577
15.11.4.249	EncoderSourceB	577
15.11.4.250	EncoderStatus	577
15.11.4.251	EncoderTimeout	577
15.11.4.252	EncoderValue	577
15.11.4.253	EncoderValueAtReset	577
15.11.4.254	EnumerationCount	578
15.11.4.255	EventAcquisitionEnd	578

15.11.4.255	EventAcquisitionEndFrameID	578
15.11.4.257	EventAcquisitionEndTimestamp	578
15.11.4.258	EventAcquisitionError	578
15.11.4.259	EventAcquisitionErrorFrameID	578
15.11.4.260	EventAcquisitionErrorTimestamp	579
15.11.4.261	EventAcquisitionStart	579
15.11.4.262	EventAcquisitionStartFrameID	579
15.11.4.263	EventAcquisitionStartTimestamp	579
15.11.4.264	EventAcquisitionTransferEnd	579
15.11.4.265	EventAcquisitionTransferEndFrameID	579
15.11.4.266	EventAcquisitionTransferEndTimestamp	580
15.11.4.267	EventAcquisitionTransferStart	580
15.11.4.268	EventAcquisitionTransferStartFrameID	580
15.11.4.269	EventAcquisitionTransferStartTimestamp	580
15.11.4.270	EventAcquisitionTrigger	580
15.11.4.271	EventAcquisitionTriggerFrameID	580
15.11.4.272	EventAcquisitionTriggerTimestamp	581
15.11.4.273	EventActionLate	581
15.11.4.274	EventActionLateFrameID	581
15.11.4.275	EventActionLateTimestamp	581
15.11.4.276	EventCounter0End	581
15.11.4.277	EventCounter0EndFrameID	581
15.11.4.278	EventCounter0EndTimestamp	582
15.11.4.279	EventCounter0Start	582
15.11.4.280	EventCounter0StartFrameID	582
15.11.4.281	EventCounter0StartTimestamp	582
15.11.4.282	EventCounter1End	582
15.11.4.283	EventCounter1EndFrameID	582
15.11.4.284	EventCounter1EndTimestamp	583
15.11.4.285	EventCounter1Start	583

15.11.4.286	EventCounter1StartFrameID	583
15.11.4.287	EventCounter1StartTimestamp	583
15.11.4.288	EventEncoder0Restarted	583
15.11.4.289	EventEncoder0RestartedFrameID	583
15.11.4.290	EventEncoder0RestartedTimestamp	584
15.11.4.291	EventEncoder0Stopped	584
15.11.4.292	EventEncoder0StoppedFrameID	584
15.11.4.293	EventEncoder0StoppedTimestamp	584
15.11.4.294	EventEncoder1Restarted	584
15.11.4.295	EventEncoder1RestartedFrameID	584
15.11.4.296	EventEncoder1RestartedTimestamp	585
15.11.4.297	EventEncoder1Stopped	585
15.11.4.298	EventEncoder1StoppedFrameID	585
15.11.4.299	EventEncoder1StoppedTimestamp	585
15.11.4.300	EventError	585
15.11.4.301	EventErrorCode	585
15.11.4.302	EventErrorFrameID	586
15.11.4.303	EventErrorTimestamp	586
15.11.4.304	EventExposureEnd	586
15.11.4.305	EventExposureEndFrameID	586
15.11.4.306	EventExposureEndTimestamp	586
15.11.4.307	EventExposureStart	586
15.11.4.308	EventExposureStartFrameID	587
15.11.4.309	EventExposureStartTimestamp	587
15.11.4.310	EventFrameBurstEnd	587
15.11.4.311	EventFrameBurstEndFrameID	587
15.11.4.312	EventFrameBurstEndTimestamp	587
15.11.4.313	EventFrameBurstStart	587
15.11.4.314	EventFrameBurstStartFrameID	588
15.11.4.315	EventFrameBurstStartTimestamp	588

15.11.4.31	EventFrameEnd	588
15.11.4.31	EventFrameEndFrameID	588
15.11.4.31	EventFrameEndTimestamp	588
15.11.4.31	EventFrameStart	588
15.11.4.32	EventFrameStartFrameID	589
15.11.4.32	EventFrameStartTimestamp	589
15.11.4.32	EventFrameTransferEnd	589
15.11.4.32	EventFrameTransferEndFrameID	589
15.11.4.32	EventFrameTransferEndTimestamp	589
15.11.4.32	EventFrameTransferStart	589
15.11.4.32	EventFrameTransferStartFrameID	590
15.11.4.32	EventFrameTransferStartTimestamp	590
15.11.4.32	EventFrameTrigger	590
15.11.4.32	EventFrameTriggerFrameID	590
15.11.4.33	EventFrameTriggerTimestamp	590
15.11.4.33	EventLine0AnyEdge	590
15.11.4.33	EventLine0AnyEdgeFrameID	591
15.11.4.33	EventLine0AnyEdgeTimestamp	591
15.11.4.33	EventLine0FallingEdge	591
15.11.4.33	EventLine0FallingEdgeFrameID	591
15.11.4.33	EventLine0FallingEdgeTimestamp	591
15.11.4.33	EventLine0RisingEdge	591
15.11.4.33	EventLine0RisingEdgeFrameID	592
15.11.4.33	EventLine0RisingEdgeTimestamp	592
15.11.4.34	EventLine1AnyEdge	592
15.11.4.34	EventLine1AnyEdgeFrameID	592
15.11.4.34	EventLine1AnyEdgeTimestamp	592
15.11.4.34	EventLine1FallingEdge	592
15.11.4.34	EventLine1FallingEdgeFrameID	593
15.11.4.34	EventLine1FallingEdgeTimestamp	593

15.11.4.346	EventLine1RisingEdge	593
15.11.4.347	EventLine1RisingEdgeFrameID	593
15.11.4.348	EventLine1RisingEdgeTimestamp	593
15.11.4.349	EventLinkSpeedChange	593
15.11.4.350	EventLinkSpeedChangeFrameID	594
15.11.4.351	EventLinkSpeedChangeTimestamp	594
15.11.4.352	EventLinkTrigger0	594
15.11.4.353	EventLinkTrigger0FrameID	594
15.11.4.354	EventLinkTrigger0Timestamp	594
15.11.4.355	EventLinkTrigger1	594
15.11.4.356	EventLinkTrigger1FrameID	595
15.11.4.357	EventLinkTrigger1Timestamp	595
15.11.4.358	EventNotification	595
15.11.4.359	EventSelector	595
15.11.4.360	EventSequencerSetChange	595
15.11.4.361	EventSequencerSetChangeFrameID	595
15.11.4.362	EventSequencerSetChangeTimestamp	596
15.11.4.363	EventSerialData	596
15.11.4.364	EventSerialDataLength	596
15.11.4.365	EventSerialPortReceive	596
15.11.4.366	EventSerialPortReceiveTimestamp	596
15.11.4.367	EventSerialReceiveOverflow	596
15.11.4.368	EventStream0TransferBlockEnd	597
15.11.4.369	EventStream0TransferBlockEndFrameID	597
15.11.4.370	EventStream0TransferBlockEndTimestamp	597
15.11.4.371	EventStream0TransferBlockStart	597
15.11.4.372	EventStream0TransferBlockStartFrameID	597
15.11.4.373	EventStream0TransferBlockStartTimestamp	597
15.11.4.374	EventStream0TransferBlockTrigger	598
15.11.4.375	EventStream0TransferBlockTriggerFrameID	598

15.11.4.375	EventStream0TransferBlockTriggerTimestamp	598
15.11.4.376	EventStream0TransferBurstEnd	598
15.11.4.377	EventStream0TransferBurstEndFrameID	598
15.11.4.378	EventStream0TransferBurstEndTimestamp	598
15.11.4.380	EventStream0TransferBurstStart	599
15.11.4.381	EventStream0TransferBurstStartFrameID	599
15.11.4.382	EventStream0TransferBurstStartTimestamp	599
15.11.4.383	EventStream0TransferEnd	599
15.11.4.384	EventStream0TransferEndFrameID	599
15.11.4.385	EventStream0TransferEndTimestamp	599
15.11.4.386	EventStream0TransferOverflow	600
15.11.4.387	EventStream0TransferOverflowFrameID	600
15.11.4.388	EventStream0TransferOverflowTimestamp	600
15.11.4.389	EventStream0TransferPause	600
15.11.4.390	EventStream0TransferPauseFrameID	600
15.11.4.391	EventStream0TransferPauseTimestamp	600
15.11.4.392	EventStream0TransferResume	601
15.11.4.393	EventStream0TransferResumeFrameID	601
15.11.4.394	EventStream0TransferResumeTimestamp	601
15.11.4.395	EventStream0TransferStart	601
15.11.4.396	EventStream0TransferStartFrameID	601
15.11.4.397	EventStream0TransferStartTimestamp	601
15.11.4.398	EventTest	602
15.11.4.399	EventTestTimestamp	602
15.11.4.400	EventTimer0End	602
15.11.4.401	EventTimer0EndFrameID	602
15.11.4.402	EventTimer0EndTimestamp	602
15.11.4.403	EventTimer0Start	602
15.11.4.404	EventTimer0StartFrameID	603
15.11.4.405	EventTimer0StartTimestamp	603

15.11.4.405	EventTimer1End	603
15.11.4.407	EventTimer1EndFrameID	603
15.11.4.408	EventTimer1EndTimestamp	603
15.11.4.409	EventTimer1Start	603
15.11.4.410	EventTimer1StartFrameID	604
15.11.4.411	EventTimer1StartTimestamp	604
15.11.4.412	ExposureActiveMode	604
15.11.4.413	ExposureAuto	604
15.11.4.414	ExposureMode	604
15.11.4.415	ExposureTime	604
15.11.4.416	ExposureTimeMode	605
15.11.4.417	ExposureTimeSelector	605
15.11.4.418	FactoryReset	605
15.11.4.419	FileAccessBuffer	605
15.11.4.420	FileAccessLength	605
15.11.4.421	FileAccessOffset	605
15.11.4.422	FileOpenMode	606
15.11.4.423	FileOperationExecute	606
15.11.4.424	FileOperationResult	606
15.11.4.425	FileOperationSelector	606
15.11.4.426	FileOperationStatus	606
15.11.4.427	FileSelector	607
15.11.4.428	FileSize	607
15.11.4.429	Gain	607
15.11.4.430	GainAuto	607
15.11.4.431	GainAutoBalance	608
15.11.4.432	GainSelector	608
15.11.4.433	Gamma	608
15.11.4.434	GammaEnable	608
15.11.4.435	SevActiveLinkCount	608

15.11.4.436evCCP	608
15.11.4.437evCurrentDefaultGateway	609
15.11.4.438evCurrentIPAddress	609
15.11.4.439evCurrentIPConfigurationDHCP	609
15.11.4.440evCurrentIPConfigurationLLA	609
15.11.4.441evCurrentIPConfigurationPersistentIP	609
15.11.4.442evCurrentPhysicalLinkConfiguration	609
15.11.4.443evCurrentSubnetMask	610
15.11.4.444evDiscoveryAckDelay	610
15.11.4.445evFirstURL	610
15.11.4.446evGVCPExtendedStatusCodes	610
15.11.4.447evGVCPExtendedStatusCodesSelector	610
15.11.4.448evGVCPHeartbeatDisable	610
15.11.4.449evGVCPPendingAck	611
15.11.4.450evGVCPPendingTimeout	611
15.11.4.451evGVSPExtendedIDMode	611
15.11.4.452evHeartbeatTimeout	611
15.11.4.453evIEEE1588	611
15.11.4.454evIEEE1588ClockAccuracy	611
15.11.4.455evIEEE1588Mode	612
15.11.4.456evIEEE1588Status	612
15.11.4.457evInterfaceSelector	612
15.11.4.458evIPConfigurationStatus	612
15.11.4.459evMACAddress	612
15.11.4.460evMCDA	612
15.11.4.461evMCPHostPort	613
15.11.4.462evMCRC	613
15.11.4.463evMCSP	613
15.11.4.464evMCTT	613
15.11.4.465evNumberOfInterfaces	613

15.11.4.466	evPAUSEFrameReception	613
15.11.4.467	evPAUSEFrameTransmission	614
15.11.4.468	evPersistentDefaultGateway	614
15.11.4.469	evPersistentIPAddress	614
15.11.4.470	evPersistentSubnetMask	614
15.11.4.471	evPhysicalLinkConfiguration	614
15.11.4.472	evPrimaryApplicationIPAddress	614
15.11.4.473	evPrimaryApplicationSocket	615
15.11.4.474	evPrimaryApplicationSwitchoverKey	615
15.11.4.475	evSCCFGAllInTransmission	615
15.11.4.476	evSCCFGExtendedChunkData	615
15.11.4.477	evSCCFGPacketResendDestination	615
15.11.4.478	evSCCFGUnconditionalStreaming	615
15.11.4.479	evSCDA	616
15.11.4.480	evSCPD	616
15.11.4.481	evSCPDDirection	616
15.11.4.482	evSCPHostPort	616
15.11.4.483	evSCPIInterfaceIndex	616
15.11.4.484	evSCPSBigEndian	616
15.11.4.485	evSCPSDoNotFragment	617
15.11.4.486	evSCPSFireTestPacket	617
15.11.4.487	evSCPSPacketSize	617
15.11.4.488	evSCSP	617
15.11.4.489	evSCZoneConfigurationLock	617
15.11.4.490	evSCZoneCount	617
15.11.4.491	evSCZoneDirectionAll	618
15.11.4.492	evSecondURL	618
15.11.4.493	evStreamChannelSelector	618
15.11.4.494	evSupportedOption	618
15.11.4.495	evSupportedOptionSelector	618

15.11.4.496	DevTimestampTickFrequency	618
15.11.4.497	GuiXmlManifestAddress	619
15.11.4.498	Height	619
15.11.4.499	HeightMax	619
15.11.4.500	ImageComponentEnable	619
15.11.4.501	ImageComponentSelector	619
15.11.4.502	ImageCompressionBitrate	619
15.11.4.503	ImageCompressionJPEGFormatOption	620
15.11.4.504	ImageCompressionMode	620
15.11.4.505	ImageCompressionQuality	620
15.11.4.506	ImageCompressionRateOption	620
15.11.4.507	IpEnable	620
15.11.4.508	LineFilterWidth	621
15.11.4.509	LineFormat	621
15.11.4.510	LineInputFilterSelector	621
15.11.4.511	LineInverter	621
15.11.4.512	LineMode	621
15.11.4.513	LinePitch	621
15.11.4.514	LineSelector	622
15.11.4.515	LineSource	622
15.11.4.516	LineStatus	622
15.11.4.517	LineStatusAll	622
15.11.4.518	LinkErrorCount	622
15.11.4.519	LinkUptime	622
15.11.4.520	LogicBlockLUTInputActivation	623
15.11.4.521	LogicBlockLUTInputSelector	623
15.11.4.522	LogicBlockLUTInputSource	623
15.11.4.523	LogicBlockLUTOutputValue	623
15.11.4.524	LogicBlockLUTOutputValueAll	623
15.11.4.525	LogicBlockLUTRowIndex	623

15.11.4.526	LogicBlockLUTSelector	624
15.11.4.527	LogicBlockSelector	624
15.11.4.528	LUTEnable	624
15.11.4.529	LUTIndex	624
15.11.4.530	LUTSelector	624
15.11.4.531	LUTValue	625
15.11.4.532	LUTValueAll	625
15.11.4.533	MaxDeviceResetTime	625
15.11.4.534	OffsetX	625
15.11.4.535	OffsetY	625
15.11.4.536	PacketResendRequestCount	626
15.11.4.537	PayloadSize	626
15.11.4.538	PixelColorFilter	626
15.11.4.539	PixelDynamicRangeMax	626
15.11.4.540	PixelDynamicRangeMin	626
15.11.4.541	PixelFormat	627
15.11.4.542	PixelFormatInfoID	627
15.11.4.543	PixelFormatInfoSelector	627
15.11.4.544	PixelSize	627
15.11.4.545	PowerSupplyCurrent	627
15.11.4.546	PowerSupplyVoltage	627
15.11.4.547	RegionDestination	628
15.11.4.548	RegionMode	628
15.11.4.549	RegionSelector	628
15.11.4.550	ReverseX	628
15.11.4.551	ReverseY	628
15.11.4.552	RgbTransformLightSource	629
15.11.4.553	Saturation	629
15.11.4.554	SaturationEnable	629
15.11.4.555	Scan3dAxisMax	629

15.11.4.555	can3dAxisMin	629
15.11.4.557	can3dCoordinateOffset	630
15.11.4.558	can3dCoordinateReferenceSelector	630
15.11.4.559	can3dCoordinateReferenceValue	630
15.11.4.560	can3dCoordinateScale	630
15.11.4.563	can3dCoordinateSelector	630
15.11.4.562	can3dCoordinateSystem	630
15.11.4.563	can3dCoordinateSystemReference	631
15.11.4.569	can3dCoordinateTransformSelector	631
15.11.4.565	can3dDistanceUnit	631
15.11.4.566	can3dInvalidDataFlag	631
15.11.4.567	can3dInvalidDataValue	631
15.11.4.568	can3dOutputMode	631
15.11.4.569	can3dTransformValue	632
15.11.4.570	sensorDescription	632
15.11.4.573	sensorDigitizationTaps	632
15.11.4.572	sensorHeight	632
15.11.4.573	sensorShutterMode	632
15.11.4.574	sensorTaps	632
15.11.4.575	sensorWidth	633
15.11.4.576	sequencerConfigurationMode	633
15.11.4.577	sequencerConfigurationValid	633
15.11.4.578	sequencerFeatureEnable	633
15.11.4.579	sequencerMode	633
15.11.4.580	sequencerPathSelector	634
15.11.4.583	sequencerSetActive	634
15.11.4.582	sequencerSetLoad	634
15.11.4.583	sequencerSetNext	634
15.11.4.584	sequencerSetSave	634
15.11.4.585	sequencerSetSelector	635

15.11.4.586	SequencerSetStart	635
15.11.4.587	SequencerSetValid	635
15.11.4.588	SequencerTriggerActivation	635
15.11.4.589	SequencerTriggerSource	635
15.11.4.590	SerialPortBaudRate	636
15.11.4.591	SerialPortDataBits	636
15.11.4.592	SerialPortParity	636
15.11.4.593	SerialPortSelector	636
15.11.4.594	SerialPortSource	636
15.11.4.595	SerialPortStopBits	636
15.11.4.596	SerialReceiveFramingErrorCount	637
15.11.4.597	SerialReceiveParityErrorCount	637
15.11.4.598	SerialReceiveQueueClear	637
15.11.4.599	SerialReceiveQueueCurrentCharacterCount	637
15.11.4.600	SerialReceiveQueueMaxCharacterCount	637
15.11.4.601	SerialTransmitQueueCurrentCharacterCount	637
15.11.4.602	SerialTransmitQueueMaxCharacterCount	638
15.11.4.603	Sharpening	638
15.11.4.604	SharpeningAuto	638
15.11.4.605	SharpeningEnable	638
15.11.4.606	SharpeningThreshold	639
15.11.4.607	SoftwareSignalPulse	639
15.11.4.608	SoftwareSignalSelector	639
15.11.4.609	SourceCount	639
15.11.4.610	SourceSelector	639
15.11.4.611	Test0001	640
15.11.4.612	TestEventGenerate	640
15.11.4.613	TestPattern	640
15.11.4.614	TestPatternGeneratorSelector	640
15.11.4.615	TestPendingAck	640

15.11.4.61TimerDelay	641
15.11.4.61TimerDuration	641
15.11.4.61TimerReset	641
15.11.4.61TimerSelector	641
15.11.4.62TimerStatus	641
15.11.4.62TimerTriggerActivation	641
15.11.4.62TimerTriggerSource	642
15.11.4.62TimerValue	642
15.11.4.62Timestamp	642
15.11.4.62TimestampLatch	642
15.11.4.62TimestampLatchValue	642
15.11.4.62TimestampReset	642
15.11.4.62BLParamsLocked	643
15.11.4.62TransferAbort	643
15.11.4.63TransferBlockCount	643
15.11.4.63TransferBurstCount	643
15.11.4.63TransferComponentSelector	643
15.11.4.63TransferControlMode	643
15.11.4.63TransferOperationMode	644
15.11.4.63TransferPause	644
15.11.4.63TransferQueueCurrentBlockCount	644
15.11.4.63TransferQueueMaxBlockCount	644
15.11.4.63TransferQueueMode	644
15.11.4.63TransferQueueOverflowCount	644
15.11.4.64TransferResume	645
15.11.4.64TransferSelector	645
15.11.4.64TransferStart	645
15.11.4.64TransferStatus	645
15.11.4.64TransferStatusSelector	645
15.11.4.64TransferStop	645

15.11.4.64TransferStreamChannel	646
15.11.4.64TransferTriggerActivation	646
15.11.4.64TransferTriggerMode	646
15.11.4.64TransferTriggerSelector	646
15.11.4.65TransferTriggerSource	646
15.11.4.65TriggerActivation	646
15.11.4.65TriggerDelay	647
15.11.4.65TriggerDivider	647
15.11.4.65TriggerEventTest	647
15.11.4.65TriggerMode	647
15.11.4.65TriggerMultiplier	647
15.11.4.65TriggerOverlap	648
15.11.4.65TriggerSelector	648
15.11.4.65TriggerSoftware	648
15.11.4.66TriggerSource	648
15.11.4.66UserOutputSelector	648
15.11.4.66UserOutputValue	649
15.11.4.66UserOutputValueAll	649
15.11.4.66UserOutputValueAllMask	649
15.11.4.66UserSetDefault	649
15.11.4.66UserSetFeatureEnable	649
15.11.4.66UserSetLoad	650
15.11.4.66UserSetSave	650
15.11.4.66UserSetSelector	650
15.11.4.67V3_3Enable	650
15.11.4.67WhiteClip	650
15.11.4.67WhiteClipSelector	651
15.11.4.67Width	651
15.11.4.67WidthMax	651
15.12CameraBase Class Reference	652

15.12.1 Detailed Description	654
15.12.2 Constructor & Destructor Documentation	654
15.12.2.1 ~CameraBase()	654
15.12.2.2 CameraBase() [1/2]	655
15.12.2.3 CameraBase() [2/2]	655
15.12.3 Member Function Documentation	655
15.12.3.1 BeginAcquisition()	655
15.12.3.2 Delnit()	655
15.12.3.3 DiscoverMaxPacketSize()	656
15.12.3.4 EndAcquisition()	656
15.12.3.5 ForceIP()	656
15.12.3.6 GetAccessMode()	657
15.12.3.7 GetBufferOwnership()	657
15.12.3.8 GetGuiXml()	657
15.12.3.9 GetNextImage()	658
15.12.3.10 GetNodeMap()	658
15.12.3.11 GetNumDataStreams()	659
15.12.3.12 GetNumImagesInUse()	659
15.12.3.13 GetTLDeviceNodeMap()	659
15.12.3.14 GetTLStreamNodeMap()	660
15.12.3.15 GetUniqueID()	660
15.12.3.16 GetUserBufferCount()	660
15.12.3.17 GetUserBufferSize()	661
15.12.3.18 GetUserBufferTotalSize()	661
15.12.3.19 Init()	662
15.12.3.20 IsInitialized()	662
15.12.3.21 IsStreaming()	662
15.12.3.22 IsValid()	663
15.12.3.23 operator=()	663
15.12.3.24 ReadPort()	663

15.12.3.25	RegisterEventHandler() [1/2]	663
15.12.3.26	RegisterEventHandler() [2/2]	664
15.12.3.27	SetBufferOwnership()	664
15.12.3.28	SetUserBuffers() [1/2]	665
15.12.3.29	SetUserBuffers() [2/2]	665
15.12.3.30	UnregisterEventHandler()	666
15.12.3.31	WritePort()	666
15.12.4	Friends And Related Function Documentation	666
15.12.4.1	InterfacelImpl	666
15.13	CameraList Class Reference	667
15.13.1	Detailed Description	668
15.13.2	Constructor & Destructor Documentation	668
15.13.2.1	CameraList() [1/2]	668
15.13.2.2	~CameraList()	668
15.13.2.3	CameraList() [2/2]	669
15.13.3	Member Function Documentation	669
15.13.3.1	Append()	669
15.13.3.2	Clear()	669
15.13.3.3	GetByDeviceID()	669
15.13.3.4	GetByIndex()	670
15.13.3.5	GetBySerial()	670
15.13.3.6	GetSize()	671
15.13.3.7	operator=()	671
15.13.3.8	operator[]()	671
15.13.3.9	RemoveByDeviceID()	671
15.13.3.10	RemoveByIndex()	672
15.13.3.11	RemoveBySerial()	672
15.14	CameraPtr Class Reference	672
15.14.1	Detailed Description	673
15.15	CategoryNode Class Reference	674

15.15.1 Detailed Description	675
15.15.2 Constructor & Destructor Documentation	675
15.15.2.1 CategoryNode() [1/2]	675
15.15.2.2 CategoryNode() [2/2]	675
15.15.2.3 ~CategoryNode()	675
15.15.3 Member Function Documentation	675
15.15.3.1 GetFeatures()	676
15.15.3.2 SetReference()	676
15.16 CChunkAdapter Class Reference	676
15.16.1 Detailed Description	677
15.16.2 Constructor & Destructor Documentation	677
15.16.2.1 ~CChunkAdapter()	677
15.16.2.2 CChunkAdapter()	677
15.16.3 Member Function Documentation	677
15.16.3.1 AttachBuffer()	677
15.16.3.2 AttachNodeMap()	678
15.16.3.3 CheckBufferLayout()	678
15.16.3.4 ClearCaches()	678
15.16.3.5 DetachBuffer()	678
15.16.3.6 DetachNodeMap()	678
15.16.3.7 UpdateBuffer()	678
15.16.4 Member Data Documentation	679
15.16.4.1 m_pChunkAdapter	679
15.17 CChunkAdapterDcam Class Reference	679
15.17.1 Detailed Description	680
15.17.2 Constructor & Destructor Documentation	680
15.17.2.1 CChunkAdapterDcam()	680
15.17.2.2 ~CChunkAdapterDcam()	680
15.17.3 Member Function Documentation	680
15.17.3.1 AttachBuffer()	681

15.17.3.2 CheckBufferLayout()	681
15.17.3.3 CheckCRC()	681
15.17.3.4 HasCRC()	681
15.18CChunkAdapterGeneric Class Reference	682
15.18.1 Constructor & Destructor Documentation	682
15.18.1.1 CChunkAdapterGeneric()	683
15.18.1.2 ~CChunkAdapterGeneric()	683
15.18.2 Member Function Documentation	683
15.18.2.1 AttachBuffer() [1/3]	683
15.18.2.2 AttachBuffer() [2/3]	683
15.18.2.3 AttachBuffer() [3/3]	683
15.18.2.4 CheckBufferLayout()	684
15.19CChunkAdapterGEV Class Reference	684
15.19.1 Detailed Description	685
15.19.2 Constructor & Destructor Documentation	685
15.19.2.1 CChunkAdapterGEV()	685
15.19.2.2 ~CChunkAdapterGEV()	685
15.19.3 Member Function Documentation	685
15.19.3.1 AttachBuffer()	686
15.19.3.2 CheckBufferLayout()	686
15.20CChunkAdapterU3V Class Reference	686
15.20.1 Detailed Description	687
15.20.2 Constructor & Destructor Documentation	687
15.20.2.1 CChunkAdapterU3V()	687
15.20.2.2 ~CChunkAdapterU3V()	688
15.20.3 Member Function Documentation	688
15.20.3.1 AttachBuffer()	688
15.20.3.2 CheckBufferLayout()	688
15.21CChunkPort Class Reference	689
15.21.1 Detailed Description	690

15.21.2 Constructor & Destructor Documentation	690
15.21.2.1 CChunkPort()	690
15.21.2.2 ~CChunkPort()	691
15.21.3 Member Function Documentation	691
15.21.3.1 AttachChunk()	691
15.21.3.2 AttachPort()	691
15.21.3.3 CheckChunkID() [1/2]	691
15.21.3.4 CheckChunkID() [2/2]	691
15.21.3.5 ClearCache()	692
15.21.3.6 DetachChunk()	692
15.21.3.7 DetachPort()	692
15.21.3.8 GetAccessMode()	692
15.21.3.9 GetChunkIDLength()	692
15.21.3.10 GetPrincipalInterfaceType()	692
15.21.3.11 GetSwapEndianess()	693
15.21.3.12 InvalidateNode()	693
15.21.3.13 Read()	693
15.21.3.14 SetPortImpl()	693
15.21.3.15 UpdateBuffer()	693
15.21.3.16 Write()	693
15.21.4 Member Data Documentation	694
15.21.4.1 m_pChunkPort	694
15.21.4.2 m_pPort	694
15.21.4.3 m_pPortAdapter	694
15.22 CCMSettings Struct Reference	694
15.22.1 Constructor & Destructor Documentation	694
15.22.1.1 CCMSettings()	694
15.22.2 Member Data Documentation	695
15.22.2.1 ColorTemperature	695
15.22.2.2 CustomCCMCode	695

15.22.2.3 Sensor	695
15.22.2.4 Type	695
15.23 CEnumerationTRef< EnumT > Class Template Reference	695
15.23.1 Detailed Description	698
15.23.2 Constructor & Destructor Documentation	698
15.23.2.1 CEnumerationTRef() [1/2]	698
15.23.2.2 CEnumerationTRef() [2/2]	698
15.23.2.3 ~CEnumerationTRef()	698
15.23.3 Member Function Documentation	698
15.23.3.1 GetCurrentEntry()	699
15.23.3.2 GetEntry() [1/2]	699
15.23.3.3 GetEntry() [2/2]	699
15.23.3.4 GetValue()	699
15.23.3.5 operator()()	700
15.23.3.6 operator=() [1/2]	700
15.23.3.7 operator=() [2/2]	700
15.23.3.8 SetEnumReference()	700
15.23.3.9 SetNumEnums()	700
15.23.3.10 SetReference()	701
15.23.3.11 SetValue()	701
15.24 CEventAdapter Class Reference	701
15.24.1 Detailed Description	702
15.24.2 Constructor & Destructor Documentation	702
15.24.2.1 CEventAdapter()	702
15.24.2.2 ~CEventAdapter()	702
15.24.3 Member Function Documentation	702
15.24.3.1 AttachNodeMap()	702
15.24.3.2 DeliverMessage()	703
15.24.3.3 DetachNodeMap()	703
15.24.4 Member Data Documentation	703

15.24.4.1 m_pEventAdapter	703
15.25 CEventAdapter1394 Class Reference	703
15.25.1 Detailed Description	704
15.25.2 Constructor & Destructor Documentation	704
15.25.2.1 CEventAdapter1394()	704
15.25.2.2 ~CEventAdapter1394()	704
15.25.3 Member Function Documentation	705
15.25.3.1 DeliverEventMessage()	705
15.25.3.2 DeliverMessage()	705
15.26 CEventAdapterGeneric Class Reference	705
15.26.1 Detailed Description	706
15.26.2 Constructor & Destructor Documentation	706
15.26.2.1 CEventAdapterGeneric()	706
15.26.2.2 ~CEventAdapterGeneric()	707
15.26.3 Member Function Documentation	707
15.26.3.1 DeliverMessage() [1/3]	707
15.26.3.2 DeliverMessage() [2/3]	707
15.26.3.3 DeliverMessage() [3/3]	707
15.27 CEventAdapterGEV Class Reference	708
15.27.1 Detailed Description	709
15.27.2 Constructor & Destructor Documentation	709
15.27.2.1 CEventAdapterGEV()	709
15.27.2.2 ~CEventAdapterGEV()	709
15.27.3 Member Function Documentation	709
15.27.3.1 DeliverEventMessage() [1/2]	709
15.27.3.2 DeliverEventMessage() [2/2]	709
15.27.3.3 DeliverMessage()	710
15.28 CEventAdapterU3V Class Reference	710
15.28.1 Detailed Description	711
15.28.2 Constructor & Destructor Documentation	711

15.28.2.1 CEventAdapterU3V()	711
15.28.2.2 ~CEventAdapterU3V()	711
15.28.3 Member Function Documentation	711
15.28.3.1 DeliverEventMessage()	711
15.28.3.2 DeliverMessage()	712
15.29 CEventPort Class Reference	712
15.29.1 Detailed Description	713
15.29.2 Constructor & Destructor Documentation	713
15.29.2.1 CEventPort()	714
15.29.2.2 ~CEventPort()	714
15.29.3 Member Function Documentation	714
15.29.3.1 AttachEvent()	714
15.29.3.2 AttachNode()	714
15.29.3.3 CheckEventID() [1/2]	714
15.29.3.4 CheckEventID() [2/2]	715
15.29.3.5 DetachEvent()	715
15.29.3.6 DetachNode()	715
15.29.3.7 GetAccessMode()	715
15.29.3.8 GetEventIDLength()	715
15.29.3.9 GetPrincipalInterfaceType()	715
15.29.3.10 GetSwapEndianess()	716
15.29.3.11 InvalidateNode()	716
15.29.3.12 Read()	716
15.29.3.13 SetPortImpl()	716
15.29.3.14 Write()	716
15.29.4 Member Data Documentation	716
15.29.4.1 m_pEventPort	717
15.29.4.2 m_pNode	717
15.29.4.3 m_pPortAdapter	717
15.30 CFeatureBag Class Reference	717

15.30.1 Detailed Description	718
15.30.2 Constructor & Destructor Documentation	718
15.30.2.1 CFeatureBag()	718
15.30.2.2 ~CFeatureBag()	718
15.30.3 Member Function Documentation	718
15.30.3.1 GetFeatureBagHandle()	718
15.30.3.2 LoadFromBag()	718
15.30.3.3 operator==()	719
15.30.3.4 PersistFeature()	719
15.30.3.5 SetInfo()	719
15.30.3.6 StoreToBag()	719
15.31 CFloatPtr Class Reference	720
15.31.1 Detailed Description	721
15.31.2 Constructor & Destructor Documentation	721
15.31.2.1 CFloatPtr() [1/2]	721
15.31.2.2 CFloatPtr() [2/2]	721
15.31.3 Member Function Documentation	721
15.31.3.1 GetEnumAlias()	721
15.31.3.2 GetIntAlias()	721
15.31.3.3 operator=()	722
15.32 CGeneric_XMLLoaderParams Class Reference	722
15.32.1 Detailed Description	722
15.32.2 Member Function Documentation	722
15.32.2.1 _Initialize()	723
15.33 CGlobalLock Class Reference	723
15.33.1 Detailed Description	723
15.33.2 Constructor & Destructor Documentation	723
15.33.2.1 CGlobalLock() [1/2]	724
15.33.2.2 CGlobalLock() [2/2]	724
15.33.2.3 ~CGlobalLock()	724

15.33.3 Member Function Documentation	724
15.33.3.1 IsValid()	724
15.33.3.2 Lock()	724
15.33.3.3 TryLock()	725
15.33.3.4 Unlock()	725
15.33.4 Member Data Documentation	725
15.33.4.1 m_DebugCount	725
15.34 CGlobalLocker Class Reference	725
15.34.1 Detailed Description	726
15.34.2 Constructor & Destructor Documentation	726
15.34.2.1 CGlobalLocker()	726
15.34.2.2 ~CGlobalLocker()	726
15.34.3 Member Function Documentation	726
15.34.3.1 UnlockEarly()	726
15.34.4 Member Data Documentation	727
15.34.4.1 m_enabled	727
15.34.4.2 m_Lock	727
15.35 ChunkData Class Reference	727
15.35.1 Detailed Description	729
15.35.2 Constructor & Destructor Documentation	729
15.35.2.1 ChunkData() [1/2]	730
15.35.2.2 ChunkData() [2/2]	730
15.35.2.3 ~ChunkData()	730
15.35.3 Member Function Documentation	730
15.35.3.1 GetBlackLevel()	730
15.35.3.2 GetCompressionMode()	730
15.35.3.3 GetCompressionRatio()	731
15.35.3.4 GetCounterValue()	731
15.35.3.5 GetCRC()	731
15.35.3.6 GetEncoderValue()	731

15.35.3.7 GetExposureEndLineStatusAll()	732
15.35.3.8 GetExposureTime()	732
15.35.3.9 GetFrameID()	732
15.35.3.10 GetGain()	732
15.35.3.11 GetHeight()	733
15.35.3.12 GetImage()	733
15.35.3.13 GetInferenceBoundingBoxResult()	733
15.35.3.14 GetInferenceConfidence()	733
15.35.3.15 GetInferenceFrameId()	734
15.35.3.16 GetInferenceResult()	734
15.35.3.17 GetLinePitch()	734
15.35.3.18 GetLineStatusAll()	734
15.35.3.19 GetOffsetX()	735
15.35.3.20 GetOffsetY()	735
15.35.3.21 GetPartSelector()	735
15.35.3.22 GetPixelDynamicRangeMax()	735
15.35.3.23 GetPixelDynamicRangeMin()	736
15.35.3.24 GetScan3dAxisMax()	736
15.35.3.25 GetScan3dAxisMin()	736
15.35.3.26 GetScan3dCoordinateOffset()	736
15.35.3.27 GetScan3dCoordinateReferenceValue()	737
15.35.3.28 GetScan3dCoordinateScale()	737
15.35.3.29 GetScan3dInvalidDataValue()	737
15.35.3.30 GetScan3dTransformValue()	737
15.35.3.31 GetScanLineSelector()	738
15.35.3.32 GetSequencerSetActive()	738
15.35.3.33 GetSerialDataLength()	738
15.35.3.34 GetStreamChannelID()	738
15.35.3.35 GetTimerValue()	739
15.35.3.36 GetTimestamp()	739

15.35.3.37	GetTimestampLatchValue()	739
15.35.3.38	GetTransferBlockID()	739
15.35.3.39	GetTransferQueueCurrentBlockCount()	740
15.35.3.40	GetWidth()	740
15.35.3.41	SetChunks()	740
15.36	Clock Class Reference	740
15.36.1	Detailed Description	741
15.36.2	Constructor & Destructor Documentation	741
15.36.2.1	Clock()	741
15.36.2.2	~Clock()	741
15.36.3	Member Function Documentation	741
15.36.3.1	Lock()	741
15.36.3.2	TryLock()	742
15.36.3.3	Unlock()	742
15.37	Clock Class Reference	742
15.37.1	Detailed Description	743
15.37.2	Constructor & Destructor Documentation	743
15.37.2.1	Clock() [1/2]	743
15.37.2.2	Clock() [2/2]	743
15.37.2.3	~Clock()	743
15.37.3	Member Function Documentation	743
15.37.3.1	Lock()	744
15.37.3.2	TryLock()	744
15.37.3.3	Unlock()	744
15.37.4	Friends And Related Function Documentation	744
15.37.4.1	NodeMap	744
15.37.5	Member Data Documentation	744
15.37.5.1	m_bOwnLock	744
15.37.5.2	m_lock	744
15.38	ClockEx Class Reference	745

15.38.1 Detailed Description	745
15.39CLOCKEX Class Reference	746
15.39.1 Detailed Description	746
15.39.2 Member Data Documentation	747
15.39.2.1 m_lockEx	747
15.40CNodeCallback Class Reference	747
15.40.1 Detailed Description	748
15.40.2 Constructor & Destructor Documentation	748
15.40.2.1 CNodeCallback()	748
15.40.2.2 ~CNodeCallback()	748
15.40.3 Member Function Documentation	748
15.40.3.1 Destroy()	748
15.40.3.2 GetCallbackType()	749
15.40.3.3 GetNode()	749
15.40.3.4 operator>()	749
15.40.4 Member Data Documentation	749
15.40.4.1 m_CallbackType	749
15.40.4.2 m_pNode	749
15.41CNodeMapFactory Class Reference	750
15.41.1 Detailed Description	751
15.41.2 Constructor & Destructor Documentation	752
15.41.2.1 CNodeMapFactory() [1/5]	752
15.41.2.2 ~CNodeMapFactory()	752
15.41.2.3 CNodeMapFactory() [2/5]	752
15.41.2.4 CNodeMapFactory() [3/5]	752
15.41.2.5 CNodeMapFactory() [4/5]	753
15.41.2.6 CNodeMapFactory() [5/5]	754
15.41.3 Member Function Documentation	754
15.41.3.1 AddInjectionData()	754
15.41.3.2 ApplyStyleSheet()	754

15.41.3.3 ClearCache()	755
15.41.3.4 CreateEmptyNodeMap()	755
15.41.3.5 CreateNodeDataFromNodeMap()	755
15.41.3.6 CreateNodeMap() [1/2]	755
15.41.3.7 CreateNodeMap() [2/2]	755
15.41.3.8 ExtractSubtree()	756
15.41.3.9 GetNodeStatistics()	756
15.41.3.10 GetSupportedSchemaVersions()	756
15.41.3.11 IsCameraDescriptionFileDataReleased()	756
15.41.3.12 IsEmpty()	756
15.41.3.13 IsLoaded()	757
15.41.3.14 IsPreprocessed()	757
15.41.3.15 LoadAndInject()	757
15.41.3.16 operator=()	757
15.41.3.17 Preprocess()	757
15.41.3.18 ReleaseCameraDescriptionFileData()	758
15.41.3.19 ToString()	758
15.41.3.20 ToXml()	758
15.42 CNodeMapRef Class Reference	758
15.42.1 Detailed Description	759
15.42.2 Constructor & Destructor Documentation	759
15.42.2.1 CNodeMapRef() [1/3]	760
15.42.2.2 CNodeMapRef() [2/3]	760
15.42.2.3 CNodeMapRef() [3/3]	760
15.42.3 Member Function Documentation	760
15.42.3.1 operator=() [1/2]	760
15.42.3.2 operator=() [2/2]	760
15.43 CNodeMapRefT< TCameraParams > Class Template Reference	761
15.43.1 Detailed Description	762
15.43.2 Member Function Documentation	763

15.43.2.1 _ClearXMLCache()	763
15.43.2.2 _Connect() [1/2]	763
15.43.2.3 _Connect() [2/2]	763
15.43.2.4 _GetDeviceName()	763
15.43.2.5 _GetNode()	764
15.43.2.6 _GetNodes()	764
15.43.2.7 _GetSupportedSchemaVersions()	764
15.43.2.8 _InvalidateNodes()	764
15.43.2.9 _LoadXMLFromFile()	764
15.43.2.10 _LoadXMLFromFileInject()	765
15.43.2.11 _LoadXMLFromString()	765
15.43.2.12 _LoadXMLFromStringInject()	765
15.43.2.13 _LoadXMLFromZIPData()	765
15.43.2.14 _LoadXMLFromZIPFile()	765
15.43.2.15 _Poll()	766
15.43.3 Member Data Documentation	766
15.43.3.1 _Ptr	766
15.44 CommandNode Class Reference	766
15.44.1 Detailed Description	767
15.44.2 Constructor & Destructor Documentation	767
15.44.2.1 CommandNode() [1/2]	768
15.44.2.2 CommandNode() [2/2]	768
15.44.2.3 ~CommandNode()	768
15.44.3 Member Function Documentation	768
15.44.3.1 Execute()	768
15.44.3.2 IsDone()	768
15.44.3.3 operator()()	769
15.44.3.4 SetReference()	769
15.45 CompressedImageInfo Struct Reference	769
15.45.1 Constructor & Destructor Documentation	769

15.45.1.1 CompressedImageInfo()	770
15.45.2 Member Data Documentation	770
15.45.2.1 compressedImageSize	770
15.45.2.2 fileName	770
15.45.2.3 imageHeight	770
15.45.2.4 imagePixelFormat	770
15.45.2.5 imageWidth	770
15.45.2.6 imageXOffset	771
15.45.2.7 imageYOffset	771
15.46Counter Class Reference	771
15.46.1 Detailed Description	771
15.46.2 Constructor & Destructor Documentation	771
15.46.2.1 Counter()	771
15.46.3 Member Function Documentation	771
15.46.3.1 GetValue()	772
15.46.3.2 IsZero()	772
15.46.3.3 operator unsigned int()	772
15.46.3.4 operator++() [1/2]	772
15.46.3.5 operator++() [2/2]	772
15.46.3.6 operator--() [1/2]	772
15.46.3.7 operator--() [2/2]	772
15.47CPointer< T, B > Class Template Reference	773
15.47.1 Detailed Description	774
15.47.2 Constructor & Destructor Documentation	774
15.47.2.1 CPointer() [1/2]	774
15.47.2.2 CPointer() [2/2]	774
15.47.2.3 ~CPointer()	774
15.47.3 Member Function Documentation	775
15.47.3.1 IsValid()	775
15.47.3.2 operator bool()	775

15.47.3.3 operator T*()	775
15.47.3.4 operator!=() [1/5]	775
15.47.3.5 operator!=() [2/5]	775
15.47.3.6 operator!=() [3/5]	776
15.47.3.7 operator!=() [4/5]	776
15.47.3.8 operator!=() [5/5]	776
15.47.3.9 operator()()	776
15.47.3.10 operator*()	776
15.47.3.11 operator->()	776
15.47.3.12 operator=()	777
15.47.3.13 operator==() [1/3]	777
15.47.3.14 operator==() [2/3]	777
15.47.3.15 operator==() [3/3]	777
15.47.4 Member Data Documentation	777
15.47.4.1 m_pT	777
15.48 CPortImpl Class Reference	778
15.48.1 Detailed Description	779
15.48.2 Constructor & Destructor Documentation	779
15.48.2.1 CPortImpl()	779
15.48.2.2 ~CPortImpl()	779
15.48.3 Member Function Documentation	779
15.48.3.1 GetAccessMode()	779
15.48.3.2 GetSwapEndianess()	780
15.48.3.3 InvalidateNode()	780
15.48.3.4 Read()	780
15.48.3.5 Replay()	780
15.48.3.6 SetPortImpl()	780
15.48.3.7 Write()	781
15.48.4 Member Data Documentation	781
15.48.4.1 m_ptrPort	781

15.49CPortWriteList Class Reference	781
15.49.1 Detailed Description	782
15.49.2 Constructor & Destructor Documentation	782
15.49.2.1 CPortWriteList()	782
15.49.2.2 ~CPortWriteList()	783
15.49.3 Member Function Documentation	783
15.49.3.1 GetCookie()	783
15.49.3.2 GetPortWriteListHandle()	783
15.49.3.3 Replay()	783
15.49.3.4 SetCookie()	783
15.49.3.5 Write()	783
15.49.4 Member Data Documentation	784
15.49.4.1 m_pWriteList	784
15.50CpuUsageInfo Struct Reference	784
15.50.1 Member Data Documentation	784
15.50.1.1 dummy	784
15.51CRegisterPortImpl Class Reference	785
15.51.1 Detailed Description	786
15.51.2 Constructor & Destructor Documentation	786
15.51.2.1 CRegisterPortImpl()	786
15.51.2.2 ~CRegisterPortImpl()	786
15.51.3 Member Function Documentation	786
15.51.3.1 GetAccessMode()	787
15.51.3.2 Read()	787
15.51.3.3 ReadRegister()	787
15.51.3.4 SetPortImpl()	787
15.51.3.5 Write()	788
15.51.3.6 WriteRegister()	788
15.52CSelectorSet Class Reference	788
15.52.1 Detailed Description	789

15.52.2 Constructor & Destructor Documentation	789
15.52.2.1 CSelectorSet()	789
15.52.2.2 ~CSelectorSet()	790
15.52.3 Member Function Documentation	790
15.52.3.1 GetSelectorList()	790
15.52.3.2 IsEmpty()	790
15.52.3.3 Restore()	790
15.52.3.4 SetFirst()	790
15.52.3.5 SetNext()	791
15.52.3.6 ToString()	791
15.53 CTestPortStruct< CDataStruct > Class Template Reference	791
15.53.1 Detailed Description	792
15.53.2 Constructor & Destructor Documentation	793
15.53.2.1 CTestPortStruct()	793
15.53.3 Member Function Documentation	793
15.53.3.1 GetAccessMode()	793
15.53.3.2 GetNumReads()	793
15.53.3.3 GetNumWrites()	793
15.53.3.4 GetPrincipalInterfaceType()	793
15.53.3.5 MemSet()	794
15.53.3.6 Read()	794
15.53.3.7 ResetStatistics()	794
15.53.3.8 Write()	794
15.53.4 Member Data Documentation	794
15.53.4.1 m_BaseAddress	794
15.53.4.2 m_NumReads	795
15.53.4.3 m_NumWrites	795
15.54 DCAM_CHECKSUM Struct Reference	795
15.54.1 Member Data Documentation	795
15.54.1.1 CRCChecksum	795

15.55DCAM_CHUNK_TRAILER Struct Reference	795
15.55.1 Member Data Documentation	796
15.55.1.1 ChunkID	796
15.55.1.2 ChunkLength	796
15.55.1.3 InverseChunkLength	796
15.56DeviceArrivalEventHandler Class Reference	796
15.56.1 Detailed Description	797
15.56.2 Constructor & Destructor Documentation	797
15.56.2.1 DeviceArrivalEventHandler()	798
15.56.2.2 ~DeviceArrivalEventHandler()	798
15.56.3 Member Function Documentation	798
15.56.3.1 OnDeviceArrival()	798
15.56.3.2 operator=()	798
15.57DeviceEventExposureEndData Struct Reference	798
15.57.1 Detailed Description	799
15.57.2 Member Data Documentation	799
15.57.2.1 frameID	799
15.58DeviceEventHandler Class Reference	799
15.58.1 Detailed Description	800
15.58.2 Constructor & Destructor Documentation	800
15.58.2.1 DeviceEventHandler()	801
15.58.2.2 ~DeviceEventHandler()	801
15.58.3 Member Function Documentation	801
15.58.3.1 GetDeviceEventId()	801
15.58.3.2 GetDeviceEventName()	801
15.58.3.3 OnDeviceEvent()	801
15.58.3.4 operator=()	802
15.59DeviceEventHandlerImpl Class Reference	802
15.59.1 Constructor & Destructor Documentation	803
15.59.1.1 DeviceEventHandlerImpl()	803

15.59.1.2 ~DeviceEventHandlerImpl()	803
15.59.2 Member Function Documentation	804
15.59.2.1 OnDeviceEvent()	804
15.60 DeviceEventInferenceData Struct Reference	804
15.60.1 Detailed Description	804
15.60.2 Member Data Documentation	804
15.60.2.1 confidence	804
15.60.2.2 frameID	805
15.60.2.3 result	805
15.61 DeviceEventUtility Class Reference	805
15.61.1 Member Function Documentation	805
15.61.1.1 ParseDeviceEventExposureEnd()	805
15.61.1.2 ParseDeviceEventInference()	806
15.62 DeviceRemovalEventHandler Class Reference	806
15.62.1 Detailed Description	807
15.62.2 Constructor & Destructor Documentation	807
15.62.2.1 DeviceRemovalEventHandler()	808
15.62.2.2 ~DeviceRemovalEventHandler()	808
15.62.3 Member Function Documentation	808
15.62.3.1 OnDeviceRemoval()	808
15.62.3.2 operator=()	808
15.63 double_autovector_t Class Reference	808
15.63.1 Detailed Description	809
15.63.2 Constructor & Destructor Documentation	809
15.63.2.1 double_autovector_t() [1/3]	809
15.63.2.2 double_autovector_t() [2/3]	809
15.63.2.3 double_autovector_t() [3/3]	809
15.63.2.4 ~double_autovector_t()	810
15.63.3 Member Function Documentation	810
15.63.3.1 operator delete()	810

15.63.3.2 operator new()	810
15.63.3.3 operator=()	810
15.63.3.4 operator[]() [1/2]	810
15.63.3.5 operator[]() [2/2]	810
15.63.3.6 size()	811
15.63.4 Member Data Documentation	811
15.63.4.1 _pCount	811
15.63.4.2 _pv	811
15.64EAccessModeClass Class Reference	811
15.64.1 Detailed Description	811
15.64.2 Member Function Documentation	811
15.64.2.1 FromString()	812
15.64.2.2 ToString() [1/2]	812
15.64.2.3 ToString() [2/2]	812
15.65ECachingModeClass Class Reference	812
15.65.1 Detailed Description	812
15.65.2 Member Function Documentation	813
15.65.2.1 FromString()	813
15.65.2.2 ToString() [1/2]	813
15.65.2.3 ToString() [2/2]	813
15.66EDisplayNotationClass Class Reference	813
15.66.1 Detailed Description	814
15.66.2 Member Function Documentation	814
15.66.2.1 FromString()	814
15.66.2.2 ToString() [1/2]	814
15.66.2.3 ToString() [2/2]	814
15.67EEndianessClass Class Reference	814
15.67.1 Detailed Description	815
15.67.2 Member Function Documentation	815
15.67.2.1 FromString()	815

15.67.2.2 ToString() [1/2]	815
15.67.2.3 ToString() [2/2]	815
15.68EGenApiSchemaVersionClass Class Reference	816
15.68.1 Detailed Description	816
15.68.2 Member Function Documentation	816
15.68.2.1 FromString()	816
15.68.2.2 ToString() [1/2]	816
15.68.2.3 ToString() [2/2]	816
15.69EInputDirectionClass Class Reference	817
15.69.1 Detailed Description	817
15.69.2 Member Function Documentation	817
15.69.2.1 FromString()	817
15.69.2.2 ToString() [1/2]	817
15.69.2.3 ToString() [2/2]	817
15.70ENameSpaceClass Class Reference	818
15.70.1 Detailed Description	818
15.70.2 Member Function Documentation	818
15.70.2.1 FromString()	818
15.70.2.2 ToString() [1/2]	818
15.70.2.3 ToString() [2/2]	818
15.71EnumEntryNode Class Reference	819
15.71.1 Detailed Description	820
15.71.2 Constructor & Destructor Documentation	820
15.71.2.1 EnumEntryNode() [1/2]	820
15.71.2.2 EnumEntryNode() [2/2]	820
15.71.2.3 ~EnumEntryNode()	820
15.71.3 Member Function Documentation	820
15.71.3.1 GetNumericValue()	821
15.71.3.2 GetSymbolic()	821
15.71.3.3 GetValue()	821

15.71.3.4 IsSelfClearing()	821
15.71.3.5 SetReference()	821
15.72EnumNode Class Reference	822
15.72.1 Detailed Description	824
15.72.2 Constructor & Destructor Documentation	824
15.72.2.1 EnumNode() [1/2]	824
15.72.2.2 EnumNode() [2/2]	824
15.72.2.3 ~EnumNode()	824
15.72.3 Member Function Documentation	824
15.72.3.1 GetCurrentEntry()	824
15.72.3.2 GetEntries()	825
15.72.3.3 GetEntry()	825
15.72.3.4 GetEntryByName()	825
15.72.3.5 GetIntValue()	825
15.72.3.6 GetSymbolics()	826
15.72.3.7 operator*()	826
15.72.3.8 operator=()	826
15.72.3.9 SetIntValue()	826
15.72.3.10SetReference()	826
15.72.4 Member Data Documentation	827
15.72.4.1 m_pEnumeration	827
15.73ERepresentationClass Class Reference	827
15.73.1 Detailed Description	827
15.73.2 Member Function Documentation	827
15.73.2.1 FromString()	828
15.73.2.2 ToString() [1/2]	828
15.73.2.3 ToString() [2/2]	828
15.74ESignClass Class Reference	828
15.74.1 Detailed Description	828
15.74.2 Member Function Documentation	829

15.74.2.1 FromString()	829
15.74.2.2 ToString() [1/2]	829
15.74.2.3 ToString() [2/2]	829
15.75ESlopeClass Class Reference	829
15.75.1 Detailed Description	830
15.75.2 Member Function Documentation	830
15.75.2.1 FromString()	830
15.75.2.2 ToString() [1/2]	830
15.75.2.3 ToString() [2/2]	830
15.76EStandardNameSpaceClass Class Reference	830
15.76.1 Detailed Description	831
15.76.2 Member Function Documentation	831
15.76.2.1 FromString()	831
15.76.2.2 ToString() [1/2]	831
15.76.2.3 ToString() [2/2]	831
15.77EventHandler Class Reference	832
15.77.1 Detailed Description	833
15.77.2 Constructor & Destructor Documentation	833
15.77.2.1 ~EventHandler()	833
15.77.2.2 EventHandler()	833
15.77.3 Member Function Documentation	833
15.77.3.1 GetEventPayloadData()	833
15.77.3.2 GetEventPayloadDataSize()	833
15.77.3.3 GetEventType()	834
15.77.3.4 operator=()	834
15.77.3.5 SetEventPayload()	834
15.77.3.6 SetEventType()	834
15.77.4 Friends And Related Function Documentation	834
15.77.4.1 EventProcessor	834
15.77.4.2 IDataStream	835

15.77.4.3 Stream	835
15.77.5 Member Data Documentation	835
15.77.5.1 m_pEventData	835
15.78EVisibilityClass Class Reference	835
15.78.1 Detailed Description	835
15.78.2 Member Function Documentation	835
15.78.2.1 FromString()	836
15.78.2.2 ToString() [1/2]	836
15.78.2.3 ToString() [2/2]	836
15.79Exception Class Reference	836
15.79.1 Detailed Description	837
15.79.2 Constructor & Destructor Documentation	838
15.79.2.1 Exception() [1/4]	838
15.79.2.2 Exception() [2/4]	838
15.79.2.3 Exception() [3/4]	838
15.79.2.4 Exception() [4/4]	839
15.79.2.5 ~Exception()	839
15.79.3 Member Function Documentation	839
15.79.3.1 GetBuildDate()	839
15.79.3.2 GetBuildTime()	839
15.79.3.3 GetError()	839
15.79.3.4 GetErrorMessage()	840
15.79.3.5 GetFileName()	840
15.79.3.6 GetFullErrorMessage()	840
15.79.3.7 GetFunctionName()	840
15.79.3.8 GetLineNumber()	840
15.79.3.9 operator!=(())	840
15.79.3.10operator=(())	841
15.79.3.11operator==(())	841
15.79.3.12what()	841

15.80EYesNoClass Class Reference	841
15.80.1 Detailed Description	841
15.80.2 Member Function Documentation	842
15.80.2.1 FromString()	842
15.80.2.2 ToString() [1/2]	842
15.80.2.3 ToString() [2/2]	842
15.81FileProtocolAdapter Class Reference	842
15.81.1 Detailed Description	843
15.81.2 Constructor & Destructor Documentation	843
15.81.2.1 FileProtocolAdapter()	843
15.81.2.2 ~FileProtocolAdapter()	843
15.81.3 Member Function Documentation	843
15.81.3.1 attach()	843
15.81.3.2 closeFile()	844
15.81.3.3 deleteFile()	844
15.81.3.4 getBufSize()	844
15.81.3.5 openFile()	845
15.81.3.6 read()	845
15.81.3.7 write()	846
15.82FloatNode Class Reference	846
15.82.1 Detailed Description	848
15.82.2 Constructor & Destructor Documentation	849
15.82.2.1 FloatNode() [1/2]	849
15.82.2.2 FloatNode() [2/2]	849
15.82.2.3 ~FloatNode()	849
15.82.3 Member Function Documentation	849
15.82.3.1 GetDisplayNotation()	849
15.82.3.2 GetDisplayPrecision()	849
15.82.3.3 GetEnumAlias()	849
15.82.3.4 GetInc()	850

15.82.3.5 GetIncMode()	850
15.82.3.6 GetIntAlias()	850
15.82.3.7 GetListOfValidValues()	850
15.82.3.8 GetMax()	850
15.82.3.9 GetMin()	850
15.82.3.10 GetRepresentation()	851
15.82.3.11 GetUnit()	851
15.82.3.12 GetValue()	851
15.82.3.13 HasInc()	851
15.82.3.14 ImposeMax()	851
15.82.3.15 ImposeMin()	852
15.82.3.16 operator()()	852
15.82.3.17 operator*()	852
15.82.3.18 operator=()	852
15.82.3.19 SetReference()	852
15.82.3.20 SetValue()	852
15.83 FloatRegNode Class Reference	853
15.83.1 Detailed Description	854
15.83.2 Constructor & Destructor Documentation	854
15.83.2.1 FloatRegNode() [1/2]	855
15.83.2.2 FloatRegNode() [2/2]	855
15.83.2.3 ~FloatRegNode()	855
15.83.3 Member Function Documentation	855
15.83.3.1 SetReference()	855
15.84 Function_NodeCallback< Function > Class Template Reference	856
15.84.1 Detailed Description	856
15.84.2 Constructor & Destructor Documentation	857
15.84.2.1 Function_NodeCallback()	857
15.84.3 Member Function Documentation	857
15.84.3.1 Destroy()	857

15.84.3.2 operator()	857
15.85gcstring Class Reference	858
15.85.1 Constructor & Destructor Documentation	859
15.85.1.1 gcstring() [1/5]	859
15.85.1.2 gcstring() [2/5]	859
15.85.1.3 gcstring() [3/5]	859
15.85.1.4 gcstring() [4/5]	859
15.85.1.5 gcstring() [5/5]	860
15.85.1.6 ~gcstring()	860
15.85.2 Member Function Documentation	860
15.85.2.1 _npos()	860
15.85.2.2 append() [1/2]	860
15.85.2.3 append() [2/2]	860
15.85.2.4 assign() [1/4]	860
15.85.2.5 assign() [2/4]	861
15.85.2.6 assign() [3/4]	861
15.85.2.7 assign() [4/4]	861
15.85.2.8 c_str()	861
15.85.2.9 compare()	861
15.85.2.10empty()	861
15.85.2.11find() [1/5]	862
15.85.2.12find() [2/5]	862
15.85.2.13find() [3/5]	862
15.85.2.14find() [4/5]	862
15.85.2.15find() [5/5]	862
15.85.2.16find_first_not_of()	862
15.85.2.17find_first_of()	863
15.85.2.18length()	863
15.85.2.19max_size()	863
15.85.2.20operator const char *()	863

15.85.2.21operator delete() [1/2]	863
15.85.2.22operator delete() [2/2]	863
15.85.2.23operator new() [1/2]	863
15.85.2.24operator new() [2/2]	864
15.85.2.25operator !=() [1/2]	864
15.85.2.26operator !=() [2/2]	864
15.85.2.27operator +=() [1/5]	864
15.85.2.28operator +=() [2/5]	864
15.85.2.29operator +=() [3/5]	864
15.85.2.30operator +=() [4/5]	864
15.85.2.31operator +=() [5/5]	865
15.85.2.32operator <()	865
15.85.2.33operator =()	865
15.85.2.34operator ==() [1/2]	865
15.85.2.35operator ==() [2/2]	865
15.85.2.36operator >()	865
15.85.2.37resize()	865
15.85.2.38size()	866
15.85.2.39substr()	866
15.85.2.40swap()	866
15.85.3 Friends And Related Function Documentation	866
15.85.3.1 operator+ [1/3]	866
15.85.3.2 operator+ [2/3]	866
15.85.3.3 operator+ [3/3]	866
15.85.4 Member Data Documentation	867
15.85.4.1 npos	867
15.86GrabInfo Struct Reference	867
15.86.1 Constructor & Destructor Documentation	867
15.86.1.1 GrabInfo()	867
15.86.2 Member Data Documentation	867

15.86.2.1 imageEventHandler	867
15.86.2.2 numImagesGrabbed	868
15.86.2.3 numIncompleteImages	868
15.86.2.4 numRemovals	868
15.87GVCP_CHUNK_TRAILER Struct Reference	868
15.87.1 Detailed Description	868
15.87.2 Member Data Documentation	868
15.87.2.1 ChunkID	868
15.87.2.2 ChunkLength	869
15.88GVCP_EVENT_ITEM Struct Reference	869
15.88.1 Detailed Description	869
15.88.2 Member Data Documentation	869
15.88.2.1 BlockId	869
15.88.2.2 EventId	869
15.88.2.3 ReservedOrEventSize	870
15.88.2.4 StreamChannelId	870
15.88.2.5 TimestampHigh	870
15.88.2.6 TimestampLow	870
15.89GVCP_EVENT_ITEM_BASIC Struct Reference	870
15.89.1 Detailed Description	870
15.89.2 Member Data Documentation	870
15.89.2.1 EventId	871
15.89.2.2 ReservedOrEventSize	871
15.90GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference	871
15.90.1 Detailed Description	871
15.90.2 Member Data Documentation	871
15.90.2.1 BlockId	871
15.90.2.2 BlockId64High	872
15.90.2.3 BlockId64Low	872
15.90.2.4 EventId	872

15.90.2.5 ReservedOrEventSize	872
15.90.2.6 StreamChannelId	872
15.90.2.7 TimestampHigh	872
15.90.2.8 TimestampLow	872
15.91GVCP_EVENT_REQUEST Struct Reference	873
15.91.1 Detailed Description	873
15.91.2 Member Data Documentation	873
15.91.2.1 Header	873
15.91.2.2 Items	873
15.92GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference	874
15.92.1 Detailed Description	874
15.92.2 Member Data Documentation	874
15.92.2.1 Header	874
15.92.2.2 Items	874
15.93GVCP_EVENTDATA_REQUEST Struct Reference	875
15.93.1 Detailed Description	875
15.93.2 Member Data Documentation	875
15.93.2.1 Data	875
15.93.2.2 Event	875
15.93.2.3 Header	876
15.94GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference	876
15.94.1 Detailed Description	876
15.94.2 Member Data Documentation	876
15.94.2.1 Data	876
15.94.2.2 Event	877
15.94.2.3 Header	877
15.95GVCP_REQUEST_HEADER Struct Reference	877
15.95.1 Detailed Description	877
15.95.2 Member Data Documentation	877
15.95.2.1 Command	877

15.95.2.2 Flags	877
15.95.2.3 Length	878
15.95.2.4 Magic	878
15.95.2.5 ReqId	878
15.96H264Option Struct Reference	878
15.96.1 Detailed Description	878
15.96.2 Constructor & Destructor Documentation	879
15.96.2.1 H264Option()	879
15.96.3 Member Data Documentation	879
15.96.3.1 bitrate	879
15.96.3.2 frameRate	879
15.96.3.3 height	879
15.96.3.4 reserved	879
15.96.3.5 width	880
15.97ICameraBase Class Reference	880
15.97.1 Detailed Description	882
15.97.2 Constructor & Destructor Documentation	882
15.97.2.1 ~ICameraBase()	882
15.97.2.2 ICameraBase() [1/2]	882
15.97.2.3 ICameraBase() [2/2]	882
15.97.3 Member Function Documentation	882
15.97.3.1 BeginAcquisition()	882
15.97.3.2 DeInit()	883
15.97.3.3 DiscoverMaxPacketSize()	883
15.97.3.4 EndAcquisition()	883
15.97.3.5 ForceIP()	883
15.97.3.6 GetAccessMode()	883
15.97.3.7 GetBufferOwnership()	883
15.97.3.8 GetGuiXml()	884
15.97.3.9 GetNextImage()	884

15.97.3.10	GetNodeMap()	884
15.97.3.11	GetNumDataStreams()	884
15.97.3.12	GetNumImagesInUse()	884
15.97.3.13	GetTLDeviceNodeMap()	884
15.97.3.14	GetTLStreamNodeMap()	885
15.97.3.15	GetUniqueID()	885
15.97.3.16	GetUserBufferCount()	885
15.97.3.17	GetUserBufferSize()	885
15.97.3.18	GetUserBufferTotalSize()	885
15.97.3.19	init()	885
15.97.3.20	IsInitialized()	886
15.97.3.21	IsStreaming()	886
15.97.3.22	IsValid()	886
15.97.3.23	operator=()	886
15.97.3.24	ReadPort()	886
15.97.3.25	RegisterEventHandler() [1/2]	886
15.97.3.26	RegisterEventHandler() [2/2]	887
15.97.3.27	SetBufferOwnership()	887
15.97.3.28	SetUserBuffers() [1/2]	887
15.97.3.29	SetUserBuffers() [2/2]	887
15.97.3.30	UnregisterEventHandler()	887
15.97.3.31	WritePort()	888
15.97.4	Friends And Related Function Documentation	888
15.97.4.1	CameraInternal	888
15.97.4.2	InterfaceImpl	888
15.97.5	Member Data Documentation	888
15.97.5.1	m_pCameraBaseData	888
15.97.5.2	TLDevice	888
15.97.5.3	TLStream	889
15.98	ICameraList Class Reference	889

15.98.1 Detailed Description	890
15.98.2 Constructor & Destructor Documentation	890
15.98.2.1 ~ICameraList()	890
15.98.2.2 ICameraList() [1/2]	890
15.98.2.3 ICameraList() [2/2]	890
15.98.3 Member Function Documentation	890
15.98.3.1 Append()	890
15.98.3.2 Clear()	891
15.98.3.3 GetByDeviceID()	891
15.98.3.4 GetByIndex()	891
15.98.3.5 GetBySerial()	891
15.98.3.6 GetSize()	891
15.98.3.7 operator=()	891
15.98.3.8 operator[]()	892
15.98.3.9 RemoveByDeviceID()	892
15.98.3.10 RemoveByIndex()	892
15.98.3.11 RemoveBySerial()	892
15.98.4 Friends And Related Function Documentation	892
15.98.4.1 CameraListImpl	892
15.98.4.2 InterfaceImpl	892
15.98.5 Member Data Documentation	893
15.98.5.1 m_pCameraListData	893
15.99 IChunkData Class Reference	893
15.99.1 Detailed Description	894
15.99.2 Constructor & Destructor Documentation	894
15.99.2.1 ~IChunkData()	894
15.99.2.2 IChunkData()	894
15.99.3 Member Function Documentation	895
15.99.3.1 GetBlackLevel()	895
15.99.3.2 GetCompressionMode()	895

15.99.3.3 GetCompressionRatio()	895
15.99.3.4 GetCounterValue()	895
15.99.3.5 GetCRC()	895
15.99.3.6 GetEncoderValue()	896
15.99.3.7 GetExposureEndLineStatusAll()	896
15.99.3.8 GetExposureTime()	896
15.99.3.9 GetFrameID()	896
15.99.3.10 GetGain()	896
15.99.3.11 GetHeight()	896
15.99.3.12 GetImage()	897
15.99.3.13 GetInferenceBoundingBoxResult()	897
15.99.3.14 GetInferenceConfidence()	897
15.99.3.15 GetInferenceFrameId()	897
15.99.3.16 GetInferenceResult()	897
15.99.3.17 GetLinePitch()	897
15.99.3.18 GetLineStatusAll()	898
15.99.3.19 GetOffsetX()	898
15.99.3.20 GetOffsetY()	898
15.99.3.21 GetPartSelector()	898
15.99.3.22 GetPixelDynamicRangeMax()	898
15.99.3.23 GetPixelDynamicRangeMin()	898
15.99.3.24 GetScan3dAxisMax()	899
15.99.3.25 GetScan3dAxisMin()	899
15.99.3.26 GetScan3dCoordinateOffset()	899
15.99.3.27 GetScan3dCoordinateReferenceValue()	899
15.99.3.28 GetScan3dCoordinateScale()	899
15.99.3.29 GetScan3dInvalidDataValue()	899
15.99.3.30 GetScan3dTransformValue()	900
15.99.3.31 GetScanLineSelector()	900
15.99.3.32 GetSequencerSetActive()	900

15.99.3.33	GetSerialDataLength()	900
15.99.3.34	GetStreamChannelID()	900
15.99.3.35	GetTimerValue()	900
15.99.3.36	GetTimestamp()	901
15.99.3.37	GetTimestampLatchValue()	901
15.99.3.38	GetTransferBlockID()	901
15.99.3.39	GetTransferQueueCurrentBlockCount()	901
15.99.3.40	GetWidth()	901
15.99.3.41	SetChunks()	901
15.100	DataStream Class Reference	902
15.100.1	Constructor & Destructor Documentation	902
15.100.1.1	~IDataStream()	903
15.100.1.2	IDataStream()	903
15.100.2	Member Function Documentation	903
15.100.2.1	AnnounceImage() [1/3]	903
15.100.2.2	AnnounceImage() [2/3]	903
15.100.2.3	AnnounceImage() [3/3]	903
15.100.2.4	AttachBuffer()	903
15.100.2.5	CleanupChunkAdapter()	904
15.100.2.6	FlushQueueAllDiscard()	904
15.100.2.7	GetBufferChunkData()	904
15.100.2.8	GetBufferInfoBool8Type()	904
15.100.2.9	GetBufferInfoPtrType()	904
15.100.2.10	GetBufferInfoSizeType()	904
15.100.2.11	GetBufferInfoUInt64Type()	905
15.100.2.12	GetDeviceNodeMap()	905
15.100.2.13	GetNextImage()	905
15.100.2.14	GetNextImageInternal()	905
15.100.2.15	GetNodeMap()	905
15.100.2.16	GetNumImagesInUse()	905

15.100.2.17	GetPort()	905
15.100.2.18	GetStreamInfoBool8Type()	906
15.100.2.19	GetStreamInfoSizeType()	906
15.100.2.20	GetStreamType()	906
15.100.2.21	HitChunkAdapter()	906
15.100.2.22	CRCCheckEnabled()	906
15.100.2.23	ImageInUse()	906
15.100.2.24	Streaming()	906
15.100.2.25	KillBufferEvent()	907
15.100.2.26	RegisterImageEventHandler()	907
15.100.2.27	ReleaseImage()	907
15.100.2.28	ResetStreamHandle()	907
15.100.2.29	RevokeImages()	907
15.100.2.30	StartStream()	907
15.100.2.31	StopStream()	907
15.100.2.32	TransportLayerStreamInfo()	908
15.100.2.33	UnregisterImageEventHandler()	908
15.100.2.34	WaitOnImageEvent()	908
15.101	DevFileStreamBase< CharType, Traits > Class Template Reference	908
15.101.1	Member Typedef Documentation	909
15.101.1.1	filebuf_type	909
15.101.1.2	ios_type	909
15.101.1.3	stream_type	910
15.101.2	Member Function Documentation	910
15.101.2.1	close()	910
15.101.2.2	is_open()	910
15.101.2.3	open()	910
15.101.2.4	rdbuf()	910
15.102	DevFileStreamBuf< CharType, Traits > Class Template Reference	911
15.102.1	Constructor & Destructor Documentation	912

15.102.1.1	DevFileStreamBuf()	912
15.102.1.2	~IDevFileStreamBuf()	912
15.102.2	Member Function Documentation	912
15.102.2.1	close()	912
15.102.2.2	is_open()	912
15.102.2.3	open()	912
15.102.2.4	backfail()	912
15.102.2.5	underflow()	913
15.103	IDeviceArrivalEventHandler Class Reference	913
15.103.1	Constructor & Destructor Documentation	914
15.103.1.1	~IDeviceArrivalEventHandler()	914
15.103.1.2	IDeviceArrivalEventHandler() [1/2]	914
15.103.1.3	IDeviceArrivalEventHandler() [2/2]	914
15.103.2	Member Function Documentation	914
15.103.2.1	OnDeviceArrival()	914
15.103.2.2	operator=()	915
15.104	IDeviceEventHandler Class Reference	915
15.104.1	Constructor & Destructor Documentation	916
15.104.1.1	~IDeviceEventHandler()	916
15.104.1.2	IDeviceEventHandler() [1/2]	916
15.104.1.3	IDeviceEventHandler() [2/2]	916
15.104.2	Member Function Documentation	916
15.104.2.1	GetDeviceEventId()	916
15.104.2.2	GetDeviceEventName()	917
15.104.2.3	OnDeviceEvent()	917
15.104.2.4	operator=()	917
15.105	IDeviceRemovalEventHandler Class Reference	917
15.105.1	Constructor & Destructor Documentation	918
15.105.1.1	~IDeviceRemovalEventHandler()	918
15.105.1.2	IDeviceRemovalEventHandler() [1/2]	918

15.105.1.3DeviceRemovalEventHandler() [2/2]	919
15.105.2Member Function Documentation	919
15.105.2.1OnDeviceRemoval()	919
15.105.2.2operator=()	919
15.106Image Class Reference	919
15.106.1Detailed Description	921
15.106.2Constructor & Destructor Documentation	921
15.106.2.1~Image()	921
15.106.2.2Image()	921
15.106.3Member Function Documentation	921
15.106.3.1CalculateStatistics()	921
15.106.3.2CheckCRC()	921
15.106.3.3Convert() [1/2]	922
15.106.3.4Convert() [2/2]	922
15.106.3.5DeepCopy()	922
15.106.3.6GetBitsPerPixel()	922
15.106.3.7GetBufferSize()	922
15.106.3.8GetChunkData()	923
15.106.3.9GetChunkLayoutId()	923
15.106.3.10GetColorProcessing()	923
15.106.3.11GetData()	923
15.106.3.12GetDataAbsoluteMax()	923
15.106.3.13GetDataAbsoluteMin()	923
15.106.3.14GetFrameID()	924
15.106.3.15GetHeight()	924
15.106.3.16GetID()	924
15.106.3.17GetImageData()	924
15.106.3.18GetImageSize()	924
15.106.3.19GetImageStatus()	924
15.106.3.20GetNumChannels()	925

15.106.3.21	GetPayloadType()	925
15.106.3.22	GetPixelFormat()	925
15.106.3.23	GetPixelFormatIntType()	925
15.106.3.24	GetPixelFormatName()	925
15.106.3.25	SetPrivateData()	925
15.106.3.26	GetStride()	926
15.106.3.27	GetTimeStamp()	926
15.106.3.28	GetTLPayloadType()	926
15.106.3.29	GetTLPixelFormat()	926
15.106.3.30	GetTLPixelFormatNamespace()	926
15.106.3.31	IsValidPayloadSize()	926
15.106.3.32	GetWidth()	927
15.106.3.33	GetXOffset()	927
15.106.3.34	GetXPadding()	927
15.106.3.35	GetYOffset()	927
15.106.3.36	GetYPadding()	927
15.106.3.37	HasCRC()	927
15.106.3.38	Compressed()	928
15.106.3.39	Incomplete()	928
15.106.3.40	InUse()	928
15.106.3.41	Release()	928
15.106.3.42	ResetImage() [1/3]	928
15.106.3.43	ResetImage() [2/3]	929
15.106.3.44	ResetImage() [3/3]	929
15.106.3.45	Save() [1/8]	929
15.106.3.46	Save() [2/8]	929
15.106.3.47	Save() [3/8]	930
15.106.3.48	Save() [4/8]	930
15.106.3.49	Save() [5/8]	930
15.106.3.50	Save() [6/8]	930

15.106.3.53	Save() [7/8]	930
15.106.3.52	Save() [8/8]	931
15.106.4	Friends And Related Function Documentation	931
15.106.4.1	Stream	931
15.107	ImageEventHandler Class Reference	931
15.107.1	Constructor & Destructor Documentation	932
15.107.1.1	~ImageEventHandler()	932
15.107.1.2	ImageEventHandler() [1/2]	932
15.107.1.3	ImageEventHandler() [2/2]	933
15.107.2	Member Function Documentation	933
15.107.2.1	OnImageEvent()	933
15.107.2.2	operator=()	933
15.108	ImageStatistics Class Reference	933
15.108.1	Detailed Description	934
15.108.2	Constructor & Destructor Documentation	934
15.108.2.1	~ImageStatistics()	934
15.108.2.2	ImageStatistics() [1/2]	934
15.108.2.3	ImageStatistics() [2/2]	935
15.108.3	Member Function Documentation	935
15.108.3.1	DisableAll()	935
15.108.3.2	EnableAll()	935
15.108.3.3	EnableGreyOnly()	935
15.108.3.4	EnableHSLOnly()	935
15.108.3.5	EnableRGBOnly()	935
15.108.3.6	GetChannelStatus()	936
15.108.3.7	GetHistogram()	936
15.108.3.8	GetMean()	936
15.108.3.9	GetNumPixelValues()	936
15.108.3.10	GetPixelValueRange()	936
15.108.3.11	GetRange()	937

15.108.3.10	GetStatistics()	937
15.108.3.11	SetChannelStatus()	937
15.109	Interface Class Reference	937
15.109.1	Detailed Description	939
15.109.2	Constructor & Destructor Documentation	939
15.109.2.1	~IInterface()	939
15.109.2.2	IInterface() [1/2]	939
15.109.2.3	IInterface() [2/2]	939
15.109.3	Member Function Documentation	939
15.109.3.1	GetCameras()	940
15.109.3.2	GetTLNodeMap()	940
15.109.3.3	IsInUse()	940
15.109.3.4	IsValid()	940
15.109.3.5	operator=()	940
15.109.3.6	RegisterEventHandler()	940
15.109.3.7	SendActionCommand()	941
15.109.3.8	UnregisterEventHandler()	941
15.109.3.9	UpdateCameras()	941
15.109.4	Friends And Related Function Documentation	941
15.109.4.1	InterfaceInternal	941
15.109.4.2	ProducerImpl	941
15.109.5	Member Data Documentation	941
15.109.5.1	m_pInterfaceData	942
15.109.5.2	TLInterface	942
15.110	InterfaceArrivalEventHandler Class Reference	942
15.110.1	Constructor & Destructor Documentation	943
15.110.1.1	~InterfaceArrivalEventHandler()	943
15.110.1.2	InterfaceArrivalEventHandler() [1/2]	943
15.110.1.3	InterfaceArrivalEventHandler() [2/2]	944
15.110.2	Member Function Documentation	944

15.110.2.1	OnInterfaceArrival()	944
15.110.2.2	operator=()	944
15.111	InterfaceEventHandler Class Reference	944
15.111.1	Constructor & Destructor Documentation	945
15.111.1.1	~InterfaceEventHandler()	945
15.111.1.2	InterfaceEventHandler() [1/2]	945
15.111.1.3	InterfaceEventHandler() [2/2]	946
15.111.2	Member Function Documentation	946
15.111.2.1	OnDeviceArrival()	946
15.111.2.2	OnDeviceRemoval()	946
15.111.2.3	operator=()	946
15.112	InterfaceList Class Reference	947
15.112.1	Detailed Description	947
15.112.2	Constructor & Destructor Documentation	948
15.112.2.1	~InterfaceList()	948
15.112.2.2	InterfaceList() [1/2]	948
15.112.2.3	InterfaceList() [2/2]	948
15.112.3	Member Function Documentation	948
15.112.3.1	Append()	948
15.112.3.2	Clear()	948
15.112.3.3	GetByIndex()	949
15.112.3.4	GetSize()	949
15.112.3.5	operator=()	949
15.112.3.6	operator[]()	949
15.112.4	Friends And Related Function Documentation	949
15.112.4.1	InterfaceListImpl	949
15.112.5	Member Data Documentation	949
15.112.5.1	m_pInterfaceListData	950
15.113	InterfaceRemovalEventHandler Class Reference	950
15.113.1	Constructor & Destructor Documentation	951

15.113.1.1~IInterfaceRemovalEventHandler()	951
15.113.1.2IInterfaceRemovalEventHandler() [1/2]	951
15.113.1.3IInterfaceRemovalEventHandler() [2/2]	951
15.113.2Member Function Documentation	951
15.113.2.1OnInterfaceRemoval()	951
15.113.2.2operator=()	952
15.114LoggingEventHandler Class Reference	952
15.114.1Constructor & Destructor Documentation	953
15.114.1.1~ILoggingEventHandler()	953
15.114.1.2LoggingEventHandler() [1/2]	953
15.114.1.3LoggingEventHandler() [2/2]	953
15.114.2Member Function Documentation	953
15.114.2.1OnLogEvent()	953
15.114.2.2operator=()	954
15.115Image Class Reference	954
15.115.1Detailed Description	958
15.115.2Constructor & Destructor Documentation	958
15.115.2.1~Image()	958
15.115.2.2Image() [1/4]	958
15.115.2.3Image() [2/4]	958
15.115.2.4Image() [3/4]	958
15.115.2.5Image() [4/4]	959
15.115.3Member Function Documentation	959
15.115.3.1CalculateStatistics()	959
15.115.3.2CheckCRC()	959
15.115.3.3Convert() [1/3]	960
15.115.3.4Convert() [2/3]	960
15.115.3.5Convert() [3/3]	961
15.115.3.6Create() [1/4]	961
15.115.3.7Create() [2/4]	961

15.115.3.8Create() [3/4]	961
15.115.3.9Create() [4/4]	962
15.115.3.10CreateShared()	963
15.115.3.11DeepCopy() [1/2]	963
15.115.3.12DeepCopy() [2/2]	963
15.115.3.13GetBitsPerPixel()	963
15.115.3.14GetBufferSize()	964
15.115.3.15GetChunkData()	964
15.115.3.16GetChunkLayoutId()	964
15.115.3.17GetColorProcessing()	965
15.115.3.18GetData()	965
15.115.3.19GetDataAbsoluteMax()	965
15.115.3.20GetDataAbsoluteMin()	966
15.115.3.21GetDefaultColorProcessing()	966
15.115.3.22GetFrameId()	966
15.115.3.23GetHeight()	967
15.115.3.24GetId()	967
15.115.3.25SetImageData()	967
15.115.3.26GetImageSize()	967
15.115.3.27GetImageStatus()	968
15.115.3.28GetImageStatusDescription()	968
15.115.3.29GetNumChannels()	968
15.115.3.30GetNumDecompressionThreads()	969
15.115.3.31GetPayloadType()	969
15.115.3.32GetPixelFormat()	969
15.115.3.33GetPixelFormatIntType()	970
15.115.3.34GetPixelFormatName()	970
15.115.3.35SetPrivateData()	970
15.115.3.36GetStride()	971
15.115.3.37GetTimeStamp()	971

15.115.3.38	GetTLPayloadType()	971
15.115.3.39	GetTLPixelFormat()	972
15.115.3.40	GetTLPixelFormatNamespace()	972
15.115.3.41	GetValidPayloadSize()	973
15.115.3.42	GetWidth()	973
15.115.3.43	GetXOffset()	973
15.115.3.44	GetXPadding()	974
15.115.3.45	GetYOffset()	974
15.115.3.46	GetYPadding()	974
15.115.3.47	HasCRC()	975
15.115.3.48	Compressed()	975
15.115.3.49	Incomplete()	975
15.115.3.50	InUse()	976
15.115.3.51	Release()	976
15.115.3.52	ResetImage() [1/3]	976
15.115.3.53	ResetImage() [2/3]	976
15.115.3.54	ResetImage() [3/3]	977
15.115.3.55	Save() [1/8]	978
15.115.3.56	Save() [2/8]	978
15.115.3.57	Save() [3/8]	978
15.115.3.58	Save() [4/8]	979
15.115.3.59	Save() [5/8]	979
15.115.3.60	Save() [6/8]	979
15.115.3.61	Save() [7/8]	980
15.115.3.62	Save() [8/8]	980
15.115.3.63	SetDefaultColorProcessing()	980
15.115.3.64	SetNumDecompressionThreads()	981
15.115.4	Friends And Related Function Documentation	981
15.115.4.1	DataStream	981
15.115.4.2	ImageConverter	981

15.115.4.3ImageConverterIpp	982
15.115.4.4ImageFiler	982
15.115.4.5ImageStatsCalculator	982
15.115.4.6ImageUtilityImpl	982
15.115.4.7ImageUtilityPolarizationImpl	982
15.115.4.8Stream	982
15.116ImageEventHandler Class Reference	983
15.116.1Detailed Description	984
15.116.2Constructor & Destructor Documentation	984
15.116.2.1ImageEventHandler()	984
15.116.2.2~ImageEventHandler()	984
15.116.3Member Function Documentation	984
15.116.3.1OnImageEvent()	984
15.116.3.2operator=()	985
15.117ImageEventHandlerImpl Class Reference	985
15.117.1Constructor & Destructor Documentation	986
15.117.1.1ImageEventHandlerImpl() [1/2]	986
15.117.1.2~ImageEventHandlerImpl() [1/2]	987
15.117.1.3ImageEventHandlerImpl() [2/2]	987
15.117.1.4~ImageEventHandlerImpl() [2/2]	987
15.117.2Member Function Documentation	987
15.117.2.1getImageCount()	987
15.117.2.2getMaxImages()	987
15.117.2.3OnImageEvent() [1/2]	987
15.117.2.4OnImageEvent() [2/2]	988
15.118ImageInfo Struct Reference	988
15.118.1Constructor & Destructor Documentation	988
15.118.1.1ImageInfo()	988
15.118.2Member Data Documentation	988
15.118.2.1imageFile	989

15.118.2.2	imageFileName	989
15.118.2.3	imageHeight	989
15.118.2.4	imageWidth	989
15.118.2.5	pixelFormat	989
15.119	ImagePtr Class Reference	990
15.119.1	Detailed Description	991
15.119.2	Constructor & Destructor Documentation	991
15.119.2.1	ImagePtr() [1/4]	991
15.119.2.2	ImagePtr() [2/4]	991
15.119.2.3	ImagePtr() [3/4]	991
15.119.2.4	ImagePtr() [4/4]	991
15.119.2.5	~ImagePtr()	992
15.119.3	Member Function Documentation	992
15.119.3.1	operator=()	992
15.120	ImageStatistics Class Reference	992
15.120.1	Detailed Description	993
15.120.2	Constructor & Destructor Documentation	994
15.120.2.1	ImageStatistics() [1/2]	994
15.120.2.2	~ImageStatistics()	994
15.120.2.3	ImageStatistics() [2/2]	994
15.120.3	Member Function Documentation	994
15.120.3.1	DisableAll()	994
15.120.3.2	EnableAll()	994
15.120.3.3	EnableGreyOnly()	995
15.120.3.4	EnableHSLOnly()	995
15.120.3.5	EnableRGBOnly()	995
15.120.3.6	GetChannelStatus()	995
15.120.3.7	GetHistogram()	996
15.120.3.8	GetMean()	996
15.120.3.9	GetNumPixelValues()	996

15.120.3.10	GetPixelValueRange()	997
15.120.3.11	GetRange()	997
15.120.3.12	GetStatistics()	997
15.120.3.13	Operator=()	998
15.120.3.14	SetChannelStatus()	998
15.120.4	Friends And Related Function Documentation	999
15.120.4.1	ImageStatsCalculator	999
15.121	ImageUtility Class Reference	999
15.121.1	Detailed Description	1000
15.121.2	Member Enumeration Documentation	1000
15.121.2.1	ImageScalingAlgorithm	1000
15.121.2.2	SourceDataRange	1000
15.121.3	Member Function Documentation	1001
15.121.3.1	CreateNormalized() [1/5]	1001
15.121.3.2	CreateNormalized() [2/5]	1001
15.121.3.3	CreateNormalized() [3/5]	1002
15.121.3.4	CreateNormalized() [4/5]	1002
15.121.3.5	CreateNormalized() [5/5]	1003
15.121.3.6	CreateScaled() [1/2]	1003
15.121.3.7	CreateScaled() [2/2]	1004
15.122	ImageUtilityCCM Class Reference	1004
15.122.1	Detailed Description	1004
15.122.2	Member Function Documentation	1004
15.122.2.1	CreateColorCorrected() [1/2]	1005
15.122.2.2	CreateColorCorrected() [2/2]	1005
15.123	ImageUtilityHeatmap Class Reference	1006
15.123.1	Detailed Description	1007
15.123.2	Member Enumeration Documentation	1007
15.123.2.1	HeatmapColor	1007
15.123.3	Member Function Documentation	1007

15.123.3.1CreateHeatmap() [1/2]	1008
15.123.3.2CreateHeatmap() [2/2]	1008
15.123.3.3GetHeatmapColorGradient()	1009
15.123.3.4GetHeatmapRange()	1009
15.123.3.5SetHeatmapColorGradient()	1009
15.123.3.6SetHeatmapRange()	1010
15.124ImageUtilityPolarization Class Reference	1010
15.124.1Detailed Description	1011
15.124.2Member Enumeration Documentation	1011
15.124.2.1PolarizationQuadrant	1012
15.124.3Member Function Documentation	1012
15.124.3.1CreateAolp() [1/2]	1012
15.124.3.2CreateAolp() [2/2]	1012
15.124.3.3CreateDolp() [1/2]	1014
15.124.3.4CreateDolp() [2/2]	1014
15.124.3.5CreateGlareReduced() [1/2]	1015
15.124.3.6CreateGlareReduced() [2/2]	1015
15.124.3.7CreateStokesS0() [1/2]	1015
15.124.3.8CreateStokesS0() [2/2]	1016
15.124.3.9CreateStokesS1() [1/2]	1016
15.124.3.10CreateStokesS1() [2/2]	1017
15.124.3.11CreateStokesS2() [1/2]	1017
15.124.3.12CreateStokesS2() [2/2]	1018
15.124.3.13ExtractPolarQuadrant() [1/2]	1018
15.124.3.14ExtractPolarQuadrant() [2/2]	1018
15.125InferenceBoundingBox Struct Reference	1020
15.125.1Detailed Description	1020
15.126InferenceBoundingBoxResult Class Reference	1020
15.126.1Detailed Description	1021
15.127InferenceBoxCircle Struct Reference	1021

15.128. InferenceBoxRect Struct Reference	1021
15.128.1 Detailed Description	1022
15.129. InferenceBoxRotatedRect Struct Reference	1022
15.130. int64_autovector_t Class Reference	1022
15.130.1 Detailed Description	1023
15.130.2 Constructor & Destructor Documentation	1023
15.130.2.1 int64_autovector_t() [1/3]	1023
15.130.2.2 int64_autovector_t() [2/3]	1023
15.130.2.3 int64_autovector_t() [3/3]	1023
15.130.2.4 ~int64_autovector_t()	1023
15.130.3 Member Function Documentation	1023
15.130.3.1 operator delete()	1024
15.130.3.2 operator new()	1024
15.130.3.3 operator=()	1024
15.130.3.4 operator[]() [1/2]	1024
15.130.3.5 operator[]() [2/2]	1024
15.130.3.6 size()	1024
15.130.4 Member Data Documentation	1024
15.130.4.1 _pCount	1025
15.130.4.2 _pv	1025
15.131. IntegerNode Class Reference	1025
15.131.1 Detailed Description	1027
15.131.2 Constructor & Destructor Documentation	1027
15.131.2.1 IntegerNode() [1/2]	1027
15.131.2.2 IntegerNode() [2/2]	1027
15.131.2.3 ~IntegerNode()	1027
15.131.3 Member Function Documentation	1027
15.131.3.1 GetFloatAlias()	1028
15.131.3.2 GetInc()	1028
15.131.3.3 GetIncMode()	1028

15.131.3.4	GetListOfValidValues()	1028
15.131.3.5	GetMax()	1028
15.131.3.6	GetMin()	1028
15.131.3.7	GetRepresentation()	1029
15.131.3.8	GetUnit()	1029
15.131.3.9	GetValue()	1029
15.131.3.10	ImposeMax()	1029
15.131.3.11	ImposeMin()	1029
15.131.3.12	operator()()	1030
15.131.3.13	operator*()	1030
15.131.3.14	operator=()	1030
15.131.3.15	SetReference()	1030
15.131.3.16	SetValue()	1030
15.132	Interface Class Reference	1031
15.132.1	Detailed Description	1032
15.132.2	Constructor & Destructor Documentation	1032
15.132.2.1	~Interface()	1032
15.132.3	Member Function Documentation	1032
15.132.3.1	GetCameras()	1033
15.132.3.2	GetTLNodeMap()	1033
15.132.3.3	IsInUse()	1033
15.132.3.4	IsValid()	1034
15.132.3.5	RegisterEventHandler()	1034
15.132.3.6	SendActionCommand()	1034
15.132.3.7	UnregisterEventHandler()	1035
15.132.3.8	UpdateCameras()	1035
15.132.4	Friends And Related Function Documentation	1036
15.132.4.1	InterfaceInternal	1036
15.133	InterfaceArrivalEventHandler Class Reference	1036
15.133.1	Detailed Description	1037

15.133.2	Constructor & Destructor Documentation	1037
15.133.2.1	InterfaceArrivalEventHandler()	1038
15.133.2.2	~InterfaceArrivalEventHandler()	1038
15.133.3	Member Function Documentation	1038
15.133.3.1	OnInterfaceArrival()	1038
15.133.3.2	operator=()	1038
15.134	InterfaceEventHandler Class Reference	1039
15.134.1	Detailed Description	1040
15.134.2	Constructor & Destructor Documentation	1040
15.134.2.1	InterfaceEventHandler()	1040
15.134.2.2	~InterfaceEventHandler()	1040
15.134.3	Member Function Documentation	1040
15.134.3.1	OnDeviceArrival()	1041
15.134.3.2	OnDeviceRemoval()	1041
15.134.3.3	operator=()	1041
15.135	InterfaceEventHandlerImpl Class Reference	1042
15.135.1	Constructor & Destructor Documentation	1043
15.135.1.1	InterfaceEventHandlerImpl() [1/3]	1044
15.135.1.2	~InterfaceEventHandlerImpl() [1/2]	1044
15.135.1.3	InterfaceEventHandlerImpl() [2/3]	1044
15.135.1.4	InterfaceEventHandlerImpl() [3/3]	1044
15.135.1.5	~InterfaceEventHandlerImpl() [2/2]	1044
15.135.2	Member Function Documentation	1044
15.135.2.1	GetInterfaceId()	1044
15.135.2.2	OnDeviceArrival() [1/2]	1045
15.135.2.3	OnDeviceArrival() [2/2]	1045
15.135.2.4	OnDeviceRemoval() [1/2]	1045
15.135.2.5	OnDeviceRemoval() [2/2]	1045
15.135.2.6	PrintGenericHandlerMessage()	1046
15.136	InterfaceList Class Reference	1046

15.136.1Detailed Description	1047
15.136.2Constructor & Destructor Documentation	1047
15.136.2.1InterfaceList() [1/2]	1047
15.136.2.2~InterfaceList()	1047
15.136.2.3InterfaceList() [2/2]	1048
15.136.3Member Function Documentation	1048
15.136.3.1Append()	1048
15.136.3.2Clear()	1048
15.136.3.3GetByIndex()	1048
15.136.3.4GetSize()	1049
15.136.3.5operator=()	1049
15.136.3.6operator[]()	1049
15.136.4Friends And Related Function Documentation	1049
15.136.4.1ProducerImpl	1050
15.136.4.2SystemImpl	1050
15.137InterfacePtr Class Reference	1050
15.137.1Detailed Description	1051
15.137.2Constructor & Destructor Documentation	1051
15.137.2.1InterfacePtr() [1/4]	1051
15.137.2.2InterfacePtr() [2/4]	1051
15.137.2.3InterfacePtr() [3/4]	1051
15.137.2.4InterfacePtr() [4/4]	1052
15.138InterfaceRemovalEventHandler Class Reference	1052
15.138.1Detailed Description	1053
15.138.2Constructor & Destructor Documentation	1053
15.138.2.1InterfaceRemovalEventHandler()	1053
15.138.2.2~InterfaceRemovalEventHandler()	1053
15.138.3Member Function Documentation	1053
15.138.3.1OnInterfaceRemoval()	1053
15.138.3.2operator=()	1054

15.139.1	IntRegNode Class Reference	1054
15.139.1	Detailed Description	1055
15.139.2	Constructor & Destructor Documentation	1055
15.139.2.1	IntRegNode() [1/2]	1056
15.139.2.2	IntRegNode() [2/2]	1056
15.139.2.3	~IntRegNode()	1056
15.139.3	Member Function Documentation	1056
15.139.3.1	SetReference()	1056
15.140.1	Info Struct Reference	1056
15.140.1	Constructor & Destructor Documentation	1057
15.140.1.1	Info()	1057
15.140.2	Member Data Documentation	1057
15.140.2.1	gateway	1057
15.140.2.2	pAddress	1057
15.140.2.3	subnetLength	1057
15.140.2.4	subnetMask	1057
15.141.1	ISystem Class Reference	1058
15.141.1	Detailed Description	1059
15.141.2	Constructor & Destructor Documentation	1059
15.141.2.1	~ISystem()	1059
15.141.2.2	ISystem() [1/2]	1059
15.141.2.3	ISystem() [2/2]	1060
15.141.3	Member Function Documentation	1060
15.141.3.1	GetCameras()	1060
15.141.3.2	GetInterfaces()	1060
15.141.3.3	GetLibraryVersion()	1060
15.141.3.4	GetLoggingEventPriorityLevel()	1060
15.141.3.5	GetTLNodeMap()	1061
15.141.3.6	IsInUse()	1061
15.141.3.7	operator=()	1061

15.141.3.8	RegisterEventHandler()	1061
15.141.3.9	RegisterInterfaceEventHandler()	1061
15.141.3.10	RegisterLoggingEventHandler()	1061
15.141.3.11	ReleaseInstance()	1062
15.141.3.12	SendActionCommand()	1062
15.141.3.13	SetLoggingEventPriorityLevel()	1062
15.141.3.14	UnregisterAllLoggingEventHandlers()	1062
15.141.3.15	UnregisterEventHandler()	1062
15.141.3.16	UnregisterInterfaceEventHandler()	1063
15.141.3.17	UnregisterLoggingEventHandler()	1063
15.141.3.18	UpdateCameras()	1063
15.141.3.19	UpdateInterfaceList()	1063
15.141.4	Friends And Related Function Documentation	1063
15.141.4.1	SystemPtrInternal	1063
15.141.5	Member Data Documentation	1063
15.141.5.1	TLSys	1064
15.142	SystemEventHandler Class Reference	1064
15.142.1	Constructor & Destructor Documentation	1065
15.142.1.1	~ISystemEventHandler()	1065
15.142.1.2	SystemEventHandler() [1/2]	1065
15.142.1.3	SystemEventHandler() [2/2]	1065
15.142.2	Member Function Documentation	1065
15.142.2.1	OnInterfaceArrival()	1065
15.142.2.2	OnInterfaceRemoval()	1066
15.142.2.3	operator=()	1066
15.143	PEGOOption Struct Reference	1066
15.143.1	Detailed Description	1066
15.143.2	Constructor & Destructor Documentation	1066
15.143.2.1	PEGOOption()	1067
15.143.3	Member Data Documentation	1067

15.143.3.1progressive	1067
15.143.3.2quality	1067
15.143.3.3reserved	1067
15.144JPG2Option Struct Reference	1067
15.144.1Detailed Description	1068
15.144.2Constructor & Destructor Documentation	1068
15.144.2.1JPG2Option()	1068
15.144.3Member Data Documentation	1068
15.144.3.1quality	1068
15.144.3.2reserved	1068
15.145LibraryVersion Struct Reference	1069
15.145.1Detailed Description	1069
15.145.2Member Data Documentation	1069
15.145.2.1build	1069
15.145.2.2major	1069
15.145.2.3minor	1069
15.145.2.4type	1070
15.146LockableObject< Object >::Lock Class Reference	1070
15.146.1Detailed Description	1070
15.146.2Constructor & Destructor Documentation	1070
15.146.2.1Lock()	1070
15.146.2.2~Lock()	1070
15.147LockableObject< Object > Class Template Reference	1071
15.147.1Detailed Description	1071
15.147.2Member Function Documentation	1072
15.147.2.1GetLock()	1072
15.147.3Friends And Related Function Documentation	1072
15.147.3.1Lock	1072
15.147.4Member Data Documentation	1072
15.147.4.1m_Lock	1072

15.148	LoggingEventData Class Reference	1072
15.148.1	Detailed Description	1073
15.148.2	Constructor & Destructor Documentation	1073
15.148.2.1	~LoggingEventData()	1073
15.148.2.2	LoggingEventData()	1074
15.148.3	Member Function Documentation	1074
15.148.3.1	GetCategoryName()	1074
15.148.3.2	GetLogMessage()	1074
15.148.3.3	GetNDC()	1074
15.148.3.4	GetPriority()	1075
15.148.3.5	GetPriorityName()	1075
15.148.3.6	GetThreadName()	1075
15.148.3.7	GetTimestamp()	1075
15.148.4	Friends And Related Function Documentation	1075
15.148.4.1	SystemImpl	1076
15.149	LoggingEventDataPtr Class Reference	1076
15.149.1	Detailed Description	1077
15.149.2	Constructor & Destructor Documentation	1077
15.149.2.1	LoggingEventDataPtr() [1/4]	1077
15.149.2.2	LoggingEventDataPtr() [2/4]	1077
15.149.2.3	LoggingEventDataPtr() [3/4]	1077
15.149.2.4	LoggingEventDataPtr() [4/4]	1077
15.150	LoggingEventHandler Class Reference	1078
15.150.1	Detailed Description	1079
15.150.2	Constructor & Destructor Documentation	1079
15.150.2.1	LoggingEventHandler()	1079
15.150.2.2	~LoggingEventHandler()	1079
15.150.3	Member Function Documentation	1079
15.150.3.1	OnLogEvent()	1079
15.150.3.2	operator=()	1080

15.151	LoggingEventHandlerImpl Class Reference	1080
15.152	Member_NodeCallback< Client, Member > Class Template Reference	1081
15.152.1	Detailed Description	1082
15.152.2	Member Typedef Documentation	1082
15.152.2.1	PMEMBERFUNC	1082
15.152.3	Constructor & Destructor Documentation	1083
15.152.3.1	Member_NodeCallback()	1083
15.152.4	Member Function Documentation	1083
15.152.4.1	Destroy()	1083
15.152.4.2	operator()()	1083
15.153	MJPGOption Struct Reference	1083
15.153.1	Detailed Description	1084
15.153.2	Constructor & Destructor Documentation	1084
15.153.2.1	MJPGOption()	1084
15.153.3	Member Data Documentation	1084
15.153.3.1	frameRate	1084
15.153.3.2	quality	1084
15.153.3.3	reserved	1085
15.154	Node Class Reference	1085
15.154.1	Detailed Description	1087
15.154.2	Constructor & Destructor Documentation	1088
15.154.2.1	Node() [1/2]	1088
15.154.2.2	Node() [2/2]	1088
15.154.2.3	~Node()	1088
15.154.3	Member Function Documentation	1088
15.154.3.1	DeregisterCallback()	1088
15.154.3.2	GetAccessMode()	1089
15.154.3.3	GetAlias()	1089
15.154.3.4	GetCachingMode()	1089
15.154.3.5	GetCastAlias()	1089

15.154.3.6	GetChildren()	1089
15.154.3.7	GetDescription()	1090
15.154.3.8	GetDeviceName()	1090
15.154.3.9	GetDisplayName()	1090
15.154.3.10	GetDocuURL()	1090
15.154.3.11	GetEventID()	1090
15.154.3.12	GetName()	1090
15.154.3.13	GetNameSpace()	1091
15.154.3.14	GetNodeHandle()	1091
15.154.3.15	GetNodeMap()	1091
15.154.3.16	GetParents()	1091
15.154.3.17	GetPollingTime()	1091
15.154.3.18	GetPrincipalInterfaceType()	1092
15.154.3.19	GetProperty()	1092
15.154.3.20	GetPropertyNames()	1092
15.154.3.21	GetSelectedFeatures()	1092
15.154.3.22	GetSelectingFeatures()	1092
15.154.3.23	GetToolTip()	1093
15.154.3.24	GetVisibility()	1093
15.154.3.25	ImposeAccessMode()	1093
15.154.3.26	ImposeVisibility()	1093
15.154.3.27	ValidateNode()	1093
15.154.3.28	AccessModeCacheable()	1093
15.154.3.29	Cachable()	1094
15.154.3.30	Deprecated()	1094
15.154.3.31	Feature()	1094
15.154.3.32	Selector()	1094
15.154.3.33	Streamable()	1094
15.154.3.34	operator!=()	1094
15.154.3.35	operator==()	1095

15.154.3.36RegisterCallback()	1095
15.154.3.37GetNodeHandle()	1095
15.154.3.38GetNodeMap()	1095
15.154.3.39GetReference() [1/2]	1095
15.154.3.40GetReference() [2/2]	1095
15.154.4Member Data Documentation	1096
15.154.4.1m_Callbacks	1096
15.154.4.2m_pNodeData	1096
15.154.4.3m_pNodeMap	1096
15.155NodeMap Class Reference	1096
15.155.1Detailed Description	1098
15.155.2Constructor & Destructor Documentation	1098
15.155.2.1NodeMap()	1099
15.155.2.2~NodeMap()	1099
15.155.3Member Function Documentation	1099
15.155.3.1ClearXMLCache()	1099
15.155.3.2Connect() [1/2]	1099
15.155.3.3Connect() [2/2]	1099
15.155.3.4Destroy()	1100
15.155.3.5GetDeviceName()	1100
15.155.3.6GetDeviceVersion()	1100
15.155.3.7GetGenApiVersion()	1100
15.155.3.8GetLock()	1100
15.155.3.9GetModelName()	1100
15.155.3.10GetNode()	1101
15.155.3.11GetNodeMapHandle()	1101
15.155.3.12GetNodes()	1101
15.155.3.13GetNumNodes()	1101
15.155.3.14GetProductGuid()	1101
15.155.3.15GetSchemaVersion()	1101

15.155.3.16	GetStandardNameSpace()	1102
15.155.3.17	GetSupportedSchemaVersions()	1102
15.155.3.18	GetToolTip()	1102
15.155.3.19	GetVendorName()	1102
15.155.3.20	GetVersionGuid()	1103
15.155.3.21	InvalidateNodes()	1103
15.155.3.22	LoadXMLFromFile()	1103
15.155.3.23	LoadXMLFromFileInject()	1103
15.155.3.24	LoadXMLFromString()	1103
15.155.3.25	LoadXMLFromStringInject()	1104
15.155.3.26	LoadXMLFromZIPData()	1104
15.155.3.27	LoadXMLFromZIPFile()	1104
15.155.3.28	Roll()	1104
15.155.4	Member Data Documentation	1104
15.155.4.1	_Ptr	1104
15.156	NodeMapFactory::NodeStatistics_t Struct Reference	1105
15.156.1	Member Data Documentation	1105
15.156.1.1	NumLinks	1105
15.156.1.2	NumNodes	1105
15.156.1.3	NumProperties	1105
15.156.1.4	NumStrings	1105
15.157	DevFileStreamBase< CharType, Traits > Class Template Reference	1106
15.157.1	Member Typedef Documentation	1107
15.157.1.1	filebuf_type	1107
15.157.1.2	os_type	1107
15.157.1.3	ostream_type	1107
15.157.2	Member Function Documentation	1107
15.157.2.1	close()	1107
15.157.2.2	is_open()	1107
15.157.2.3	open()	1107

15.157.2.4	dbuf()	1108
15.158	DevFileStreamBuf< CharType, Traits > Class Template Reference	1108
15.158.1	Constructor & Destructor Documentation	1109
15.158.1.1	IODevFileStreamBuf()	1109
15.158.1.2	~ODevFileStreamBuf()	1109
15.158.2	Member Function Documentation	1109
15.158.2.1	close()	1109
15.158.2.2	is_open()	1109
15.158.2.3	open()	1110
15.158.2.4	overflow()	1110
15.158.2.5	sync()	1110
15.158.2.6	sputn()	1110
15.159	PGMOption Struct Reference	1110
15.159.1	Detailed Description	1111
15.159.2	Constructor & Destructor Documentation	1111
15.159.2.1	PGMOption()	1111
15.159.3	Member Data Documentation	1111
15.159.3.1	binaryFile	1111
15.159.3.2	reserved	1111
15.160	PNGOption Struct Reference	1111
15.160.1	Detailed Description	1112
15.160.2	Constructor & Destructor Documentation	1112
15.160.2.1	PNGOption()	1112
15.160.3	Member Data Documentation	1112
15.160.3.1	compressionLevel	1112
15.160.3.2	interlaced	1112
15.160.3.3	reserved	1113
15.161	PortNode Class Reference	1113
15.161.1	Detailed Description	1115
15.161.2	Constructor & Destructor Documentation	1115

15.161.2.1PortNode() [1/2]	1115
15.161.2.2PortNode() [2/2]	1115
15.161.2.3~PortNode()	1115
15.161.3Member Function Documentation	1115
15.161.3.1CacheChunkData()	1115
15.161.3.2GetChunkID()	1116
15.161.3.3GetPortHandle()	1116
15.161.3.4GetSwapEndianness()	1116
15.161.3.5Read()	1116
15.161.3.6Replay()	1116
15.161.3.7SetPortImpl()	1117
15.161.3.8SetReference() [1/3]	1117
15.161.3.9SetReference() [2/3]	1117
15.161.3.10SetReference() [3/3]	1117
15.161.3.11StartRecording()	1117
15.161.3.12StopRecording()	1118
15.161.3.13Write()	1118
15.162PortRecorder Class Reference	1118
15.162.1Detailed Description	1119
15.162.2Constructor & Destructor Documentation	1120
15.162.2.1PortRecorder()	1120
15.162.2.2~PortRecorder()	1120
15.162.3Member Function Documentation	1120
15.162.3.1GetAccessMode()	1120
15.162.3.2Read()	1120
15.162.3.3Replay()	1121
15.162.3.4SetReference()	1121
15.162.3.5StartRecording()	1121
15.162.3.6StopRecording()	1121
15.162.3.7Write()	1122

15.163.1	PortReplay Class Reference	1122
15.163.1.1	Detailed Description	1123
15.163.1.2	Constructor & Destructor Documentation	1123
15.163.1.2.1	PortReplay()	1124
15.163.1.2.2	~PortReplay()	1124
15.163.1.3	Member Function Documentation	1124
15.163.1.3.1	GetAccessMode()	1124
15.163.1.3.2	GetPortReplayHandle()	1124
15.163.1.3.3	Read()	1124
15.163.1.3.4	Replay()	1125
15.163.1.3.5	SetReference()	1125
15.163.1.3.6	Write()	1125
15.163.2	PPMOption Struct Reference	1125
15.163.2.1	Detailed Description	1126
15.163.2.2	Constructor & Destructor Documentation	1126
15.163.2.2.1	PPMOption()	1126
15.163.2.3	Member Data Documentation	1126
15.163.2.3.1	binaryFile	1126
15.163.2.3.2	reserved	1126
15.164.1	RegisterNode Class Reference	1127
15.164.1.1	Detailed Description	1128
15.164.1.2	Constructor & Destructor Documentation	1128
15.164.1.2.1	RegisterNode() [1/2]	1129
15.164.1.2.2	RegisterNode() [2/2]	1129
15.164.1.2.3	~RegisterNode()	1129
15.164.1.3	Member Function Documentation	1129
15.164.1.3.1	Get()	1129
15.164.1.3.2	GetAddress()	1129
15.164.1.3.3	GetLength()	1130
15.164.1.3.4	Set()	1130

15.165.3.5SetReference()	1130
15.166SingleChunkData_t Struct Reference	1130
15.166.1Member Data Documentation	1131
15.166.1.1ChunkID	1131
15.166.1.2ChunkLength	1131
15.166.1.3ChunkOffset	1131
15.167SingleChunkDataStr_t Struct Reference	1131
15.167.1Member Data Documentation	1131
15.167.1.1ChunkID	1131
15.167.1.2ChunkLength	1132
15.167.1.3ChunkOffset	1132
15.168SpinTestCamera Class Reference	1132
15.169SpinVideo Class Reference	1133
15.169.1Detailed Description	1133
15.169.2Constructor & Destructor Documentation	1133
15.169.2.1SpinVideo()	1133
15.169.2.2~SpinVideo()	1134
15.169.3Member Function Documentation	1134
15.169.3.1Append()	1134
15.169.3.2Close()	1134
15.169.3.3Open() [1/3]	1134
15.169.3.4Open() [2/3]	1135
15.169.3.5Open() [3/3]	1135
15.169.3.6SetMaximumFileSize()	1136
15.170StringNode Class Reference	1136
15.170.1Detailed Description	1138
15.170.2Constructor & Destructor Documentation	1138
15.170.2.1StringNode() [1/2]	1138
15.170.2.2StringNode() [2/2]	1138
15.170.2.3~StringNode()	1138

15.170.3Member Function Documentation	1139
15.170.3.1GetMaxLength()	1139
15.170.3.2GetValue()	1139
15.170.3.3operator()()	1139
15.170.3.4operator*()	1139
15.170.3.5operator=()	1140
15.170.3.6SetReference()	1140
15.170.3.7SetValue()	1140
15.171StringRegNode Class Reference	1140
15.171.1Detailed Description	1142
15.171.2Constructor & Destructor Documentation	1142
15.171.2.1StringRegNode() [1/2]	1143
15.171.2.2StringRegNode() [2/2]	1143
15.171.2.3~StringRegNode()	1143
15.171.3Member Function Documentation	1143
15.171.3.1SetReference()	1143
15.172System Class Reference	1144
15.172.1Detailed Description	1145
15.172.2Constructor & Destructor Documentation	1146
15.172.2.1~System()	1146
15.172.2.2System()	1146
15.172.3Member Function Documentation	1146
15.172.3.1GetCameras()	1146
15.172.3.2GetInstance()	1147
15.172.3.3GetInterfaces()	1147
15.172.3.4GetLibraryVersion()	1148
15.172.3.5GetLoggingEventPriorityLevel()	1148
15.172.3.6GetTLNodeMap()	1148
15.172.3.7IsInUse()	1149
15.172.3.8RegisterEventHandler()	1149

15.172.3.9	RegisterInterfaceEventHandler()	1149
15.172.3.10	RegisterLoggingEventHandler()	1150
15.172.3.11	ReleaseInstance()	1150
15.172.3.12	SendActionCommand()	1150
15.172.3.13	SetLoggingEventPriorityLevel()	1151
15.172.3.14	UnregisterAllLoggingEventHandlers()	1152
15.172.3.15	UnregisterEventHandler()	1152
15.172.3.16	UnregisterInterfaceEventHandler()	1152
15.172.3.17	UnregisterLoggingEventHandler()	1153
15.172.3.18	UpdateCameras()	1153
15.172.3.19	UpdateInterfaceList()	1153
15.173	SystemEventHandler Class Reference	1154
15.173.1	Detailed Description	1155
15.173.2	Constructor & Destructor Documentation	1155
15.173.2.1	SystemEventHandler()	1156
15.173.2.2	~SystemEventHandler()	1156
15.173.3	Member Function Documentation	1156
15.173.3.1	OnInterfaceArrival()	1156
15.173.3.2	OnInterfaceRemoval()	1156
15.173.3.3	operator=()	1157
15.174	SystemEventHandlerImpl Class Reference	1157
15.174.1	Constructor & Destructor Documentation	1158
15.174.1.1	SystemEventHandlerImpl()	1158
15.174.1.2	~SystemEventHandlerImpl()	1159
15.174.2	Member Function Documentation	1159
15.174.2.1	LockEventHandlerMutex()	1159
15.174.2.2	OnInterfaceArrival()	1159
15.174.2.3	OnInterfaceRemoval()	1159
15.174.2.4	RegisterAllInterfaceEvents()	1160
15.174.2.5	RegisterInterfaceEventToSystem()	1160

15.174.2.6UnlockEventHandlerMutex()	1160
15.174.2.7UnregisterAllInterfaceEvents()	1160
15.174.2.8UnregisterInterfaceEventFromSystem()	1160
15.175SystemPtr Class Reference	1161
15.175.1Detailed Description	1162
15.175.2Constructor & Destructor Documentation	1162
15.175.2.1SystemPtr() [1/4]	1162
15.175.2.2SystemPtr() [2/4]	1162
15.175.2.3SystemPtr() [3/4]	1162
15.175.2.4SystemPtr() [4/4]	1162
15.175.2.5~SystemPtr()	1163
15.176TIFFOption Struct Reference	1163
15.176.1Detailed Description	1163
15.176.2Member Enumeration Documentation	1163
15.176.2.1CompressionMethod	1163
15.176.3Constructor & Destructor Documentation	1164
15.176.3.1TIFFOption()	1164
15.176.4Member Data Documentation	1164
15.176.4.1compression	1164
15.176.4.2reserved	1164
15.177TransportLayerDevice Class Reference	1165
15.177.1Detailed Description	1167
15.177.2Constructor & Destructor Documentation	1167
15.177.2.1TransportLayerDevice() [1/2]	1167
15.177.2.2~TransportLayerDevice()	1167
15.177.2.3TransportLayerDevice() [2/2]	1167
15.177.3Friends And Related Function Documentation	1167
15.177.3.1CameraBase	1167
15.177.3.2CameraInternal	1168
15.177.3.3CameraBase	1168

15.177.4Member Data Documentation	1168
15.177.4.1DeviceAccessStatus	1168
15.177.4.2DeviceCurrentSpeed	1168
15.177.4.3DeviceDisplayName	1168
15.177.4.4DeviceDriverVersion	1168
15.177.4.5DeviceEndianessMechanism	1169
15.177.4.6DeviceID	1169
15.177.4.7DeviceInstanceId	1169
15.177.4.8DeviceIsUpdater	1169
15.177.4.9DeviceLinkSpeed	1169
15.177.4.10DeviceLocation	1169
15.177.4.11DeviceModelName	1170
15.177.4.12DeviceMulticastMonitorMode	1170
15.177.4.13DevicePortId	1170
15.177.4.14DeviceSerialNumber	1170
15.177.4.15DeviceType	1170
15.177.4.16DeviceU3VProtocol	1170
15.177.4.17DeviceUserID	1171
15.177.4.18DeviceVendorName	1171
15.177.4.19DeviceVersion	1171
15.177.4.20GenICamXMLLocation	1171
15.177.4.21GenICamXMLPath	1171
15.177.4.22DevCCP	1171
15.177.4.23DevDeviceAutoForceIP	1172
15.177.4.24DevDeviceDiscoverMaximumPacketSize	1172
15.177.4.25DevDeviceForceGateway	1172
15.177.4.26DevDeviceForceIP	1172
15.177.4.27DevDeviceForceIPAddress	1172
15.177.4.28DevDeviceForceSubnetMask	1172
15.177.4.29DevDeviceGateway	1173

15.177.4.30	DevDeviceIPAddress	1173
15.177.4.31	DevDevicesWrongSubnet	1173
15.177.4.32	DevDeviceMACAddress	1173
15.177.4.33	DevDeviceMaximumPacketSize	1173
15.177.4.34	DevDeviceMaximumRetryCount	1173
15.177.4.35	DevDeviceModelsBigEndian	1174
15.177.4.36	DevDevicePort	1174
15.177.4.37	DevDeviceReadAndWriteTimeout	1174
15.177.4.38	DevDeviceSubnetMask	1174
15.177.4.39	DevVersionMajor	1174
15.177.4.40	DevVersionMinor	1174
15.177.4.41	UIXMLLocation	1175
15.177.4.42	UIXMLPath	1175
15.178	TransportLayerInterface Class Reference	1175
15.178.1	Detailed Description	1177
15.178.2	Constructor & Destructor Documentation	1178
15.178.2.1	TransportLayerInterface() [1/2]	1178
15.178.2.2	~TransportLayerInterface()	1178
15.178.2.3	TransportLayerInterface() [2/2]	1178
15.178.3	Friends And Related Function Documentation	1178
15.178.3.1	Interface	1178
15.178.3.2	Interface	1178
15.178.3.3	InterfaceInternal	1178
15.178.4	Member Data Documentation	1178
15.178.4.1	ActionCommand	1179
15.178.4.2	DeviceAccessStatus	1179
15.178.4.3	DeviceCount	1179
15.178.4.4	DeviceID	1179
15.178.4.5	DeviceModelName	1179
15.178.4.6	DeviceSelector	1179

15.178.4.7DeviceSerialNumber	1180
15.178.4.8DeviceUnlock	1180
15.178.4.9DeviceUpdateList	1180
15.178.4.10DeviceVendorName	1180
15.178.4.11FilterDriverStatus	1180
15.178.4.12DevActionDeviceKey	1180
15.178.4.13DevActionGroupKey	1181
15.178.4.14DevActionGroupMask	1181
15.178.4.15DevActionTime	1181
15.178.4.16DevDeviceAutoForceIP	1181
15.178.4.17DevDeviceForceGateway	1181
15.178.4.18DevDeviceForceIP	1181
15.178.4.19DevDeviceForceIPAddress	1182
15.178.4.20DevDeviceForceSubnetMask	1182
15.178.4.21DevDeviceGateway	1182
15.178.4.22DevDeviceIPAddress	1182
15.178.4.23DevDeviceMACAddress	1182
15.178.4.24DevDeviceSubnetMask	1182
15.178.4.25DevInterfaceGateway	1183
15.178.4.26DevInterfaceGatewaySelector	1183
15.178.4.27DevInterfaceMACAddress	1183
15.178.4.28DevInterfaceMTU	1183
15.178.4.29DevInterfaceReceiveLinkSpeed	1183
15.178.4.30DevInterfaceSubnetIPAddress	1183
15.178.4.31DevInterfaceSubnetMask	1184
15.178.4.32DevInterfaceSubnetSelector	1184
15.178.4.33DevInterfaceTransmitLinkSpeed	1184
15.178.4.34HostAdapterDriverVersion	1184
15.178.4.35HostAdapterName	1184
15.178.4.36HostAdapterVendor	1184

15.178.4.37	compatibleDeviceCount	1185
15.178.4.38	compatibleDeviceID	1185
15.178.4.39	compatibleDeviceModelName	1185
15.178.4.40	compatibleDeviceSelector	1185
15.178.4.41	compatibleDeviceVendorName	1185
15.178.4.42	compatibleGevDeviceIPAddress	1185
15.178.4.43	compatibleGevDeviceMACAddress	1186
15.178.4.44	compatibleGevDeviceSubnetMask	1186
15.178.4.45	interfaceDisplayName	1186
15.178.4.46	interfaceID	1186
15.178.4.47	interfaceType	1186
15.178.4.48	BOEStatus	1186
15.179	TransportLayerStream Class Reference	1187
15.179.1	Detailed Description	1188
15.179.2	Constructor & Destructor Documentation	1188
15.179.2.1	TransportLayerStream() [1/2]	1188
15.179.2.2	~TransportLayerStream()	1189
15.179.2.3	TransportLayerStream() [2/2]	1189
15.179.3	Friends And Related Function Documentation	1189
15.179.3.1	CameraBase	1189
15.179.3.2	CameraInternal	1189
15.179.3.3	CameraBase	1189
15.179.4	Member Data Documentation	1189
15.179.4.1	GevFailedPacketCount	1189
15.179.4.2	GevMaximumNumberResendRequests	1190
15.179.4.3	GevPacketResendMode	1190
15.179.4.4	GevPacketResendTimeout	1190
15.179.4.5	GevResendPacketCount	1190
15.179.4.6	GevResendRequestCount	1190
15.179.4.7	GevTotalIPacketCount	1190

15.179.4.8	StreamAnnounceBufferMinimum	1191
15.179.4.9	StreamAnnouncedBufferCount	1191
15.179.4.10	StreamBlockTransferSize	1191
15.179.4.11	StreamBufferAlignment	1191
15.179.4.12	StreamBufferCountManual	1191
15.179.4.13	StreamBufferCountMax	1191
15.179.4.14	StreamBufferCountMode	1192
15.179.4.15	StreamBufferCountResult	1192
15.179.4.16	StreamBufferHandlingMode	1192
15.179.4.17	StreamChunkCountMaximum	1192
15.179.4.18	StreamCRCCheckEnable	1192
15.179.4.19	StreamDeliveredFrameCount	1192
15.179.4.20	StreamDroppedFrameCount	1193
15.179.4.21	StreamFailedBufferCount	1193
15.179.4.22	StreamID	1193
15.179.4.23	StreamInputBufferCount	1193
15.179.4.24	StreamIsGrabbing	1193
15.179.4.25	StreamLostFrameCount	1193
15.179.4.26	StreamOutputBufferCount	1194
15.179.4.27	StreamStartedFrameCount	1194
15.179.4.28	StreamType	1194
15.180	TransportLayerSystem Class Reference	1194
15.180.1	Detailed Description	1196
15.180.2	Constructor & Destructor Documentation	1196
15.180.2.1	TransportLayerSystem() [1/2]	1196
15.180.2.2	~TransportLayerSystem()	1196
15.180.2.3	TransportLayerSystem() [2/2]	1196
15.180.3	Friends And Related Function Documentation	1196
15.180.3.1	ISystem	1197
15.180.3.2	System	1197

15.180.3.3SystemPtrInternal	1197
15.180.4Member Data Documentation	1197
15.180.4.1EnumerateGEVInterfaces	1197
15.180.4.2EnumerateUSBInterfaces	1197
15.180.4.3GenTLSFNCVersionMajor	1197
15.180.4.4GenTLSFNCVersionMinor	1198
15.180.4.5GenTLSFNCVersionSubMinor	1198
15.180.4.6GenTLVersionMajor	1198
15.180.4.7GenTLVersionMinor	1198
15.180.4.8GevInterfaceDefaultGateway	1198
15.180.4.9GevInterfaceDefaultIPAddress	1198
15.180.4.10GevInterfaceDefaultSubnetMask	1199
15.180.4.11GevInterfaceMACAddress	1199
15.180.4.12GevVersionMajor	1199
15.180.4.13GevVersionMinor	1199
15.180.4.14InterfaceDisplayName	1199
15.180.4.15InterfaceID	1199
15.180.4.16InterfaceSelector	1200
15.180.4.17InterfaceUpdateList	1200
15.180.4.18LDisplayName	1200
15.180.4.19LFileName	1200
15.180.4.20LID	1200
15.180.4.21LModelName	1200
15.180.4.22LPath	1201
15.180.4.23LType	1201
15.180.4.24LVendorName	1201
15.180.4.25LVersion	1201
15.18U3V_CHUNK_TRAILER Struct Reference	1201
15.181.1Detailed Description	1202
15.181.2Member Data Documentation	1202

15.181.2.1ChunkID	1202
15.181.2.2ChunkLength	1202
15.1823V_COMMAND_HEADER Struct Reference	1202
15.182.1Detailed Description	1202
15.182.2Member Data Documentation	1202
15.182.2.1CommandId	1203
15.182.2.2Flags	1203
15.182.2.3Length	1203
15.182.2.4Prefix	1203
15.182.2.5ReqId	1203
15.1833V_EVENT_DATA Struct Reference	1203
15.183.1Detailed Description	1203
15.183.2Member Data Documentation	1204
15.183.2.1EventId	1204
15.183.2.2Reserved	1204
15.183.2.3Timestamp	1204
15.1843V_EVENT_MESSAGE Struct Reference	1204
15.184.1Detailed Description	1205
15.184.2Member Data Documentation	1205
15.184.2.1CommandHeader	1205
15.184.2.2EventData	1205
15.185ValueNode Class Reference	1205
15.185.1Detailed Description	1206
15.185.2Constructor & Destructor Documentation	1206
15.185.2.1ValueNode() [1/2]	1207
15.185.2.2ValueNode() [2/2]	1207
15.185.2.3~ValueNode()	1207
15.185.3Member Function Documentation	1207
15.185.3.1FromString()	1207
15.185.3.2GetNode()	1207
15.185.3.3IsValueCacheValid()	1208
15.185.3.4SetReference()	1208
15.185.3.5ToString()	1208
15.186Version_t Struct Reference	1208
15.186.1Detailed Description	1209
15.186.2Member Data Documentation	1209
15.186.2.1Major	1209
15.186.2.2Minor	1209
15.186.2.3SubMinor	1209

16 File Documentation	1211
16.1 doc/spindocs/C++/GettingStarted.dox File Reference	1211
16.2 doc/spindocs/C++/ProgrammerGuide.dox File Reference	1211
16.3 doc/spindocs/shared/Benefits.dox File Reference	1211
16.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference	1211
16.5 doc/spindocs/shared/GenICamGenTL.dox File Reference	1211
16.6 doc/spindocs/shared/Licensing.dox File Reference	1211
16.7 doc/spindocs/shared/Maintenance.dox File Reference	1211
16.8 include/AdapterConfig.h File Reference	1211
16.8.1 Macro Definition Documentation	1213
16.8.1.1 ADAPTERCONFIG_API	1213
16.9 include/AVIRecorder.h File Reference	1213
16.10include/BasePtr.h File Reference	1214
16.11include/Camera.h File Reference	1215
16.12include/CameraBase.h File Reference	1215
16.13include/CameraDefs.h File Reference	1216
16.14include/CameraList.h File Reference	1248
16.15include/CameraPtr.h File Reference	1249
16.16include/ChunkData.h File Reference	1249
16.17include/ChunkDataInference.h File Reference	1250
16.18include/DeviceArrivalEventHandler.h File Reference	1251
16.19include/DeviceEventHandler.h File Reference	1252
16.20include/DeviceEventUtility.h File Reference	1253
16.21include/DeviceRemovalEventHandler.h File Reference	1254
16.22include/EventHandler.h File Reference	1255
16.23include/Exception.h File Reference	1255
16.24include/Image.h File Reference	1256
16.25include/ImageEventHandler.h File Reference	1257
16.26include/ImagePtr.h File Reference	1257
16.27include/ImageStatistics.h File Reference	1258

16.28include/ImageUtility.h File Reference	1259
16.29include/ImageUtilityCCM.h File Reference	1260
16.30include/ImageUtilityHeatmap.h File Reference	1260
16.31include/ImageUtilityPolarization.h File Reference	1261
16.32include/Interface.h File Reference	1262
16.33include/Interface/ICameraBase.h File Reference	1262
16.34include/Interface/ICameraList.h File Reference	1263
16.35include/Interface/IChunkData.h File Reference	1264
16.36include/Interface/IDeviceArrivalEventHandler.h File Reference	1265
16.37include/Interface/IDeviceEventHandler.h File Reference	1266
16.38include/Interface/IDeviceRemovalEventHandler.h File Reference	1266
16.39include/Interface/IImage.h File Reference	1267
16.40include/Interface/IImageEventHandler.h File Reference	1268
16.41include/Interface/IImageStatistics.h File Reference	1268
16.42include/Interface/IInterface.h File Reference	1269
16.43include/Interface/IInterfaceArrivalEventHandler.h File Reference	1270
16.44include/Interface/IInterfaceEventHandler.h File Reference	1271
16.45include/Interface/IInterfaceList.h File Reference	1271
16.46include/Interface/IInterfaceRemovalEventHandler.h File Reference	1272
16.47include/Interface/ILoggingEventHandler.h File Reference	1273
16.48include/Interface/IStream.h File Reference	1273
16.49include/Interface/ISystem.h File Reference	1274
16.50include/Interface/ISystemEventHandler.h File Reference	1275
16.51include/InterfaceArrivalEventHandler.h File Reference	1276
16.52include/InterfaceEventHandler.h File Reference	1277
16.53include/InterfaceList.h File Reference	1278
16.54include/InterfacePtr.h File Reference	1278
16.55include/InterfaceRemovalEventHandler.h File Reference	1279
16.56include/LoggingEventData.h File Reference	1280
16.57include/LoggingEventDataPtr.h File Reference	1280

16.58include/LoggingEventHandler.h File Reference	1281
16.59include/SpinGenApi/Autovector.h File Reference	1282
16.60include/SpinGenApi/Base.h File Reference	1283
16.61include/SpinGenApi/BooleanNode.h File Reference	1284
16.62include/SpinGenApi/CategoryNode.h File Reference	1285
16.63include/SpinGenApi/ChunkAdapter.h File Reference	1286
16.64include/SpinGenApi/ChunkAdapterDcam.h File Reference	1287
16.65include/SpinGenApi/ChunkAdapterGeneric.h File Reference	1288
16.66include/SpinGenApi/ChunkAdapterGEV.h File Reference	1289
16.67include/SpinGenApi/ChunkAdapterU3V.h File Reference	1290
16.68include/SpinGenApi/ChunkPort.h File Reference	1291
16.69include/SpinGenApi/CommandNode.h File Reference	1291
16.70include/SpinGenApi/Compatibility.h File Reference	1292
16.70.1 Macro Definition Documentation	1292
16.70.1.1 FMT_I64	1292
16.71include/SpinGenApi/Container.h File Reference	1293
16.72include/SpinGenApi/Counter.h File Reference	1293
16.73include/SpinGenApi/EnumClasses.h File Reference	1294
16.74include/SpinGenApi/EnumEntryNode.h File Reference	1295
16.75include/SpinGenApi/EnumNode.h File Reference	1296
16.76include/SpinGenApi/EnumNodeT.h File Reference	1297
16.77include/SpinGenApi/EventAdapter.h File Reference	1297
16.78include/SpinGenApi/EventAdapter1394.h File Reference	1298
16.79include/SpinGenApi/EventAdapterGeneric.h File Reference	1299
16.80include/SpinGenApi/EventAdapterGEV.h File Reference	1299
16.81include/SpinGenApi/EventAdapterU3V.h File Reference	1300
16.82include/SpinGenApi/EventPort.h File Reference	1301
16.83include/SpinGenApi/Filestream.h File Reference	1302
16.84include/SpinGenApi/FloatNode.h File Reference	1303
16.85include/SpinGenApi/FloatRegNode.h File Reference	1304

16.86include/SpinGenApi/GCBase.h File Reference	1305
16.87include/SpinGenApi/GCString.h File Reference	1305
16.87.1 Macro Definition Documentation	1306
16.87.1.1 GCSTRING_NPOS	1306
16.87.2 Function Documentation	1306
16.87.2.1 operator<<()	1307
16.87.2.2 operator>>()	1307
16.88include/SpinGenApi/GCStringVector.h File Reference	1307
16.89include/SpinGenApi/GCSynch.h File Reference	1308
16.90include/SpinGenApi/GCTypes.h File Reference	1309
16.90.1 Macro Definition Documentation	1310
16.90.1.1 __STDC_CONSTANT_MACROS	1310
16.90.1.2 __STDC_LIMIT_MACROS	1310
16.90.1.3 GC_INT32_MAX	1310
16.90.1.4 GC_INT32_MIN	1310
16.90.1.5 GC_INT64_MAX	1310
16.90.1.6 GC_INT64_MIN	1311
16.90.1.7 GC_INT8_MAX	1311
16.90.1.8 GC_INT8_MIN	1311
16.90.1.9 GC_UINT32_MAX	1311
16.90.1.10GC_UINT64_MAX	1311
16.90.1.11GC_UINT8_MAX	1311
16.91include/SpinGenApi/GCUtilities.h File Reference	1312
16.91.1 Macro Definition Documentation	1313
16.91.1.1 __ERR__	1314
16.91.1.2 __LINE_STR__	1314
16.91.1.3 __LOCATION__	1314
16.91.1.4 __OUTPUT_FORMATER__	1314
16.91.1.5 __TODO__	1314
16.91.1.6 __WARN__	1314

16.91.1.7 _TO_STRING	1314
16.91.1.8 EXPAND_TO_STRINGISE	1315
16.91.1.9 GC_COUNTOF	1315
16.91.1.10GENICAM_DEPRECATED	1315
16.91.1.11GENICAM_UNUSED	1315
16.91.1.12USE_TEMP_CACHE_FILE [1/2]	1315
16.91.1.13USE_TEMP_CACHE_FILE [2/2]	1315
16.92include/SpinGenApi/IBoolean.h File Reference	1316
16.93include/SpinGenApi/ICategory.h File Reference	1317
16.94include/SpinGenApi/IChunkPort.h File Reference	1318
16.95include/SpinGenApi/ICommand.h File Reference	1319
16.96include/SpinGenApi/IDestroy.h File Reference	1320
16.97include/SpinGenApi/IDeviceInfo.h File Reference	1321
16.98include/SpinGenApi/IEnumEntry.h File Reference	1322
16.99include/SpinGenApi/IEnumeration.h File Reference	1323
16.100include/SpinGenApi/IEnumerationT.h File Reference	1324
16.101include/SpinGenApi/IFloat.h File Reference	1325
16.102include/SpinGenApi/IInteger.h File Reference	1327
16.103include/SpinGenApi/INode.h File Reference	1328
16.104include/SpinGenApi/INodeMap.h File Reference	1331
16.105include/SpinGenApi/INodeMapDyn.h File Reference	1333
16.106include/SpinGenApi/IntegerNode.h File Reference	1334
16.107include/SpinGenApi/IntRegNode.h File Reference	1335
16.108include/SpinGenApi/IPort.h File Reference	1336
16.109include/SpinGenApi/IPortConstruct.h File Reference	1337
16.110include/SpinGenApi/IPortRecorder.h File Reference	1338
16.111include/SpinGenApi/IRegister.h File Reference	1339
16.112include/SpinGenApi/ISelector.h File Reference	1340
16.113include/SpinGenApi/ISelectorDigit.h File Reference	1341
16.114include/SpinGenApi/IString.h File Reference	1342

16.115	include/SpinGenApi/IValue.h File Reference	1343
16.116	include/SpinGenApi/Node.h File Reference	1344
16.117	include/SpinGenApi/NodeCallback.h File Reference	1345
16.118	include/SpinGenApi/NodeCallbackImpl.h File Reference	1347
16.119	include/SpinGenApi/NodeMap.h File Reference	1347
16.120	include/SpinGenApi/NodeMapFactory.h File Reference	1348
16.121	include/SpinGenApi/NodeMapRef.h File Reference	1349
16.122	include/SpinGenApi/Persistence.h File Reference	1350
16.123	include/SpinGenApi/Pointer.h File Reference	1351
16.124	include/SpinGenApi/PortImpl.h File Reference	1353
16.125	include/SpinGenApi/PortNode.h File Reference	1354
16.126	include/SpinGenApi/PortRecorder.h File Reference	1355
16.127	include/SpinGenApi/PortReplay.h File Reference	1356
16.128	include/SpinGenApi/PortWriteList.h File Reference	1357
16.129	include/SpinGenApi/Reference.h File Reference	1358
16.130	include/SpinGenApi/RegisterNode.h File Reference	1359
16.131	include/SpinGenApi/RegisterPortImpl.h File Reference	1359
16.132	include/SpinGenApi/SelectorSet.h File Reference	1360
16.133	include/SpinGenApi/SpinnakerGenApi.h File Reference	1360
16.134	include/SpinGenApi/SpinTestCamera.h File Reference	1361
16.135	include/SpinGenApi/StringNode.h File Reference	1361
16.136	include/SpinGenApi/StringRegNode.h File Reference	1362
16.137	include/SpinGenApi/StructPort.h File Reference	1363
16.138	include/SpinGenApi/Synch.h File Reference	1363
16.139	include/SpinGenApi/Types.h File Reference	1364
16.139.1	Macro Definition Documentation	1367
16.139.1.1	interface	1367
16.140	include/SpinGenApi/ValueNode.h File Reference	1367
16.141	include/Spinnaker.h File Reference	1368
16.142	include/SpinnakerDefs.h File Reference	1368

16.143	include/SpinnakerDirectShow.h File Reference	1373
16.143.1	Variable Documentation	1375
16.143.1.1	MAX_LENGTH	1375
16.144	include/SpinnakerPlatform.h File Reference	1375
16.145	include/SpinUpdate.h File Reference	1376
16.145.1	Macro Definition Documentation	1376
16.145.1.1	SPINUPDATE_API	1376
16.145.2	Function Documentation	1377
16.145.2.1	GetErrorMessage()	1377
16.145.2.2	SetMessageCallback()	1377
16.145.2.3	SetProgressCallback()	1377
16.145.2.4	UpdateFirmware()	1377
16.145.2.5	UpdateFirmwareConsole()	1377
16.145.2.6	UpdateFirmwareGUI()	1378
16.145.3	Variable Documentation	1378
16.145.3.1	UpdatorMessageCallback	1378
16.145.3.2	UpdatorProgressCallback	1378
16.146	include/SpinVideo.h File Reference	1378
16.147	include/SpinVideoDefs.h File Reference	1379
16.148	include/System.h File Reference	1380
16.148.1	Macro Definition Documentation	1381
16.148.1.1	FLIR_SPINNAKER_VERSION_BUILD	1381
16.148.1.2	FLIR_SPINNAKER_VERSION_MAJOR	1381
16.148.1.3	FLIR_SPINNAKER_VERSION_MINOR	1381
16.148.1.4	FLIR_SPINNAKER_VERSION_TYPE	1381
16.149	include/SystemEventHandler.h File Reference	1382
16.150	include/SystemPtr.h File Reference	1382
16.151	include/TransportLayerDefs.h File Reference	1383
16.152	include/TransportLayerDevice.h File Reference	1385
16.153	include/TransportLayerInterface.h File Reference	1385

16.154	include/TransportLayerStream.h File Reference	1386
16.155	include/TransportLayerSystem.h File Reference	1387
16.156	rc/Acquisition/Acquisition.cpp File Reference	1387
16.156	Function Documentation	1387
16.156.1.1	AcquireImages()	1388
16.156.1.2	main()	1388
16.156.1.3	PrintDeviceInfo()	1388
16.156.1.4	RunSingleCamera()	1388
16.157	rc/Acquisition/resource.h File Reference	1388
16.158	rc/AcquisitionMultipleCameraRecovery/resource.h File Reference	1388
16.159	rc/AcquisitionMultipleCamerasWriteToFile/resource.h File Reference	1388
16.160	rc/AcquisitionMultipleThread/resource.h File Reference	1388
16.161	rc/ActionCommand/resource.h File Reference	1388
16.162	rc/BufferHandling/resource.h File Reference	1388
16.163	rc/ChunkData/resource.h File Reference	1388
16.164	rc/Compression/resource.h File Reference	1389
16.165	rc/CounterAndTimer/resource.h File Reference	1389
16.166	rc/DeviceEvents/resource.h File Reference	1389
16.167	rc/Enumeration/resource.h File Reference	1389
16.168	rc/Enumeration_QuickSpin/resource.h File Reference	1389
16.169	rc/EnumerationEvents/resource.h File Reference	1389
16.170	rc/ExceptionHandling/resource.h File Reference	1389
16.171	rc/Exposure/resource.h File Reference	1389
16.172	rc/Exposure_QuickSpin/resource.h File Reference	1389
16.173	rc/FileAccess_QuickSpin/resource.h File Reference	1389
16.174	rc/GigEVisionPerformance/resource.h File Reference	1389
16.175	rc/ImageEvents/resource.h File Reference	1389
16.176	rc/ImageFormatControl/resource.h File Reference	1389
16.177	rc/ImageFormatControl_QuickSpin/resource.h File Reference	1389
16.178	rc/Inference/resource.h File Reference	1389

16.179rc/Logging/resource.h File Reference	1389
16.180rc/LogicBlock/resource.h File Reference	1389
16.181rc/LookupTable/resource.h File Reference	1390
16.182rc/NodeMapCallback/resource.h File Reference	1390
16.183rc/NodeMapInfo/resource.h File Reference	1390
16.184rc/Polarization/resource.h File Reference	1390
16.185rc/SaveToAvi/resource.h File Reference	1390
16.186rc/Sequencer/resource.h File Reference	1390
16.187rc/SerialRxTx/resource.h File Reference	1390
16.188rc/Trigger/resource.h File Reference	1390
16.189rc/Trigger_QuickSpin/resource.h File Reference	1390
16.190rc/Acquisition/stdafx.cpp File Reference	1390
16.191rc/ActionCommand/stdafx.cpp File Reference	1391
16.192rc/BufferHandling/stdafx.cpp File Reference	1391
16.193rc/CounterAndTimer/stdafx.cpp File Reference	1392
16.194rc/DeviceEvents/stdafx.cpp File Reference	1392
16.195rc/Enumeration/stdafx.cpp File Reference	1393
16.196rc/Enumeration_QuickSpin/stdafx.cpp File Reference	1393
16.197rc/ExceptionHandling/stdafx.cpp File Reference	1394
16.198rc/Exposure/stdafx.cpp File Reference	1394
16.199rc/Exposure_QuickSpin/stdafx.cpp File Reference	1395
16.200rc/FileAccess_QuickSpin/stdafx.cpp File Reference	1396
16.201rc/GigEVisionPerformance/stdafx.cpp File Reference	1397
16.202rc/NodeMapInfo/stdafx.cpp File Reference	1397
16.203rc/Sequencer/stdafx.cpp File Reference	1398
16.204rc/SerialRxTx/stdafx.cpp File Reference	1398
16.205rc/Acquisition/stdafx.h File Reference	1399
16.206rc/ActionCommand/stdafx.h File Reference	1400
16.207rc/BufferHandling/stdafx.h File Reference	1401
16.208rc/CounterAndTimer/stdafx.h File Reference	1402

16.209rc/DeviceEvents/stdafx.h File Reference	1403
16.210rc/Enumeration/stdafx.h File Reference	1404
16.211rc/Enumeration_QuickSpin/stdafx.h File Reference	1405
16.212rc/ExceptionHandling/stdafx.h File Reference	1406
16.213rc/Exposure/stdafx.h File Reference	1407
16.214rc/Exposure_QuickSpin/stdafx.h File Reference	1408
16.215rc/FileAccess_QuickSpin/stdafx.h File Reference	1409
16.216rc/GigEVisionPerformance/stdafx.h File Reference	1410
16.217rc/ImageFormatControl/stdafx.h File Reference	1411
16.218rc/ImageFormatControl_QuickSpin/stdafx.h File Reference	1411
16.219rc/NodeMapInfo/stdafx.h File Reference	1412
16.220rc/Polarization/stdafx.h File Reference	1413
16.221rc/Sequencer/stdafx.h File Reference	1413
16.222rc/SerialRxTx/stdafx.h File Reference	1414
16.223rc/Acquisition/targetver.h File Reference	1415
16.224rc/ActionCommand/targetver.h File Reference	1416
16.225rc/BufferHandling/targetver.h File Reference	1417
16.226rc/CounterAndTimer/targetver.h File Reference	1418
16.227rc/DeviceEvents/targetver.h File Reference	1419
16.228rc/Enumeration/targetver.h File Reference	1420
16.229rc/Enumeration_QuickSpin/targetver.h File Reference	1421
16.230rc/ExceptionHandling/targetver.h File Reference	1422
16.231rc/Exposure/targetver.h File Reference	1423
16.232rc/Exposure_QuickSpin/targetver.h File Reference	1424
16.233rc/FileAccess_QuickSpin/targetver.h File Reference	1425
16.234rc/GigEVisionPerformance/targetver.h File Reference	1426
16.235rc/NodeMapInfo/targetver.h File Reference	1427
16.236rc/Sequencer/targetver.h File Reference	1428
16.237rc/SerialRxTx/targetver.h File Reference	1429
16.238rc/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp File Reference	1429

16.238. Function Documentation	1430
16.238.1.1ConfigureCamera()	1430
16.238.1.2ConfigureUserSet1()	1430
16.238.1.3GetDeviceSerial()	1430
16.238.1.4main()	1430
16.238.1.5PrintExampleStatistics()	1431
16.238.1.6RefreshCameraList()	1431
16.238.1.7ResetCameraUserSetToDefault()	1431
16.238.1.8SleepyWrapper()	1431
16.238.2/variable Documentation	1431
16.238.2.1cameraGrabInfoMap	1431
16.238.2.2globalCamList	1431
16.239rc/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp File Reference	1432
16.239. Function Documentation	1432
16.239.1.1AcquireImagesAndSaveToFile()	1432
16.239.1.2ConfigureCameras()	1433
16.239.1.3CreateFiles()	1433
16.239.1.4main()	1433
16.239.1.5PrintDeviceInfo()	1433
16.239.1.6RetrieveImagesFromFiles()	1433
16.239.1.7RunCameras()	1433
16.239.2/variable Documentation	1433
16.239.2.1imageInfos	1434
16.239.2.2k_numImages	1434
16.239.2.3DestinationDirectory	1434
16.240rc/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference	1434
16.240. Function Documentation	1434
16.240.1.1AcquireImages()	1434
16.240.1.2main()	1435
16.240.1.3PrintDeviceInfo()	1435

16.240.1.4RunMultipleCameras()	1435
16.241src/ActionCommand/ActionCommand.cpp File Reference	1435
16.241.1Function Documentation	1436
16.241.1.1AcquireImages()	1436
16.241.1.2ConfigureActionControl()	1436
16.241.1.3ConfigureChunkData()	1436
16.241.1.4ConfigureIEEE1588()	1436
16.241.1.5ConfigureInterface()	1436
16.241.1.6ConfigureOtherNodes()	1436
16.241.1.7ConfigureTrigger()	1437
16.241.1.8main()	1437
16.241.1.9PrintDeviceInfo()	1437
16.241.1.10RunMultipleCameras()	1437
16.241.1.11SleepyWrapper()	1437
16.242src/BufferHandling/BufferHandling.cpp File Reference	1437
16.242.1Macro Definition Documentation	1438
16.242.1.1k_numLoops	1438
16.242.1.2numBuffers	1438
16.242.1.3z_numTriggers	1438
16.242.2Function Documentation	1438
16.242.2.1AcquireImages()	1438
16.242.2.2ConfigureTrigger()	1439
16.242.2.3GrabNextImageByTrigger()	1439
16.242.2.4main()	1439
16.242.2.5PrintDeviceInfo()	1439
16.242.2.6ResetTrigger()	1439
16.242.2.7RunSingleCamera()	1439
16.242.2.8SleepyWrapper()	1439
16.243src/ChunkData/ChunkData.cpp File Reference	1440
16.243.1Enumeration Type Documentation	1440

16.243.1.1chunkDataType	1440
16.243.2Function Documentation	1441
16.243.2.1AcquireImages()	1441
16.243.2.2ConfigureChunkData()	1441
16.243.2.3DisableChunkData()	1441
16.243.2.4DisplayChunkData() [1/2]	1441
16.243.2.5DisplayChunkData() [2/2]	1441
16.243.2.6main()	1441
16.243.2.7PrintDeviceInfo()	1442
16.243.2.8RunSingleCamera()	1442
16.243.3Variable Documentation	1442
16.243.3.1chosenChunkData	1442
16.244src/Compression/Compression.cpp File Reference	1442
16.244.1Function Documentation	1443
16.244.1.1AcquireImages()	1443
16.244.1.2DisableImageChunkData()	1443
16.244.1.3DisableImageCompression()	1443
16.244.1.4EnableImageChunkData()	1443
16.244.1.5EnableImageCompression()	1443
16.244.1.6main()	1444
16.244.1.7PrintDeviceInfo()	1444
16.244.1.8ProcessCompressedImagesFromFile()	1444
16.244.1.9RunSingleCamera()	1444
16.244.2Variable Documentation	1444
16.244.2.1enableChunkData	1444
16.245src/CounterAndTimer/CounterAndTimer.cpp File Reference	1444
16.245.1Function Documentation	1445
16.245.1.1AcquireImages()	1445
16.245.1.2ConfigureDigitalIO()	1445
16.245.1.3ConfigureExposureandTrigger()	1445

16.245.1.4	main()	1445
16.245.1.5	PrintDeviceInfo()	1446
16.245.1.6	ResetTrigger()	1446
16.245.1.7	RunSingleCamera()	1446
16.245.1.8	SetupCounterAndTimer()	1446
16.246	src/DeviceEvents/DeviceEvents.cpp File Reference	1446
16.246.1	Enumeration Type Documentation	1447
16.246.1.1	eventType	1447
16.246.2	Function Documentation	1447
16.246.2.1	AcquireImages()	1447
16.246.2.2	ConfigureDeviceEvents()	1447
16.246.2.3	InferenceAvailable()	1448
16.246.2.4	main()	1448
16.246.2.5	PrintDeviceInfo()	1448
16.246.2.6	ResetDeviceEvents()	1448
16.246.2.7	RunSingleCamera()	1448
16.246.3	Variable Documentation	1448
16.246.3.1	chosenEvent	1448
16.247	src/Enumeration/Enumeration.cpp File Reference	1449
16.247.1	Function Documentation	1449
16.247.1.1	main()	1449
16.247.1.2	QueryInterface()	1449
16.248	src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp File Reference	1449
16.248.1	Function Documentation	1450
16.248.1.1	main()	1450
16.248.1.2	QueryInterface()	1450
16.249	src/EnumerationEvents/EnumerationEvents.cpp File Reference	1450
16.249.1	Function Documentation	1450
16.249.1.1	CheckGevEnabled()	1451
16.249.1.2	main()	1451

16.250rc/ExceptionHandling/ExceptionHandling.cpp File Reference	1451
16.250.1 Enumeration Type Documentation	1451
16.250.1.1 exceptionType	1451
16.250.2 Function Documentation	1452
16.250.2.1 causeSpinnakerException()	1452
16.250.2.2 causeStandardException()	1452
16.250.2.3 main()	1452
16.250.3 Variable Documentation	1452
16.250.3.1 chosenException	1452
16.251rc/Exposure/Exposure.cpp File Reference	1452
16.251.1 Function Documentation	1453
16.251.1.1 AcquireImages()	1453
16.251.1.2 ConfigureExposure()	1453
16.251.1.3 main()	1453
16.251.1.4 PrintDeviceInfo()	1453
16.251.1.5 ResetExposure()	1453
16.251.1.6 RunSingleCamera()	1454
16.252rc/Exposure_QuickSpin/Exposure_QuickSpin.cpp File Reference	1454
16.252.1 Function Documentation	1454
16.252.1.1 AcquireImages()	1454
16.252.1.2 ConfigureExposure()	1454
16.252.1.3 main()	1455
16.252.1.4 PrintDeviceInfo()	1455
16.252.1.5 ResetExposure()	1455
16.252.1.6 RunSingleCamera()	1455
16.253rc/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp File Reference	1455
16.253.1 Function Documentation	1456
16.253.1.1 AcquireImages()	1456
16.253.1.2 CloseFile()	1456
16.253.1.3 DownloadImage()	1456

16.253.1.4ExecuteDeleteCommand()	1456
16.253.1.5ExecuteReadCommand()	1456
16.253.1.6ExecuteWriteCommand()	1457
16.253.1.7InitializeSystem()	1457
16.253.1.8main()	1457
16.253.1.9OpenFileToRead()	1457
16.253.1.10OpenFileToWrite()	1457
16.253.1.11PrintDebugMessage()	1457
16.253.1.12PrintDeviceInfo()	1458
16.253.1.13PrintResultMessage()	1458
16.253.1.14PrintUsage()	1458
16.253.1.15ploadImage()	1458
16.253.2Variable Documentation	1458
16.253.2.1_enableDebug	1458
16.253.2.2_fileSelector	1458
16.254rc/GigEVisionPerformance/CpuUtil.cpp File Reference	1458
16.255rc/GigEVisionPerformance/CpuUtil.h File Reference	1459
16.256rc/GigEVisionPerformance/GigEVisionPerformance.cpp File Reference	1460
16.256Function Documentation	1461
16.256.1.1AcquireImages()	1461
16.256.1.2EnableManualFramerate()	1461
16.256.1.3getCameraCategory()	1461
16.256.1.4main()	1462
16.256.1.5ParseArguments()	1462
16.256.1.6PrintAllNodes()	1462
16.256.1.7PrintCPUUsage()	1462
16.256.1.8PrintDataStreamInfo()	1462
16.256.1.9PrintDeviceInfo()	1462
16.256.1.10PrintUsage()	1462
16.256.1.11RunSingleCamera()	1463

16.256.1.1	SetFrameRate()	1463
16.256.2	Variable Documentation	1463
16.256.2.1	argBayerRG	1463
16.256.2.2	argDuration	1463
16.256.2.3	argMaxFrames	1463
16.256.2.4	argNumImages	1463
16.256.2.5	argPacketDelay	1463
16.256.2.6	argPacketSize	1464
16.256.2.7	argPrintUsage	1464
16.256.2.8	argRelease	1464
16.256.2.9	argUserSetFrames	1464
16.256.2.10	argCpuUsageInfo	1464
16.256.2.11	argRelease	1464
16.256.2.12	argNumImagesToGrab	1464
16.256.2.13	argPacketDelayToSet	1464
16.256.2.14	argPacketSizeToSet	1465
16.256.2.15	argPixelFormatToSet	1465
16.256.2.16	argTestDuration	1465
16.256.2.17	argUseDuration	1465
16.256.2.18	argUseMaxFramerate	1465
16.256.2.19	argUserSetFramerate	1465
16.257	src/GigEVisionPerformance/GigEVisionPerformance.h File Reference	1465
16.258	src/ImageEvents/ImageEvents.cpp File Reference	1466
16.258	Function Documentation	1466
16.258.1.1	AcquireImages()	1466
16.258.1.2	ConfigureImageEvents()	1466
16.258.1.3	main()	1467
16.258.1.4	PrintDeviceInfo()	1467
16.258.1.5	ResetImageEvents()	1467
16.258.1.6	RunSingleCamera()	1467

16.258.1.7SleepyWrapper()	1467
16.258.1.8WaitForImages()	1467
16.259rc/ImageFormatControl/ImageFormatControl.cpp File Reference	1468
16.259. Function Documentation	1468
16.259.1.1AcquireImages()	1468
16.259.1.2ConfigureCustomImageSettings()	1468
16.259.1.3main()	1468
16.259.1.4PrintDeviceInfo()	1469
16.259.1.5RunSingleCamera()	1469
16.260rc/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp File Reference	1469
16.260. Function Documentation	1469
16.260.1.1AcquireImages()	1469
16.260.1.2ConfigureCustomImageSettings()	1470
16.260.1.3main()	1470
16.260.1.4PrintDeviceInfo()	1470
16.260.1.5RunSingleCamera()	1470
16.261rc/Inference/Inference.cpp File Reference	1470
16.261. Enumeration Type Documentation	1471
16.261.1.1FileUploadPersistence	1471
16.261.1.2InferenceNetworkType	1472
16.261.2 Function Documentation	1472
16.261.2.1AcquireImages()	1472
16.261.2.2CameraCloseFile()	1472
16.261.2.3CameraDeleteFile()	1472
16.261.2.4CameraOpenFile()	1472
16.261.2.5CameraWriteToFile()	1473
16.261.2.6ConfigureChunkData()	1473
16.261.2.7ConfigureInference()	1473
16.261.2.8ConfigureTestPattern()	1473
16.261.2.9ConfigureTrigger()	1473

16.261.2.1DeleteFileOnCamera()	1473
16.261.2.1DisableChunkData()	1474
16.261.2.1DisableTrigger()	1474
16.261.2.1DisplayChunkData()	1474
16.261.2.1LabelClassification()	1474
16.261.2.1LabelDetection()	1474
16.261.2.1LoadFileIntoMemory()	1474
16.261.2.1Main()	1475
16.261.2.1PrintDeviceInfo()	1475
16.261.2.1RunSingleCamera()	1475
16.261.2.2SetChunkEnable()	1475
16.261.2.2UploadFileToCamera()	1475
16.261.3/variable Documentation	1475
16.261.3.1arrayLabelClassification	1475
16.261.3.2arrayLabelDetection	1476
16.261.3.3chosenFileUploadPersistence	1476
16.261.3.4chosenInferenceNetworkType	1476
16.261.3.5injectedImageFilePath	1476
16.261.3.6injectedImageHeight	1476
16.261.3.7injectedImageWidth	1477
16.261.3.8networkFilePath	1477
16.262rc/Logging/Logging.cpp File Reference	1477
16.262.1Function Documentation	1477
16.262.1.1main()	1478
16.262.2/variable Documentation	1478
16.262.2.1k_LoggingLevel	1478
16.263rc/LogicBlock/LogicBlock.cpp File Reference	1478
16.263.1Function Documentation	1478
16.263.1.1AcquireImages()	1478
16.263.1.2ConfigureLogicBlock()	1479

16.263.1.3ConfigureTrigger()	1479
16.263.1.4GrabTwoImages()	1479
16.263.1.5main()	1479
16.263.1.6PrintDeviceInfo()	1479
16.263.1.7ResetExposure()	1479
16.263.1.8ResetTrigger()	1479
16.263.1.9RunSingleCamera()	1480
16.264rc/LookupTable/LookupTable.cpp File Reference	1480
16.264. Function Documentation	1480
16.264.1.1AcquireImages()	1480
16.264.1.2ConfigureLookupTables()	1480
16.264.1.3main()	1481
16.264.1.4PrintDeviceInfo()	1481
16.264.1.5PrintRetrieveNodeFailure()	1481
16.264.1.6ResetLookupTables()	1481
16.264.1.7RunSingleCamera()	1481
16.265rc/NodeMapCallback/NodeMapCallback.cpp File Reference	1481
16.265. Function Documentation	1482
16.265.1.1ChangeHeightAndGain()	1482
16.265.1.2ConfigureCallbacks()	1482
16.265.1.3main()	1482
16.265.1.4OnGainNodeUpdate()	1482
16.265.1.5OnHeightNodeUpdate()	1483
16.265.1.6PrintDeviceInfo()	1483
16.265.1.7ResetCallbacks()	1483
16.265.1.8RunSingleCamera()	1483
16.266rc/NodeMapInfo/NodeMapInfo.cpp File Reference	1483
16.266. Enumeration Type Documentation	1484
16.266.1.1readType	1484
16.266.2Function Documentation	1484

16.266.2.1	Indent()	1484
16.266.2.2	main()	1485
16.266.2.3	PrintBooleanNode()	1485
16.266.2.4	PrintCategoryNodeAndAllFeatures()	1485
16.266.2.5	PrintCommandNode()	1485
16.266.2.6	PrintEnumerationNodeAndCurrentEntry()	1485
16.266.2.7	PrintEnumerationSelector()	1485
16.266.2.8	PrintFloatNode()	1486
16.266.2.9	PrintIntegerNode()	1486
16.266.2.10	PrintNode()	1486
16.266.2.11	PrintStringNode()	1486
16.266.2.12	PrintValueNode()	1486
16.266.2.13	RunSingleCamera()	1486
16.266.3	Variable Documentation	1486
16.266.3.1	chosenRead	1487
16.266.3.2	maxChars	1487
16.267	src/Polarization/Polarization.cpp File Reference	1487
16.267	Function Documentation	1488
16.267.1.1	AcquireImages()	1488
16.267.1.2	ConfigureStream()	1488
16.267.1.3	CreateAndSaveAolpDolpImages()	1488
16.267.1.4	CreateAndSaveGlareReducedImage()	1488
16.267.1.5	CreateAndSaveStokesImages()	1488
16.267.1.6	CreateHeatmapImages()	1488
16.267.1.7	CreateNormalizedImage()	1489
16.267.1.8	ExtractAndSavePolarQuadImages()	1489
16.267.1.9	GetQuadFileNameAppendage()	1489
16.267.1.10	main()	1489
16.267.1.11	PrintDeviceInfo()	1489
16.267.1.12	RunSingleCamera()	1489

16.267.1.1	SaveImage()	1490
16.267.2	Variable Documentation	1490
16.267.2.1	PixelFormatColor	1490
16.268	src/SaveToAvi/SaveToAvi.cpp File Reference	1490
16.268.1	Enumeration Type Documentation	1490
16.268.1.1	videoType	1490
16.268.2	Function Documentation	1491
16.268.2.1	AcquireImages()	1491
16.268.2.2	main()	1491
16.268.2.3	PrintDeviceInfo()	1491
16.268.2.4	RunSingleCamera()	1491
16.268.2.5	SaveVectorToVideo()	1491
16.268.3	Variable Documentation	1492
16.268.3.1	chosenVideoType	1492
16.269	src/Sequencer/Sequencer.cpp File Reference	1492
16.269.1	Function Documentation	1492
16.269.1.1	AcquireImages()	1492
16.269.1.2	ConfigureSequencerPartOne()	1493
16.269.1.3	ConfigureSequencerPartTwo()	1493
16.269.1.4	main()	1493
16.269.1.5	PrintDeviceInfo()	1493
16.269.1.6	PrintRetrieveNodeFailure()	1493
16.269.1.7	ResetSequencer()	1493
16.269.1.8	RunSingleCamera()	1493
16.269.1.9	SetSingleState()	1494
16.270	src/SerialRxTx/SerialRxTx.cpp File Reference	1494
16.270.1	Macro Definition Documentation	1494
16.270.1.1	COM_PORT_COUNT_MAX	1495
16.270.1.2	DATA_BITS	1495
16.270.1.3	MILLISECOND	1495

16.270.1.4	SERIAL_PORT_BAUD_RATE	1495
16.270.1.5	SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND	1495
16.270.1.6	SERIAL_PORT_DELAY	1495
16.270.1.7	SERIAL_PORT_PARITY_BITS	1495
16.270.1.8	SERIAL_PORT_STOP_BITS	1495
16.270.1.9	TWO_SECOND_DELAY	1496
16.270.2	Function Documentation	1496
16.270.2.1	CleanUp()	1496
16.270.2.2	ConfigureDevice()	1496
16.270.2.3	main()	1496
16.270.2.4	PrintDeviceInfo()	1496
16.270.2.5	RunSingleCamera()	1496
16.270.2.6	SerialRx()	1497
16.270.2.7	SerialTx()	1497
16.271	src/Trigger/Trigger.cpp File Reference	1497
16.271.1	Enumeration Type Documentation	1498
16.271.1.1	triggerType	1498
16.271.2	Function Documentation	1499
16.271.2.1	AcquireImages()	1499
16.271.2.2	ConfigureTrigger()	1499
16.271.2.3	GrabNextImageByTrigger()	1499
16.271.2.4	main()	1499
16.271.2.5	PrintDeviceInfo()	1499
16.271.2.6	ResetTrigger()	1500
16.271.2.7	RunSingleCamera()	1500
16.271.3	Variable Documentation	1500
16.271.3.1	chosenTrigger	1500
16.272	src/Trigger_QuickSpin/Trigger_QuickSpin.cpp File Reference	1500
16.272.1	Enumeration Type Documentation	1501
16.272.1.1	triggerType	1501
16.272.2	Function Documentation	1501
16.272.2.1	AcquireImages()	1501
16.272.2.2	ConfigureTrigger()	1501
16.272.2.3	GrabNextImageByTrigger()	1501
16.272.2.4	main()	1502
16.272.2.5	PrintDeviceInfo()	1502
16.272.2.6	ResetTrigger()	1502
16.272.2.7	RunSingleCamera()	1502
16.272.3	Variable Documentation	1502
16.272.3.1	chosenTrigger	1502

17 Example Documentation	1503
17.1 Acquisition.cpp	1503
17.2 AcquisitionMultipleCameraRecovery.cpp	1503
17.3 AcquisitionMultipleCamerasWriteToFiles.cpp	1504
17.4 AcquisitionMultipleThread.cpp	1504
17.5 ActionCommand.cpp	1504
17.6 BufferHandling.cpp	1504
17.7 ChunkData.cpp	1505
17.8 Compression.cpp	1505
17.9 CounterAndTimer.cpp	1505
17.10DeviceEvents.cpp	1505
17.11Enumeration.cpp	1506
17.12Enumeration_QuickSpin.cpp	1506
17.13EnumerationEvents.cpp	1506
17.14ExceptionHandling.cpp	1507
17.15Exposure.cpp	1507
17.16Exposure_QuickSpin.cpp	1507
17.17FileAccess_Quickspin.cpp	1508
17.18GigEVisionPerformance.cpp	1508
17.19ImageEvents.cpp	1508
17.20ImageFormatControl.cpp	1508
17.21ImageFormatControl_QuickSpin.cpp	1509
17.22Inference.cpp	1509
17.23Logging.cpp	1509
17.24LogicBlock.cpp	1510
17.25LookupTable.cpp	1510
17.26NodeMapCallback.cpp	1510
17.27NodeMapInfo.cpp	1511
17.28Polarization.cpp	1511
17.29SaveToAvi.cpp	1511
17.30Sequencer.cpp	1512
17.31SerialRxTx.cpp	1512
17.32Trigger.cpp	1513
17.33Trigger_QuickSpin.cpp	1513
Index	1515

Chapter 1

Getting Started

The [Spinnaker](#) application programming interface (API) is used to interface with FLIR's USB3 Vision and GigE Vision cameras.

- [Benefits of Spinnaker](#)
- [Software Licensing Information](#)
- [Software Maintenance Policy](#)
- [FlyCapture2 Feature Comparison with Spinnaker](#)
- [Programmer's Guide](#)
- [SpinViewGuide](#)
- [Working with GenICam GenTL Devices](#)

Chapter 2

Programmer's Guide

Please see (<http://softwareservices.flir.com/Spinnaker/latest/page2.html>) for the latest version of this document

Chapter 3

Benefits of Spinnaker

Please see (<http://softwareservices.flir.com/Spinnaker/latest/index.html>) for the latest version of this document

Chapter 4

FlyCapture2 Feature Comparison with Spinnaker

Please see (<http://softwareservices.flir.com/Spinnaker/latest/page3.html>) for the latest version of this document

Chapter 5

Working with GenICam GenTL Devices

5.1 GenTL Overview

FLIR GenTL Producer is a software driver that implements the GenICam™ GenTL 1.5 standard (<https://www.emva.org/>). It allows users to enumerate, communicate and stream from FLIR GigE Vision and USB3 Vision devices in a generic way independent from the underlying transport technology. This allows third-party software such as MATLAB (<https://www.mathworks.com>) and other software libraries to work with FLIR devices in a transport layer agnostic way. These applications are referred to as "GenTL Consumers," which directly use one or more GenTL Producers.

NOTE: Consumer applications must be aware of differences in device capabilities and be prepared to handle specific device models differently.

5.2 Installation

In order to use a FLIR GenTL producer, it needs to be properly registered and installed on the system. **The FLIR Producer comes packaged with the full Spinnaker SDK installer as of 2.x or newer.**

The GenTL Producer is provided as a platform dependent, dynamic loadable library file with the `.cti` ("Common Transport Interface") extension.

The Spinnaker SDK installer stores the folder paths for 32-bit and 64-bit GenTL Producers (`.cti` files) in environment variables named `GENICAM_GENTL32_PATH` and `GENICAM_GENTL64_PATH`, respectively. If there are multiple GenTL Producers installed on the system, path entries must be separated by `;` on Windows and `:` on UNIX-like systems.

NOTE: A 32bit GenTL consumer application will require a 32-bit GenTL producer and a 64-bit application will require a 64-bit producer library.

5.3 Troubleshooting

5.3.1 Enable FLIR GenTL Logging

FLIR GenTL Logging can be enabled if a configuration file with the name "log4cpp.gentl.property" resides in the path of where the consumer application executes from. For MATLAB, this is where the working directory is set and may default to the "Downloads" folder on Windows.

Sample log4cpp.gentl.property configuration file:

```
# FLIR GenTL Property Configuration file
log4cpp.rootCategory=ERROR, rootAppender
log4cpp.category.GenTLCategory=ERROR, GenTLCategory

log4cpp.appender.rootAppender=ConsoleAppender
log4cpp.appender.rootAppender.layout=PatternLayout
log4cpp.appender.rootAppender.layout.ConversionPattern=[%p] %d [%t] %m%n

log4cpp.appender.GenTLCategory=RollingFileAppender
log4cpp.appender.GenTLCategory.fileName=$(ALLUSERSPROFILE)\Spinnaker\Logs\GenTL.log
log4cpp.appender.GenTLCategory.append=true
log4cpp.appender.GenTLCategory.maxFileSize=1000000
log4cpp.appender.GenTLCategory.maxBackupIndex=5
log4cpp.appender.GenTLCategory.layout=PatternLayout
log4cpp.appender.GenTLCategory.layout.ConversionPattern=[%p] %d [%t] %m%n
```

5.3.2 USB3 Device Image Tearing

Image tearing could occur with certain USB3 host controllers when streaming with a GenTL producer. To work around the issue, make sure the size of each buffer announced to the FLIR GenTL producer is 1024 bytes aligned. The size of each buffer should be $(\text{bufferSize} + 1024 - 1) / 1024 * 1024$ where 1024 is the USB3 packet transfer size.

For more information about image tearing causes and solutions, please refer to: <https://www.flir.com/support-center/iis/machine-vision/application-note/image-tearing-causes-and-solutions/>

Chapter 6

Software Licensing Information

Table 6.1 License table

Component	License
Spinnaker	Copyright (c) 2001-2020 FLIR Systems, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FLIR SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
GenICam	GenICam License
AdapterManager	The Code Project Open License (CPO-OL)
Make ListView.ScrollIntoView Scroll the Item into the Center of the ListView	WP:CC-BY-SA License
Work with Bitmaps Faster in C#	The Code Project Open License (CPO-OL) 1.02
FreeImage	FreeImage public license
Boost	Boost Software License
Libusb	LGPLv2.1 License
Libraw1394	LGPLv2.0 License
FFMPEG	LGPLv2.1 License
log4Net	Apache license 2.0
log4Cpp	LGPL License

The licenses mentioned above can also be found in the [Spinnaker](#) installed license folder.

Chapter 7

Software Maintenance Policy

7.1 GenTL Overview

This document outlines the FLIR maintenance policy for [Spinnaker](#) Software Development Kit (SDK). FLIR regularly provides SDK updates that may contain support for new or updated features, enhancements, updated drivers, updated examples, bug fixes or documentation updates. Updates may also address changes with introducing and/or deprecating language runtimes, operating systems and dependencies.

We recommend users to stay up-to-date with SDK releases to keep up with the latest features, bug fixes and performance improvements. Continued use of an unsupported SDK version is not recommended and is done at the user's discretion.

[Spinnaker](#) SDK releases are published through our website and can be found here: <https://www.flir.ca/products/spinnaker-sdk/>

7.2 Platform Support Policy

7.2.1 Windows Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against the latest available version of Windows x86/x64. The latest three versions of Visual Studio compiler toolchain are supported on Windows. Only the latest compiler toolchain on the latest available version of Windows are being actively tested.

7.2.2 Linux Desktop Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against the latest two LTS versions of Ubuntu x86/x64. Only the latest x64 LTS version of Ubuntu is being actively tested.

7.2.3 Linux Embedded Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against the latest supported LTS version of Ubuntu ARMHF/ARM64 for a specific board. Only the latest LTS Ubuntu version on an ARM64 board is being actively tested. Contact sales if you need support for a specific embedded board.

7.2.4 MacOS Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against MacOS Mojave (10.14). Contact sales if you need newer MacOS support.

7.3 Versioning Policy

[Spinnaker](#) SDK releases use a modified semantic versioning scheme and is indicated by four sets of numbers separated by periods:

MAJOR.MINOR.0.PATCH

- MAJOR: Version change that can include incompatible API changes
- MINOR: Version change that adds functionality in a backwards-compatible manner
- PATCH: Version change with backwards-compatible fixes

Reference: <https://www.flir.com/support-center/iis/machine-vision/knowledge-base/flir-machi>

Chapter 8

Module Index

8.1 Modules

Here is a list of all modules:

Spinnaker Classes	42
AVI Recorder Class	46
BasePtr Class	49
Camera Class	50
Camera Base Class	51
CameraDefs Class	52
Camera List Class	168
CameraPtr Class	169
ChunkData Class	171
Chunk Data Inference Class	172
Exception Class	184
Image Class	185
ImagePtr Class	187
ImageStatistics Class	188
Image Utility Class	189
Image Utility CCM Class	190
Image Utility Heatmap Class	191
Image Utility Polarization Class	192
Interface Class	193
InterfaceList Class	196
InterfacePtr Class	197
Spinnaker Video Class	237
System Class	239
SystemPtr Class	241
Camera Base Interface Class	255
IChunkData Class	256
IImage Class	257
IImageStatistics Class	258
IInterface Class	259
IInterfaceList Class	260
ISystem Class	261
Spinnaker EventHandler Classes	178
DeviceArrivalEventHandler Class	180
DeviceEventHandler Class	181
DeviceRemovalEventHandler Class	182

EventHandler Class	183
ImageEventHandler Class	186
InterfaceArrivalEventHandler Class	194
InterfaceEventHandler Class	195
InterfaceRemovalEventHandler Class	198
Logging EventHandler Class	199
LoggingEventDataPtr Class	200
LoggingEventHandler Class	201
SystemEventHandler Class	240
Spinnaker Headers	202
Spinnaker.h	219
Spinnaker Definitions	220
SpinnakerDirectShow.h	232
Spinnaker Platform	236
Spinnaker Video Definitions	238
Spinnaker QuickSpin Classes	242
TransportLayerDefs Class	243
TransportLayerDevice Class	251
TransportLayerInterface Class	252
TransportLayerStream Class	253
TransportLayerSystem Class	254
Spinnaker GenApi Classes	262
AutoVector Class	273
BooleanNode Class	277
CategoryNode Class	278
ChunkAdapter Class	279
ChunkAdapterDcam Class	280
ChunkAdapterGeneric Class	281
ChunkAdapterGEV Class	282
ChunkPort Class	283
CommandNode Class	284
Container Class	285
Counter Class	286
EnumClasses Class	287
EnumEntryNode Class	289
EnumNode Class	290
EnumNodeT Class	291
EventAdapter Class	292
EventAdapter1394 Class	293
EventAdapterGeneric Class	294
EventAdapterGEV Class	295
EventAdapterU3V Class	296
EventPort Class	297
Filestream Class	298
FloatNode Class	299
FloatRegNode Class	300
GCString Class	301
GCSynch Class	302
GCTypes Class	303
IntegerNode Class	354
IntRegNode Class	355
IString Class	367
IValue Class	368
Node Class	370
NodeCallback Class	371
NodeMap Class	374
NodeMapFactory Class	375

NodeMapRef Class	377
Persistence Class	378
Pointer Class	379
PortImpl Class	385
PortNode Class	386
PortRecorder Class	387
PortReplay Class	388
PortWriteList Class	389
RegisterNode Class	391
RegisterPortImpl Class	392
SelectorSet Class	393
SpinTestCamera Class	394
StringNode Class	395
StringRegNode Class	396
StructPort Class	397
Synch Class	398
ValueNode Class	410
ChunkAdapterU3V Class	411
IPortRecorder Interface	359
Spinnaker GenApi Interfaces	274
IBase Interface	276
IBoolean Interface	311
ICategory Interfaces	313
IChunkPort Interface	314
ICommand Interface	316
IDestroy Interface	318
IDeviceInfo Interface	319
IEnumEntry Interface	322
IEnumeration Interface	324
IEnumerationT Interface	327
IFloat Interface	329
IInteger Interface	333
INode Interface	335
INodeMap Interface	346
INodeMapDyn Interface	349
IPort Interface	356
IPortConstruct Interface	358
IPortRecorder Interface	359
IRegister Interfaces	361
ISelector Interface	363
ISelectorDigit Interface	364
Reference Interfaces	390
Spinnaker GenApi Utilities	304
GCUilities Utility	305
Spinnaker GenApi Enums	399
Types Enums	400

Chapter 9

Namespace Index

9.1 Namespace List

Here is a list of all namespaces with brief descriptions:

AdapterConfig	413
Conversion	417
CpuUtil	418
PerformanceCounter	418
SecondsCounter	419
Spinnaker	421
Spinnaker::GenApi	460
Spinnaker::GenICam	477
Spinnaker::Video	479

Chapter 10

Hierarchical Index

10.1 Class Hierarchy

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

ActionCommandResult	481
AdapterInfo	483
AttachStatistics_t	487
AutoLock	488
AutoLock	488
AVIOption	489
BasePtr< T, B >	490
BasePtr< Camera, ICameraBase >	490
CameraPtr	672
BasePtr< IImage >	490
ImagePtr	990
BasePtr< IInterface >	490
InterfacePtr	1050
BasePtr< ISystem >	490
SystemPtr	1161
BasePtr< LoggingEventData >	490
LoggingEventDataPtr	1076
basic_istream	
IDevFileStreamBase< CharType, Traits >	908
basic_ostream	
ODevFileStreamBase< CharType, Traits >	1106
basic_streambuf	
IDevFileStreamBuf< CharType, Traits >	911
ODevFileStreamBuf< CharType, Traits >	1108
BMPOption	494
CChunkAdapter	676
CChunkAdapterDcam	679
CChunkAdapterGeneric	682
CChunkAdapterGEV	684
CChunkAdapterU3V	686
CCMSettings	694
CDataStruct	
CTestPortStruct< CDataStruct >	791

CEventAdapter	701
CEventAdapter1394	703
CEventAdapterGeneric	705
CEventAdapterGEV	708
CEventAdapterU3V	710
CGeneric_XMLLoaderParams	722
CNodeMapRefT< GenApi::CGeneric_XMLLoaderParams >	761
CNodeMapRef	758
CGlobalLock	723
CGlobalLockUnlocker	725
CLock	740
CLockEx	745
CLock	742
CLockEx	746
CNodeCallback	747
Function_NodeCallback< Function >	856
Member_NodeCallback< Client, Member >	1081
CNodeMapFactory	750
CompressedImageInfo	769
Counter	771
CPointer< T, B >	773
CPointer< IFloat, IBase >	773
CFloatPtr	720
CPointer< INode, IBase >	773
CpuUsageInfo	784
DCAM_CHECKSUM	795
DCAM_CHUNK_TRAILER	795
DeviceEventExposureEndData	798
DeviceEventInferenceData	804
DeviceEventUtility	805
double_autovector_t	808
EAccessModeClass	811
ECachingModeClass	812
EDisplayNotationClass	813
EEndianessClass	814
EGenApiSchemaVersionClass	816
EInputDirectionClass	817
ENamespaceClass	818
ERepresentationClass	827
ESignClass	828
ESlopeClass	829
EStandardNameSpaceClass	830
EventHandler	832
IDeviceArrivalEventHandler	913
DeviceArrivalEventHandler	796
IInterfaceEventHandler	944
InterfaceEventHandler	1039
InterfaceEventHandlerImpl	1042
InterfaceEventHandlerImpl	1042
IDeviceEventHandler	915
DeviceEventHandler	799
DeviceEventHandlerImpl	802
IDeviceRemovalEventHandler	917
DeviceRemovalEventHandler	806
IInterfaceEventHandler	944
IImageEventHandler	931

ImageEventHandler	983
ImageEventHandlerImpl	985
ImageEventHandlerImpl	985
IInterfaceArrivalEventHandler	942
InterfaceArrivalEventHandler	1036
ISystemEventHandler	1064
SystemEventHandler	1154
SystemEventHandlerImpl	1157
IInterfaceRemovalEventHandler	950
InterfaceRemovalEventHandler	1052
ISystemEventHandler	1064
ILoggingEventHandler	952
LoggingEventHandler	1078
LoggingEventHandlerImpl	1080
EVisibilityClass	835
exception	
AdapterConfigException	482
Exception	836
EYesNoClass	841
FileProtocolAdapter	842
gcstring	858
GrabInfo	867
GVCP_CHUNK_TRAILER	868
GVCP_EVENT_ITEM	869
GVCP_EVENT_ITEM_BASIC	870
GVCP_EVENT_ITEM_EXTENDED_ID	871
GVCP_EVENT_REQUEST	873
GVCP_EVENT_REQUEST_EXTENDED_ID	874
GVCP_EVENTDATA_REQUEST	875
GVCP_EVENTDATA_REQUEST_EXTENDED_ID	876
GVCP_REQUEST_HEADER	877
H264Option	878
IBoolean	
BooleanNode	496
ICameraBase	880
CameraBase	652
Camera	499
ICameraList	889
CameraList	667
ICategory	
CategoryNode	674
IChunkData	893
ChunkData	727
IChunkPort	
PortNode	1113
PortReplay	1122
PortRecorder	1118
ICommand	
CommandNode	766
IDataStream	902
IDeviceInfo	
NodeMap	1096
SpinTestCamera	1132
IEnumEntry	
EnumEntryNode	819
IEnumeration	

EnumNode	822
CEnumerationTRef< EnumT >	695
IEnumerationT	
CEnumerationTRef< EnumT >	695
IFloat	
FloatNode	846
FloatRegNode	853
Image	919
Image	954
ImageStatistics	933
ImageStatistics	992
Integer	
IntegerNode	1025
IntRegNode	1054
IInterface	937
Interface	1031
IInterfaceList	947
InterfaceList	1046
ImageInfo	988
ImageUtility	999
ImageUtilityCCM	1004
ImageUtilityHeatmap	1006
ImageUtilityPolarization	1010
InferenceBoundingBox	1020
InferenceBoundingBoxResult	1020
InferenceBoxCircle	1021
InferenceBoxRect	1021
InferenceBoxRotatedRect	1022
INode	
Node	1085
CSelectorSet	788
PortNode	1113
ValueNode	1205
BooleanNode	496
CategoryNode	674
CommandNode	766
EnumEntryNode	819
EnumNode	822
FloatNode	846
IntegerNode	1025
RegisterNode	1127
FloatRegNode	853
IntRegNode	1054
StringRegNode	1140
StringNode	1136
StringRegNode	1140
INodeMap	
NodeMap	1096
int64_autovector_t	1022
IPersistScript	
CFeatureBag	717
IpInfo	1056
IPortConstruct	
CChunkPort	689
CEventPort	712
CPortImpl	778

CRegisterPortImpl	785
CTestPortStruct< CDataStruct >	791
PortNode	1113
IPortRecorder	
PortNode	1113
PortRecorder	1118
IPortReplay	
CPortImpl	778
PortReplay	1122
IPortWriteList	
CPortWriteList	781
IRegister	
RegisterNode	1127
IString	
StringNode	1136
ISystem	1058
System	1144
IValue	
ValueNode	1205
JPEGOption	1066
JPG2Option	1067
LibraryVersion	1069
LockableObject< Object >::Lock	1070
LockableObject< Object >	1071
LoggingEventData	1072
MJPEGOption	1083
CNodeMapFactory::NodeStatistics_t	1105
PGMOption	1110
PNGOption	1111
PPMOption	1125
SingleChunkData_t	1130
SingleChunkDataStr_t	1131
SpinVideo	1133
TIFFOption	1163
TransportLayerDevice	1165
TransportLayerInterface	1175
TransportLayerStream	1187
TransportLayerSystem	1194
U3V_CHUNK_TRAILER	1201
U3V_COMMAND_HEADER	1202
U3V_EVENT_DATA	1203
U3V_EVENT_MESSAGE	1204
Version_t	1208
TCameraParams	
CNodeMapRefT< TCameraParams >	761

Chapter 11

Class Index

11.1 Class List

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

ActionCommandResult	
Action Command Result	481
AdapterConfigException	482
AdapterInfo	483
AttachStatistics_t	
Delivers information about the attached chunks and nodes	487
AutoLock	488
AutoLock	488
AVIOption	
Options for saving AVI files	489
BasePtr< T, B >	
The base class of the SystemPtr , CameraPtr , InterfacePtr , ImagePtr and LoggingEventDataPtr objects	490
BMPOption	
Options for saving Bitmap image	494
BooleanNode	
Interface for string properties	496
Camera	
The camera object class	499
CameraBase	
The base class for the camera object	652
CameraList	
Used to hold a list of camera objects	667
CameraPtr	
A reference tracked pointer to a camera object	672
CategoryNode	
Interface for string properties	674
CChunkAdapter	
Connects a chunked buffer to a node map	676
CChunkAdapterDcam	
Connects a chunked DCAM buffer to a node map	679
CChunkAdapterGeneric	682
CChunkAdapterGEV	
Connects a chunked DCAM buffer to a node map	684
CChunkAdapterU3V	
Connects a chunked U3V buffer to a node map	686

CChunkPort	
Port attachable to a chunk in a buffer	689
CCMSettings	694
CEnumerationTRef< EnumT >	
Interface for string properties	695
CEventAdapter	
Delivers Events to ports	701
CEventAdapter1394	
Distribute the events to the node map	703
CEventAdapterGeneric	
Connects a generic event to a node map	705
CEventAdapterGEV	
Connects a GigE Event to a node map	708
CEventAdapterU3V	
Connects a U3V Event to a node map	710
CEventPort	
Port attachable to an event	712
CFeatureBag	
Bag holding streamable features of a nodetree	717
CFloatPtr	
SmartPointer for IFloat interface pointer	720
CGeneric_XMLLoaderParams	
Empty base class used by class CNodeMapRef as generic template argument	722
CGlobalLock	
Named global lock which can be used over process boundaries	723
CGlobalLockUnlocker	
Unlocks the global lock object on destruction	725
ChunkData	
The chunk data which contains additional information about an image	727
CLock	
A lock class	740
CLock	
A lock class	742
CLockEx	
This class is for testing purposes only	745
CLockEx	
This class is for testing purposes only	746
CNodeCallback	
Callback body instance for INode pointers	747
CNodeMapFactory	
The node map factory is used for creating node maps from camera description files	750
CNodeMapRef	
SmartPointer for NodeMaps with create function	758
CNodeMapRefT< TCameraParams >	
SmartPointer template for NodeMaps with create function	761
CommandNode	
Interface for string properties	766
CompressedImageInfo	769
Counter	
Definition of a simple Counter class	771
CPointer< T, B >	
Encapsulates a GenApi pointer dealing with the dynamic_cast automatically	773
CPortImpl	
Standard implementation for a port	778
CPortWriteList	
Container holding a list of port write commands	781
CpuUsageInfo	784

CRegisterPortImpl	Standard implementation for a port using a register based transport layer	785
CSelectorSet	The set of selectors selecting a given node	788
CTestPortStruct< CDataStruct >	Implements a register spaces based on a C++ struct	791
DCAM_CHECKSUM		795
DCAM_CHUNK_TRAILER		795
DeviceArrivalEventHandler	An event handler for capturing the device arrival event	796
DeviceEventExposureEndData	Data Fields for Device Event payload for EventExposureEnd	798
DeviceEventHandler	A handler to device events	799
DeviceEventHandlerImpl		802
DeviceEventInferenceData	Data Fields for Device Event payload for EventInference	804
DeviceEventUtility		805
DeviceRemovalEventHandler	An event handler for capturing the device removal event	806
double_autovector_t	Vector of doubles with reference counting	808
EAccessModeClass	Holds conversion methods for the access mode enumeration	811
ECachingModeClass	Holds conversion methods for the caching mode enumeration	812
EDisplayNotationClass	Holds conversion methods for the notation type of floats	813
EEndianessClass	Holds conversion methods for the endianess enumeration	814
EGenApiSchemaVersionClass	Helper class converting EGenApiSchemaVersion from and to string	816
EInputDirectionClass	Holds conversion methods for the notation type of floats	817
ENameSpaceClass	Holds conversion methods for the namespace enumeration	818
EnumEntryNode	Interface for string properties	819
EnumNode	Interface for string properties	822
ERepresentationClass	Holds conversion methods for the representation enumeration	827
ESignClass	Holds conversion methods for the sign enumeration	828
ESlopeClass	Holds conversion methods for the converter formulas	829
EStandardNameSpaceClass	Holds conversion methods for the standard namespace enumeration	830
EventHandler	The base class for all event handler types	832
EVisibilityClass	Holds conversion methods for the visibility enumeration	835
Exception	The Exception object represents an error that is returned from the library	836
EYesNoClass	Holds conversion methods for the standard namespace enumeration	841

FileProtocolAdapter	
Adapter between the <code>std::iostreambuf</code> and the SFNC Features representing the device file system	842
FloatNode	
Interface for string properties	846
FloatRegNode	
Interface for string properties	853
Function_NodeCallback< Function >	
Container for a function pointer	856
gcstring	858
GrabInfo	867
GVCP_CHUNK_TRAILER	
Header of a GVCP request packet	868
GVCP_EVENT_ITEM	
Layout of a GVCP event item (Extended ID flag not set)	869
GVCP_EVENT_ITEM_BASIC	
Layout of a GVCP event item (common to all types)	870
GVCP_EVENT_ITEM_EXTENDED_ID	
Layout of a GVCP event item (Extended ID flag set)	871
GVCP_EVENT_REQUEST	
Layout of a GVCP event request packet (Extended ID flag not set)	873
GVCP_EVENT_REQUEST_EXTENDED_ID	
Layout of a GVCP event request packet (Extended ID flag set)	874
GVCP_EVENTDATA_REQUEST	
Layout of a GVCP event data request packet (Extended ID flag not set)	875
GVCP_EVENTDATA_REQUEST_EXTENDED_ID	
Layout of a GVCP event data request packet (Extended ID flag set)	876
GVCP_REQUEST_HEADER	
Header of a GVCP request packet	877
H264Option	
Options for saving H264 files	878
ICameraBase	
The interface file for base class for the camera object	880
ICameraList	
Used to hold a list of camera objects	889
IChunkData	
The Interface file for ChunkData	893
IDataStream	902
IDevFileStreamBase< CharType, Traits >	908
IDevFileStreamBuf< CharType, Traits >	911
IDeviceArrivalEventHandler	913
IDeviceEventHandler	915
IDeviceRemovalEventHandler	917
Image	
The interface file for Image	919
ImageEventHandler	931
ImageStatistics	
The interface file for image statistics	933
IInterface	
The interface file for Interface	937
IInterfaceArrivalEventHandler	942
IInterfaceEventHandler	944
IInterfaceList	
The interface file for InterfaceList class	947
IInterfaceRemovalEventHandler	950
ILoggingEventHandler	952
Image	
The image object class	954

ImageEventHandler	
A handler for capturing image arrival events	983
ImageEventHandlerImpl	985
ImageInfo	988
ImagePtr	
A reference tracked pointer to an image object	990
ImageStatistics	
Represents image statistics for an image	992
ImageUtility	
Static helper functions for the image object class	999
ImageUtilityCCM	
Static function to create color corrected images from an image object	1004
ImageUtilityHeatmap	
Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16	1006
ImageUtilityPolarization	
Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8	1010
InferenceBoundingBox	
Inference Bounding Boxes data structure	1020
InferenceBoundingBoxResult	
An inference bounding boxes object which holds information about the detected bounding boxes	1020
InferenceBoxCircle	1021
InferenceBoxRect	
Inference Bounding Box Type Data Structures	1021
InferenceBoxRotatedRect	1022
int64_autovector_t	
Vector of integers with reference counting	1022
IntegerNode	
Interface for string properties	1025
Interface	
An interface object which holds a list of cameras	1031
InterfaceArrivalEventHandler	
An event handler for capturing the interface arrival event	1036
InterfaceEventHandler	
A handler to device arrival and removal events on all interfaces	1039
InterfaceEventHandlerImpl	1042
InterfaceList	
A list of the available interfaces on the system	1046
InterfacePtr	
A reference tracked pointer to the interface object	1050
InterfaceRemovalEventHandler	
An event handler for capturing the interface removal event	1052
IntRegNode	
Interface for string properties	1054
IpInfo	1056
ISystem	
The interface file for System	1058
ISystemEventHandler	1064
JPEGOption	
Options for saving JPEG image	1066
JPG2Option	
Options for saving JPEG2000 image	1067
LibraryVersion	
Provides easier access to the current version of Spinnaker	1069
LockableObject< Object >::Lock	
A scopelevel Lock class	1070
LockableObject< Object >	
Instance-Lock for an object	1071

LoggingEventData	
The LoggingEventData object	1072
LoggingEventDataPtr	
A reference tracked pointer to the LoggingEvent object	1076
LoggingEventHandler	
An event handler for capturing the device logging event	1078
LoggingEventHandlerImpl	1080
Member_NodeCallback< Client, Member >	
Container for a member function pointer	1081
MJPGOption	
Options for saving MJPG files	1083
Node	
Class common to all nodes	1085
NodeMap	
Smart pointer template for NodeMaps with create function	1096
CNodeMapFactory::NodeStatistics_t	1105
ODevFileStreamBase< CharType, Traits >	1106
ODevFileStreamBuf< CharType, Traits >	1108
PGMOption	
Options for saving PGM images	1110
PNGOption	
Options for saving PNG images	1111
PortNode	
Interface for value properties	1113
PortRecorder	
Interface for recording write commands on a port	1118
PortReplay	
Interface for replaying write commands on a port	1122
PPMOption	
Options for saving PPM images	1125
RegisterNode	
Interface for string properties	1127
SingleChunkData_t	1130
SingleChunkDataStr_t	1131
SpinTestCamera	1132
SpinVideo	
Provides the functionality for the user to record images to an AVI/MP4 file	1133
StringNode	
Interface for string properties	1136
StringRegNode	
Interface for string properties	1140
System	
The system object is used to retrieve the list of interfaces and cameras available	1144
SystemEventHandler	
A handler to interface arrival and removal events on the system	1154
SystemEventHandlerImpl	1157
SystemPtr	
A reference tracked pointer to a system object	1161
TIFFOption	
Options for saving TIFF images	1163
TransportLayerDevice	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1165
TransportLayerInterface	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1175

TransportLayerStream	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1187
TransportLayerSystem	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1194
U3V_CHUNK_TRAILER	
Header of a GVCP request packet	1201
U3V_COMMAND_HEADER	
U3V/GenCP command header	1202
U3V_EVENT_DATA	
U3V/GenCP EVENT_CMD specific command data	1203
U3V_EVENT_MESSAGE	
Entire event data message (without the variable-sized data field)	1204
ValueNode	
Interface for value properties	1205
Version_t	
Version	1208

Chapter 12

File Index

12.1 File List

Here is a list of all files with brief descriptions:

include/AdapterConfig.h	1211
include/AVIRecorder.h	1213
include/BasePtr.h	1214
include/Camera.h	1215
include/CameraBase.h	1215
include/CameraDefs.h	1216
include/CameraList.h	1248
include/CameraPtr.h	1249
include/ChunkData.h	1249
include/ChunkDataInference.h	1250
include/DeviceArrivalEventHandler.h	1251
include/DeviceEventHandler.h	1252
include/DeviceEventUtility.h	1253
include/DeviceRemovalEventHandler.h	1254
include/EventHandler.h	1255
include/Exception.h	1255
include/Image.h	1256
include/ImageEventHandler.h	1257
include/ImagePtr.h	1257
include/ImageStatistics.h	1258
include/ImageUtility.h	1259
include/ImageUtilityCCM.h	1260
include/ImageUtilityHeatmap.h	1260
include/ImageUtilityPolarization.h	1261
include/Interface.h	1262
include/InterfaceArrivalEventHandler.h	1276
include/InterfaceEventHandler.h	1277
include/InterfaceList.h	1278
include/InterfacePtr.h	1278
include/InterfaceRemovalEventHandler.h	1279
include/LoggingEventData.h	1280
include/LoggingEventDataPtr.h	1280
include/LoggingEventHandler.h	1281
include/Spinnaker.h	1368
include/SpinnakerDefs.h	1368

include/SpinnakerDirectShow.h	1373
include/SpinnakerPlatform.h	1375
include/SpinUpdate.h	1376
include/SpinVideo.h	1378
include/SpinVideoDefs.h	1379
include/System.h	1380
include/SystemEventHandler.h	1382
include/SystemPtr.h	1382
include/TransportLayerDefs.h	1383
include/TransportLayerDevice.h	1385
include/TransportLayerInterface.h	1385
include/TransportLayerStream.h	1386
include/TransportLayerSystem.h	1387
include/Interface/ICameraBase.h	1262
include/Interface/ICameraList.h	1263
include/Interface/IChunkData.h	1264
include/Interface/IDeviceArrivalEventHandler.h	1265
include/Interface/IDeviceEventHandler.h	1266
include/Interface/IDeviceRemovalEventHandler.h	1266
include/Interface/IImage.h	1267
include/Interface/IImageEventHandler.h	1268
include/Interface/IImageStatistics.h	1268
include/Interface/IInterface.h	1269
include/Interface/IInterfaceArrivalEventHandler.h	1270
include/Interface/IInterfaceEventHandler.h	1271
include/Interface/IInterfaceList.h	1271
include/Interface/IInterfaceRemovalEventHandler.h	1272
include/Interface/ILoggingEventHandler.h	1273
include/Interface/IStream.h	1273
include/Interface/ISystem.h	1274
include/Interface/ISystemEventHandler.h	1275
include/SpinGenApi/Autovector.h	1282
include/SpinGenApi/Base.h	1283
include/SpinGenApi/BooleanNode.h	1284
include/SpinGenApi/CategoryNode.h	1285
include/SpinGenApi/ChunkAdapter.h	1286
include/SpinGenApi/ChunkAdapterDcam.h	1287
include/SpinGenApi/ChunkAdapterGeneric.h	1288
include/SpinGenApi/ChunkAdapterGEV.h	1289
include/SpinGenApi/ChunkAdapterU3V.h	1290
include/SpinGenApi/ChunkPort.h	1291
include/SpinGenApi/CommandNode.h	1291
include/SpinGenApi/Compatibility.h	1292
include/SpinGenApi/Container.h	1293
include/SpinGenApi/Counter.h	1293
include/SpinGenApi/EnumClasses.h	1294
include/SpinGenApi/EnumEntryNode.h	1295
include/SpinGenApi/EnumNode.h	1296
include/SpinGenApi/EnumNodeT.h	1297
include/SpinGenApi/EventAdapter.h	1297
include/SpinGenApi/EventAdapter1394.h	1298
include/SpinGenApi/EventAdapterGeneric.h	1299
include/SpinGenApi/EventAdapterGEV.h	1299
include/SpinGenApi/EventAdapterU3V.h	1300
include/SpinGenApi/EventPort.h	1301
include/SpinGenApi/Filestream.h	1302
include/SpinGenApi/FloatNode.h	1303
include/SpinGenApi/FloatRegNode.h	1304

include/SpinGenApi/GCBase.h	1305
include/SpinGenApi/GCString.h	1305
include/SpinGenApi/GCStringVector.h	1307
include/SpinGenApi/GCSynch.h	1308
include/SpinGenApi/GCTypes.h	1309
include/SpinGenApi/GCUtilities.h	1312
include/SpinGenApi/IBoolean.h	1316
include/SpinGenApi/ICategory.h	1317
include/SpinGenApi/IChunkPort.h	1318
include/SpinGenApi/ICommand.h	1319
include/SpinGenApi/IDestroy.h	1320
include/SpinGenApi/IDeviceInfo.h	1321
include/SpinGenApi/IEnumEntry.h	1322
include/SpinGenApi/IEnumeration.h	1323
include/SpinGenApi/IEnumerationT.h	1324
include/SpinGenApi/IFloat.h	1325
include/SpinGenApi/IInteger.h	1327
include/SpinGenApi/INode.h	1328
include/SpinGenApi/INodeMap.h	1331
include/SpinGenApi/INodeMapDyn.h	1333
include/SpinGenApi/IntegerNode.h	1334
include/SpinGenApi/IntRegNode.h	1335
include/SpinGenApi/IPort.h	1336
include/SpinGenApi/IPortConstruct.h	1337
include/SpinGenApi/IPortRecorder.h	1338
include/SpinGenApi/IRegister.h	1339
include/SpinGenApi/ISelector.h	1340
include/SpinGenApi/ISelectorDigit.h	1341
include/SpinGenApi/IString.h	1342
include/SpinGenApi/IValue.h	1343
include/SpinGenApi/Node.h	1344
include/SpinGenApi/NodeCallback.h	1345
include/SpinGenApi/NodeCallbackImpl.h	1347
include/SpinGenApi/NodeMap.h	1347
include/SpinGenApi/NodeMapFactory.h	1348
include/SpinGenApi/NodeMapRef.h	1349
include/SpinGenApi/Persistence.h	1350
include/SpinGenApi/Pointer.h	1351
include/SpinGenApi/PortImpl.h	1353
include/SpinGenApi/PortNode.h	1354
include/SpinGenApi/PortRecorder.h	1355
include/SpinGenApi/PortReplay.h	1356
include/SpinGenApi/PortWriteList.h	1357
include/SpinGenApi/Reference.h	1358
include/SpinGenApi/RegisterNode.h	1359
include/SpinGenApi/RegisterPortImpl.h	1359
include/SpinGenApi/SelectorSet.h	1360
include/SpinGenApi/SpinnakerGenApi.h	1360
include/SpinGenApi/SpinTestCamera.h	1361
include/SpinGenApi/StringNode.h	1361
include/SpinGenApi/StringRegNode.h	1362
include/SpinGenApi/StructPort.h	1363
include/SpinGenApi/Synch.h	1363
include/SpinGenApi/Types.h	1364
include/SpinGenApi/ValueNode.h	1367
src/Acquisition/Acquisition.cpp	1387
src/Acquisition/resource.h	1388
src/Acquisition/stdafx.cpp	1390

src/Acquisition/stdafx.h	1399
src/Acquisition/targetver.h	1415
src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp	1429
src/AcquisitionMultipleCameraRecovery/resource.h	1388
src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp	1432
src/AcquisitionMultipleCamerasWriteToFile/resource.h	1388
src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp	1434
src/AcquisitionMultipleThread/resource.h	1388
src/ActionCommand/ActionCommand.cpp	1435
src/ActionCommand/resource.h	1388
src/ActionCommand/stdafx.cpp	1391
src/ActionCommand/stdafx.h	1400
src/ActionCommand/targetver.h	1416
src/BufferHandling/BufferHandling.cpp	1437
src/BufferHandling/resource.h	1388
src/BufferHandling/stdafx.cpp	1391
src/BufferHandling/stdafx.h	1401
src/BufferHandling/targetver.h	1417
src/ChunkData/ChunkData.cpp	1440
src/ChunkData/resource.h	1388
src/Compression/Compression.cpp	1442
src/Compression/resource.h	1389
src/CounterAndTimer/CounterAndTimer.cpp	1444
src/CounterAndTimer/resource.h	1389
src/CounterAndTimer/stdafx.cpp	1392
src/CounterAndTimer/stdafx.h	1402
src/CounterAndTimer/targetver.h	1418
src/DeviceEvents/DeviceEvents.cpp	1446
src/DeviceEvents/resource.h	1389
src/DeviceEvents/stdafx.cpp	1392
src/DeviceEvents/stdafx.h	1403
src/DeviceEvents/targetver.h	1419
src/Enumeration/Enumeration.cpp	1449
src/Enumeration/resource.h	1389
src/Enumeration/stdafx.cpp	1393
src/Enumeration/stdafx.h	1404
src/Enumeration/targetver.h	1420
src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp	1449
src/Enumeration_QuickSpin/resource.h	1389
src/Enumeration_QuickSpin/stdafx.cpp	1393
src/Enumeration_QuickSpin/stdafx.h	1405
src/Enumeration_QuickSpin/targetver.h	1421
src/EnumerationEvents/EnumerationEvents.cpp	1450
src/EnumerationEvents/resource.h	1389
src/ExceptionHandling/ExceptionHandling.cpp	1451
src/ExceptionHandling/resource.h	1389
src/ExceptionHandling/stdafx.cpp	1394
src/ExceptionHandling/stdafx.h	1406
src/ExceptionHandling/targetver.h	1422
src/Exposure/Exposure.cpp	1452
src/Exposure/resource.h	1389
src/Exposure/stdafx.cpp	1394
src/Exposure/stdafx.h	1407
src/Exposure/targetver.h	1423
src/Exposure_QuickSpin/Exposure_QuickSpin.cpp	1454
src/Exposure_QuickSpin/resource.h	1389
src/Exposure_QuickSpin/stdafx.cpp	1395
src/Exposure_QuickSpin/stdafx.h	1408

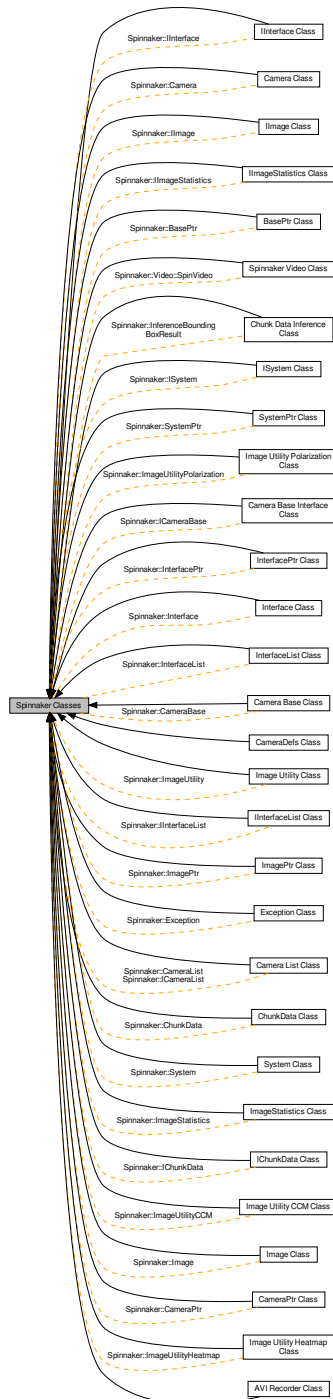
src/Exposure_QuickSpin/targetver.h	1424
src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp	1455
src/FileAccess_QuickSpin/resource.h	1389
src/FileAccess_QuickSpin/stdafx.cpp	1396
src/FileAccess_QuickSpin/stdafx.h	1409
src/FileAccess_QuickSpin/targetver.h	1425
src/GigEVisionPerformance/CpuUtil.cpp	1458
src/GigEVisionPerformance/CpuUtil.h	1459
src/GigEVisionPerformance/GigEVisionPerformance.cpp	1460
src/GigEVisionPerformance/GigEVisionPerformance.h	1465
src/GigEVisionPerformance/resource.h	1389
src/GigEVisionPerformance/stdafx.cpp	1397
src/GigEVisionPerformance/stdafx.h	1410
src/GigEVisionPerformance/targetver.h	1426
src/ImageEvents/ImageEvents.cpp	1466
src/ImageEvents/resource.h	1389
src/ImageFormatControl/ImageFormatControl.cpp	1468
src/ImageFormatControl/resource.h	1389
src/ImageFormatControl/stdafx.h	1411
src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp	1469
src/ImageFormatControl_QuickSpin/resource.h	1389
src/ImageFormatControl_QuickSpin/stdafx.h	1411
src/Inference/Inference.cpp	1470
src/Inference/resource.h	1389
src/Logging/Logging.cpp	1477
src/Logging/resource.h	1389
src/LogicBlock/LogicBlock.cpp	1478
src/LogicBlock/resource.h	1389
src/LookupTable/LookupTable.cpp	1480
src/LookupTable/resource.h	1390
src/NodeMapCallback/NodeMapCallback.cpp	1481
src/NodeMapCallback/resource.h	1390
src/NodeMapInfo/NodeMapInfo.cpp	1483
src/NodeMapInfo/resource.h	1390
src/NodeMapInfo/stdafx.cpp	1397
src/NodeMapInfo/stdafx.h	1412
src/NodeMapInfo/targetver.h	1427
src/Polarization/Polarization.cpp	1487
src/Polarization/resource.h	1390
src/Polarization/stdafx.h	1413
src/SaveToAvi/resource.h	1390
src/SaveToAvi/SaveToAvi.cpp	1490
src/Sequencer/resource.h	1390
src/Sequencer/Sequencer.cpp	1492
src/Sequencer/stdafx.cpp	1398
src/Sequencer/stdafx.h	1413
src/Sequencer/targetver.h	1428
src/SerialRxTx/resource.h	1390
src/SerialRxTx/SerialRxTx.cpp	1494
src/SerialRxTx/stdafx.cpp	1398
src/SerialRxTx/stdafx.h	1414
src/SerialRxTx/targetver.h	1429
src/Trigger/resource.h	1390
src/Trigger/Trigger.cpp	1497
src/Trigger_QuickSpin/resource.h	1390
src/Trigger_QuickSpin/Trigger_QuickSpin.cpp	1500

Chapter 13

Module Documentation

13.1 Spinnaker Classes

Collaboration diagram for Spinnaker Classes:



Modules

- [AVI Recorder Class](#)
- [BasePtr Class](#)

- [Camera Class](#)
- [Camera Base Class](#)
- [CameraDefs Class](#)
- [Camera List Class](#)
- [CameraPtr Class](#)
- [ChunkData Class](#)
- [Chunk Data Inference Class](#)
- [Exception Class](#)
- [Image Class](#)
- [ImagePtr Class](#)
- [ImageStatistics Class](#)
- [Image Utility Class](#)
- [Image Utility CCM Class](#)
- [Image Utility Heatmap Class](#)
- [Image Utility Polarization Class](#)
- [Interface Class](#)
- [InterfaceList Class](#)
- [InterfacePtr Class](#)
- [Spinnaker Video Class](#)
- [System Class](#)
- [SystemPtr Class](#)
- [Camera Base Interface Class](#)
- [IChunkData Class](#)
- [IImage Class](#)
- [IImageStatistics Class](#)
- [IInterface Class](#)
- [IInterfaceList Class](#)
- [ISystem Class](#)

Classes

- class [BasePtr< T, B >](#)
The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.
- class [Camera](#)
The camera object class.
- class [CameraBase](#)
The base class for the camera object.
- class [CameraList](#)
Used to hold a list of camera objects.
- class [CameraPtr](#)
A reference tracked pointer to a camera object.
- class [ChunkData](#)
The chunk data which contains additional information about an image.
- struct [InferenceBoxRect](#)
Inference Bounding Box Type Data Structures.
- struct [InferenceBoxCircle](#)
- struct [InferenceBoxRotatedRect](#)
- struct [InferenceBoundingBox](#)
Inference Bounding Boxes data structure.
- class [InferenceBoundingBoxResult](#)
An inference bounding boxes object which holds information about the detected bounding boxes.
- class [Exception](#)

- The [Exception](#) object represents an error that is returned from the library.
- class [Image](#)

The image object class.
 - class [ImagePtr](#)

A reference tracked pointer to an image object.
 - class [ImageStatistics](#)

Represents image statistics for an image.
 - class [ImageUtility](#)

Static helper functions for the image object class.
 - class [ImageUtilityCCM](#)

Static function to create color corrected images from an image object.
 - class [ImageUtilityHeatmap](#)

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.
 - class [ImageUtilityPolarization](#)

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.
 - class [Interface](#)

An interface object which holds a list of cameras.
 - class [InterfaceList](#)

A list of the available interfaces on the system.
 - class [InterfacePtr](#)

A reference tracked pointer to the interface object.
 - class [SpinVideo](#)

Provides the functionality for the user to record images to an AVI/MP4 file.
 - class [System](#)

The system object is used to retrieve the list of interfaces and cameras available.
 - class [SystemPtr](#)

A reference tracked pointer to a system object.
 - class [ICameraBase](#)

The interface file for base class for the camera object.
 - class [ICameraList](#)

Used to hold a list of camera objects.
 - class [IChunkData](#)

The [Interface](#) file for [ChunkData](#).
 - class [IImage](#)

The interface file for [Image](#).
 - class [IImageStatistics](#)

The interface file for image statistics.
 - class [IInterface](#)

The interface file for [Interface](#).
 - class [IInterfaceList](#)

The interface file for [InterfaceList](#) class.
 - class [ISystem](#)

The interface file for [System](#).

Enumerations

- enum [InferenceBoxType](#) {
 - [INFERENCE_BOX_TYPE_RECTANGLE](#) = 0,
 - [INFERENCE_BOX_TYPE_CIRCLE](#) = 1,
 - [INFERENCE_BOX_TYPE_ROTATED_RECTANGLE](#) = 2 }

Inference Bounding Box Type.

13.1.1 Detailed Description

13.1.2 Enumeration Type Documentation

13.1.2.1 InferenceBoxType

enum [InferenceBoxType](#)

Inference Bounding Box Type.

Enumerator

INFERENCE_BOX_TYPE_RECTANGLE	
INFERENCE_BOX_TYPE_CIRCLE	
INFERENCE_BOX_TYPE_ROTATED_RECTANGLE	

13.2 AVI Recorder Class

Collaboration diagram for AVI Recorder Class:



Functions

- class [DEPRECATED_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API A↔ VRecorder
Provides the functionality for the user to record images to an AVI file.

13.2.1 Detailed Description

13.2.2 Function Documentation

13.2.2.1 DEPRECATED_CLASS()

```

class Spinnaker::DEPRECATED_CLASS (
    "AVIRecorder is deprecated,
    use SpinVideo instead." )
  
```

Provides the functionality for the user to record images to an AVI file.

NOTE: This class is deprecated and replaced by SpinVideo. Refer to [SpinVideo.h](#) instead. Default constructor.

Default destructor.

Open an AVI file in preparation for writing Images to disk. The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the AVI file.
<i>pOption</i>	Options to apply to the AVI file.

See also

AVIClose()

Open an MJPEG AVI file in preparation for writing Images to disk. The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the AVI file.
<i>pOption</i>	MJPEG options to apply to the AVI file.

See also

AVIClose()
MJPGOption

Open an H264 MP4 file in preparation for writing Images to disk. The size of MP4 files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the MP4 file.
<i>pOption</i>	H264 options to apply to the MP4 file.

See also

AVIClose()
H264Option

Append an image to the AVI/MP4 file.

Parameters

<i>pImage</i>	The image to append.

Close the AVI/MP4 file.

See also

AVIOpen()

Set the maximum file size (in megabytes) of a AVI/MP4 file. A new AVI/MP4 file is created automatically when file size limit is reached. Setting a maximum size of 0 indicates no limit on file size.

Parameters

<i>size</i>	The maximum AVI file size in MB.

See also

`AVIAppend( ImagePtr pImage)`

13.3 BasePtr Class

Collaboration diagram for BasePtr Class:



Classes

- class `BasePtr< T, B >`

The base class of the `SystemPtr`, `CameraPtr`, `InterfacePtr`, `ImagePtr` and `LoggingEventDataPtr` objects.

Functions

- `template<class T, class B >`
`bool operator== (const std::nullptr_t, const BasePtr< T, B > &rhs)`
Pointer equal.

13.3.1 Detailed Description

13.3.2 Function Documentation

13.3.2.1 operator==()

```

bool Spinnaker::operator== (
    const std::nullptr_t ,
    const BasePtr< T, B > & rhs ) [inline]
  
```

Pointer equal.

13.4 Camera Class

Collaboration diagram for Camera Class:



Classes

- class [Camera](#)
The camera object class.

13.4.1 Detailed Description

13.5 Camera Base Class

Collaboration diagram for Camera Base Class:



Classes

- class [CameraBase](#)
The base class for the camera object.

13.5.1 Detailed Description

13.6 CameraDefs Class

Collaboration diagram for CameraDefs Class:



Enumerations

- enum `LUTSelectorEnums` {
`LUTSelector_LUT1`,
`NUM_LUTSELECTOR` }

The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.

- enum `ExposureModeEnums` {
`ExposureMode_Timed`,
`ExposureMode_TriggerWidth`,
`NUM_EXPOSUREMODE` }
- enum `AcquisitionModeEnums` {
`AcquisitionMode_Continuous`,
`AcquisitionMode_SingleFrame`,
`AcquisitionMode_MultiFrame`,
`NUM_ACQUISITIONMODE` }
- enum `TriggerSourceEnums` {
`TriggerSource_Software`,
`TriggerSource_Line0`,
`TriggerSource_Line1`,
`TriggerSource_Line2`,
`TriggerSource_Line3`,
`TriggerSource_UserOutput0`,
`TriggerSource_UserOutput1`,
`TriggerSource_UserOutput2`,
`TriggerSource_UserOutput3`,
`TriggerSource_Counter0Start`,
`TriggerSource_Counter1Start`,
`TriggerSource_Counter0End`,
`TriggerSource_Counter1End`,
`TriggerSource_LogicBlock0`,
`TriggerSource_LogicBlock1`,
`TriggerSource_Action0`,
`NUM_TRIGGERSOURCE` }
- enum `TriggerActivationEnums` {
`TriggerActivation_LevelLow`,
`TriggerActivation_LevelHigh`,
`TriggerActivation_FallingEdge`,
`TriggerActivation_RisingEdge`,
`TriggerActivation_AnyEdge`,
`NUM_TRIGGERACTIVATION` }

- enum SensorShutterModeEnums {
SensorShutterMode_Global,
SensorShutterMode_Rolling,
SensorShutterMode_GlobalReset,
NUM_SENSORSHUTTERMODE }
- enum TriggerModeEnums {
TriggerMode_Off,
TriggerMode_On,
NUM_TRIGGERMODE }
- enum TriggerOverlapEnums {
TriggerOverlap_Off,
TriggerOverlap_ReadOut,
TriggerOverlap_PreviousFrame,
NUM_TRIGGEROVERLAP }
- enum TriggerSelectorEnums {
TriggerSelector_AcquisitionStart,
TriggerSelector_FrameStart,
TriggerSelector_FrameBurstStart,
NUM_TRIGGERSELECTOR }
- enum ExposureAutoEnums {
ExposureAuto_Off,
ExposureAuto_Once,
ExposureAuto_Continuous,
NUM_EXPOSUREAUTO }
- enum EventSelectorEnums {
EventSelector_Error,
EventSelector_ExposureEnd,
EventSelector_SerialPortReceive,
NUM_EVENTSELECTOR }
- enum EventNotificationEnums {
EventNotification_On,
EventNotification_Off,
NUM_EVENTNOTIFICATION }
- enum LogicBlockSelectorEnums {
LogicBlockSelector_LogicBlock0,
LogicBlockSelector_LogicBlock1,
NUM_LOGICBLOCKSELECTOR }
- enum LogicBlockLUTInputActivationEnums {
LogicBlockLUTInputActivation_LevelLow,
LogicBlockLUTInputActivation_LevelHigh,
LogicBlockLUTInputActivation_FallingEdge,
LogicBlockLUTInputActivation_RisingEdge,
LogicBlockLUTInputActivation_AnyEdge,
NUM_LOGICBLOCKLUTINPUTACTIVATION }
- enum LogicBlockLUTInputSelectorEnums {
LogicBlockLUTInputSelector_Input0,
LogicBlockLUTInputSelector_Input1,
LogicBlockLUTInputSelector_Input2,
LogicBlockLUTInputSelector_Input3,
NUM_LOGICBLOCKLUTINPUTSELECTOR }
- enum LogicBlockLUTInputSourceEnums {
LogicBlockLUTInputSource_Zero,
LogicBlockLUTInputSource_Line0,
LogicBlockLUTInputSource_Line1,
LogicBlockLUTInputSource_Line2,
LogicBlockLUTInputSource_Line3,
LogicBlockLUTInputSource_UserOutput0,
LogicBlockLUTInputSource_UserOutput1,

```

LogicBlockLUTInputSource_UserOutput2,
LogicBlockLUTInputSource_UserOutput3,
LogicBlockLUTInputSource_Counter0Start,
LogicBlockLUTInputSource_Counter1Start,
LogicBlockLUTInputSource_Counter0End,
LogicBlockLUTInputSource_Counter1End,
LogicBlockLUTInputSource_LogicBlock0,
LogicBlockLUTInputSource_LogicBlock1,
LogicBlockLUTInputSource_ExposureStart,
LogicBlockLUTInputSource_ExposureEnd,
LogicBlockLUTInputSource_FrameTriggerWait,
LogicBlockLUTInputSource_AcquisitionActive,
NUM_LOGICBLOCKLUTINPUTSOURCE }

• enum LogicBlockLUTSelectorEnums {
    LogicBlockLUTSelector_Value,
    LogicBlockLUTSelector_Enable,
    NUM_LOGICBLOCKLUTSELECTOR }

• enum ColorTransformationSelectorEnums {
    ColorTransformationSelector_RGBtoRGB,
    ColorTransformationSelector_RGBtoYUV,
    NUM_COLORTRANSFORMATIONSELECTOR }

• enum RgbTransformLightSourceEnums {
    RgbTransformLightSource_General,
    RgbTransformLightSource_Tungsten2800K,
    RgbTransformLightSource_WarmFluorescent3000K,
    RgbTransformLightSource_CoolFluorescent4000K,
    RgbTransformLightSource_Daylight5000K,
    RgbTransformLightSource_Cloudy6500K,
    RgbTransformLightSource_Shade8000K,
    RgbTransformLightSource_Custom,
    NUM_RGBTRANSFORMLIGHTSOURCE }

• enum ColorTransformationValueSelectorEnums {
    ColorTransformationValueSelector_Gain00,
    ColorTransformationValueSelector_Gain01,
    ColorTransformationValueSelector_Gain02,
    ColorTransformationValueSelector_Gain10,
    ColorTransformationValueSelector_Gain11,
    ColorTransformationValueSelector_Gain12,
    ColorTransformationValueSelector_Gain20,
    ColorTransformationValueSelector_Gain21,
    ColorTransformationValueSelector_Gain22,
    ColorTransformationValueSelector_Offset0,
    ColorTransformationValueSelector_Offset1,
    ColorTransformationValueSelector_Offset2,
    NUM_COLORTRANSFORMATIONVALUESELECTOR }

• enum DeviceRegistersEndiannessEnums {
    DeviceRegistersEndianness_Little,
    DeviceRegistersEndianness_Big,
    NUM_DEVICEREGISTERSENDIANNES }

• enum DeviceScanTypeEnums {
    DeviceScanType_Areascan,
    NUM_DEVICESCANTYPE }

• enum DeviceCharacterSetEnums {
    DeviceCharacterSet_UTF8,
    DeviceCharacterSet_ASCII,
    NUM_DEVICECHARACTERSET }

• enum DeviceTLTypeEnums {
    DeviceTLType_GigEVision,

```

```

DeviceTLType_CameraLink,
DeviceTLType_CameraLinkHS,
DeviceTLType_CoaXPress,
DeviceTLType_USB3Vision,
DeviceTLType_Custom,
NUM_DEVICE_TLTYPE }

• enum DevicePowerSupplySelectorEnums {
DevicePowerSupplySelector_External,
NUM_DEVICE_POWER_SUPPLY_SELECTOR }

• enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICE_TEMPERATURE_SELECTOR }

• enum DeviceIndicatorModeEnums {
DeviceIndicatorMode_Inactive,
DeviceIndicatorMode_Active,
DeviceIndicatorMode_ErrorStatus,
NUM_DEVICE_INDICATOR_MODE }

• enum AutoExposureControlPriorityEnums {
AutoExposureControlPriority_Gain,
AutoExposureControlPriority_ExposureTime,
NUM_AUTO_EXPOSURE_CONTROL_PRIORITY }

• enum AutoExposureMeteringModeEnums {
AutoExposureMeteringMode_Average,
AutoExposureMeteringMode_Spot,
AutoExposureMeteringMode_Partial,
AutoExposureMeteringMode_CenterWeighted,
AutoExposureMeteringMode_HistogramPeak,
NUM_AUTO_EXPOSURE_METERING_MODE }

• enum BalanceWhiteAutoProfileEnums {
BalanceWhiteAutoProfile_Indoor,
BalanceWhiteAutoProfile_Outdoor,
NUM_BALANCE_WHITE_AUTO_PROFILE }

• enum AutoAlgorithmSelectorEnums {
AutoAlgorithmSelector_Awb,
AutoAlgorithmSelector_Ae,
NUM_AUTO_ALGORITHM_SELECTOR }

• enum AutoExposureTargetGreyValueAutoEnums {
AutoExposureTargetGreyValueAuto_Off,
AutoExposureTargetGreyValueAuto_Continuous,
NUM_AUTO_EXPOSURE_TARGET_GREY_VALUE_AUTO }

• enum AutoExposureLightingModeEnums {
AutoExposureLightingMode_AutoDetect,
AutoExposureLightingMode_Backlight,
AutoExposureLightingMode_Frontlight,
AutoExposureLightingMode_Normal,
NUM_AUTO_EXPOSURE_LIGHTING_MODE }

• enum GevIEEE1588StatusEnums {
GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,
GevIEEE1588Status_Listening,
GevIEEE1588Status_PreMaster,
GevIEEE1588Status_Master,
GevIEEE1588Status_Passive,
GevIEEE1588Status_Uncalibrated,
GevIEEE1588Status_Slave,
NUM_GEV_IEEE1588_STATUS }

```

- enum `GevIEEE1588ModeEnums` {
`GevIEEE1588Mode_Auto`,
`GevIEEE1588Mode_SlaveOnly`,
`NUM_GEVIEEE1588MODE` }
- enum `GevIEEE1588ClockAccuracyEnums` {
`GevIEEE1588ClockAccuracy_Unknown`,
`NUM_GEVIEEE1588CLOCKACCURACY` }
- enum `GevCCPEnums` {
`GevCCP_OpenAccess`,
`GevCCP_ExclusiveAccess`,
`GevCCP_ControlAccess`,
`NUM_GEVCCP` }
- enum `GevSupportedOptionSelectorEnums` {
`GevSupportedOptionSelector_UserDefinedName`,
`GevSupportedOptionSelector_SerialNumber`,
`GevSupportedOptionSelector_HeartbeatDisable`,
`GevSupportedOptionSelector_LinkSpeed`,
`GevSupportedOptionSelector_CCPApplicationSocket`,
`GevSupportedOptionSelector_ManifestTable`,
`GevSupportedOptionSelector_TestData`,
`GevSupportedOptionSelector_DiscoveryAckDelay`,
`GevSupportedOptionSelector_DiscoveryAckDelayWritable`,
`GevSupportedOptionSelector_ExtendedStatusCodes`,
`GevSupportedOptionSelector_Action`,
`GevSupportedOptionSelector_PendingAck`,
`GevSupportedOptionSelector_EventData`,
`GevSupportedOptionSelector_Event`,
`GevSupportedOptionSelector_PacketResend`,
`GevSupportedOptionSelector_WriteMem`,
`GevSupportedOptionSelector_CommandsConcatenation`,
`GevSupportedOptionSelector_IPConfigurationLLA`,
`GevSupportedOptionSelector_IPConfigurationDHCP`,
`GevSupportedOptionSelector_IPConfigurationPersistentIP`,
`GevSupportedOptionSelector_StreamChannelSourceSocket`,
`GevSupportedOptionSelector_MessageChannelSourceSocket`,
`NUM_GEVSUPPORTEDOPTIONSELECTOR` }
- enum `BlackLevelSelectorEnums` {
`BlackLevelSelector_All`,
`BlackLevelSelector_Analog`,
`BlackLevelSelector_Digital`,
`NUM_BLACKLEVELSELECTOR` }
- enum `BalanceWhiteAutoEnums` {
`BalanceWhiteAuto_Off`,
`BalanceWhiteAuto_Once`,
`BalanceWhiteAuto_Continuous`,
`NUM_BALANCEWHITEAUTO` }
- enum `GainAutoEnums` {
`GainAuto_Off`,
`GainAuto_Once`,
`GainAuto_Continuous`,
`NUM_GAINAUTO` }
- enum `BalanceRatioSelectorEnums` {
`BalanceRatioSelector_Red`,
`BalanceRatioSelector_Blue`,
`NUM_BALANCERATIOSELECTOR` }
- enum `GainSelectorEnums` {
`GainSelector_All`,
`NUM_GAINSELECTOR` }

- enum DefectCorrectionModeEnums {
DefectCorrectionMode_Average,
DefectCorrectionMode_Highlight,
DefectCorrectionMode_Zero,
NUM_DEFECTCORRECTIONMODE }
- enum UserSetSelectorEnums {
UserSetSelector_Default,
UserSetSelector_UserSet0,
UserSetSelector_UserSet1,
NUM_USERSETSELECTOR }
- enum UserSetDefaultEnums {
UserSetDefault_Default,
UserSetDefault_UserSet0,
UserSetDefault_UserSet1,
NUM_USERSETDEFAULT }
- enum SerialPortBaudRateEnums {
SerialPortBaudRate_Baud300,
SerialPortBaudRate_Baud600,
SerialPortBaudRate_Baud1200,
SerialPortBaudRate_Baud2400,
SerialPortBaudRate_Baud4800,
SerialPortBaudRate_Baud9600,
SerialPortBaudRate_Baud14400,
SerialPortBaudRate_Baud19200,
SerialPortBaudRate_Baud38400,
SerialPortBaudRate_Baud57600,
SerialPortBaudRate_Baud115200,
SerialPortBaudRate_Baud230400,
SerialPortBaudRate_Baud460800,
SerialPortBaudRate_Baud921600,
NUM_SERIALPORTBAUDRATE }
- enum SerialPortParityEnums {
SerialPortParity_None,
SerialPortParity_Odd,
SerialPortParity_Even,
SerialPortParity_Mark,
SerialPortParity_Space,
NUM_SERIALPORTPARITY }
- enum SerialPortSelectorEnums {
SerialPortSelector_SerialPort0,
NUM_SERIALPORTSELECTOR }
- enum SerialPortStopBitsEnums {
SerialPortStopBits_Bits1,
SerialPortStopBits_Bits1AndAHalf,
SerialPortStopBits_Bits2,
NUM_SERIALPORTSTOPBITS }
- enum SerialPortSourceEnums {
SerialPortSource_Line0,
SerialPortSource_Line1,
SerialPortSource_Line2,
SerialPortSource_Line3,
SerialPortSource_Off,
NUM_SERIALPORTSOURCE }
- enum SequencerModeEnums {
SequencerMode_Off,
SequencerMode_On,
NUM_SEQUENCERMODE }

- enum SequencerConfigurationValidEnums {
SequencerConfigurationValid_No,
SequencerConfigurationValid_Yes,
NUM_SEQUENCERCONFIGURATIONVALID }
- enum SequencerSetValidEnums {
SequencerSetValid_No,
SequencerSetValid_Yes,
NUM_SEQUENCERSETVALID }
- enum SequencerTriggerActivationEnums {
SequencerTriggerActivation_RisingEdge,
SequencerTriggerActivation_FallingEdge,
SequencerTriggerActivation_AnyEdge,
SequencerTriggerActivation_LevelHigh,
SequencerTriggerActivation_LevelLow,
NUM_SEQUENCERTRIGGERACTIVATION }
- enum SequencerConfigurationModeEnums {
SequencerConfigurationMode_Off,
SequencerConfigurationMode_On,
NUM_SEQUENCERCONFIGURATIONMODE }
- enum SequencerTriggerSourceEnums {
SequencerTriggerSource_Off,
SequencerTriggerSource_FrameStart,
NUM_SEQUENCERTRIGGERSOURCE }
- enum TransferQueueModeEnums {
TransferQueueMode_FirstInFirstOut,
NUM_TRANSFERQUEUEMODE }
- enum TransferOperationModeEnums {
TransferOperationMode_Continuous,
TransferOperationMode_MultiBlock,
NUM_TRANSFEROPERATIONMODE }
- enum TransferControlModeEnums {
TransferControlMode_Basic,
TransferControlMode_Automatic,
TransferControlMode_UserControlled,
NUM_TRANSFERCONTROLMODE }
- enum ChunkGainSelectorEnums {
ChunkGainSelector_All,
ChunkGainSelector_Red,
ChunkGainSelector_Green,
ChunkGainSelector_Blue,
NUM_CHUNKGAINSELECTOR }
- enum ChunkSelectorEnums {
ChunkSelector_Image,
ChunkSelector_CRC,
ChunkSelector_FrameID,
ChunkSelector_OffsetX,
ChunkSelector_OffsetY,
ChunkSelector_Width,
ChunkSelector_Height,
ChunkSelector_ExposureTime,
ChunkSelector_Gain,
ChunkSelector_BlackLevel,
ChunkSelector_PixelFormat,
ChunkSelector_Timestamp,
ChunkSelector_SequencerSetActive,
ChunkSelector_SerialData,
ChunkSelector_ExposureEndLineStatusAll,
NUM_CHUNKSELECTOR }

- enum ChunkBlackLevelSelectorEnums {
 ChunkBlackLevelSelector_All,
 NUM_CHUNKBLACKLEVELSELECTOR }
- enum ChunkPixelFormatEnums {
 ChunkPixelFormat_Mono8,
 ChunkPixelFormat_Mono12Packed,
 ChunkPixelFormat_Mono16,
 ChunkPixelFormat_RGB8Packed,
 ChunkPixelFormat_YUV422Packed,
 ChunkPixelFormat_BayerGR8,
 ChunkPixelFormat_BayerRG8,
 ChunkPixelFormat_BayerGB8,
 ChunkPixelFormat_BayerBG8,
 ChunkPixelFormat_YCbCr601_422_8_CbYCrY,
 NUM_CHUNKPIXELFORMAT }
- enum FileOperationStatusEnums {
 FileOperationStatus_Success,
 FileOperationStatus_Failure,
 FileOperationStatus_Overflow,
 NUM_FILEOPERATIONSTATUS }
- enum FileOpenModeEnums {
 FileOpenMode_Read,
 FileOpenMode_Write,
 FileOpenMode_ReadWrite,
 NUM_FILEOPENMODE }
- enum FileOperationSelectorEnums {
 FileOperationSelector_Open,
 FileOperationSelector_Close,
 FileOperationSelector_Read,
 FileOperationSelector_Write,
 FileOperationSelector_Delete,
 NUM_FILEOPERATIONSELECTOR }
- enum FileSelectorEnums {
 FileSelector_UserSetDefault,
 FileSelector_UserSet0,
 FileSelector_UserSet1,
 FileSelector_UserFile1,
 FileSelector_SerialPort0,
 NUM_FILESELECTOR }
- enum BinningSelectorEnums {
 BinningSelector_All,
 BinningSelector_Sensor,
 BinningSelector_ISP,
 NUM_BINNINGSELECTOR }
- enum TestPatternGeneratorSelectorEnums {
 TestPatternGeneratorSelector_Sensor,
 TestPatternGeneratorSelector_PipelineStart,
 NUM_TESTPATTERNGENERATORSELECTOR }
- enum CompressionSaturationPriorityEnums {
 CompressionSaturationPriority_DropFrame,
 CompressionSaturationPriority_ReduceFrameRate,
 NUM_COMPRESSIONSATURATIONPRIORITY }
- enum TestPatternEnums {
 TestPattern_Off,
 TestPattern_Increment,
 TestPattern_SensorTestPattern,
 NUM_TESTPATTERN }

- enum PixelColorFilterEnums {
PixelColorFilter_None,
PixelColorFilter_BayerRG,
PixelColorFilter_BayerGB,
PixelColorFilter_BayerGR,
PixelColorFilter_BayerBG,
NUM_PIXELCOLORFILTER }
- enum AdcBitDepthEnums {
AdcBitDepth_Bit8,
AdcBitDepth_Bit10,
AdcBitDepth_Bit12,
AdcBitDepth_Bit14,
NUM_ADCBITDEPTH }
- enum DecimationHorizontalModeEnums {
DecimationHorizontalMode_Discard,
NUM_DECIMATIONHORIZONTALMODE }
- enum BinningVerticalModeEnums {
BinningVerticalMode_Sum,
BinningVerticalMode_Average,
NUM_BINNINGVERTICALMODE }
- enum PixelSizeEnums {
PixelSize_Bpp1,
PixelSize_Bpp2,
PixelSize_Bpp4,
PixelSize_Bpp8,
PixelSize_Bpp10,
PixelSize_Bpp12,
PixelSize_Bpp14,
PixelSize_Bpp16,
PixelSize_Bpp20,
PixelSize_Bpp24,
PixelSize_Bpp30,
PixelSize_Bpp32,
PixelSize_Bpp36,
PixelSize_Bpp48,
PixelSize_Bpp64,
PixelSize_Bpp96,
NUM_PIXELSIZE }
- enum DecimationSelectorEnums {
DecimationSelector_All,
DecimationSelector_Sensor,
NUM_DECIMATIONSELECTOR }
- enum ImageCompressionModeEnums {
ImageCompressionMode_Off,
ImageCompressionMode_Lossless,
NUM_IMAGECOMPRESSIONMODE }
- enum BinningHorizontalModeEnums {
BinningHorizontalMode_Sum,
BinningHorizontalMode_Average,
NUM_BINNINGHORIZONTALMODE }
- enum PixelFormatEnums {
PixelFormat_Mono8,
PixelFormat_Mono16,
PixelFormat_RGB8Packed,
PixelFormat_BayerGR8,
PixelFormat_BayerRG8,
PixelFormat_BayerGB8,
PixelFormat_BayerBG8,

PixelFormat_BayerGR16,
PixelFormat_BayerRG16,
PixelFormat_BayerGB16,
PixelFormat_BayerBG16,
PixelFormat_Mono12Packed,
PixelFormat_BayerGR12Packed,
PixelFormat_BayerRG12Packed,
PixelFormat_BayerGB12Packed,
PixelFormat_BayerBG12Packed,
PixelFormat_YUV411Packed,
PixelFormat_YUV422Packed,
PixelFormat_YUV444Packed,
PixelFormat_Mono12p,
PixelFormat_BayerGR12p,
PixelFormat_BayerRG12p,
PixelFormat_BayerGB12p,
PixelFormat_BayerBG12p,
PixelFormat_YCbCr8,
PixelFormat_YCbCr422_8,
PixelFormat_YCbCr411_8,
PixelFormat_BGR8,
PixelFormat_BGRa8,
PixelFormat_Mono10Packed,
PixelFormat_BayerGR10Packed,
PixelFormat_BayerRG10Packed,
PixelFormat_BayerGB10Packed,
PixelFormat_BayerBG10Packed,
PixelFormat_Mono10p,
PixelFormat_BayerGR10p,
PixelFormat_BayerRG10p,
PixelFormat_BayerGB10p,
PixelFormat_BayerBG10p,
PixelFormat_Mono1p,
PixelFormat_Mono2p,
PixelFormat_Mono4p,
PixelFormat_Mono8s,
PixelFormat_Mono10,
PixelFormat_Mono12,
PixelFormat_Mono14,
PixelFormat_Mono16s,
PixelFormat_Mono32f,
PixelFormat_BayerBG10,
PixelFormat_BayerBG12,
PixelFormat_BayerGB10,
PixelFormat_BayerGB12,
PixelFormat_BayerGR10,
PixelFormat_BayerGR12,
PixelFormat_BayerRG10,
PixelFormat_BayerRG12,
PixelFormat_RGBa8,
PixelFormat_RGBa10,
PixelFormat_RGBa10p,
PixelFormat_RGBa12,
PixelFormat_RGBa12p,
PixelFormat_RGBa14,
PixelFormat_RGBa16,
PixelFormat_RGB8,
PixelFormat_RGB8_Planar,

PixelFormat_RGB10,
PixelFormat_RGB10_Planar,
PixelFormat_RGB10p,
PixelFormat_RGB10p32,
PixelFormat_RGB12,
PixelFormat_RGB12_Planar,
PixelFormat_RGB12p,
PixelFormat_RGB14,
PixelFormat_RGB16,
PixelFormat_RGB16s,
PixelFormat_RGB32f,
PixelFormat_RGB16_Planar,
PixelFormat_RGB565p,
PixelFormat_BGRa10,
PixelFormat_BGRa10p,
PixelFormat_BGRa12,
PixelFormat_BGRa12p,
PixelFormat_BGRa14,
PixelFormat_BGRa16,
PixelFormat_RGBa32f,
PixelFormat_BGR10,
PixelFormat_BGR10p,
PixelFormat_BGR12,
PixelFormat_BGR12p,
PixelFormat_BGR14,
PixelFormat_BGR16,
PixelFormat_BGR565p,
PixelFormat_R8,
PixelFormat_R10,
PixelFormat_R12,
PixelFormat_R16,
PixelFormat_G8,
PixelFormat_G10,
PixelFormat_G12,
PixelFormat_G16,
PixelFormat_B8,
PixelFormat_B10,
PixelFormat_B12,
PixelFormat_B16,
PixelFormat_Coord3D_ABC8,
PixelFormat_Coord3D_ABC8_Planar,
PixelFormat_Coord3D_ABC10p,
PixelFormat_Coord3D_ABC10p_Planar,
PixelFormat_Coord3D_ABC12p,
PixelFormat_Coord3D_ABC12p_Planar,
PixelFormat_Coord3D_ABC16,
PixelFormat_Coord3D_ABC16_Planar,
PixelFormat_Coord3D_ABC32f,
PixelFormat_Coord3D_ABC32f_Planar,
PixelFormat_Coord3D_AC8,
PixelFormat_Coord3D_AC8_Planar,
PixelFormat_Coord3D_AC10p,
PixelFormat_Coord3D_AC10p_Planar,
PixelFormat_Coord3D_AC12p,
PixelFormat_Coord3D_AC12p_Planar,
PixelFormat_Coord3D_AC16,
PixelFormat_Coord3D_AC16_Planar,
PixelFormat_Coord3D_AC32f,

PixelFormat_Coord3D_AC32f_Planar,
PixelFormat_Coord3D_A8,
PixelFormat_Coord3D_A10p,
PixelFormat_Coord3D_A12p,
PixelFormat_Coord3D_A16,
PixelFormat_Coord3D_A32f,
PixelFormat_Coord3D_B8,
PixelFormat_Coord3D_B10p,
PixelFormat_Coord3D_B12p,
PixelFormat_Coord3D_B16,
PixelFormat_Coord3D_B32f,
PixelFormat_Coord3D_C8,
PixelFormat_Coord3D_C10p,
PixelFormat_Coord3D_C12p,
PixelFormat_Coord3D_C16,
PixelFormat_Coord3D_C32f,
PixelFormat_Confidence1,
PixelFormat_Confidence1p,
PixelFormat_Confidence8,
PixelFormat_Confidence16,
PixelFormat_Confidence32f,
PixelFormat_BiColorBGRG8,
PixelFormat_BiColorBGRG10,
PixelFormat_BiColorBGRG10p,
PixelFormat_BiColorBGRG12,
PixelFormat_BiColorBGRG12p,
PixelFormat_BiColorRGBG8,
PixelFormat_BiColorRGBG10,
PixelFormat_BiColorRGBG10p,
PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,
PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,
PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,
PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,

PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,
PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,
PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,
PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,


```

PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

• enum DecimationVerticalModeEnums {
    DecimationVerticalMode_Discard,
    NUM_DECIMATIONVERTICALMODE }

• enum LineModeEnums {
    LineMode_Input,
    LineMode_Output,
    NUM_LINEMODE }

• enum LineSourceEnums {
    LineSource_Off,
    LineSource_Line0,
    LineSource_Line1,
    LineSource_Line2,
    LineSource_Line3,
    LineSource_UserOutput0,
    LineSource_UserOutput1,
    LineSource_UserOutput2,
    LineSource_UserOutput3,
    LineSource_Counter0Active,
    LineSource_Counter1Active,
    LineSource_LogicBlock0,
    LineSource_LogicBlock1,
    LineSource_ExposureActive,
    LineSource_FrameTriggerWait,
    LineSource_SerialPort0,
    LineSource_PPSSignal,
    LineSource_AllPixel,
    LineSource_AnyPixel,
    NUM_LINESOURCE }

• enum LineInputFilterSelectorEnums {
    LineInputFilterSelector_Deglintch,
    LineInputFilterSelector_Debounce,
    NUM_LINEINPUTFILTERSELECTOR }

• enum UserOutputSelectorEnums {
    UserOutputSelector_UserOutput0,
    UserOutputSelector_UserOutput1,
    UserOutputSelector_UserOutput2,
    UserOutputSelector_UserOutput3,
    NUM_USEROUTPUTSELECTOR }

• enum LineFormatEnums {
    LineFormat_NoConnect,
    LineFormat_TriState,
    LineFormat_TTL,
    LineFormat_LVDS,
    LineFormat_RS422,

```

```

LineFormat_OptoCoupled,
LineFormat_OpenDrain,
NUM_LINEFORMAT }
• enum LineSelectorEnums {
LineSelector_Line0,
LineSelector_Line1,
LineSelector_Line2,
LineSelector_Line3,
NUM_LINESELECTOR }
• enum ExposureActiveModeEnums {
ExposureActiveMode_Line1,
ExposureActiveMode_AnyPixels,
ExposureActiveMode_AllPixels,
NUM_EXPOSUREACTIVEMODE }
• enum CounterTriggerActivationEnums {
CounterTriggerActivation_LevelLow,
CounterTriggerActivation_LevelHigh,
CounterTriggerActivation_FallingEdge,
CounterTriggerActivation_RisingEdge,
CounterTriggerActivation_AnyEdge,
NUM_COUNTERTRIGGERACTIVATION }
• enum CounterSelectorEnums {
CounterSelector_Counter0,
CounterSelector_Counter1,
NUM_COUNTERSELECTOR }
• enum CounterStatusEnums {
CounterStatus_CounterIdle,
CounterStatus_CounterTriggerWait,
CounterStatus_CounterActive,
CounterStatus_CounterCompleted,
CounterStatus_CounterOverflow,
NUM_COUNTERSTATUS }
• enum CounterTriggerSourceEnums {
CounterTriggerSource_Off,
CounterTriggerSource_Line0,
CounterTriggerSource_Line1,
CounterTriggerSource_Line2,
CounterTriggerSource_Line3,
CounterTriggerSource_UserOutput0,
CounterTriggerSource_UserOutput1,
CounterTriggerSource_UserOutput2,
CounterTriggerSource_UserOutput3,
CounterTriggerSource_Counter0Start,
CounterTriggerSource_Counter1Start,
CounterTriggerSource_Counter0End,
CounterTriggerSource_Counter1End,
CounterTriggerSource_LogicBlock0,
CounterTriggerSource_LogicBlock1,
CounterTriggerSource_ExposureStart,
CounterTriggerSource_ExposureEnd,
CounterTriggerSource_FrameTriggerWait,
NUM_COUNTERTRIGGERSOURCE }
• enum CounterResetSourceEnums {
CounterResetSource_Off,
CounterResetSource_Line0,
CounterResetSource_Line1,
CounterResetSource_Line2,
CounterResetSource_Line3,

```

```

CounterResetSource_UserOutput0,
CounterResetSource_UserOutput1,
CounterResetSource_UserOutput2,
CounterResetSource_UserOutput3,
CounterResetSource_Counter0Start,
CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }
• enum CounterEventSourceEnums {
CounterEventSource_Off,
CounterEventSource_MHzTick,
CounterEventSource_Line0,
CounterEventSource_Line1,
CounterEventSource_Line2,
CounterEventSource_Line3,
CounterEventSource_UserOutput0,
CounterEventSource_UserOutput1,
CounterEventSource_UserOutput2,
CounterEventSource_UserOutput3,
CounterEventSource_Counter0Start,
CounterEventSource_Counter1Start,
CounterEventSource_Counter0End,
CounterEventSource_Counter1End,
CounterEventSource_LogicBlock0,
CounterEventSource_LogicBlock1,
CounterEventSource_ExposureStart,
CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }
• enum CounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }
• enum CounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }
• enum DeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICETYPE }
• enum DeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,

```

```

NUM_DEVICECONNECTIONSTATUS }

• enum DeviceLinkThroughputLimitModeEnums {
    DeviceLinkThroughputLimitMode_On,
    DeviceLinkThroughputLimitMode_Off,
    NUM_DEVICELINKTHROUGHPUTLIMITMODE }

• enum DeviceLinkHeartbeatModeEnums {
    DeviceLinkHeartbeatMode_On,
    DeviceLinkHeartbeatMode_Off,
    NUM_DEVICELINKHEARTBEATMODE }

• enum DeviceStreamChannelTypeEnums {
    DeviceStreamChannelType_Transmitter,
    DeviceStreamChannelType_Receiver,
    NUM_DEVICESTREAMCHANNELTYPE }

• enum DeviceStreamChannelEndiannessEnums {
    DeviceStreamChannelEndianness_Big,
    DeviceStreamChannelEndianness_Little,
    NUM_DEVICESTREAMCHANNELENDIANNESS }

• enum DeviceClockSelectorEnums {
    DeviceClockSelector_Sensor,
    DeviceClockSelector_SensorDigitization,
    DeviceClockSelector_CameraLink,
    NUM_DEVICECLOCKSELECTOR }

• enum DeviceSerialPortSelectorEnums {
    DeviceSerialPortSelector_CameraLink,
    NUM_DEVICESERIALPORTSELECTOR }

• enum DeviceSerialPortBaudRateEnums {
    DeviceSerialPortBaudRate_Baud9600,
    DeviceSerialPortBaudRate_Baud19200,
    DeviceSerialPortBaudRate_Baud38400,
    DeviceSerialPortBaudRate_Baud57600,
    DeviceSerialPortBaudRate_Baud115200,
    DeviceSerialPortBaudRate_Baud230400,
    DeviceSerialPortBaudRate_Baud460800,
    DeviceSerialPortBaudRate_Baud921600,
    NUM_DEVICESERIALPORTBAUDRATE }

• enum SensorTapsEnums {
    SensorTaps_One,
    SensorTaps_Two,
    SensorTaps_Three,
    SensorTaps_Four,
    SensorTaps_Eight,
    SensorTaps_Ten,
    NUM_SENSORTAPS }

• enum SensorDigitizationTapsEnums {
    SensorDigitizationTaps_One,
    SensorDigitizationTaps_Two,
    SensorDigitizationTaps_Three,
    SensorDigitizationTaps_Four,
    SensorDigitizationTaps_Eight,
    SensorDigitizationTaps_Ten,
    NUM_SENSORDIGITIZATIONTAPS }

• enum RegionSelectorEnums {
    RegionSelector_Region0,
    RegionSelector_Region1,
    RegionSelector_Region2,
    RegionSelector_All,
    NUM_REGIONSELECTOR }

```

- enum RegionModeEnums {
RegionMode_Off,
RegionMode_On,
NUM_REGIONMODE }
- enum RegionDestinationEnums {
RegionDestination_Stream0,
RegionDestination_Stream1,
RegionDestination_Stream2,
NUM_REGIONDESTINATION }
- enum ImageComponentSelectorEnums {
ImageComponentSelector_Intensity,
ImageComponentSelector_Color,
ImageComponentSelector_Infrared,
ImageComponentSelector_Ultraviolet,
ImageComponentSelector_Range,
ImageComponentSelector_Disparity,
ImageComponentSelector_Confidence,
ImageComponentSelector_Scatter,
NUM_IMAGECOMPONENTSELECTOR }
- enum PixelFormatInfoSelectorEnums {
PixelFormatInfoSelector_Mono1p,
PixelFormatInfoSelector_Mono2p,
PixelFormatInfoSelector_Mono4p,
PixelFormatInfoSelector_Mono8,
PixelFormatInfoSelector_Mono8s,
PixelFormatInfoSelector_Mono10,
PixelFormatInfoSelector_Mono10p,
PixelFormatInfoSelector_Mono12,
PixelFormatInfoSelector_Mono12p,
PixelFormatInfoSelector_Mono14,
PixelFormatInfoSelector_Mono16,
PixelFormatInfoSelector_Mono16s,
PixelFormatInfoSelector_Mono32f,
PixelFormatInfoSelector_BayerBG8,
PixelFormatInfoSelector_BayerBG10,
PixelFormatInfoSelector_BayerBG10p,
PixelFormatInfoSelector_BayerBG12,
PixelFormatInfoSelector_BayerBG12p,
PixelFormatInfoSelector_BayerBG16,
PixelFormatInfoSelector_BayerGB8,
PixelFormatInfoSelector_BayerGB10,
PixelFormatInfoSelector_BayerGB10p,
PixelFormatInfoSelector_BayerGB12,
PixelFormatInfoSelector_BayerGB12p,
PixelFormatInfoSelector_BayerGB16,
PixelFormatInfoSelector_BayerGR8,
PixelFormatInfoSelector_BayerGR10,
PixelFormatInfoSelector_BayerGR10p,
PixelFormatInfoSelector_BayerGR12,
PixelFormatInfoSelector_BayerGR12p,
PixelFormatInfoSelector_BayerGR16,
PixelFormatInfoSelector_BayerRG8,
PixelFormatInfoSelector_BayerRG10,
PixelFormatInfoSelector_BayerRG10p,
PixelFormatInfoSelector_BayerRG12,
PixelFormatInfoSelector_BayerRG12p,
PixelFormatInfoSelector_BayerRG16,
PixelFormatInfoSelector_RGBa8,

[PixelFormatInfoSelector_RGBA10,](#)
[PixelFormatInfoSelector_RGBA10p,](#)
[PixelFormatInfoSelector_RGBA12,](#)
[PixelFormatInfoSelector_RGBA12p,](#)
[PixelFormatInfoSelector_RGBA14,](#)
[PixelFormatInfoSelector_RGBA16,](#)
[PixelFormatInfoSelector_RGB8,](#)
[PixelFormatInfoSelector_RGB8_Planar,](#)
[PixelFormatInfoSelector_RGB10,](#)
[PixelFormatInfoSelector_RGB10_Planar,](#)
[PixelFormatInfoSelector_RGB10p,](#)
[PixelFormatInfoSelector_RGB10p32,](#)
[PixelFormatInfoSelector_RGB12,](#)
[PixelFormatInfoSelector_RGB12_Planar,](#)
[PixelFormatInfoSelector_RGB12p,](#)
[PixelFormatInfoSelector_RGB14,](#)
[PixelFormatInfoSelector_RGB16,](#)
[PixelFormatInfoSelector_RGB16s,](#)
[PixelFormatInfoSelector_RGB32f,](#)
[PixelFormatInfoSelector_RGB16_Planar,](#)
[PixelFormatInfoSelector_RGB565p,](#)
[PixelFormatInfoSelector_BGRa8,](#)
[PixelFormatInfoSelector_BGRa10,](#)
[PixelFormatInfoSelector_BGRa10p,](#)
[PixelFormatInfoSelector_BGRa12,](#)
[PixelFormatInfoSelector_BGRa12p,](#)
[PixelFormatInfoSelector_BGRa14,](#)
[PixelFormatInfoSelector_BGRa16,](#)
[PixelFormatInfoSelector_RGBA32f,](#)
[PixelFormatInfoSelector_BGR8,](#)
[PixelFormatInfoSelector_BGR10,](#)
[PixelFormatInfoSelector_BGR10p,](#)
[PixelFormatInfoSelector_BGR12,](#)
[PixelFormatInfoSelector_BGR12p,](#)
[PixelFormatInfoSelector_BGR14,](#)
[PixelFormatInfoSelector_BGR16,](#)
[PixelFormatInfoSelector_BGR565p,](#)
[PixelFormatInfoSelector_R8,](#)
[PixelFormatInfoSelector_R10,](#)
[PixelFormatInfoSelector_R12,](#)
[PixelFormatInfoSelector_R16,](#)
[PixelFormatInfoSelector_G8,](#)
[PixelFormatInfoSelector_G10,](#)
[PixelFormatInfoSelector_G12,](#)
[PixelFormatInfoSelector_G16,](#)
[PixelFormatInfoSelector_B8,](#)
[PixelFormatInfoSelector_B10,](#)
[PixelFormatInfoSelector_B12,](#)
[PixelFormatInfoSelector_B16,](#)
[PixelFormatInfoSelector_Coord3D_ABC8,](#)
[PixelFormatInfoSelector_Coord3D_ABC8_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC10p,](#)
[PixelFormatInfoSelector_Coord3D_ABC10p_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC12p,](#)
[PixelFormatInfoSelector_Coord3D_ABC12p_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC16,](#)
[PixelFormatInfoSelector_Coord3D_ABC16_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC32f,](#)

PixelFormatInfoSelector_Coord3D_ABC32f_Planar,
PixelFormatInfoSelector_Coord3D_AC8,
PixelFormatInfoSelector_Coord3D_AC8_Planar,
PixelFormatInfoSelector_Coord3D_AC10p,
PixelFormatInfoSelector_Coord3D_AC10p_Planar,
PixelFormatInfoSelector_Coord3D_AC12p,
PixelFormatInfoSelector_Coord3D_AC12p_Planar,
PixelFormatInfoSelector_Coord3D_AC16,
PixelFormatInfoSelector_Coord3D_AC16_Planar,
PixelFormatInfoSelector_Coord3D_AC32f,
PixelFormatInfoSelector_Coord3D_AC32f_Planar,
PixelFormatInfoSelector_Coord3D_A8,
PixelFormatInfoSelector_Coord3D_A10p,
PixelFormatInfoSelector_Coord3D_A12p,
PixelFormatInfoSelector_Coord3D_A16,
PixelFormatInfoSelector_Coord3D_A32f,
PixelFormatInfoSelector_Coord3D_B8,
PixelFormatInfoSelector_Coord3D_B10p,
PixelFormatInfoSelector_Coord3D_B12p,
PixelFormatInfoSelector_Coord3D_B16,
PixelFormatInfoSelector_Coord3D_B32f,
PixelFormatInfoSelector_Coord3D_C8,
PixelFormatInfoSelector_Coord3D_C10p,
PixelFormatInfoSelector_Coord3D_C12p,
PixelFormatInfoSelector_Coord3D_C16,
PixelFormatInfoSelector_Coord3D_C32f,
PixelFormatInfoSelector_Confidence1,
PixelFormatInfoSelector_Confidence1p,
PixelFormatInfoSelector_Confidence8,
PixelFormatInfoSelector_Confidence16,
PixelFormatInfoSelector_Confidence32f,
PixelFormatInfoSelector_BiColorBGRG8,
PixelFormatInfoSelector_BiColorBGRG10,
PixelFormatInfoSelector_BiColorBGRG10p,
PixelFormatInfoSelector_BiColorBGRG12,
PixelFormatInfoSelector_BiColorBGRG12p,
PixelFormatInfoSelector_BiColorRGBG8,
PixelFormatInfoSelector_BiColorRGBG10,
PixelFormatInfoSelector_BiColorRGBG10p,
PixelFormatInfoSelector_BiColorRGBG12,
PixelFormatInfoSelector_BiColorRGBG12p,
PixelFormatInfoSelector_SCF1WBWG8,
PixelFormatInfoSelector_SCF1WBWG10,
PixelFormatInfoSelector_SCF1WBWG10p,
PixelFormatInfoSelector_SCF1WBWG12,
PixelFormatInfoSelector_SCF1WBWG12p,
PixelFormatInfoSelector_SCF1WBWG14,
PixelFormatInfoSelector_SCF1WBWG16,
PixelFormatInfoSelector_SCF1WGWB8,
PixelFormatInfoSelector_SCF1WGWB10,
PixelFormatInfoSelector_SCF1WGWB10p,
PixelFormatInfoSelector_SCF1WGWB12,
PixelFormatInfoSelector_SCF1WGWB12p,
PixelFormatInfoSelector_SCF1WGWB14,
PixelFormatInfoSelector_SCF1WGWB16,
PixelFormatInfoSelector_SCF1WGWR8,
PixelFormatInfoSelector_SCF1WGWR10,
PixelFormatInfoSelector_SCF1WGWR10p,

PixelFormatInfoSelector_SCF1WGWR12,
PixelFormatInfoSelector_SCF1WGWR12p,
PixelFormatInfoSelector_SCF1WGWR14,
PixelFormatInfoSelector_SCF1WGWR16,
PixelFormatInfoSelector_SCF1WRWG8,
PixelFormatInfoSelector_SCF1WRWG10,
PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,
PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,


```

PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

• enum DeinterlacingEnums {
    Deinterlacing_Off,
    Deinterlacing_LineDuplication,
    Deinterlacing_Weave,
    NUM_DEINTERLACING }

• enum ImageCompressionRateOptionEnums {
    ImageCompressionRateOption_FixBitrate,
    ImageCompressionRateOption_FixQuality,
    NUM_IMAGECOMPRESSIONRATEOPTION }

• enum ImageCompressionJPEGFormatOptionEnums {
    ImageCompressionJPEGFormatOption_Lossless,
    ImageCompressionJPEGFormatOption_BaselineStandard,
    ImageCompressionJPEGFormatOption_BaselineOptimized,
    ImageCompressionJPEGFormatOption_Progressive,
    NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }

• enum AcquisitionStatusSelectorEnums {
    AcquisitionStatusSelector_AcquisitionTriggerWait,
    AcquisitionStatusSelector_AcquisitionActive,
    AcquisitionStatusSelector_AcquisitionTransfer,
    AcquisitionStatusSelector_FrameTriggerWait,
    AcquisitionStatusSelector_FrameActive,
    AcquisitionStatusSelector_ExposureActive,
    NUM_ACQUISITIONSTATUSSELECTOR }

• enum ExposureTimeModeEnums {
    ExposureTimeMode_Common,
    ExposureTimeMode_Individual,
    NUM_EXPOSURETIMEMODE }

• enum ExposureTimeSelectorEnums {
    ExposureTimeSelector_Common,
    ExposureTimeSelector_Red,
    ExposureTimeSelector_Green,
    ExposureTimeSelector_Blue,
    ExposureTimeSelector_Cyan,
    ExposureTimeSelector_Magenta,
    ExposureTimeSelector_Yellow,
    ExposureTimeSelector_Infrared,
    ExposureTimeSelector_Ultraviolet,
    ExposureTimeSelector_Stage1,

```

```

    ExposureTimeSelector_Stage2,
    NUM_EXPOSURETIMESELECTOR }
• enum GainAutoBalanceEnums {
    GainAutoBalance_Off,
    GainAutoBalance_Once,
    GainAutoBalance_Continuous,
    NUM_GAINAUTOBALANCE }
• enum BlackLevelAutoEnums {
    BlackLevelAuto_Off,
    BlackLevelAuto_Once,
    BlackLevelAuto_Continuous,
    NUM_BLACKLEVELAUTO }
• enum BlackLevelAutoBalanceEnums {
    BlackLevelAutoBalance_Off,
    BlackLevelAutoBalance_Once,
    BlackLevelAutoBalance_Continuous,
    NUM_BLACKLEVELAUTOBALANCE }
• enum WhiteClipSelectorEnums {
    WhiteClipSelector_All,
    WhiteClipSelector_Red,
    WhiteClipSelector_Green,
    WhiteClipSelector_Blue,
    WhiteClipSelector_Y,
    WhiteClipSelector_U,
    WhiteClipSelector_V,
    WhiteClipSelector_Tap1,
    WhiteClipSelector_Tap2,
    NUM_WHITECLIPSELECTOR }
• enum TimerSelectorEnums {
    TimerSelector_Timer0,
    TimerSelector_Timer1,
    TimerSelector_Timer2,
    NUM_TIMERSELECTOR }
• enum TimerStatusEnums {
    TimerStatus_TimerIdle,
    TimerStatus_TimerTriggerWait,
    TimerStatus_TimerActive,
    TimerStatus_TimerCompleted,
    NUM_TIMERSTATUS }
• enum TimerTriggerSourceEnums {
    TimerTriggerSource_Off,
    TimerTriggerSource_AcquisitionTrigger,
    TimerTriggerSource_AcquisitionStart,
    TimerTriggerSource_AcquisitionEnd,
    TimerTriggerSource_FrameTrigger,
    TimerTriggerSource_FrameStart,
    TimerTriggerSource_FrameEnd,
    TimerTriggerSource_FrameBurstStart,
    TimerTriggerSource_FrameBurstEnd,
    TimerTriggerSource_LineTrigger,
    TimerTriggerSource_LineStart,
    TimerTriggerSource_LineEnd,
    TimerTriggerSource_ExposureStart,
    TimerTriggerSource_ExposureEnd,
    TimerTriggerSource_Line0,
    TimerTriggerSource_Line1,
    TimerTriggerSource_Line2,
    TimerTriggerSource_UserOutput0,

```

```

TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }
• enum TimerTriggerActivationEnums {
TimerTriggerActivation_RisingEdge,
TimerTriggerActivation_FallingEdge,
TimerTriggerActivation_AnyEdge,
TimerTriggerActivation_LevelHigh,
TimerTriggerActivation_LevelLow,
NUM_TIMERTRIGGERACTIVATION }
• enum EncoderSelectorEnums {
EncoderSelector_Encoder0,
EncoderSelector_Encoder1,
EncoderSelector_Encoder2,
NUM_ENCODERSELECTOR }
• enum EncoderSourceAEnums {
EncoderSourceA_Off,
EncoderSourceA_Line0,
EncoderSourceA_Line1,
EncoderSourceA_Line2,
NUM_ENCODERSOURCEA }
• enum EncoderSourceBEnums {
EncoderSourceB_Off,
EncoderSourceB_Line0,
EncoderSourceB_Line1,
EncoderSourceB_Line2,
NUM_ENCODERSOURCEB }
• enum EncoderModeEnums {
EncoderMode_FourPhase,
EncoderMode_HighResolution,
NUM_ENCODERMODE }
• enum EncoderOutputModeEnums {
EncoderOutputMode_Off,
EncoderOutputMode_PositionUp,

```

```

EncoderOutputMode_PositionDown,
EncoderOutputMode_DirectionUp,
EncoderOutputMode_DirectionDown,
EncoderOutputMode_Motion,
NUM_ENCODEROUTPUTMODE }

• enum EncoderStatusEnums {
EncoderStatus_EncoderUp,
EncoderStatus_EncoderDown,
EncoderStatus_EncoderIdle,
EncoderStatus_EncoderStatic,
NUM_ENCODERSTATUS }

• enum EncoderResetSourceEnums {
EncoderResetSource_Off,
EncoderResetSource_AcquisitionTrigger,
EncoderResetSource_AcquisitionStart,
EncoderResetSource_AcquisitionEnd,
EncoderResetSource_FrameTrigger,
EncoderResetSource_FrameStart,
EncoderResetSource_FrameEnd,
EncoderResetSource_ExposureStart,
EncoderResetSource_ExposureEnd,
EncoderResetSource_Line0,
EncoderResetSource_Line1,
EncoderResetSource_Line2,
EncoderResetSource_Counter0Start,
EncoderResetSource_Counter1Start,
EncoderResetSource_Counter2Start,
EncoderResetSource_Counter0End,
EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }

• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,

```

```

SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }

• enum ActionUnconditionalModeEnums {
  ActionUnconditionalMode_Off,
  ActionUnconditionalMode_On,
  NUM_ACTIONUNCONDITIONALMODE }

• enum SourceSelectorEnums {
  SourceSelector_Source0,
  SourceSelector_Source1,
  SourceSelector_Source2,
  SourceSelector_All,
  NUM_SOURCESELECTOR }

• enum TransferSelectorEnums {
  TransferSelector_Stream0,
  TransferSelector_Stream1,
  TransferSelector_Stream2,
  TransferSelector_All,
  NUM_TRANSFERSELECTOR }

• enum TransferTriggerSelectorEnums {
  TransferTriggerSelector_TransferStart,
  TransferTriggerSelector_TransferStop,
  TransferTriggerSelector_TransferAbort,
  TransferTriggerSelector_TransferPause,
  TransferTriggerSelector_TransferResume,
  TransferTriggerSelector_TransferActive,
  TransferTriggerSelector_TransferBurstStart,
  TransferTriggerSelector_TransferBurstStop,
  NUM_TRANSFERTRIGGERSELECTOR }

• enum TransferTriggerModeEnums {
  TransferTriggerMode_Off,
  TransferTriggerMode_On,
  NUM_TRANSFERTRIGGERMODE }

• enum TransferTriggerSourceEnums {
  TransferTriggerSource_Line0,
  TransferTriggerSource_Line1,
  TransferTriggerSource_Line2,
  TransferTriggerSource_Counter0Start,
  TransferTriggerSource_Counter1Start,
  TransferTriggerSource_Counter2Start,
  TransferTriggerSource_Counter0End,
  TransferTriggerSource_Counter1End,
  TransferTriggerSource_Counter2End,
  TransferTriggerSource_Timer0Start,
  TransferTriggerSource_Timer1Start,
  TransferTriggerSource_Timer2Start,
  TransferTriggerSource_Timer0End,
  TransferTriggerSource_Timer1End,
  TransferTriggerSource_Timer2End,
  TransferTriggerSource_SoftwareSignal0,
  TransferTriggerSource_SoftwareSignal1,
  TransferTriggerSource_SoftwareSignal2,
  TransferTriggerSource_Action0,
  TransferTriggerSource_Action1,
  TransferTriggerSource_Action2,
  NUM_TRANSFERTRIGGERSOURCE }

• enum TransferTriggerActivationEnums {
  TransferTriggerActivation_RisingEdge,

```

```

TransferTriggerActivation_FallingEdge,
TransferTriggerActivation_AnyEdge,
TransferTriggerActivation_LevelHigh,
TransferTriggerActivation_LevelLow,
NUM_TRANSFERTRIGGERACTIVATION }

• enum TransferStatusSelectorEnums {
TransferStatusSelector_Streaming,
TransferStatusSelector_Paused,
TransferStatusSelector_Stopping,
TransferStatusSelector_Stopped,
TransferStatusSelector_QueueOverflow,
NUM_TRANSFERSTATUSSELECTOR }

• enum TransferComponentSelectorEnums {
TransferComponentSelector_Red,
TransferComponentSelector_Green,
TransferComponentSelector_Blue,
TransferComponentSelector_All,
NUM_TRANSFERCOMPONENTSELECTOR }

• enum Scan3dDistanceUnitEnums {
Scan3dDistanceUnit_Millimeter,
Scan3dDistanceUnit_Inch,
NUM_SCAN3DDISTANCEUNIT }

• enum Scan3dCoordinateSystemEnums {
Scan3dCoordinateSystem_Cartesian,
Scan3dCoordinateSystem_Spherical,
Scan3dCoordinateSystem_Cylindrical,
NUM_SCAN3DCOORDINATESYSTEM }

• enum Scan3dOutputModeEnums {
Scan3dOutputMode_UncalibratedC,
Scan3dOutputMode_CalibratedABC_Grid,
Scan3dOutputMode_CalibratedABC_PointCloud,
Scan3dOutputMode_CalibratedAC,
Scan3dOutputMode_CalibratedAC_Linescan,
Scan3dOutputMode_CalibratedC,
Scan3dOutputMode_CalibratedC_Linescan,
Scan3dOutputMode_RectifiedC,
Scan3dOutputMode_RectifiedC_Linescan,
Scan3dOutputMode_DisparityC,
Scan3dOutputMode_DisparityC_Linescan,
NUM_SCAN3DOUTPUTMODE }

• enum Scan3dCoordinateSystemReferenceEnums {
Scan3dCoordinateSystemReference_Anchor,
Scan3dCoordinateSystemReference_Transformed,
NUM_SCAN3DCOORDINATESYSTEMREFERENCE }

• enum Scan3dCoordinateSelectorEnums {
Scan3dCoordinateSelector_CoordinateA,
Scan3dCoordinateSelector_CoordinateB,
Scan3dCoordinateSelector_CoordinateC,
NUM_SCAN3DCOORDINATESELECTOR }

• enum Scan3dCoordinateTransformSelectorEnums {
Scan3dCoordinateTransformSelector_RotationX,
Scan3dCoordinateTransformSelector_RotationY,
Scan3dCoordinateTransformSelector_RotationZ,
Scan3dCoordinateTransformSelector_TranslationX,
Scan3dCoordinateTransformSelector_TranslationY,
Scan3dCoordinateTransformSelector_TranslationZ,
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }

```

- enum Scan3dCoordinateReferenceSelectorEnums {
 Scan3dCoordinateReferenceSelector_RotationX,
 Scan3dCoordinateReferenceSelector_RotationY,
 Scan3dCoordinateReferenceSelector_RotationZ,
 Scan3dCoordinateReferenceSelector_TranslationX,
 Scan3dCoordinateReferenceSelector_TranslationY,
 Scan3dCoordinateReferenceSelector_TranslationZ,
 NUM_SCAN3DCOORDINATEREFERENCESELECTOR }
- enum ChunkImageComponentEnums {
 ChunkImageComponent_Intensity,
 ChunkImageComponent_Color,
 ChunkImageComponent_Infrared,
 ChunkImageComponent_Ultraviolet,
 ChunkImageComponent_Range,
 ChunkImageComponent_Disparity,
 ChunkImageComponent_Confidence,
 ChunkImageComponent_Scatter,
 NUM_CHUNKIMAGECOMPONENT }
- enum ChunkCounterSelectorEnums {
 ChunkCounterSelector_Counter0,
 ChunkCounterSelector_Counter1,
 ChunkCounterSelector_Counter2,
 NUM_CHUNKCOUNTERSELECTOR }
- enum ChunkTimerSelectorEnums {
 ChunkTimerSelector_Timer0,
 ChunkTimerSelector_Timer1,
 ChunkTimerSelector_Timer2,
 NUM_CHUNKTIMERSELECTOR }
- enum ChunkEncoderSelectorEnums {
 ChunkEncoderSelector_Encoder0,
 ChunkEncoderSelector_Encoder1,
 ChunkEncoderSelector_Encoder2,
 NUM_CHUNKENCODERSELECTOR }
- enum ChunkEncoderStatusEnums {
 ChunkEncoderStatus_EncoderUp,
 ChunkEncoderStatus_EncoderDown,
 ChunkEncoderStatus_EncoderIdle,
 ChunkEncoderStatus_EncoderStatic,
 NUM_CHUNKENCODERSTATUS }
- enum ChunkExposureTimeSelectorEnums {
 ChunkExposureTimeSelector_Common,
 ChunkExposureTimeSelector_Red,
 ChunkExposureTimeSelector_Green,
 ChunkExposureTimeSelector_Blue,
 ChunkExposureTimeSelector_Cyan,
 ChunkExposureTimeSelector_Magenta,
 ChunkExposureTimeSelector_Yellow,
 ChunkExposureTimeSelector_Infrared,
 ChunkExposureTimeSelector_Ultraviolet,
 ChunkExposureTimeSelector_Stage1,
 ChunkExposureTimeSelector_Stage2,
 NUM_CHUNKEXPOSURETIMESELECTOR }
- enum ChunkSourceIDEnums {
 ChunkSourceID_Source0,
 ChunkSourceID_Source1,
 ChunkSourceID_Source2,
 NUM_CHUNKSOURCEID }

- `enum ChunkRegionIDEnums {`
`ChunkRegionID_Region0,`
`ChunkRegionID_Region1,`
`ChunkRegionID_Region2,`
`NUM_CHUNKREGIONID }`
- `enum ChunkTransferStreamIDEnums {`
`ChunkTransferStreamID_Stream0,`
`ChunkTransferStreamID_Stream1,`
`ChunkTransferStreamID_Stream2,`
`ChunkTransferStreamID_Stream3,`
`NUM_CHUNKTRANSFERSTREAMID }`
- `enum ChunkScan3dDistanceUnitEnums {`
`ChunkScan3dDistanceUnit_Millimeter,`
`ChunkScan3dDistanceUnit_Inch,`
`NUM_CHUNKSCAN3DDISTANCEUNIT }`
- `enum ChunkScan3dOutputModeEnums {`
`ChunkScan3dOutputMode_UncalibratedC,`
`ChunkScan3dOutputMode_CalibratedABC_Grid,`
`ChunkScan3dOutputMode_CalibratedABC_PointCloud,`
`ChunkScan3dOutputMode_CalibratedAC,`
`ChunkScan3dOutputMode_CalibratedAC_Linescan,`
`ChunkScan3dOutputMode_CalibratedC,`
`ChunkScan3dOutputMode_CalibratedC_Linescan,`
`ChunkScan3dOutputMode_RectifiedC,`
`ChunkScan3dOutputMode_RectifiedC_Linescan,`
`ChunkScan3dOutputMode_DisparityC,`
`ChunkScan3dOutputMode_DisparityC_Linescan,`
`NUM_CHUNKSCAN3DOUTPUTMODE }`
- `enum ChunkScan3dCoordinateSystemEnums {`
`ChunkScan3dCoordinateSystem_Cartesian,`
`ChunkScan3dCoordinateSystem_Spherical,`
`ChunkScan3dCoordinateSystem_Cylindrical,`
`NUM_CHUNKSCAN3DCOORDINATESYSTEM }`
- `enum ChunkScan3dCoordinateSystemReferenceEnums {`
`ChunkScan3dCoordinateSystemReference_Anchor,`
`ChunkScan3dCoordinateSystemReference_Transformed,`
`NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }`
- `enum ChunkScan3dCoordinateSelectorEnums {`
`ChunkScan3dCoordinateSelector_CoordinateA,`
`ChunkScan3dCoordinateSelector_CoordinateB,`
`ChunkScan3dCoordinateSelector_CoordinateC,`
`NUM_CHUNKSCAN3DCOORDINATESELECTOR }`
- `enum ChunkScan3dCoordinateTransformSelectorEnums {`
`ChunkScan3dCoordinateTransformSelector_RotationX,`
`ChunkScan3dCoordinateTransformSelector_RotationY,`
`ChunkScan3dCoordinateTransformSelector_RotationZ,`
`ChunkScan3dCoordinateTransformSelector_TranslationX,`
`ChunkScan3dCoordinateTransformSelector_TranslationY,`
`ChunkScan3dCoordinateTransformSelector_TranslationZ,`
`NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }`
- `enum ChunkScan3dCoordinateReferenceSelectorEnums {`
`ChunkScan3dCoordinateReferenceSelector_RotationX,`
`ChunkScan3dCoordinateReferenceSelector_RotationY,`
`ChunkScan3dCoordinateReferenceSelector_RotationZ,`
`ChunkScan3dCoordinateReferenceSelector_TranslationX,`
`ChunkScan3dCoordinateReferenceSelector_TranslationY,`
`ChunkScan3dCoordinateReferenceSelector_TranslationZ,`
`NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }`

- enum DeviceTapGeometryEnums {
 - DeviceTapGeometry_Geometry_1X_1Y,
 - DeviceTapGeometry_Geometry_1X2_1Y,
 - DeviceTapGeometry_Geometry_1X2_1Y2,
 - DeviceTapGeometry_Geometry_2X_1Y,
 - DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
 - DeviceTapGeometry_Geometry_2XE_1Y2,
 - DeviceTapGeometry_Geometry_2XM_1Y,
 - DeviceTapGeometry_Geometry_2XM_1Y2,
 - DeviceTapGeometry_Geometry_1X_1Y2,
 - DeviceTapGeometry_Geometry_1X_2YE,
 - DeviceTapGeometry_Geometry_1X3_1Y,
 - DeviceTapGeometry_Geometry_3X_1Y,
 - DeviceTapGeometry_Geometry_1X,
 - DeviceTapGeometry_Geometry_1X2,
 - DeviceTapGeometry_Geometry_2X,
 - DeviceTapGeometry_Geometry_2XE,
 - DeviceTapGeometry_Geometry_2XM,
 - DeviceTapGeometry_Geometry_1X3,
 - DeviceTapGeometry_Geometry_3X,
 - DeviceTapGeometry_Geometry_1X4_1Y,
 - DeviceTapGeometry_Geometry_4X_1Y,
 - DeviceTapGeometry_Geometry_2X2_1Y,
 - DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
 - DeviceTapGeometry_Geometry_1X2_2YE,
 - DeviceTapGeometry_Geometry_2X_2YE,
 - DeviceTapGeometry_Geometry_2XE_2YE,
 - DeviceTapGeometry_Geometry_2XM_2YE,
 - DeviceTapGeometry_Geometry_1X4,
 - DeviceTapGeometry_Geometry_4X,
 - DeviceTapGeometry_Geometry_2X2,
 - DeviceTapGeometry_Geometry_2X2E,
 - DeviceTapGeometry_Geometry_2X2M,
 - DeviceTapGeometry_Geometry_1X8_1Y,
 - DeviceTapGeometry_Geometry_8X_1Y,
 - DeviceTapGeometry_Geometry_4X2_1Y,
 - DeviceTapGeometry_Geometry_2X2E_2YE,
 - DeviceTapGeometry_Geometry_1X8,
 - DeviceTapGeometry_Geometry_8X,
 - DeviceTapGeometry_Geometry_4X2,
 - DeviceTapGeometry_Geometry_4X2E,
 - DeviceTapGeometry_Geometry_4X2E_1Y,
 - DeviceTapGeometry_Geometry_1X10_1Y,
 - DeviceTapGeometry_Geometry_10X_1Y,
 - DeviceTapGeometry_Geometry_1X10,
 - DeviceTapGeometry_Geometry_10X,
 - NUM_DEVICETAPGEOMETRY }
- enum GevPhysicalLinkConfigurationEnums {
 - GevPhysicalLinkConfiguration_SingleLink,
 - GevPhysicalLinkConfiguration_MultiLink,
 - GevPhysicalLinkConfiguration_StaticLAG,
 - GevPhysicalLinkConfiguration_DynamicLAG,
 - NUM_GEVPHYSICALLINKCONFIGURATION }
- enum GevCurrentPhysicalLinkConfigurationEnums {
 - GevCurrentPhysicalLinkConfiguration_SingleLink,
 - GevCurrentPhysicalLinkConfiguration_MultiLink,
 - GevCurrentPhysicalLinkConfiguration_StaticLAG,
 - GevCurrentPhysicalLinkConfiguration_DynamicLAG,

```

NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

• enum GevIPConfigurationStatusEnums {
    GevIPConfigurationStatus_None,
    GevIPConfigurationStatus_PersistentIP,
    GevIPConfigurationStatus_DHCP,
    GevIPConfigurationStatus_LLA,
    GevIPConfigurationStatus_ForceIP,
    NUM_GEVIPCONFIGURATIONSTATUS }

• enum GevGVCPEExtendedStatusCodesSelectorEnums {
    GevGVCPEExtendedStatusCodesSelector_Version1_1,
    GevGVCPEExtendedStatusCodesSelector_Version2_0,
    NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }

• enum GevGVSPExtendedIDModeEnums {
    GevGVSPExtendedIDMode_Off,
    GevGVSPExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDIDMODE }

• enum ClConfigurationEnums {
    ClConfiguration_Base,
    ClConfiguration_Medium,
    ClConfiguration_Full,
    ClConfiguration_DualBase,
    ClConfiguration_EightyBit,
    NUM_CLCONFIGURATION }

• enum ClTimeSlotsCountEnums {
    ClTimeSlotsCount_One,
    ClTimeSlotsCount_Two,
    ClTimeSlotsCount_Three,
    NUM_CLTIMESLOTSCOUNT }

• enum CxpLinkConfigurationStatusEnums {
    CxpLinkConfigurationStatus_None,
    CxpLinkConfigurationStatus_Pending,
    CxpLinkConfigurationStatus_CXP1_X1,
    CxpLinkConfigurationStatus_CXP2_X1,
    CxpLinkConfigurationStatus_CXP3_X1,
    CxpLinkConfigurationStatus_CXP5_X1,
    CxpLinkConfigurationStatus_CXP6_X1,
    CxpLinkConfigurationStatus_CXP1_X2,
    CxpLinkConfigurationStatus_CXP2_X2,
    CxpLinkConfigurationStatus_CXP3_X2,
    CxpLinkConfigurationStatus_CXP5_X2,
    CxpLinkConfigurationStatus_CXP6_X2,
    CxpLinkConfigurationStatus_CXP1_X3,
    CxpLinkConfigurationStatus_CXP2_X3,
    CxpLinkConfigurationStatus_CXP3_X3,
    CxpLinkConfigurationStatus_CXP5_X3,
    CxpLinkConfigurationStatus_CXP6_X3,
    CxpLinkConfigurationStatus_CXP1_X4,
    CxpLinkConfigurationStatus_CXP2_X4,
    CxpLinkConfigurationStatus_CXP3_X4,
    CxpLinkConfigurationStatus_CXP5_X4,
    CxpLinkConfigurationStatus_CXP6_X4,
    CxpLinkConfigurationStatus_CXP1_X5,
    CxpLinkConfigurationStatus_CXP2_X5,
    CxpLinkConfigurationStatus_CXP3_X5,
    CxpLinkConfigurationStatus_CXP5_X5,
    CxpLinkConfigurationStatus_CXP6_X5,
    CxpLinkConfigurationStatus_CXP1_X6,
    CxpLinkConfigurationStatus_CXP2_X6,

```

```

CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

• enum CxpLinkConfigurationPreferredEnums {
CxpLinkConfigurationPreferred_CXP1_X1,
CxpLinkConfigurationPreferred_CXP2_X1,
CxpLinkConfigurationPreferred_CXP3_X1,
CxpLinkConfigurationPreferred_CXP5_X1,
CxpLinkConfigurationPreferred_CXP6_X1,
CxpLinkConfigurationPreferred_CXP1_X2,
CxpLinkConfigurationPreferred_CXP2_X2,
CxpLinkConfigurationPreferred_CXP3_X2,
CxpLinkConfigurationPreferred_CXP5_X2,
CxpLinkConfigurationPreferred_CXP6_X2,
CxpLinkConfigurationPreferred_CXP1_X3,
CxpLinkConfigurationPreferred_CXP2_X3,
CxpLinkConfigurationPreferred_CXP3_X3,
CxpLinkConfigurationPreferred_CXP5_X3,
CxpLinkConfigurationPreferred_CXP6_X3,
CxpLinkConfigurationPreferred_CXP1_X4,
CxpLinkConfigurationPreferred_CXP2_X4,
CxpLinkConfigurationPreferred_CXP3_X4,
CxpLinkConfigurationPreferred_CXP5_X4,
CxpLinkConfigurationPreferred_CXP6_X4,
CxpLinkConfigurationPreferred_CXP1_X5,
CxpLinkConfigurationPreferred_CXP2_X5,
CxpLinkConfigurationPreferred_CXP3_X5,
CxpLinkConfigurationPreferred_CXP5_X5,
CxpLinkConfigurationPreferred_CXP6_X5,
CxpLinkConfigurationPreferred_CXP1_X6,
CxpLinkConfigurationPreferred_CXP2_X6,
CxpLinkConfigurationPreferred_CXP3_X6,
CxpLinkConfigurationPreferred_CXP5_X6,
CxpLinkConfigurationPreferred_CXP6_X6,
NUM_CXPLINKCONFIGURATIONPREFERRED }

• enum CxpLinkConfigurationEnums {
CxpLinkConfiguration_Auto,
CxpLinkConfiguration_CXP1_X1,
CxpLinkConfiguration_CXP2_X1,
CxpLinkConfiguration_CXP3_X1,
CxpLinkConfiguration_CXP5_X1,
CxpLinkConfiguration_CXP6_X1,
CxpLinkConfiguration_CXP1_X2,
CxpLinkConfiguration_CXP2_X2,
CxpLinkConfiguration_CXP3_X2,
CxpLinkConfiguration_CXP5_X2,
CxpLinkConfiguration_CXP6_X2,
CxpLinkConfiguration_CXP1_X3,
CxpLinkConfiguration_CXP2_X3,
CxpLinkConfiguration_CXP3_X3,
CxpLinkConfiguration_CXP5_X3,
CxpLinkConfiguration_CXP6_X3,
CxpLinkConfiguration_CXP1_X4,
CxpLinkConfiguration_CXP2_X4,
CxpLinkConfiguration_CXP3_X4,
CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,

```

```

CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }
• enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
• enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

```

13.6.1 Detailed Description

13.6.2 Enumeration Type Documentation

13.6.2.1 AcquisitionModeEnums

```
enum AcquisitionModeEnums
```

< Sets the acquisition mode of the device. Continuous: acquires images continuously. Multi Frame: acquires a specified number of images before stopping acquisition. Single Frame: acquires 1 image before stopping acquisition.

Enumerator

AcquisitionMode_Continuous	
AcquisitionMode_SingleFrame	
AcquisitionMode_MultiFrame	
NUM_ACQUISITIONMODE	

13.6.2.2 AcquisitionStatusSelectorEnums

```
enum AcquisitionStatusSelectorEnums
```

< Selects the internal acquisition signal to read using AcquisitionStatus.

Enumerator

AcquisitionStatusSelector_AcquisitionTriggerWait	Device is currently waiting for a trigger for the capture of one or many frames.
AcquisitionStatusSelector_AcquisitionActive	Device is currently doing an acquisition of one or many frames.
AcquisitionStatusSelector_AcquisitionTransfer	Device is currently transferring an acquisition of one or many frames.
AcquisitionStatusSelector_FrameTriggerWait	Device is currently waiting for a frame start trigger.
AcquisitionStatusSelector_FrameActive	Device is currently doing the capture of a frame.
AcquisitionStatusSelector_ExposureActive	Device is doing the exposure of a frame.
NUM_ACQUISITIONSTATUSSELECTION	

13.6.2.3 ActionUnconditionalModeEnums

enum [ActionUnconditionalModeEnums](#)

< Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

Enumerator

ActionUnconditionalMode_Off	Unconditional mode is disabled.
ActionUnconditionalMode_On	Unconditional mode is enabled.
NUM_ACTIONUNCONDITIONALMODE	

13.6.2.4 AdcBitDepthEnums

enum [AdcBitDepthEnums](#)

< Selects which ADC bit depth to use. A higher ADC bit depth results in better image quality but slower maximum frame rate.

Enumerator

AdcBitDepth_Bit8	
AdcBitDepth_Bit10	
AdcBitDepth_Bit12	
AdcBitDepth_Bit14	
NUM_ADCBITDEPTH	

13.6.2.5 AutoAlgorithmSelectorEnums

enum `AutoAlgorithmSelectorEnums`

< Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

Enumerator

<code>AutoAlgorithmSelector_Awb</code>	Selects the Auto White Balance algorithm.
<code>AutoAlgorithmSelector_Ae</code>	Selects the Auto Exposure algorithm.
<code>NUM_AUTOALGORITHMSELECTOR</code>	

13.6.2.6 AutoExposureControlPriorityEnums

enum `AutoExposureControlPriorityEnums`

< Selects whether to adjust gain or exposure first. When gain priority is selected, the camera fixes the gain to 0 dB, and the exposure is adjusted according to the target grey level. If the maximum exposure is reached before the target grey level is hit, the gain starts to change to meet the target. This mode is used to have the minimum noise. When exposure priority is selected, the camera sets the exposure to a small value (default is 5 ms). The gain is adjusted according to the target grey level. If maximum gain is reached before the target grey level is hit, the exposure starts to change to meet the target. This mode is used to capture fast motion.

Enumerator

<code>AutoExposureControlPriority_Gain</code>	
<code>AutoExposureControlPriority_ExposureTime</code>	
<code>NUM_AUTOEXPOSURECONTROLPRIORITY</code>	

13.6.2.7 AutoExposureLightingModeEnums

enum `AutoExposureLightingModeEnums`

< Selects a lighting mode: Backlight, Frontlight or Normal (default). a. Backlight compensation: used when a strong light is coming from the back of the object. b. Frontlight compensation: used when a strong light is shining in the front of the object while the background is dark. c. Normal lighting: used when the object is not under backlight or frontlight conditions. When normal lighting is selected, metering modes are available.

Enumerator

<code>AutoExposureLightingMode_AutoDetect</code>	
<code>AutoExposureLightingMode_Backlight</code>	
<code>AutoExposureLightingMode_Frontlight</code>	
<code>AutoExposureLightingMode_Normal</code>	
<code>NUM_AUTOEXPOSURELIGHTINGMODE</code>	

13.6.2.8 AutoExposureMeteringModeEnums

enum `AutoExposureMeteringModeEnums`

< Selects a metering mode: average, spot, or partial metering. a. Average: Measures the light from the entire scene uniformly to determine the final exposure value. Every portion of the exposed area has the same contribution. b. Spot: Measures a small area (about 3%) in the center of the scene while the rest of the scene is ignored. This mode is used when the scene has a high contrast and the object of interest is relatively small. c. Partial: Measures the light from a larger area (about 11%) in the center of the scene. This mode is used when very dark or bright regions appear at the edge of the frame. Note: Metering mode is available only when Lighting Mode Selector is Normal.

Enumerator

<code>AutoExposureMeteringMode_Average</code>	
<code>AutoExposureMeteringMode_Spot</code>	
<code>AutoExposureMeteringMode_Partial</code>	
<code>AutoExposureMeteringMode_CenterWeighted</code>	
<code>AutoExposureMeteringMode_HistogramPeak</code>	
<code>NUM_AUTOEXPOSUREMETERINGMODE</code>	

13.6.2.9 AutoExposureTargetGreyValueAutoEnums

enum `AutoExposureTargetGreyValueAutoEnums`

< This indicates whether the target image grey level is automatically set by the camera or manually set by the user. Note that the target grey level is in the linear domain before gamma correction is applied.

Enumerator

<code>AutoExposureTargetGreyValueAuto_Off</code>	Target grey value is manually controlled
<code>AutoExposureTargetGreyValueAuto_Continuous</code>	Target grey value is constantly adapted by the device to maximize the dynamic range.
<code>NUM_AUTOEXPOSURETARGETGREYVALUEAUTO</code>	

13.6.2.10 BalanceRatioSelectorEnums

enum `BalanceRatioSelectorEnums`

< Selects a balance ratio to configure once a balance ratio control has been selected.

Enumerator

BalanceRatioSelector_Red	Selects the red balance ratio control for adjustment. The red balance ratio is relative to the green channel.
BalanceRatioSelector_Blue	Selects the blue balance ratio control for adjustment. The blue balance ratio is relative to the green channel.
NUM_BALANCERATIOSELECTOR	

13.6.2.11 BalanceWhiteAutoEnums

enum `BalanceWhiteAutoEnums`

< White Balance compensates for color shifts caused by different lighting conditions. It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

Enumerator

BalanceWhiteAuto_Off	Sets operation mode to Off, which is manual control.
BalanceWhiteAuto_Once	Sets operation mode to once. Once runs for a number of iterations and then sets White Balance Auto to Off.
BalanceWhiteAuto_Continuous	Sets operation mode to continuous. Continuous automatically adjusts values if the colors are imbalanced.
NUM_BALANCEWHITEAUTO	

13.6.2.12 BalanceWhiteAutoProfileEnums

enum `BalanceWhiteAutoProfileEnums`

< Selects the profile used by BalanceWhiteAuto.

Enumerator

BalanceWhiteAutoProfile_Indoor	Indoor auto white balance Profile. Can be used to compensate for artificial lighting.
BalanceWhiteAutoProfile_Outdoor	Outdoor auto white balance profile. Designed for scenes with natural lighting.
NUM_BALANCEWHITEAUTOPROFILE	

13.6.2.13 BinningHorizontalModeEnums

enum `BinningHorizontalModeEnums`

<

Enumerator

BinningHorizontalMode_Sum	The response from the combined horizontal cells is added, resulting in increased sensitivity (a brighter image).
BinningHorizontalMode_Average	The response from the combined horizontal cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning.
NUM_BINNINGHORIZONTALMODE	

13.6.2.14 BinningSelectorEnums

```
enum BinningSelectorEnums
```

< Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

Enumerator

BinningSelector_All	The total amount of binning to be performed on the captured sensor data.
BinningSelector_Sensor	The portion of binning to be performed on the sensor directly.
BinningSelector_ISP	The portion of binning to be performed by the image signal processing engine (ISP) outside of the sensor. Note: the ISP can be disabled.
NUM_BINNINGSELECTOR	

13.6.2.15 BinningVerticalModeEnums

```
enum BinningVerticalModeEnums
```

<

Enumerator

BinningVerticalMode_Sum	The response from the combined vertical cells is added, resulting in increased sensitivity (a brighter image).
BinningVerticalMode_Average	The response from the combined vertical cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning.
NUM_BINNINGVERTICALMODE	

13.6.2.16 BlackLevelAutoBalanceEnums

```
enum BlackLevelAutoBalanceEnums
```

< Controls the mode for automatic black level balancing between the sensor color channels or taps. The black level coefficients of each channel are adjusted so they are matched.

Enumerator

BlackLevelAutoBalance_Off	Black level tap balancing is user controlled using BlackLevel.
BlackLevelAutoBalance_Once	Black level tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.
BlackLevelAutoBalance_Continuous	Black level tap balancing is constantly adjusted by the device.
NUM_BLACKLEVELAUTOBALANCE	

13.6.2.17 BlackLevelAutoEnums

enum [BlackLevelAutoEnums](#)

< Controls the mode for automatic black level adjustment. The exact algorithm used to implement this adjustment is device-specific.

Enumerator

BlackLevelAuto_Off	Analog black level is user controlled using BlackLevel.
BlackLevelAuto_Once	Analog black level is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.
BlackLevelAuto_Continuous	Analog black level is constantly adjusted by the device.
NUM_BLACKLEVELAUTO	

13.6.2.18 BlackLevelSelectorEnums

enum [BlackLevelSelectorEnums](#)

< Selects which black level to control. Only All can be set by the user. Analog and Digital are read-only.

Enumerator

BlackLevelSelector_All	
BlackLevelSelector_Analog	
BlackLevelSelector_Digital	
NUM_BLACKLEVELSELECTOR	

13.6.2.19 ChunkBlackLevelSelectorEnums

enum [ChunkBlackLevelSelectorEnums](#)

< Selects which black level to retrieve

Enumerator

ChunkBlackLevelSelector_All	
NUM_CHUNKBLACKLEVELSELECTOR	

13.6.2.20 ChunkCounterSelectorEnums

enum [ChunkCounterSelectorEnums](#)

< Selects which counter to retrieve data from.

Enumerator

ChunkCounterSelector_Counter0	Selects the counter 0.
ChunkCounterSelector_Counter1	Selects the counter 1.
ChunkCounterSelector_Counter2	Selects the counter 2.
NUM_CHUNKCOUNTERSELECTOR	

13.6.2.21 ChunkEncoderSelectorEnums

enum [ChunkEncoderSelectorEnums](#)

< Selects which Encoder to retrieve data from.

Enumerator

ChunkEncoderSelector_Encoder0	Selects the first Encoder.
ChunkEncoderSelector_Encoder1	Selects the first Encoder.
ChunkEncoderSelector_Encoder2	Selects the second Encoder.
NUM_CHUNKENCODERSELECTOR	

13.6.2.22 ChunkEncoderStatusEnums

enum [ChunkEncoderStatusEnums](#)

< Returns the motion status of the selected encoder.

Enumerator

ChunkEncoderStatus_EncoderUp	The encoder counter last incremented.
ChunkEncoderStatus_EncoderDown	The encoder counter last decremented.
ChunkEncoderStatus_EncoderIdle	The encoder is not active.
ChunkEncoderStatus_EncoderStatic	No motion within the EncoderTimeout time.
NUM_CHUNKENCODERSTATUS	

13.6.2.23 ChunkExposureTimeSelectorEnums

enum `ChunkExposureTimeSelectorEnums`

< Selects which exposure time is read by the ChunkExposureTime feature.

Enumerator

ChunkExposureTimeSelector_Common	Selects the common ExposureTime.
ChunkExposureTimeSelector_Red	Selects the red common ExposureTime.
ChunkExposureTimeSelector_Green	Selects the green ExposureTime.
ChunkExposureTimeSelector_Blue	Selects the blue ExposureTime.
ChunkExposureTimeSelector_Cyan	Selects the cyan common ExposureTime..
ChunkExposureTimeSelector_Magenta	Selects the magenta ExposureTime..
ChunkExposureTimeSelector_Yellow	Selects the yellow ExposureTime..
ChunkExposureTimeSelector_Infrared	Selects the infrared ExposureTime.
ChunkExposureTimeSelector_Ultraviolet	Selects the ultraviolet ExposureTime.
ChunkExposureTimeSelector_Stage1	Selects the first stage ExposureTime.
ChunkExposureTimeSelector_Stage2	Selects the second stage ExposureTime.
NUM_CHUNKEXPOSURETIMESELECTOR	

13.6.2.24 ChunkGainSelectorEnums

enum `ChunkGainSelectorEnums`

< Selects which gain to retrieve

Enumerator

ChunkGainSelector_All	
ChunkGainSelector_Red	
ChunkGainSelector_Green	
ChunkGainSelector_Blue	
NUM_CHUNKGAINSELECTOR	

13.6.2.25 ChunkImageComponentEnums

enum `ChunkImageComponentEnums`

< Returns the component of the payload image. This can be used to identify the image component of a generic part in a multipart transfer.

Enumerator

ChunkImageComponent_Intensity	The image data is the intensity component.
ChunkImageComponent_Color	The image data is color component.
ChunkImageComponent_Infrared	The image data is infrared component.
ChunkImageComponent_Ultraviolet	The image data is the ultraviolet component.
ChunkImageComponent_Range	The image data is the range (distance) component.
ChunkImageComponent_Disparity	The image data is the disparity component.
ChunkImageComponent_Confidence	The image data is the confidence map component.
ChunkImageComponent_Scatter	The image data is the scatter component.
NUM_CHUNKIMAGECOMPONENT	

13.6.2.26 ChunkPixelFormatEnums

enum [ChunkPixelFormatEnums](#)

< Format of the pixel provided by the camera

Enumerator

ChunkPixelFormat_Mono8	
ChunkPixelFormat_Mono12Packed	
ChunkPixelFormat_Mono16	
ChunkPixelFormat_RGB8Packed	
ChunkPixelFormat_YUV422Packed	
ChunkPixelFormat_BayerGR8	
ChunkPixelFormat_BayerRG8	
ChunkPixelFormat_BayerGB8	
ChunkPixelFormat_BayerBG8	
ChunkPixelFormat_YCbCr601_422_8_CbYCrY	
NUM_CHUNKPIXELFORMAT	

13.6.2.27 ChunkRegionIDEnums

enum [ChunkRegionIDEnums](#)

< Returns the identifier of Region that the image comes from.

Enumerator

ChunkRegionID_Region0	Image comes from the Region 0.
ChunkRegionID_Region1	Image comes from the Region 1.
ChunkRegionID_Region2	Image comes from the Region 2.
NUM_CHUNKREGIONID	

13.6.2.28 ChunkScan3dCoordinateReferenceSelectorEnums

enum [ChunkScan3dCoordinateReferenceSelectorEnums](#)

< Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

Enumerator

ChunkScan3dCoordinateReferenceSelector_RotationX	Rotation around X axis.
ChunkScan3dCoordinateReferenceSelector_RotationY	Rotation around Y axis.
ChunkScan3dCoordinateReferenceSelector_RotationZ	Rotation around Z axis.
ChunkScan3dCoordinateReferenceSelector_TranslationX	X axis translation.
ChunkScan3dCoordinateReferenceSelector_TranslationY	Y axis translation.
ChunkScan3dCoordinateReferenceSelector_TranslationZ	Z axis translation.
NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR	

13.6.2.29 ChunkScan3dCoordinateSelectorEnums

enum [ChunkScan3dCoordinateSelectorEnums](#)

< Selects which Coordinate to retrieve data from.

Enumerator

ChunkScan3dCoordinateSelector_CoordinateA	The first (X or Theta) coordinate
ChunkScan3dCoordinateSelector_CoordinateB	The second (Y or Phi) coordinate
ChunkScan3dCoordinateSelector_CoordinateC	The third (Z or Rho) coordinate.
NUM_CHUNKSCAN3DCOORDINATESELECTOR	

13.6.2.30 ChunkScan3dCoordinateSystemEnums

enum [ChunkScan3dCoordinateSystemEnums](#)

< Returns the Coordinate [System](#) of the image included in the payload.

Enumerator

ChunkScan3dCoordinateSystem_Cartesian	Default value. 3-axis orthogonal, right-hand X-Y-Z.
ChunkScan3dCoordinateSystem_Spherical	A Theta-Phi-Rho coordinate system.
ChunkScan3dCoordinateSystem_Cylindrical	A Theta-Y-Rho coordinate system.
NUM_CHUNKSCAN3DCOORDINATESYSTEM	

13.6.2.31 ChunkScan3dCoordinateSystemReferenceEnums

enum [ChunkScan3dCoordinateSystemReferenceEnums](#)

< Returns the Coordinate [System](#) Position of the image included in the payload.

Enumerator

ChunkScan3dCoordinateSystemReference_Anchor	Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.
ChunkScan3dCoordinateSystemReference_↔ Transformed	Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices.
NUM_CHUNKSCAN3DCOORDINATESYSTEMREF↔ ERENCE	

13.6.2.32 ChunkScan3dCoordinateTransformSelectorEnums

enum [ChunkScan3dCoordinateTransformSelectorEnums](#)

< Selector for transform values.

Enumerator

ChunkScan3dCoordinateTransformSelector_RotationX	Rotation around X axis.
ChunkScan3dCoordinateTransformSelector_RotationY	Rotation around Y axis.
ChunkScan3dCoordinateTransformSelector_RotationZ	Rotation around Z axis.
ChunkScan3dCoordinateTransformSelector_TranslationX	Translation along X axis.
ChunkScan3dCoordinateTransformSelector_TranslationY	Translation along Y axis.
ChunkScan3dCoordinateTransformSelector_TranslationZ	Translation along Z axis.
NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR	

13.6.2.33 ChunkScan3dDistanceUnitEnums

enum [ChunkScan3dDistanceUnitEnums](#)

< Returns the Distance Unit of the payload image.

Enumerator

ChunkScan3dDistanceUnit_Millimeter	Default value. Distance values are in millimeter units.
ChunkScan3dDistanceUnit_Inch	Distance values are in inch units.
NUM_CHUNKSCAN3DDISTANCEUNIT	

13.6.2.34 ChunkScan3dOutputModeEnums

enum [ChunkScan3dOutputModeEnums](#)

< Returns the Calibrated Mode of the payload image.

Enumerator

ChunkScan3dOutputMode_UncalibratedC	Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.
ChunkScan3dOutputMode_CalibratedABC_Grid	3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.
ChunkScan3dOutputMode_CalibratedABC_Point↔ Cloud	3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.
ChunkScan3dOutputMode_CalibratedAC	2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.
ChunkScan3dOutputMode_CalibratedAC_Linescan	2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.
ChunkScan3dOutputMode_CalibratedC	Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.
ChunkScan3dOutputMode_CalibratedC_Linescan	Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.
ChunkScan3dOutputMode_RectifiedC	Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats.
ChunkScan3dOutputMode_RectifiedC_Linescan	Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D_C pixels. The B (Y) axis comes from the encoder chunk value.
ChunkScan3dOutputMode_DisparityC	Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.
ChunkScan3dOutputMode_DisparityC_Linescan	Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value.
NUM_CHUNKSCAN3DOUTPUTMODE	

13.6.2.35 ChunkSelectorEnums

enum [ChunkSelectorEnums](#)

< Selects which chunk data to enable or disable.

Enumerator

ChunkSelector_Image	
ChunkSelector_CRC	
ChunkSelector_FrameID	
ChunkSelector_OffsetX	
ChunkSelector_OffsetY	
ChunkSelector_Width	
ChunkSelector_Height	
ChunkSelector_ExposureTime	
ChunkSelector_Gain	
ChunkSelector_BlackLevel	
ChunkSelector_PixelFormat	
ChunkSelector_Timestamp	
ChunkSelector_SequencerSetActive	
ChunkSelector_SerialData	
ChunkSelector_ExposureEndLineStatusAll	
NUM_CHUNKSELECTOR	

13.6.2.36 ChunkSourceIDEnums

enum [ChunkSourceIDEnums](#)

< Returns the identifier of Source that the image comes from.

Enumerator

ChunkSourceID_Source0	Image comes from the Source 0.
ChunkSourceID_Source1	Image comes from the Source 1.
ChunkSourceID_Source2	Image comes from the Source 2.
NUM_CHUNKSOURCEID	

13.6.2.37 ChunkTimerSelectorEnums

enum [ChunkTimerSelectorEnums](#)

< Selects which Timer to retrieve data from.

Enumerator

ChunkTimerSelector_Timer0	Selects the first Timer.
ChunkTimerSelector_Timer1	Selects the first Timer.
ChunkTimerSelector_Timer2	Selects the second Timer.
NUM_CHUNKTIMERSELECTOR	

13.6.2.38 ChunkTransferStreamIDEnums

enum [ChunkTransferStreamIDEnums](#)

< Returns identifier of the stream that generated this block.

Enumerator

ChunkTransferStreamID_Stream0	Data comes from Stream0.
ChunkTransferStreamID_Stream1	Data comes from Stream1.
ChunkTransferStreamID_Stream2	Data comes from Stream2.
ChunkTransferStreamID_Stream3	Data comes from Stream3.
NUM_CHUNKTRANSFERSTREAMID	

13.6.2.39 ClConfigurationEnums

enum [ClConfigurationEnums](#)

< This [Camera](#) Link specific feature describes the configuration used by the camera. It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitization, Taps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera.

Enumerator

ClConfiguration_Base	Standard base configuration described by the Camera Link standard.
ClConfiguration_Medium	Standard medium configuration described by the Camera Link standard.
ClConfiguration_Full	Standard full configuration described by the Camera Link standard.
ClConfiguration_DualBase	The camera streams the data from multiple taps (that do not fit in the standard base configuration) through two Camera Link base ports. It is responsibility of the application or frame grabber to reconstruct the full image. Only one of the ports (fixed) serves as the "master" for serial communication and triggering.
ClConfiguration_EightyBit	Standard 80-bit configuration with 10 taps of 8 bits or 8 taps of 10 bits, as described by the Camera Link standard.
NUM_CLCONFIGURATION	

13.6.2.40 CTimeSlotsCountEnums

```
enum CTimeSlotsCountEnums
```

< This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Enumerator

CTimeSlotsCount_One	One
CTimeSlotsCount_Two	Two
CTimeSlotsCount_Three	Three
NUM_CLTIMESLOTSCOUNT	

13.6.2.41 ColorTransformationSelectorEnums

```
enum ColorTransformationSelectorEnums
```

< Selects which Color Transformation module is controlled by the various Color Transformation features

Enumerator

ColorTransformationSelector_RGBtoRGB	
ColorTransformationSelector_RGBtoYUV	
NUM_COLORTRANSFORMATIONSELECTOR	

13.6.2.42 ColorTransformationValueSelectorEnums

```
enum ColorTransformationValueSelectorEnums
```

< Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module

Enumerator

ColorTransformationValueSelector_Gain00	
ColorTransformationValueSelector_Gain01	
ColorTransformationValueSelector_Gain02	
ColorTransformationValueSelector_Gain10	
ColorTransformationValueSelector_Gain11	
ColorTransformationValueSelector_Gain12	
ColorTransformationValueSelector_Gain20	
ColorTransformationValueSelector_Gain21	
ColorTransformationValueSelector_Gain22	
ColorTransformationValueSelector_Offset0	
ColorTransformationValueSelector_Offset1	
ColorTransformationValueSelector_Offset2	
NUM_COLORTRANSFORMATIONVALUESELECTOR	

13.6.2.43 CompressionSaturationPriorityEnums

enum [CompressionSaturationPriorityEnums](#)

< When FrameRate is enabled, camera drops frames if datarate is saturated. If FrameRate is disabled, camera adjusts the framerate to match the maximum achievable datarate.

Enumerator

CompressionSaturationPriority_DropFrame	Frames which will cause the MaxDatarateThreshold to be exceeded will not be transmitted. Requires FrameRateEnable to be True
CompressionSaturationPriority_ReduceFrameRate	AcquisitionFrameRate is dynamically adjusted to the highest possible value without exceeding the MaxDatarateThreshold.
NUM_COMPRESSIONSATURATIONPRIORITY	

13.6.2.44 CounterEventActivationEnums

enum [CounterEventActivationEnums](#)

< Selects the activation mode of the event to increment the Counter.

Enumerator

CounterEventActivation_LevelLow	
CounterEventActivation_LevelHigh	
CounterEventActivation_FallingEdge	
CounterEventActivation_RisingEdge	
CounterEventActivation_AnyEdge	
NUM_COUNTEREVENTACTIVATION	

13.6.2.45 CounterEventSourceEnums

enum [CounterEventSourceEnums](#)

< Selects the event that will increment the counter

Enumerator

CounterEventSource_Off	Off
CounterEventSource_MHzTick	MHzTick
CounterEventSource_Line0	Line0

Enumerator

CounterEventSource_Line1	Line1
CounterEventSource_Line2	Line2
CounterEventSource_Line3	Line3
CounterEventSource_UserOutput0	UserOutput0
CounterEventSource_UserOutput1	UserOutput1
CounterEventSource_UserOutput2	UserOutput2
CounterEventSource_UserOutput3	UserOutput3
CounterEventSource_Counter0Start	Counter0Start
CounterEventSource_Counter1Start	Counter1Start
CounterEventSource_Counter0End	Counter0End
CounterEventSource_Counter1End	Counter1End
CounterEventSource_LogicBlock0	LogicBlock0
CounterEventSource_LogicBlock1	LogicBlock1
CounterEventSource_ExposureStart	ExposureStart
CounterEventSource_ExposureEnd	ExposureEnd
CounterEventSource_FrameTriggerWait	FrameTriggerWait
NUM_COUNTEREVENTSOURCE	

13.6.2.46 CounterResetActivationEnums

enum CounterResetActivationEnums

< Selects the Activation mode of the Counter Reset Source signal.

Enumerator

CounterResetActivation_LevelLow	
CounterResetActivation_LevelHigh	
CounterResetActivation_FallingEdge	
CounterResetActivation_RisingEdge	
CounterResetActivation_AnyEdge	
NUM_COUNTERRESETACTIVATION	

13.6.2.47 CounterResetSourceEnums

enum CounterResetSourceEnums

< Selects the signal that will be the source to reset the counter.

Enumerator

CounterResetSource_Off	Off

Enumerator

CounterResetSource_Line0	Line0
CounterResetSource_Line1	Line1
CounterResetSource_Line2	Line2
CounterResetSource_Line3	Line3
CounterResetSource_UserOutput0	UserOutput0
CounterResetSource_UserOutput1	UserOutput1
CounterResetSource_UserOutput2	UserOutput2
CounterResetSource_UserOutput3	UserOutput3
CounterResetSource_Counter0Start	Counter0Start
CounterResetSource_Counter1Start	Counter1Start
CounterResetSource_Counter0End	Counter0End
CounterResetSource_Counter1End	Counter1End
CounterResetSource_LogicBlock0	LogicBlock0
CounterResetSource_LogicBlock1	LogicBlock1
CounterResetSource_ExposureStart	ExposureStart
CounterResetSource_ExposureEnd	ExposureEnd
CounterResetSource_FrameTriggerWait	FrameTriggerWait
NUM_COUNTERRESETSOURCE	

13.6.2.48 CounterSelectorEnums

```
enum CounterSelectorEnums
```

< Selects which counter to configure

Enumerator

CounterSelector_Counter0	
CounterSelector_Counter1	
NUM_COUNTERSELECTOR	

13.6.2.49 CounterStatusEnums

```
enum CounterStatusEnums
```

< Returns the current status of the counter.

Enumerator

CounterStatus_CounterIdle	The counter is idle.
CounterStatus_CounterTriggerWait	The counter is waiting for a start trigger.
CounterStatus_CounterActive	The counter is counting for the specified duration.
CounterStatus_CounterCompleted	The counter reached the CounterDuration count.
CounterStatus_CounterOverflow	The counter reached its maximum possible count.
NUM_COUNTERSTATUS	

13.6.2.50 CounterTriggerActivationEnums

```
enum CounterTriggerActivationEnums
```

< Selects the activation mode of the trigger to start the counter.

Enumerator

CounterTriggerActivation_LevelLow	
CounterTriggerActivation_LevelHigh	
CounterTriggerActivation_FallingEdge	
CounterTriggerActivation_RisingEdge	
CounterTriggerActivation_AnyEdge	
NUM_COUNTERTRIGGERACTIVATION	

13.6.2.51 CounterTriggerSourceEnums

```
enum CounterTriggerSourceEnums
```

< Selects the source of the trigger to start the counter

Enumerator

CounterTriggerSource_Off	Off
CounterTriggerSource_Line0	Line0
CounterTriggerSource_Line1	Line1
CounterTriggerSource_Line2	Line2
CounterTriggerSource_Line3	Line3
CounterTriggerSource_UserOutput0	UserOutput0
CounterTriggerSource_UserOutput1	UserOutput1
CounterTriggerSource_UserOutput2	UserOutput2
CounterTriggerSource_UserOutput3	UserOutput3
CounterTriggerSource_Counter0Start	Counter0Start
CounterTriggerSource_Counter1Start	Counter1Start
CounterTriggerSource_Counter0End	Counter0End
CounterTriggerSource_Counter1End	Counter1End
CounterTriggerSource_LogicBlock0	LogicBlock0
CounterTriggerSource_LogicBlock1	LogicBlock1
CounterTriggerSource_ExposureStart	ExposureStart
CounterTriggerSource_ExposureEnd	ExposureEnd
CounterTriggerSource_FrameTriggerWait	FrameTriggerWait
NUM_COUNTERTRIGGERSOURCE	

13.6.2.52 CxpConnectionTestModeEnums

enum [CxpConnectionTestModeEnums](#)

< Enables the test mode for an individual physical connection of the Device.

Enumerator

CxpConnectionTestMode_Off	Off
CxpConnectionTestMode_Mode1	Mode 1
NUM_CXP_CONNECTION_TEST_MODE	

13.6.2.53 CxpLinkConfigurationEnums

enum [CxpLinkConfigurationEnums](#)

< This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device. In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus.

Enumerator

CxpLinkConfiguration_Auto	Sets Automatic discovery for the Link Configuration.
CxpLinkConfiguration_CXP1_X1	Force the Link to 1 Connection operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X1	Force the Link to 1 Connection operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X1	Force the Link to 1 Connection operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X1	Force the Link to 1 Connection operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X1	Force the Link to 1 Connection operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X2	Force the Link to 2 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X2	Force the Link to 2 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X2	Force the Link to 2 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X2	Force the Link to 2 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X2	Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X3	Force the Link to 3 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X3	Force the Link to 3 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X3	Force the Link to 3 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X3	Force the Link to 3 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X3	Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X4	Force the Link to 4 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X4	Force the Link to 4 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X4	Force the Link to 4 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X4	Force the Link to 4 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X4	Force the Link to 4 Connections operating at CXP-6 speed (6.25 Gbps).

Enumerator

CxpLinkConfiguration_CXP1_X5	Force the Link to 5 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X5	Force the Link to 5 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X5	Force the Link to 5 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X5	Force the Link to 5 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X5	Force the Link to 5 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X6	Force the Link to 6 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X6	Force the Link to 6 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X6	Force the Link to 6 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X6	Force the Link to 6 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X6	Force the Link to 6 Connections operating at CXP-6 speed (6.25 Gbps).
NUM_CXPLINKCONFIGURATION	

13.6.2.54 CxpLinkConfigurationPreferredEnums

```
enum CxpLinkConfigurationPreferredEnums
```

< Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

Enumerator

CxpLinkConfigurationPreferred_CXP1_X1	1 Connection operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X1	1 Connection operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X1	1 Connection operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X1	1 Connection operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X1	1 Connection operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X2	2 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X2	2 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X2	2 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X2	2 Connections operating at CXP-4 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X2	3 Connections operating at CXP-5 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X3	3 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X3	3 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X3	3 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X3	3 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X3	3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X4	4 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X4	4 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X4	4 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X4	4 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X4	4 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X5	5 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X5	5 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X5	5 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X5	5 Connections operating at CXP-5 speed (5.00 Gbps).

Enumerator

CxpLinkConfigurationPreferred_CXP6_X5	5 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X6	6 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X6	6 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X6	6 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X6	6 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X6	6 Connections operating at CXP-6 speed (6.25 Gbps).
NUM_CXPLINKCONFIGURATIONPREFERRED	

13.6.2.55 CxpLinkConfigurationStatusEnums

```
enum CxpLinkConfigurationStatusEnums
```

< This feature indicates the current and active Link configuration used by the Device.

Enumerator

CxpLinkConfigurationStatus_None	The Link configuration of the Device is unknown. Either the configuration operation has failed or there is nothing connected.
CxpLinkConfigurationStatus_Pending	The Device is in the process of configuring the Link. The Link cannot be used yet.
CxpLinkConfigurationStatus_CXP1_X1	1 Connection operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X1	1 Connection operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X1	1 Connection operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X1	1 Connection operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X1	1 Connection operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X2	2 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X2	2 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X2	2 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X2	2 Connections operating at CXP-4 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X2	3 Connections operating at CXP-5 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X3	3 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X3	3 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X3	3 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X3	3 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X3	3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X4	4 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X4	4 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X4	4 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X4	4 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X4	4 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X5	5 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X5	5 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X5	5 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X5	5 Connections operating at CXP-5 speed (5.00 Gbps).

Enumerator

CxpLinkConfigurationStatus_CXP6_X5	5 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X6	6 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X6	6 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X6	6 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X6	6 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X6	6 Connections operating at CXP-6 speed (6.25 Gbps).
NUM_CXPLINKCONFIGURATIONSTATUS	

13.6.2.56 CxpPoCxpStatusEnums

```
enum CxpPoCxpStatusEnums
```

< Returns the Power over CoaXPress (PoCXP) status of the Device.

Enumerator

CxpPoCxpStatus_Auto	Normal automatic PoCXP operation.
CxpPoCxpStatus_Off	PoCXP is forced off.
CxpPoCxpStatus_Tripped	The Link has shut down because of an over-current trip.
NUM_CXPPOCXPSTATUS	

13.6.2.57 DecimationHorizontalModeEnums

```
enum DecimationHorizontalModeEnums
```

< The mode used to reduce the horizontal resolution when DecimationHorizontal is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Enumerator

DecimationHorizontalMode_Discard	The value of every Nth pixel is kept, others are discarded.
NUM_DECIMATIONHORIZONTALMODE	

13.6.2.58 DecimationSelectorEnums

```
enum DecimationSelectorEnums
```

< Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

Enumerator

DecimationSelector_All	The total amount of decimation to be performed on the captured image data.
DecimationSelector_Sensor	The portion of decimation to be performed on the sensor directly. Currently this is the only decimation layer available and hence is identical to the "All" layer. All decimation modification should therefore be done via the "All" layer only.
NUM_DECIMATIONSELECTOR	

13.6.2.59 DecimationVerticalModeEnums

enum `DecimationVerticalModeEnums`

< The mode used to reduce the vertical resolution when DecimationVertical is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Enumerator

DecimationVerticalMode_Discard	The value of every Nth pixel is kept, others are discarded.
NUM_DECIMATIONVERTICALMODE	

13.6.2.60 DefectCorrectionModeEnums

enum `DefectCorrectionModeEnums`

< Controls the method used for replacing defective pixels.

Enumerator

DefectCorrectionMode_Average	Pixels are replaced with the average of their neighbours. This is the normal mode of operation.
DefectCorrectionMode_Highlight	Pixels are replaced with the maximum pixel value (i.e., 255 for 8-bit images). Can be used for debugging the table.
DefectCorrectionMode_Zero	Pixels are replaced by the value zero. Can be used for testing the table.
NUM_DEFECTCORRECTIONMODE	

13.6.2.61 DeinterlacingEnums

enum `DeinterlacingEnums`

< Controls how the device performs de-interlacing.

Enumerator

Deinterlacing_Off	The device doesn't perform de-interlacing.
Deinterlacing_LineDuplication	The device performs de-interlacing by outputting each line of each field twice.
Deinterlacing_Weave	The device performs de-interlacing by interleaving the lines of all fields.
NUM_DEINTERLACING	

13.6.2.62 DeviceCharacterSetEnums

enum [DeviceCharacterSetEnums](#)

< Character set used by the strings of the device's bootstrap registers.

Enumerator

DeviceCharacterSet_UTF8	
DeviceCharacterSet_ASCII	
NUM_DEVICECHARACTERSET	

13.6.2.63 DeviceClockSelectorEnums

enum [DeviceClockSelectorEnums](#)

< Selects the clock frequency to access from the device.

Enumerator

DeviceClockSelector_Sensor	Clock frequency of the image sensor of the camera.
DeviceClockSelector_SensorDigitization	Clock frequency of the camera A/D conversion stage.
DeviceClockSelector_CameraLink	Frequency of the Camera Link clock.
NUM_DEVICECLOCKSELECTOR	

13.6.2.64 DeviceConnectionStatusEnums

enum [DeviceConnectionStatusEnums](#)

< Indicates the status of the specified Connection.

Enumerator

DeviceConnectionStatus_Active	Connection is in use.
DeviceConnectionStatus_Inactive	Connection is not in use.
NUM_DEVICECONNECTIONSTATUS	

13.6.2.65 DeviceIndicatorModeEnums

enum `DeviceIndicatorModeEnums`

< Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

Enumerator

DeviceIndicatorMode_Inactive	
DeviceIndicatorMode_Active	
DeviceIndicatorMode_ErrorStatus	
NUM_DEVICEINDICATORMODE	

13.6.2.66 DeviceLinkHeartbeatModeEnums

enum `DeviceLinkHeartbeatModeEnums`

< Activate or deactivate the Link's heartbeat.

Enumerator

DeviceLinkHeartbeatMode_On	Enables the Link heartbeat.
DeviceLinkHeartbeatMode_Off	Disables the Link heartbeat.
NUM_DEVICELINKHEARTBEATMODE	

13.6.2.67 DeviceLinkThroughputLimitModeEnums

enum `DeviceLinkThroughputLimitModeEnums`

< Controls if the DeviceLinkThroughputLimit is active. When disabled, lower level TL specific features are expected to control the throughput. When enabled, DeviceLinkThroughputLimit controls the overall throughput.

Enumerator

DeviceLinkThroughputLimitMode_On	Enables the DeviceLinkThroughputLimit feature.
DeviceLinkThroughputLimitMode_Off	Disables the DeviceLinkThroughputLimit feature.
NUM_DEVICELINKTHROUGHPUTLIMITMODE	

13.6.2.68 DevicePowerSupplySelectorEnums

enum [DevicePowerSupplySelectorEnums](#)

< Selects the power supply source to control or read.

Enumerator

DevicePowerSupplySelector_External	
NUM_DEVICEPOWERSUPPLYSELECTOR	

13.6.2.69 DeviceRegistersEndiannessEnums

enum [DeviceRegistersEndiannessEnums](#)

< Endianness of the registers of the device.

Enumerator

DeviceRegistersEndianness_Little	
DeviceRegistersEndianness_Big	
NUM_DEVICEREGISTERSENDIANNES	

13.6.2.70 DeviceScanTypeEnums

enum [DeviceScanTypeEnums](#)

< Scan type of the sensor of the device.

Enumerator

DeviceScanType_Areasan	
NUM_DEVICEESCANTYPE	

13.6.2.71 DeviceSerialPortBaudRateEnums

enum [DeviceSerialPortBaudRateEnums](#)

< This feature controls the baud rate used by the selected serial port.

Enumerator

DeviceSerialPortBaudRate_Baud9600	Serial port speed of 9600 baud.
DeviceSerialPortBaudRate_Baud19200	Serial port speed of 19200 baud.
DeviceSerialPortBaudRate_Baud38400	Serial port speed of 38400 baud.
DeviceSerialPortBaudRate_Baud57600	Serial port speed of 57600 baud.
DeviceSerialPortBaudRate_Baud115200	Serial port speed of 115200 baud.
DeviceSerialPortBaudRate_Baud230400	Serial port speed of 230400 baud.
DeviceSerialPortBaudRate_Baud460800	Serial port speed of 460800 baud.
DeviceSerialPortBaudRate_Baud921600	Serial port speed of 921600 baud.
NUM_DEVICESERIALPORTBAUDRATE	

13.6.2.72 DeviceSerialPortSelectorEnums

```
enum DeviceSerialPortSelectorEnums
```

< Selects which serial port of the device to control.

Enumerator

DeviceSerialPortSelector_CameraLink	Serial port associated to the Camera link connection.
NUM_DEVICESERIALPORTSELECTOR	

13.6.2.73 DeviceStreamChannelEndiannessEnums

```
enum DeviceStreamChannelEndiannessEnums
```

< Endianness of multi-byte pixel data for this stream.

Enumerator

DeviceStreamChannelEndianness_Big	Stream channel data is big Endian.
DeviceStreamChannelEndianness_Little	Stream channel data is little Endian.
NUM_DEVICESTREAMCHANNELENDIANNESS	

13.6.2.74 DeviceStreamChannelTypeEnums

```
enum DeviceStreamChannelTypeEnums
```

< Reports the type of the stream channel.

Enumerator

DeviceStreamChannelType_Transmitter	Data stream transmitter channel.
DeviceStreamChannelType_Receiver	Data stream receiver channel.
NUM_DEVICESTREAMCHANNELTYPE	

13.6.2.75 DeviceTapGeometryEnums

enum [DeviceTapGeometryEnums](#)

< This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

Enumerator

DeviceTapGeometry_Geometry_1X_1Y	Geometry_1X_1Y
DeviceTapGeometry_Geometry_1X2_1Y	Geometry_1X2_1Y
DeviceTapGeometry_Geometry_1X2_1Y2	Geometry_1X2_1Y2
DeviceTapGeometry_Geometry_2X_1Y	Geometry_2X_1Y
DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y	Geometry_2X_1Y2Geometry_2XE_1Y
DeviceTapGeometry_Geometry_2XE_1Y2	Geometry_2XE_1Y2
DeviceTapGeometry_Geometry_2XM_1Y	Geometry_2XM_1Y
DeviceTapGeometry_Geometry_2XM_1Y2	Geometry_2XM_1Y2
DeviceTapGeometry_Geometry_1X_1Y2	Geometry_1X_1Y2
DeviceTapGeometry_Geometry_1X_2YE	Geometry_1X_2YE
DeviceTapGeometry_Geometry_1X3_1Y	Geometry_1X3_1Y
DeviceTapGeometry_Geometry_3X_1Y	Geometry_3X_1Y
DeviceTapGeometry_Geometry_1X	Geometry_1X
DeviceTapGeometry_Geometry_1X2	Geometry_1X2
DeviceTapGeometry_Geometry_2X	Geometry_2X
DeviceTapGeometry_Geometry_2XE	Geometry_2XE
DeviceTapGeometry_Geometry_2XM	Geometry_2XM
DeviceTapGeometry_Geometry_1X3	Geometry_1X3
DeviceTapGeometry_Geometry_3X	Geometry_3X
DeviceTapGeometry_Geometry_1X4_1Y	Geometry_1X4_1Y
DeviceTapGeometry_Geometry_4X_1Y	Geometry_4X_1Y
DeviceTapGeometry_Geometry_2X2_1Y	Geometry_2X2_1Y
DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y	Geometry_2X2E_1YGeometry_2X2M_1Y
DeviceTapGeometry_Geometry_1X2_2YE	Geometry_1X2_2YE
DeviceTapGeometry_Geometry_2X_2YE	Geometry_2X_2YE
DeviceTapGeometry_Geometry_2XE_2YE	Geometry_2XE_2YE
DeviceTapGeometry_Geometry_2XM_2YE	Geometry_2XM_2YE
DeviceTapGeometry_Geometry_1X4	Geometry_1X4
DeviceTapGeometry_Geometry_4X	Geometry_4X
DeviceTapGeometry_Geometry_2X2	Geometry_2X2
DeviceTapGeometry_Geometry_2X2E	Geometry_2X2E

Enumerator

DeviceTapGeometry_Geometry_2X2M	Geometry_2X2M
DeviceTapGeometry_Geometry_1X8_1Y	Geometry_1X8_1Y
DeviceTapGeometry_Geometry_8X_1Y	Geometry_8X_1Y
DeviceTapGeometry_Geometry_4X2_1Y	Geometry_4X2_1Y
DeviceTapGeometry_Geometry_2X2E_2YE	Geometry_2X2E_2YE
DeviceTapGeometry_Geometry_1X8	Geometry_1X8
DeviceTapGeometry_Geometry_8X	Geometry_8X
DeviceTapGeometry_Geometry_4X2	Geometry_4X2
DeviceTapGeometry_Geometry_4X2E	Geometry_4X2E
DeviceTapGeometry_Geometry_4X2E_1Y	Geometry_4X2E_1Y
DeviceTapGeometry_Geometry_1X10_1Y	Geometry_1X10_1Y
DeviceTapGeometry_Geometry_10X_1Y	Geometry_10X_1Y
DeviceTapGeometry_Geometry_1X10	Geometry_1X10
DeviceTapGeometry_Geometry_10X	Geometry_10X
NUM_DEVICETAPGEOMETRY	

13.6.2.76 DeviceTemperatureSelectorEnums

enum [DeviceTemperatureSelectorEnums](#)

< Selects the location within the device, where the temperature will be measured.

Enumerator

DeviceTemperatureSelector_Sensor	
NUM_DEVICETEMPERATURESELECTOR	

13.6.2.77 DeviceTLTypeEnums

enum [DeviceTLTypeEnums](#)

< Transport Layer type of the device.

Enumerator

DeviceTLType_GigEVision	
DeviceTLType_CameraLink	
DeviceTLType_CameraLinkHS	
DeviceTLType_CoaXPress	
DeviceTLType_USB3Vision	
DeviceTLType_Custom	
NUM_DEVICETLTYPE	

13.6.2.78 DeviceTypeEnums

enum [DeviceTypeEnums](#)

< Returns the device type.

Enumerator

DeviceType_Transmitter	Data stream transmitter device.
DeviceType_Receiver	Data stream receiver device.
DeviceType_Transceiver	Data stream receiver and transmitter device.
DeviceType_Peripheral	Controllable device (with no data stream handling).
NUM_DEVICETYPE	

13.6.2.79 EncoderModeEnums

enum [EncoderModeEnums](#)

< Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.

Enumerator

EncoderMode_FourPhase	The counter increments or decrements 1 for every full quadrature cycle with jitter filtering.
EncoderMode_HighResolution	The counter increments or decrements every quadrature phase for high resolution counting, but without jitter filtering.
NUM_ENCODERMODE	

13.6.2.80 EncoderOutputModeEnums

enum [EncoderOutputModeEnums](#)

< Selects the conditions for the Encoder interface to generate a valid Encoder output signal.

Enumerator

EncoderOutputMode_Off	No output pulse are generated.
EncoderOutputMode_PositionUp	Output pulses are generated at all new positions in the positive direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started.

Enumerator

EncoderOutputMode_PositionDown	Output pulses are generated at all new positions in the negative direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started.
EncoderOutputMode_DirectionUp	Output pulses are generated at all position increments in the positive direction while ignoring negative direction motion.
EncoderOutputMode_DirectionDown	Output pulses are generated at all position increments in the negative direction while ignoring positive direction motion.
EncoderOutputMode_Motion	Output pulses are generated at all motion increments in both directions.
NUM_ENCODEROUTPUTMODE	

13.6.2.81 EncoderResetActivationEnums

enum [EncoderResetActivationEnums](#)

< Selects the Activation mode of the Encoder Reset Source signal.

Enumerator

EncoderResetActivation_RisingEdge	Resets the Encoder on the Rising Edge of the signal.
EncoderResetActivation_FallingEdge	Resets the Encoder on the Falling Edge of the signal.
EncoderResetActivation_AnyEdge	Resets the Encoder on the Falling or rising Edge of the selected signal.
EncoderResetActivation_LevelHigh	Resets the Encoder as long as the selected signal level is High.
EncoderResetActivation_LevelLow	Resets the Encoder as long as the selected signal level is Low.
NUM_ENCODERRESETACTIVATION	

13.6.2.82 EncoderResetSourceEnums

enum [EncoderResetSourceEnums](#)

< Selects the signals that will be the source to reset the Encoder.

Enumerator

EncoderResetSource_Off	Disable the Encoder Reset trigger.
EncoderResetSource_AcquisitionTrigger	Resets with the reception of the Acquisition Trigger.
EncoderResetSource_AcquisitionStart	Resets with the reception of the Acquisition Start.
EncoderResetSource_AcquisitionEnd	Resets with the reception of the Acquisition End.
EncoderResetSource_FrameTrigger	Resets with the reception of the Frame Start Trigger.
EncoderResetSource_FrameStart	Resets with the reception of the Frame Start.
EncoderResetSource_FrameEnd	Resets with the reception of the Frame End.
EncoderResetSource_ExposureStart	Resets with the reception of the Exposure Start.

Enumerator

EncoderResetSource_ExposureEnd	Resets with the reception of the Exposure End.
EncoderResetSource_Line0	Resets by the chosen I/O Line.
EncoderResetSource_Line1	Resets by the chosen I/O Line.
EncoderResetSource_Line2	Resets by the chosen I/O Line.
EncoderResetSource_Counter0Start	Resets with the reception of the Counter Start.
EncoderResetSource_Counter1Start	Resets with the reception of the Counter Start.
EncoderResetSource_Counter2Start	Resets with the reception of the Counter Start.
EncoderResetSource_Counter0End	Resets with the reception of the Counter End.
EncoderResetSource_Counter1End	Resets with the reception of the Counter End.
EncoderResetSource_Counter2End	Resets with the reception of the Counter End.
EncoderResetSource_Timer0Start	Resets with the reception of the Timer Start.
EncoderResetSource_Timer1Start	Resets with the reception of the Timer Start.
EncoderResetSource_Timer2Start	Resets with the reception of the Timer Start.
EncoderResetSource_Timer0End	Resets with the reception of the Timer End.
EncoderResetSource_Timer1End	Resets with the reception of the Timer End.
EncoderResetSource_Timer2End	Resets with the reception of the Timer End.
EncoderResetSource_UserOutput0	Resets by the chosen User Output bit.
EncoderResetSource_UserOutput1	Resets by the chosen User Output bit.
EncoderResetSource_UserOutput2	Resets by the chosen User Output bit.
EncoderResetSource_SoftwareSignal0	Resets on the reception of the Software Signal.
EncoderResetSource_SoftwareSignal1	Resets on the reception of the Software Signal.
EncoderResetSource_SoftwareSignal2	Resets on the reception of the Software Signal.
EncoderResetSource_Action0	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
EncoderResetSource_Action1	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
EncoderResetSource_Action2	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
EncoderResetSource_LinkTrigger0	Resets on the reception of the chosen Link Trigger (received from the transport layer).
EncoderResetSource_LinkTrigger1	Resets on the reception of the chosen Link Trigger (received from the transport layer).
EncoderResetSource_LinkTrigger2	Resets on the reception of the chosen Link Trigger (received from the transport layer).
NUM_ENCODERRESETSOURCE	

13.6.2.83 EncoderSelectorEnums

```
enum EncoderSelectorEnums
```

< Selects which Encoder to configure.

Enumerator

EncoderSelector_Encoder0	Selects Encoder 0.
EncoderSelector_Encoder1	Selects Encoder 1.
EncoderSelector_Encoder2	Selects Encoder 2.
NUM_ENCODERSELECTOR	

13.6.2.84 EncoderSourceAEnums

enum [EncoderSourceAEnums](#)

< Selects the signal which will be the source of the A input of the Encoder.

Enumerator

EncoderSourceA_Off	Counter is stopped.
EncoderSourceA_Line0	Encoder Forward input is taken from the chosen I/O Line.
EncoderSourceA_Line1	Encoder Forward input is taken from the chosen I/O Line.
EncoderSourceA_Line2	Encoder Forward input is taken from the chosen I/O Line.
NUM_ENCODERSOURCEA	

13.6.2.85 EncoderSourceBEnums

enum [EncoderSourceBEnums](#)

< Selects the signal which will be the source of the B input of the Encoder.

Enumerator

EncoderSourceB_Off	Counter is stopped.
EncoderSourceB_Line0	Encoder Reverse input is taken from the chosen I/O Line..
EncoderSourceB_Line1	Encoder Reverse input is taken from the chosen I/O Line..
EncoderSourceB_Line2	Encoder Reverse input is taken from the chosen I/O Line..
NUM_ENCODERSOURCEB	

13.6.2.86 EncoderStatusEnums

enum [EncoderStatusEnums](#)

< Returns the motion status of the encoder.

Enumerator

EncoderStatus_EncoderUp	The encoder counter last incremented.
EncoderStatus_EncoderDown	The encoder counter last decremented.
EncoderStatus_EncoderIdle	The encoder is not active.
EncoderStatus_EncoderStatic	No motion within the EncoderTimeout time.
NUM_ENCODERSTATUS	

13.6.2.87 EventNotificationEnums

enum `EventNotificationEnums`

< Enables/Disables the selected event.

Enumerator

EventNotification_On	
EventNotification_Off	
NUM_EVENTNOTIFICATION	

13.6.2.88 EventSelectorEnums

enum `EventSelectorEnums`

< Selects which Event to enable or disable.

Enumerator

EventSelector_Error	
EventSelector_ExposureEnd	
EventSelector_SerialPortReceive	
NUM_EVENTSELECTOR	

13.6.2.89 ExposureActiveModeEnums

enum `ExposureActiveModeEnums`

< Control sensor active exposure mode.

Enumerator

ExposureActiveMode_Line1	
ExposureActiveMode_AnyPixels	
ExposureActiveMode_AllPixels	
NUM_EXPOSUREACTIVEMODE	

13.6.2.90 ExposureAutoEnums

enum [ExposureAutoEnums](#)

< Sets the automatic exposure mode

Enumerator

ExposureAuto_Off	Exposure time is manually controlled using ExposureTime
ExposureAuto_Once	Exposure time is adapted once by the device. Once it has converged, it returns to the Off state.
ExposureAuto_Continuous	Exposure time is constantly adapted by the device to maximize the dynamic range.
NUM_EXPOSUREAUTO	

13.6.2.91 ExposureModeEnums

enum [ExposureModeEnums](#)

< Sets the operation mode of the Exposure.

Enumerator

ExposureMode_Timed	Timed exposure. The exposure time is set using the ExposureTime or ExposureAuto features and the exposure starts with the FrameStart or LineStart.
ExposureMode_TriggerWidth	Uses the width of the current Frame trigger signal pulse to control the exposure time.
NUM_EXPOSUREMODE	

13.6.2.92 ExposureTimeModeEnums

enum [ExposureTimeModeEnums](#)

< Sets the configuration mode of the ExposureTime feature.

Enumerator

ExposureTimeMode_Common	The exposure time is common to all the color components. The common ExposureTime value to use can be set selecting it with ExposureTimeSelector[Common].
ExposureTimeMode_Individual	The exposure time is individual for each color component. Each individual ExposureTime values to use can be set by selecting them with ExposureTimeSelector.
NUM_EXPOSURETIMEMODE	

13.6.2.93 ExposureTimeSelectorEnums

enum `ExposureTimeSelectorEnums`

< Selects which exposure time is controlled by the ExposureTime feature. This allows for independent control over the exposure components.

Enumerator

ExposureTimeSelector_Common	Selects the common ExposureTime.
ExposureTimeSelector_Red	Selects the red common ExposureTime.
ExposureTimeSelector_Green	Selects the green ExposureTime.
ExposureTimeSelector_Blue	Selects the blue ExposureTime.
ExposureTimeSelector_Cyan	Selects the cyan common ExposureTime.
ExposureTimeSelector_Magenta	Selects the magenta ExposureTime.
ExposureTimeSelector_Yellow	Selects the yellow ExposureTime.
ExposureTimeSelector_Infrared	Selects the infrared ExposureTime.
ExposureTimeSelector_Ultraviolet	Selects the ultraviolet ExposureTime.
ExposureTimeSelector_Stage1	Selects the first stage ExposureTime.
ExposureTimeSelector_Stage2	Selects the second stage ExposureTime.
NUM_EXPOSURETIMESELECTOR	

13.6.2.94 FileOpenModeEnums

enum `FileOpenModeEnums`

< The mode of the file when it is opened. The file can be opened for reading, writing or both. This must be set before opening the file.

Enumerator

FileOpenMode_Read	
FileOpenMode_Write	
FileOpenMode_ReadWrite	
NUM_FILEOPENMODE	

13.6.2.95 FileOperationSelectorEnums

enum `FileOperationSelectorEnums`

< Sets operation to execute on the selected file when the execute command is given.

Enumerator

FileOperationSelector_Open	
FileOperationSelector_Close	
FileOperationSelector_Read	
FileOperationSelector_Write	
FileOperationSelector_Delete	
NUM_FILEOPERATIONSELECTOR	

13.6.2.96 FileOperationStatusEnums

enum `FileOperationStatusEnums`

< Represents the file operation execution status.

Enumerator

FileOperationStatus_Success	File Operation was successful.
FileOperationStatus_Failure	File Operation failed.
FileOperationStatus_Overflow	An overflow occurred while executing the File Operation.
NUM_FILEOPERATIONSTATUS	

13.6.2.97 FileSelectorEnums

enum `FileSelectorEnums`

< Selects which file is being operated on. This must be set before performing any file operations.

Enumerator

FileSelector_UserSetDefault	
FileSelector_UserSet0	
FileSelector_UserSet1	
FileSelector_UserFile1	
FileSelector_SerialPort0	
NUM_FILESELECTOR	

13.6.2.98 GainAutoBalanceEnums

enum `GainAutoBalanceEnums`

< Sets the mode for automatic gain balancing between the sensor color channels or taps. The gain coefficients of each channel or tap are adjusted so they are matched.

Enumerator

GainAutoBalance_Off	Gain tap balancing is user controlled using Gain .
GainAutoBalance_Once	Gain tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.
GainAutoBalance_Continuous	Gain tap balancing is constantly adjusted by the device.
NUM_GAINAUTOBALANCE	

13.6.2.99 GainAutoEnums

enum [GainAutoEnums](#)

< Sets the automatic gain mode. Set to Off for manual control. Set to Once for a single automatic adjustment then return to Off. Set to Continuous for constant adjustment. In automatic modes, the camera adjusts the gain to maximize the dynamic range.

Enumerator

GainAuto_Off	Gain is manually controlled
GainAuto_Once	Gain is adapted once by the device. Once it has converged, it returns to the Off state.
GainAuto_Continuous	Gain is constantly adapted by the device to maximize the dynamic range.
NUM_GAINAUTO	

13.6.2.100 GainSelectorEnums

enum [GainSelectorEnums](#)

< Selects which gain to control. The All selection is a total amplification across all channels (or taps).

Enumerator

GainSelector_All	
NUM_GAINSELECTOR	

13.6.2.101 GevCCPEnums

enum [GevCCPEnums](#)

< Controls the device access privilege of an application.

Enumerator

GevCCP_OpenAccess	
GevCCP_ExclusiveAccess	
GevCCP_ControlAccess	
NUM_GEVCCP	

13.6.2.102 GevCurrentPhysicalLinkConfigurationEnums

```
enum GevCurrentPhysicalLinkConfigurationEnums
```

< Indicates the current physical link configuration of the device.

Enumerator

GevCurrentPhysicalLinkConfiguration_SingleLink	Single Link
GevCurrentPhysicalLinkConfiguration_MultiLink	Multi Link
GevCurrentPhysicalLinkConfiguration_StaticLAG	Static LAG
GevCurrentPhysicalLinkConfiguration_DynamicLAG	Dynamic LAG
NUM_GEVCURRENTPHYSICALLINKCONFIGURATION	

13.6.2.103 GevGVCPExtendedStatusCodesSelectorEnums

```
enum GevGVCPExtendedStatusCodesSelectorEnums
```

< Selects the GigE Vision version to control extended status codes for.

Enumerator

GevGVCPExtendedStatusCodesSelector_Version1_1	Version 1 1
GevGVCPExtendedStatusCodesSelector_Version2_0	Version 2 0
NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR	

13.6.2.104 GevGVSPExtendedIDModeEnums

```
enum GevGVSPExtendedIDModeEnums
```

< Enables the extended IDs mode.

Enumerator

GevGVSPExtendedIDMode_Off	Off
GevGVSPExtendedIDMode_On	On
NUM_GEVGVSPEXTENDEDIDMODE	

13.6.2.105 GevIEEE1588ClockAccuracyEnums

enum [GevIEEE1588ClockAccuracyEnums](#)

< Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

Enumerator

GevIEEE1588ClockAccuracy_Unknown	Unknown Accuracy
NUM_GEVIEEE1588CLOCKACCURACY	

13.6.2.106 GevIEEE1588ModeEnums

enum [GevIEEE1588ModeEnums](#)

< Provides the mode of the IEEE 1588 clock.

Enumerator

GevIEEE1588Mode_Auto	Automatic
GevIEEE1588Mode_SlaveOnly	Slave Only
NUM_GEVIEEE1588MODE	

13.6.2.107 GevIEEE1588StatusEnums

enum [GevIEEE1588StatusEnums](#)

< Provides the status of the IEEE 1588 clock.

Enumerator

GevIEEE1588Status_Initializing	Initializing
GevIEEE1588Status_Faulty	Faulty
GevIEEE1588Status_Disabled	Disabled

Enumerator

GevIEEE1588Status_Listening	Listening
GevIEEE1588Status_PreMaster	Pre Master
GevIEEE1588Status_Master	Master
GevIEEE1588Status_Passive	Passive
GevIEEE1588Status_Uncalibrated	Uncalibrated
GevIEEE1588Status_Slave	Slave
NUM_GEVIEEE1588STATUS	

13.6.2.108 GevIPConfigurationStatusEnums

enum [GevIPConfigurationStatusEnums](#)

< Reports the current IP configuration status.

Enumerator

GevIPConfigurationStatus_None	None
GevIPConfigurationStatus_PersistentIP	Persistent IP
GevIPConfigurationStatus_DHCP	DHCP
GevIPConfigurationStatus_LLA	LLA
GevIPConfigurationStatus_ForceIP	Force IP
NUM_GEVIPCONFIGURATIONSTATUS	

13.6.2.109 GevPhysicalLinkConfigurationEnums

enum [GevPhysicalLinkConfigurationEnums](#)

< Controls the principal physical link configuration to use on next restart/power-up of the device.

Enumerator

GevPhysicalLinkConfiguration_SingleLink	Single Link
GevPhysicalLinkConfiguration_MultiLink	Multi Link
GevPhysicalLinkConfiguration_StaticLAG	Static LAG
GevPhysicalLinkConfiguration_DynamicLAG	Dynamic LAG
NUM_GEVPHYSICALLINKCONFIGURATION	

13.6.2.110 `GevSupportedOptionSelectorEnums`

```
enum GevSupportedOptionSelectorEnums
```

< Selects the GEV option to interrogate for existing support.

Enumerator

<code>GevSupportedOptionSelector_UserDefinedName</code>	
<code>GevSupportedOptionSelector_SerialNumber</code>	
<code>GevSupportedOptionSelector_HeartbeatDisable</code>	
<code>GevSupportedOptionSelector_LinkSpeed</code>	
<code>GevSupportedOptionSelector_CCPApplicationSocket</code>	
<code>GevSupportedOptionSelector_ManifestTable</code>	
<code>GevSupportedOptionSelector_TestData</code>	
<code>GevSupportedOptionSelector_DiscoveryAckDelay</code>	
<code>GevSupportedOptionSelector_DiscoveryAckDelayWritable</code>	
<code>GevSupportedOptionSelector_ExtendedStatusCodes</code>	
<code>GevSupportedOptionSelector_Action</code>	
<code>GevSupportedOptionSelector_PendingAck</code>	
<code>GevSupportedOptionSelector_EventData</code>	
<code>GevSupportedOptionSelector_Event</code>	
<code>GevSupportedOptionSelector_PacketResend</code>	
<code>GevSupportedOptionSelector_WriteMem</code>	
<code>GevSupportedOptionSelector_CommandsConcatenation</code>	
<code>GevSupportedOptionSelector_IPConfigurationLLA</code>	
<code>GevSupportedOptionSelector_IPConfigurationDHCP</code>	
<code>GevSupportedOptionSelector_IPConfigurationPersistentIP</code>	
<code>GevSupportedOptionSelector_StreamChannelSourceSocket</code>	
<code>GevSupportedOptionSelector_MessageChannelSourceSocket</code>	
<code>NUM_GEVSUPPORTEDOPTIONSELECTOR</code>	

13.6.2.111 `ImageComponentSelectorEnums`

```
enum ImageComponentSelectorEnums
```

< Selects a component to activate data streaming from.

Enumerator

<code>ImageComponentSelector_Intensity</code>	The acquisition of intensity of the reflected light is controlled.
<code>ImageComponentSelector_Color</code>	The acquisition of color of the reflected light is controlled
<code>ImageComponentSelector_Infrared</code>	The acquisition of non-visible infrared light is controlled.
<code>ImageComponentSelector_Ultraviolet</code>	The acquisition of non-visible ultraviolet light is controlled.
<code>ImageComponentSelector_Range</code>	The acquisition of range (distance) data is controlled. The data produced may be only range (2.5D) or a point cloud 3D coordinates depending on the Scan3dControl.

Enumerator

ImageComponentSelector_Disparity	The acquisition of stereo camera disparity data is controlled. Disparity is a more specific range format approximately inversely proportional to distance. Disparity is typically given in pixel units.
ImageComponentSelector_Confidence	The acquisition of confidence map of the acquired image is controlled. Confidence data may be binary (0 - invalid) or an integer where 0 is invalid and increasing value is increased confidence in the data in the corresponding pixel. If floating point representation is used the confidence image is normalized to the range [0,1], for integer representation the maximum possible integer represents maximum confidence.
ImageComponentSelector_Scatter	The acquisition of data measuring how much light is scattered around the reflected light. In processing this is used as an additional intensity image, often together with the standard intensity.
NUM_IMAGECOMPONENTSELECTOR	

13.6.2.112 ImageCompressionJPEGFormatOptionEnums

enum `ImageCompressionJPEGFormatOptionEnums`

< When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

Enumerator

ImageCompressionJPEGFormatOption_Lossless	Selects lossless JPEG compression based on a predictive coding model.
ImageCompressionJPEGFormatOption_Baseline↔ Standard	Indicates this is a baseline sequential (single-scan) DCT-based JPEG.
ImageCompressionJPEGFormatOption_Baseline↔ Optimized	Provides optimized color and slightly better compression than baseline standard by using custom Huffman tables optimized after statistical analysis of the image content.
ImageCompressionJPEGFormatOption_Progressive	Indicates this is a progressive (multi-scan) DCT-based JPEG.
NUM_IMAGECOMPRESSIONJPEGFORMATOPT↔ ION	

13.6.2.113 ImageCompressionModeEnums

enum `ImageCompressionModeEnums`

<

Enumerator

ImageCompressionMode_Off	
ImageCompressionMode_Lossless	
NUM_IMAGECOMPRESSIONMODE	

13.6.2.114 ImageCompressionRateOptionEnums

enum [ImageCompressionRateOptionEnums](#)

< Two rate controlling options are offered: fixed bit rate or fixed quality. The exact implementation to achieve one or the other is vendor-specific.

Enumerator

ImageCompressionRateOption_FixBitrate	Output stream follows a constant bit rate. Allows easy bandwidth management on the link.
ImageCompressionRateOption_FixQuality	Output stream has a constant image quality. Can be used when image processing algorithms are sensitive to image degradation caused by excessive data compression.
NUM_IMAGECOMPRESSIONRATEOPTION	

13.6.2.115 LineFormatEnums

enum [LineFormatEnums](#)

< Displays the current electrical format of the selected physical input or output Line.

Enumerator

LineFormat_NoConnect	
LineFormat_TriState	
LineFormat_TTL	
LineFormat_LVDS	
LineFormat_RS422	
LineFormat_OptoCoupled	
LineFormat_OpenDrain	
NUM_LINEFORMAT	

13.6.2.116 LineInputFilterSelectorEnums

enum [LineInputFilterSelectorEnums](#)

< Selects the kind of input filter to configure: Deglitch or Debounce.

Enumerator

LineInputFilterSelector_Deglitch	
LineInputFilterSelector_Debounce	
NUM_LINEINPUTFILTERSELECTOR	

13.6.2.117 LineModeEnums

enum [LineModeEnums](#)

< Controls if the physical Line is used to Input or Output a signal.

Enumerator

LineMode_Input	
LineMode_Output	
NUM_LINEMODE	

13.6.2.118 LineSelectorEnums

enum [LineSelectorEnums](#)

< Selects the physical line (or pin) of the external device connector to configure

Enumerator

LineSelector_Line0	
LineSelector_Line1	
LineSelector_Line2	
LineSelector_Line3	
NUM_LINESELECTOR	

13.6.2.119 LineSourceEnums

enum [LineSourceEnums](#)

< Selects which internal acquisition or I/O source signal to output on the selected line. LineMode must be Output.

Enumerator

LineSource_Off	
LineSource_Line0	
LineSource_Line1	
LineSource_Line2	
LineSource_Line3	
LineSource_UserOutput0	
LineSource_UserOutput1	
LineSource_UserOutput2	
LineSource_UserOutput3	
LineSource_Counter0Active	
LineSource_Counter1Active	
LineSource_LogicBlock0	
LineSource_LogicBlock1	
LineSource_ExposureActive	
LineSource_FrameTriggerWait	
LineSource_SerialPort0	
LineSource_PPSSignal	
LineSource_AllPixel	
LineSource_AnyPixel	
NUM_LINESOURCE	

13.6.2.120 LogicBlockLUTInputActivationEnums

```
enum LogicBlockLUTInputActivationEnums
```

< Selects the activation mode of the Logic Input Source signal.

Enumerator

LogicBlockLUTInputActivation_LevelLow	
LogicBlockLUTInputActivation_LevelHigh	
LogicBlockLUTInputActivation_FallingEdge	
LogicBlockLUTInputActivation_RisingEdge	
LogicBlockLUTInputActivation_AnyEdge	
NUM_LOGICBLOCKLUTINPUTACTIVATION	

13.6.2.121 LogicBlockLUTInputSelectorEnums

```
enum LogicBlockLUTInputSelectorEnums
```

< Controls which LogicBlockLUT Input Source & Activation to access.

Enumerator

LogicBlockLUTInputSelector_Input0	
LogicBlockLUTInputSelector_Input1	
LogicBlockLUTInputSelector_Input2	
LogicBlockLUTInputSelector_Input3	
NUM_LOGICBLOCKLUTINPUTSELECTOR	

13.6.2.122 LogicBlockLUTInputSourceEnums

enum [LogicBlockLUTInputSourceEnums](#)

< Selects the source for the input into the Logic LUT.

Enumerator

LogicBlockLUTInputSource_Zero	Zero
LogicBlockLUTInputSource_Line0	Line0
LogicBlockLUTInputSource_Line1	Line1
LogicBlockLUTInputSource_Line2	Line2
LogicBlockLUTInputSource_Line3	Line3
LogicBlockLUTInputSource_UserOutput0	UserOutput0
LogicBlockLUTInputSource_UserOutput1	UserOutput1
LogicBlockLUTInputSource_UserOutput2	UserOutput2
LogicBlockLUTInputSource_UserOutput3	UserOutput3
LogicBlockLUTInputSource_Counter0Start	Counter0Start
LogicBlockLUTInputSource_Counter1Start	Counter1Start
LogicBlockLUTInputSource_Counter0End	Counter0End
LogicBlockLUTInputSource_Counter1End	Counter1End
LogicBlockLUTInputSource_LogicBlock0	LogicBlock0
LogicBlockLUTInputSource_LogicBlock1	LogicBlock1
LogicBlockLUTInputSource_ExposureStart	ExposureStart
LogicBlockLUTInputSource_ExposureEnd	ExposureEnd
LogicBlockLUTInputSource_FrameTriggerWait	FrameTriggerWait
LogicBlockLUTInputSource_AcquisitionActive	AcquisitionActive
NUM_LOGICBLOCKLUTINPUTSOURCE	

13.6.2.123 LogicBlockLUTSelectorEnums

enum [LogicBlockLUTSelectorEnums](#)

< Selects which LogicBlock LUT to configure

Enumerator

LogicBlockLUTSelector_Value	
LogicBlockLUTSelector_Enable	
NUM_LOGICBLOCKLUTSELECTOR	

13.6.2.124 LogicBlockSelectorEnums

enum `LogicBlockSelectorEnums`

< Selects which LogicBlock to configure

Enumerator

LogicBlockSelector_LogicBlock0	
LogicBlockSelector_LogicBlock1	
NUM_LOGICBLOCKSELECTOR	

13.6.2.125 LUTSelectorEnums

enum `LUTSelectorEnums`

The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.

< Selects which LUT to control.

Enumerator

LUTSelector_LUT1	This LUT is for re-mapping pixels of all formats (mono, Bayer, red, green and blue).
NUM_LUTSELECTOR	

13.6.2.126 PixelColorFilterEnums

enum `PixelColorFilterEnums`

< Type of color filter that is applied to the image. Only applies to Bayer pixel formats. All others have no color filter.

Enumerator

PixelColorFilter_None	No color filter.
PixelColorFilter_BayerRG	Bayer Red Green filter.

Enumerator

PixelColorFilter_BayerGB	Bayer Green Blue filter.
PixelColorFilter_BayerGR	Bayer Green Red filter.
PixelColorFilter_BayerBG	Bayer Blue Green filter.
NUM_PIXELCOLORFILTER	

13.6.2.127 PixelFormatEnums

enum [PixelFormatEnums](#)

< Format of the pixel provided by the camera.

Enumerator

PixelFormat_Mono8	
PixelFormat_Mono16	
PixelFormat_RGB8Packed	
PixelFormat_BayerGR8	
PixelFormat_BayerRG8	
PixelFormat_BayerGB8	
PixelFormat_BayerBG8	
PixelFormat_BayerGR16	
PixelFormat_BayerRG16	
PixelFormat_BayerGB16	
PixelFormat_BayerBG16	
PixelFormat_Mono12Packed	
PixelFormat_BayerGR12Packed	
PixelFormat_BayerRG12Packed	
PixelFormat_BayerGB12Packed	
PixelFormat_BayerBG12Packed	
PixelFormat_YUV411Packed	
PixelFormat_YUV422Packed	
PixelFormat_YUV444Packed	
PixelFormat_Mono12p	
PixelFormat_BayerGR12p	
PixelFormat_BayerRG12p	
PixelFormat_BayerGB12p	
PixelFormat_BayerBG12p	
PixelFormat_YCbCr8	
PixelFormat_YCbCr422_8	
PixelFormat_YCbCr411_8	
PixelFormat_BGR8	
PixelFormat_BGRa8	
PixelFormat_Mono10Packed	
PixelFormat_BayerGR10Packed	
PixelFormat_BayerRG10Packed	

Enumerator

PixelFormat_BayerGB10Packed	
PixelFormat_BayerBG10Packed	
PixelFormat_Mono10p	
PixelFormat_BayerGR10p	
PixelFormat_BayerRG10p	
PixelFormat_BayerGB10p	
PixelFormat_BayerBG10p	
PixelFormat_Mono1p	Monochrome 1-bit packed
PixelFormat_Mono2p	Monochrome 2-bit packed
PixelFormat_Mono4p	Monochrome 4-bit packed
PixelFormat_Mono8s	Monochrome 8-bit signed
PixelFormat_Mono10	Monochrome 10-bit unpacked
PixelFormat_Mono12	Monochrome 12-bit unpacked
PixelFormat_Mono14	Monochrome 14-bit unpacked
PixelFormat_Mono16s	Monochrome 16-bit signed
PixelFormat_Mono32f	Monochrome 32-bit float
PixelFormat_BayerBG10	Bayer Blue-Green 10-bit unpacked
PixelFormat_BayerBG12	Bayer Blue-Green 12-bit unpacked
PixelFormat_BayerGB10	Bayer Green-Blue 10-bit unpacked
PixelFormat_BayerGB12	Bayer Green-Blue 12-bit unpacked
PixelFormat_BayerGR10	Bayer Green-Red 10-bit unpacked
PixelFormat_BayerGR12	Bayer Green-Red 12-bit unpacked
PixelFormat_BayerRG10	Bayer Red-Green 10-bit unpacked
PixelFormat_BayerRG12	Bayer Red-Green 12-bit unpacked
PixelFormat_RGBa8	Red-Green-Blue-alpha 8-bit
PixelFormat_RGBa10	Red-Green-Blue-alpha 10-bit unpacked
PixelFormat_RGBa10p	Red-Green-Blue-alpha 10-bit packed
PixelFormat_RGBa12	Red-Green-Blue-alpha 12-bit unpacked
PixelFormat_RGBa12p	Red-Green-Blue-alpha 12-bit packed
PixelFormat_RGBa14	Red-Green-Blue-alpha 14-bit unpacked
PixelFormat_RGBa16	Red-Green-Blue-alpha 16-bit
PixelFormat_RGB8	Red-Green-Blue 8-bit
PixelFormat_RGB8_Planar	Red-Green-Blue 8-bit planar
PixelFormat_RGB10	Red-Green-Blue 10-bit unpacked
PixelFormat_RGB10_Planar	Red-Green-Blue 10-bit unpacked planar
PixelFormat_RGB10p	Red-Green-Blue 10-bit packed
PixelFormat_RGB10p32	Red-Green-Blue 10-bit packed into 32-bit
PixelFormat_RGB12	Red-Green-Blue 12-bit unpacked
PixelFormat_RGB12_Planar	Red-Green-Blue 12-bit unpacked planar
PixelFormat_RGB12p	Red-Green-Blue 12-bit packed
PixelFormat_RGB14	Red-Green-Blue 14-bit unpacked
PixelFormat_RGB16	Red-Green-Blue 16-bit
PixelFormat_RGB16s	Red-Green-Blue 16-bit signed
PixelFormat_RGB32f	Red-Green-Blue 32-bit float
PixelFormat_RGB16_Planar	Red-Green-Blue 16-bit planar
PixelFormat_RGB565p	Red-Green-Blue 5/6/5-bit packed
PixelFormat_BGRa10	Blue-Green-Red-alpha 10-bit unpacked

Enumerator

PixelFormat_BGRa10p	Blue-Green-Red-alpha 10-bit packed
PixelFormat_BGRa12	Blue-Green-Red-alpha 12-bit unpacked
PixelFormat_BGRa12p	Blue-Green-Red-alpha 12-bit packed
PixelFormat_BGRa14	Blue-Green-Red-alpha 14-bit unpacked
PixelFormat_BGRa16	Blue-Green-Red-alpha 16-bit
PixelFormat_RGBa32f	Red-Green-Blue-alpha 32-bit float
PixelFormat_BGR10	Blue-Green-Red 10-bit unpacked
PixelFormat_BGR10p	Blue-Green-Red 10-bit packed
PixelFormat_BGR12	Blue-Green-Red 12-bit unpacked
PixelFormat_BGR12p	Blue-Green-Red 12-bit packed
PixelFormat_BGR14	Blue-Green-Red 14-bit unpacked
PixelFormat_BGR16	Blue-Green-Red 16-bit
PixelFormat_BGR565p	Blue-Green-Red 5/6/5-bit packed
PixelFormat_R8	Red 8-bit
PixelFormat_R10	Red 10-bit
PixelFormat_R12	Red 12-bit
PixelFormat_R16	Red 16-bit
PixelFormat_G8	Green 8-bit
PixelFormat_G10	Green 10-bit
PixelFormat_G12	Green 12-bit
PixelFormat_G16	Green 16-bit
PixelFormat_B8	Blue 8-bit
PixelFormat_B10	Blue 10-bit
PixelFormat_B12	Blue 12-bit
PixelFormat_B16	Blue 16-bit
PixelFormat_Coord3D_ABC8	3D coordinate A-B-C 8-bit
PixelFormat_Coord3D_ABC8_Planar	3D coordinate A-B-C 8-bit planar
PixelFormat_Coord3D_ABC10p	3D coordinate A-B-C 10-bit packed
PixelFormat_Coord3D_ABC10p_Planar	3D coordinate A-B-C 10-bit packed planar
PixelFormat_Coord3D_ABC12p	3D coordinate A-B-C 12-bit packed
PixelFormat_Coord3D_ABC12p_Planar	3D coordinate A-B-C 12-bit packed planar
PixelFormat_Coord3D_ABC16	3D coordinate A-B-C 16-bit
PixelFormat_Coord3D_ABC16_Planar	3D coordinate A-B-C 16-bit planar
PixelFormat_Coord3D_ABC32f	3D coordinate A-B-C 32-bit floating point
PixelFormat_Coord3D_ABC32f_Planar	3D coordinate A-B-C 32-bit floating point planar
PixelFormat_Coord3D_AC8	3D coordinate A-C 8-bit
PixelFormat_Coord3D_AC8_Planar	3D coordinate A-C 8-bit planar
PixelFormat_Coord3D_AC10p	3D coordinate A-C 10-bit packed
PixelFormat_Coord3D_AC10p_Planar	3D coordinate A-C 10-bit packed planar
PixelFormat_Coord3D_AC12p	3D coordinate A-C 12-bit packed
PixelFormat_Coord3D_AC12p_Planar	3D coordinate A-C 12-bit packed planar
PixelFormat_Coord3D_AC16	3D coordinate A-C 16-bit
PixelFormat_Coord3D_AC16_Planar	3D coordinate A-C 16-bit planar
PixelFormat_Coord3D_AC32f	3D coordinate A-C 32-bit floating point
PixelFormat_Coord3D_AC32f_Planar	3D coordinate A-C 32-bit floating point planar
PixelFormat_Coord3D_A8	3D coordinate A 8-bit
PixelFormat_Coord3D_A10p	3D coordinate A 10-bit packed

Enumerator

PixelFormat_Coord3D_A12p	3D coordinate A 12-bit packed
PixelFormat_Coord3D_A16	3D coordinate A 16-bit
PixelFormat_Coord3D_A32f	3D coordinate A 32-bit floating point
PixelFormat_Coord3D_B8	3D coordinate B 8-bit
PixelFormat_Coord3D_B10p	3D coordinate B 10-bit packed
PixelFormat_Coord3D_B12p	3D coordinate B 12-bit packed
PixelFormat_Coord3D_B16	3D coordinate B 16-bit
PixelFormat_Coord3D_B32f	3D coordinate B 32-bit floating point
PixelFormat_Coord3D_C8	3D coordinate C 8-bit
PixelFormat_Coord3D_C10p	3D coordinate C 10-bit packed
PixelFormat_Coord3D_C12p	3D coordinate C 12-bit packed
PixelFormat_Coord3D_C16	3D coordinate C 16-bit
PixelFormat_Coord3D_C32f	3D coordinate C 32-bit floating point
PixelFormat_Confidence1	Confidence 1-bit unpacked
PixelFormat_Confidence1p	Confidence 1-bit packed
PixelFormat_Confidence8	Confidence 8-bit
PixelFormat_Confidence16	Confidence 16-bit
PixelFormat_Confidence32f	Confidence 32-bit floating point
PixelFormat_BiColorBGRG8	Bi-color Blue/Green - Red/Green 8-bit
PixelFormat_BiColorBGRG10	Bi-color Blue/Green - Red/Green 10-bit unpacked
PixelFormat_BiColorBGRG10p	Bi-color Blue/Green - Red/Green 10-bit packed
PixelFormat_BiColorBGRG12	Bi-color Blue/Green - Red/Green 12-bit unpacked
PixelFormat_BiColorBGRG12p	Bi-color Blue/Green - Red/Green 12-bit packed
PixelFormat_BiColorRGBG8	Bi-color Red/Green - Blue/Green 8-bit
PixelFormat_BiColorRGBG10	Bi-color Red/Green - Blue/Green 10-bit unpacked
PixelFormat_BiColorRGBG10p	Bi-color Red/Green - Blue/Green 10-bit packed
PixelFormat_BiColorRGBG12	Bi-color Red/Green - Blue/Green 12-bit unpacked
PixelFormat_BiColorRGBG12p	Bi-color Red/Green - Blue/Green 12-bit packed
PixelFormat_SCF1WBWG8	Sparse Color Filter #1 White-Blue-White-Green 8-bit
PixelFormat_SCF1WBWG10	Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked
PixelFormat_SCF1WBWG10p	Sparse Color Filter #1 White-Blue-White-Green 10-bit packed
PixelFormat_SCF1WBWG12	Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked
PixelFormat_SCF1WBWG12p	Sparse Color Filter #1 White-Blue-White-Green 12-bit packed
PixelFormat_SCF1WBWG14	Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked
PixelFormat_SCF1WBWG16	Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked
PixelFormat_SCF1WGWB8	Sparse Color Filter #1 White-Green-White-Blue 8-bit
PixelFormat_SCF1WGWB10	Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked
PixelFormat_SCF1WGWB10p	Sparse Color Filter #1 White-Green-White-Blue 10-bit packed
PixelFormat_SCF1WGWB12	Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked
PixelFormat_SCF1WGWB12p	Sparse Color Filter #1 White-Green-White-Blue 12-bit packed
PixelFormat_SCF1WGWB14	Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked
PixelFormat_SCF1WGWB16	Sparse Color Filter #1 White-Green-White-Blue 16-bit
PixelFormat_SCF1WGWR8	Sparse Color Filter #1 White-Green-White-Red 8-bit
PixelFormat_SCF1WGWR10	Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked
PixelFormat_SCF1WGWR10p	Sparse Color Filter #1 White-Green-White-Red 10-bit packed
PixelFormat_SCF1WGWR12	Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked
PixelFormat_SCF1WGWR12p	Sparse Color Filter #1 White-Green-White-Red 12-bit packed

Enumerator

PixelFormat_SCF1WGWR14	Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked
PixelFormat_SCF1WGWR16	Sparse Color Filter #1 White-Green-White-Red 16-bit
PixelFormat_SCF1WRWG8	Sparse Color Filter #1 White-Red-White-Green 8-bit
PixelFormat_SCF1WRWG10	Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked
PixelFormat_SCF1WRWG10p	Sparse Color Filter #1 White-Red-White-Green 10-bit packed
PixelFormat_SCF1WRWG12	Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked
PixelFormat_SCF1WRWG12p	Sparse Color Filter #1 White-Red-White-Green 12-bit packed
PixelFormat_SCF1WRWG14	Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked
PixelFormat_SCF1WRWG16	Sparse Color Filter #1 White-Red-White-Green 16-bit
PixelFormat_YCbCr8_CbYCr	YCbCr 4:4:4 8-bit
PixelFormat_YCbCr10_CbYCr	YCbCr 4:4:4 10-bit unpacked
PixelFormat_YCbCr10p_CbYCr	YCbCr 4:4:4 10-bit packed
PixelFormat_YCbCr12_CbYCr	YCbCr 4:4:4 12-bit unpacked
PixelFormat_YCbCr12p_CbYCr	YCbCr 4:4:4 12-bit packed
PixelFormat_YCbCr411_8_CbYYCrYY	YCbCr 4:1:1 8-bit
PixelFormat_YCbCr422_8_CbYCrY	YCbCr 4:2:2 8-bit
PixelFormat_YCbCr422_10	YCbCr 4:2:2 10-bit unpacked
PixelFormat_YCbCr422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked
PixelFormat_YCbCr422_10p	YCbCr 4:2:2 10-bit packed
PixelFormat_YCbCr422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed
PixelFormat_YCbCr422_12	YCbCr 4:2:2 12-bit unpacked
PixelFormat_YCbCr422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked
PixelFormat_YCbCr422_12p	YCbCr 4:2:2 12-bit packed
PixelFormat_YCbCr422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed
PixelFormat_YCbCr601_8_CbYCr	YCbCr 4:4:4 8-bit BT.601
PixelFormat_YCbCr601_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.601
PixelFormat_YCbCr601_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.601
PixelFormat_YCbCr601_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.601
PixelFormat_YCbCr601_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.601
PixelFormat_YCbCr601_411_8_CbYYCrYY	YCbCr 4:1:1 8-bit BT.601
PixelFormat_YCbCr601_422_8	YCbCr 4:2:2 8-bit BT.601
PixelFormat_YCbCr601_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.601
PixelFormat_YCbCr601_422_10	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormat_YCbCr601_422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormat_YCbCr601_422_10p	YCbCr 4:2:2 10-bit packed BT.601
PixelFormat_YCbCr601_422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed BT.601
PixelFormat_YCbCr601_422_12	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormat_YCbCr601_422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormat_YCbCr601_422_12p	YCbCr 4:2:2 12-bit packed BT.601
PixelFormat_YCbCr601_422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed BT.601
PixelFormat_YCbCr709_8_CbYCr	YCbCr 4:4:4 8-bit BT.709
PixelFormat_YCbCr709_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.709
PixelFormat_YCbCr709_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.709
PixelFormat_YCbCr709_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.709
PixelFormat_YCbCr709_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.709
PixelFormat_YCbCr709_411_8_CbYYCrYY	YCbCr 4:1:1 8-bit BT.709
PixelFormat_YCbCr709_422_8	YCbCr 4:2:2 8-bit BT.709

Enumerator

PixelFormat_YCbCr709_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.709
PixelFormat_YCbCr709_422_10	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormat_YCbCr709_422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormat_YCbCr709_422_10p	YCbCr 4:2:2 10-bit packed BT.709
PixelFormat_YCbCr709_422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed BT.709
PixelFormat_YCbCr709_422_12	YCbCr 4:2:2 12-bit unpacked BT.709
PixelFormat_YCbCr709_422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked BT.709
PixelFormat_YCbCr709_422_12p	YCbCr 4:2:2 12-bit packed BT.709
PixelFormat_YCbCr709_422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed BT.709
PixelFormat_YUV8_UYV	YUV 4:4:4 8-bit
PixelFormat_YUV411_8_UYYVYY	YUV 4:1:1 8-bit
PixelFormat_YUV422_8	YUV 4:2:2 8-bit
PixelFormat_YUV422_8_UYVY	YUV 4:2:2 8-bit
PixelFormat_Polarized8	Monochrome Polarized 8-bit
PixelFormat_Polarized10p	Monochrome Polarized 10-bit packed
PixelFormat_Polarized12p	Monochrome Polarized 12-bit packed
PixelFormat_Polarized16	Monochrome Polarized 16-bit
PixelFormat_BayerRGPolarized8	Polarized Bayer Red Green filter 8-bit
PixelFormat_BayerRGPolarized10p	Polarized Bayer Red Green filter 10-bit packed
PixelFormat_BayerRGPolarized12p	Polarized Bayer Red Green filter 12-bit packed
PixelFormat_BayerRGPolarized16	Polarized Bayer Red Green filter 16-bit
PixelFormat_LLCMono8	Lossless Compression Monochrome 8-bit
PixelFormat_LLCBayerRG8	Lossless Compression Bayer Red Green filter 8-bit
PixelFormat_JPEGMono8	JPEG Monochrome 8-bit
PixelFormat_JPEGColor8	JPEG Color 8-bit
PixelFormat_Raw16	Raw 16 bit.
PixelFormat_Raw8	Raw bit.
PixelFormat_R12_Jpeg	Red 12-bit JPEG.
PixelFormat_GR12_Jpeg	Green Red 12-bit JPEG.
PixelFormat_GB12_Jpeg	Green Blue 12-bit JPEG.
PixelFormat_B12_Jpeg	Blue 12-bit packed JPEG.
UNKNOWN_PIXELFORMAT	
NUM_PIXELFORMAT	

13.6.2.128 PixelFormatInfoSelectorEnums

```
enum PixelFormatInfoSelectorEnums
```

< Select the pixel format for which the information will be returned.

Enumerator

PixelFormatInfoSelector_Mono1p	Monochrome 1-bit packed
PixelFormatInfoSelector_Mono2p	Monochrome 2-bit packed
PixelFormatInfoSelector_Mono4p	Monochrome 4-bit packed

Enumerator

PixelFormatInfoSelector_Mono8	Monochrome 8-bit
PixelFormatInfoSelector_Mono8s	Monochrome 8-bit signed
PixelFormatInfoSelector_Mono10	Monochrome 10-bit unpacked
PixelFormatInfoSelector_Mono10p	Monochrome 10-bit packed
PixelFormatInfoSelector_Mono12	Monochrome 12-bit unpacked
PixelFormatInfoSelector_Mono12p	Monochrome 12-bit packed
PixelFormatInfoSelector_Mono14	Monochrome 14-bit unpacked
PixelFormatInfoSelector_Mono16	Monochrome 16-bit
PixelFormatInfoSelector_Mono16s	Monochrome 16-bit signed
PixelFormatInfoSelector_Mono32f	Monochrome 32-bit float
PixelFormatInfoSelector_BayerBG8	Bayer Blue-Green 8-bit
PixelFormatInfoSelector_BayerBG10	Bayer Blue-Green 10-bit unpacked
PixelFormatInfoSelector_BayerBG10p	Bayer Blue-Green 10-bit packed
PixelFormatInfoSelector_BayerBG12	Bayer Blue-Green 12-bit unpacked
PixelFormatInfoSelector_BayerBG12p	Bayer Blue-Green 12-bit packed
PixelFormatInfoSelector_BayerBG16	Bayer Blue-Green 16-bit
PixelFormatInfoSelector_BayerGB8	Bayer Green-Blue 8-bit
PixelFormatInfoSelector_BayerGB10	Bayer Green-Blue 10-bit unpacked
PixelFormatInfoSelector_BayerGB10p	Bayer Green-Blue 10-bit packed
PixelFormatInfoSelector_BayerGB12	Bayer Green-Blue 12-bit unpacked
PixelFormatInfoSelector_BayerGB12p	Bayer Green-Blue 12-bit packed
PixelFormatInfoSelector_BayerGB16	Bayer Green-Blue 16-bit
PixelFormatInfoSelector_BayerGR8	Bayer Green-Red 8-bit
PixelFormatInfoSelector_BayerGR10	Bayer Green-Red 10-bit unpacked
PixelFormatInfoSelector_BayerGR10p	Bayer Green-Red 10-bit packed
PixelFormatInfoSelector_BayerGR12	Bayer Green-Red 12-bit unpacked
PixelFormatInfoSelector_BayerGR12p	Bayer Green-Red 12-bit packed
PixelFormatInfoSelector_BayerGR16	Bayer Green-Red 16-bit
PixelFormatInfoSelector_BayerRG8	Bayer Red-Green 8-bit
PixelFormatInfoSelector_BayerRG10	Bayer Red-Green 10-bit unpacked
PixelFormatInfoSelector_BayerRG10p	Bayer Red-Green 10-bit packed
PixelFormatInfoSelector_BayerRG12	Bayer Red-Green 12-bit unpacked
PixelFormatInfoSelector_BayerRG12p	Bayer Red-Green 12-bit packed
PixelFormatInfoSelector_BayerRG16	Bayer Red-Green 16-bit
PixelFormatInfoSelector_RGBa8	Red-Green-Blue-alpha 8-bit
PixelFormatInfoSelector_RGBa10	Red-Green-Blue-alpha 10-bit unpacked
PixelFormatInfoSelector_RGBa10p	Red-Green-Blue-alpha 10-bit packed
PixelFormatInfoSelector_RGBa12	Red-Green-Blue-alpha 12-bit unpacked
PixelFormatInfoSelector_RGBa12p	Red-Green-Blue-alpha 12-bit packed
PixelFormatInfoSelector_RGBa14	Red-Green-Blue-alpha 14-bit unpacked
PixelFormatInfoSelector_RGBa16	Red-Green-Blue-alpha 16-bit
PixelFormatInfoSelector_RGB8	Red-Green-Blue 8-bit
PixelFormatInfoSelector_RGB8_Planar	Red-Green-Blue 8-bit planar
PixelFormatInfoSelector_RGB10	Red-Green-Blue 10-bit unpacked
PixelFormatInfoSelector_RGB10_Planar	Red-Green-Blue 10-bit unpacked planar
PixelFormatInfoSelector_RGB10p	Red-Green-Blue 10-bit packed
PixelFormatInfoSelector_RGB10p32	Red-Green-Blue 10-bit packed into 32-bit

Enumerator

PixelFormatInfoSelector_RGB12	Red-Green-Blue 12-bit unpacked
PixelFormatInfoSelector_RGB12_Planar	Red-Green-Blue 12-bit unpacked planar
PixelFormatInfoSelector_RGB12p	Red-Green-Blue 12-bit packed
PixelFormatInfoSelector_RGB14	Red-Green-Blue 14-bit unpacked
PixelFormatInfoSelector_RGB16	Red-Green-Blue 16-bit
PixelFormatInfoSelector_RGB16s	Red-Green-Blue 16-bit signed
PixelFormatInfoSelector_RGB32f	Red-Green-Blue 32-bit float
PixelFormatInfoSelector_RGB16_Planar	Red-Green-Blue 16-bit planar
PixelFormatInfoSelector_RGB565p	Red-Green-Blue 5/6/5-bit packed
PixelFormatInfoSelector_BGRa8	Blue-Green-Red-alpha 8-bit
PixelFormatInfoSelector_BGRa10	Blue-Green-Red-alpha 10-bit unpacked
PixelFormatInfoSelector_BGRa10p	Blue-Green-Red-alpha 10-bit packed
PixelFormatInfoSelector_BGRa12	Blue-Green-Red-alpha 12-bit unpacked
PixelFormatInfoSelector_BGRa12p	Blue-Green-Red-alpha 12-bit packed
PixelFormatInfoSelector_BGRa14	Blue-Green-Red-alpha 14-bit unpacked
PixelFormatInfoSelector_BGRa16	Blue-Green-Red-alpha 16-bit
PixelFormatInfoSelector_RGBA32f	Red-Green-Blue-alpha 32-bit float
PixelFormatInfoSelector_BGR8	Blue-Green-Red 8-bit
PixelFormatInfoSelector_BGR10	Blue-Green-Red 10-bit unpacked
PixelFormatInfoSelector_BGR10p	Blue-Green-Red 10-bit packed
PixelFormatInfoSelector_BGR12	Blue-Green-Red 12-bit unpacked
PixelFormatInfoSelector_BGR12p	Blue-Green-Red 12-bit packed
PixelFormatInfoSelector_BGR14	Blue-Green-Red 14-bit unpacked
PixelFormatInfoSelector_BGR16	Blue-Green-Red 16-bit
PixelFormatInfoSelector_BGR565p	Blue-Green-Red 5/6/5-bit packed
PixelFormatInfoSelector_R8	Red 8-bit
PixelFormatInfoSelector_R10	Red 10-bit
PixelFormatInfoSelector_R12	Red 12-bit
PixelFormatInfoSelector_R16	Red 16-bit
PixelFormatInfoSelector_G8	Green 8-bit
PixelFormatInfoSelector_G10	Green 10-bit
PixelFormatInfoSelector_G12	Green 12-bit
PixelFormatInfoSelector_G16	Green 16-bit
PixelFormatInfoSelector_B8	Blue 8-bit
PixelFormatInfoSelector_B10	Blue 10-bit
PixelFormatInfoSelector_B12	Blue 12-bit
PixelFormatInfoSelector_B16	Blue 16-bit
PixelFormatInfoSelector_Coord3D_ABC8	3D coordinate A-B-C 8-bit
PixelFormatInfoSelector_Coord3D_ABC8_Planar	3D coordinate A-B-C 8-bit planar
PixelFormatInfoSelector_Coord3D_ABC10p	3D coordinate A-B-C 10-bit packed
PixelFormatInfoSelector_Coord3D_ABC10p_Planar	3D coordinate A-B-C 10-bit packed planar
PixelFormatInfoSelector_Coord3D_ABC12p	3D coordinate A-B-C 12-bit packed
PixelFormatInfoSelector_Coord3D_ABC12p_Planar	3D coordinate A-B-C 12-bit packed planar
PixelFormatInfoSelector_Coord3D_ABC16	3D coordinate A-B-C 16-bit
PixelFormatInfoSelector_Coord3D_ABC16_Planar	3D coordinate A-B-C 16-bit planar
PixelFormatInfoSelector_Coord3D_ABC32f	3D coordinate A-B-C 32-bit floating point
PixelFormatInfoSelector_Coord3D_ABC32f_Planar	3D coordinate A-B-C 32-bit floating point planar

Enumerator

PixelFormatInfoSelector_Coord3D_AC8	3D coordinate A-C 8-bit
PixelFormatInfoSelector_Coord3D_AC8_Planar	3D coordinate A-C 8-bit planar
PixelFormatInfoSelector_Coord3D_AC10p	3D coordinate A-C 10-bit packed
PixelFormatInfoSelector_Coord3D_AC10p_Planar	3D coordinate A-C 10-bit packed planar
PixelFormatInfoSelector_Coord3D_AC12p	3D coordinate A-C 12-bit packed
PixelFormatInfoSelector_Coord3D_AC12p_Planar	3D coordinate A-C 12-bit packed planar
PixelFormatInfoSelector_Coord3D_AC16	3D coordinate A-C 16-bit
PixelFormatInfoSelector_Coord3D_AC16_Planar	3D coordinate A-C 16-bit planar
PixelFormatInfoSelector_Coord3D_AC32f	3D coordinate A-C 32-bit floating point
PixelFormatInfoSelector_Coord3D_AC32f_Planar	3D coordinate A-C 32-bit floating point planar
PixelFormatInfoSelector_Coord3D_A8	3D coordinate A 8-bit
PixelFormatInfoSelector_Coord3D_A10p	3D coordinate A 10-bit packed
PixelFormatInfoSelector_Coord3D_A12p	3D coordinate A 12-bit packed
PixelFormatInfoSelector_Coord3D_A16	3D coordinate A 16-bit
PixelFormatInfoSelector_Coord3D_A32f	3D coordinate A 32-bit floating point
PixelFormatInfoSelector_Coord3D_B8	3D coordinate B 8-bit
PixelFormatInfoSelector_Coord3D_B10p	3D coordinate B 10-bit packed
PixelFormatInfoSelector_Coord3D_B12p	3D coordinate B 12-bit packed
PixelFormatInfoSelector_Coord3D_B16	3D coordinate B 16-bit
PixelFormatInfoSelector_Coord3D_B32f	3D coordinate B 32-bit floating point
PixelFormatInfoSelector_Coord3D_C8	3D coordinate C 8-bit
PixelFormatInfoSelector_Coord3D_C10p	3D coordinate C 10-bit packed
PixelFormatInfoSelector_Coord3D_C12p	3D coordinate C 12-bit packed
PixelFormatInfoSelector_Coord3D_C16	3D coordinate C 16-bit
PixelFormatInfoSelector_Coord3D_C32f	3D coordinate C 32-bit floating point
PixelFormatInfoSelector_Confidence1	Confidence 1-bit unpacked
PixelFormatInfoSelector_Confidence1p	Confidence 1-bit packed
PixelFormatInfoSelector_Confidence8	Confidence 8-bit
PixelFormatInfoSelector_Confidence16	Confidence 16-bit
PixelFormatInfoSelector_Confidence32f	Confidence 32-bit floating point
PixelFormatInfoSelector_BiColorBGRG8	Bi-color Blue/Green - Red/Green 8-bit
PixelFormatInfoSelector_BiColorBGRG10	Bi-color Blue/Green - Red/Green 10-bit unpacked
PixelFormatInfoSelector_BiColorBGRG10p	Bi-color Blue/Green - Red/Green 10-bit packed
PixelFormatInfoSelector_BiColorBGRG12	Bi-color Blue/Green - Red/Green 12-bit unpacked
PixelFormatInfoSelector_BiColorBGRG12p	Bi-color Blue/Green - Red/Green 12-bit packed
PixelFormatInfoSelector_BiColorRGBG8	Bi-color Red/Green - Blue/Green 8-bit
PixelFormatInfoSelector_BiColorRGBG10	Bi-color Red/Green - Blue/Green 10-bit unpacked
PixelFormatInfoSelector_BiColorRGBG10p	Bi-color Red/Green - Blue/Green 10-bit packed
PixelFormatInfoSelector_BiColorRGBG12	Bi-color Red/Green - Blue/Green 12-bit unpacked
PixelFormatInfoSelector_BiColorRGBG12p	Bi-color Red/Green - Blue/Green 12-bit packed
PixelFormatInfoSelector_SCF1WBWG8	Sparse Color Filter #1 White-Blue-White-Green 8-bit
PixelFormatInfoSelector_SCF1WBWG10	Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked
PixelFormatInfoSelector_SCF1WBWG10p	Sparse Color Filter #1 White-Blue-White-Green 10-bit packed
PixelFormatInfoSelector_SCF1WBWG12	Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked

Enumerator

PixelFormatInfoSelector_SCF1WBWG12p	Sparse Color Filter #1 White-Blue-White-Green 12-bit packed
PixelFormatInfoSelector_SCF1WBWG14	Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked
PixelFormatInfoSelector_SCF1WBWG16	Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked
PixelFormatInfoSelector_SCF1WGWB8	Sparse Color Filter #1 White-Green-White-Blue 8-bit
PixelFormatInfoSelector_SCF1WGWB10	Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked
PixelFormatInfoSelector_SCF1WGWB10p	Sparse Color Filter #1 White-Green-White-Blue 10-bit packed
PixelFormatInfoSelector_SCF1WGWB12	Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked
PixelFormatInfoSelector_SCF1WGWB12p	Sparse Color Filter #1 White-Green-White-Blue 12-bit packed
PixelFormatInfoSelector_SCF1WGWB14	Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked
PixelFormatInfoSelector_SCF1WGWB16	Sparse Color Filter #1 White-Green-White-Blue 16-bit
PixelFormatInfoSelector_SCF1WGWR8	Sparse Color Filter #1 White-Green-White-Red 8-bit
PixelFormatInfoSelector_SCF1WGWR10	Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked
PixelFormatInfoSelector_SCF1WGWR10p	Sparse Color Filter #1 White-Green-White-Red 10-bit packed
PixelFormatInfoSelector_SCF1WGWR12	Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked
PixelFormatInfoSelector_SCF1WGWR12p	Sparse Color Filter #1 White-Green-White-Red 12-bit packed
PixelFormatInfoSelector_SCF1WGWR14	Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked
PixelFormatInfoSelector_SCF1WGWR16	Sparse Color Filter #1 White-Green-White-Red 16-bit
PixelFormatInfoSelector_SCF1WRWG8	Sparse Color Filter #1 White-Red-White-Green 8-bit
PixelFormatInfoSelector_SCF1WRWG10	Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked
PixelFormatInfoSelector_SCF1WRWG10p	Sparse Color Filter #1 White-Red-White-Green 10-bit packed
PixelFormatInfoSelector_SCF1WRWG12	Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked
PixelFormatInfoSelector_SCF1WRWG12p	Sparse Color Filter #1 White-Red-White-Green 12-bit packed
PixelFormatInfoSelector_SCF1WRWG14	Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked
PixelFormatInfoSelector_SCF1WRWG16	Sparse Color Filter #1 White-Red-White-Green 16-bit
PixelFormatInfoSelector_YCbCr8	YCbCr 4:4:4 8-bit
PixelFormatInfoSelector_YCbCr8_CbYCr	YCbCr 4:4:4 8-bit
PixelFormatInfoSelector_YCbCr10_CbYCr	YCbCr 4:4:4 10-bit unpacked
PixelFormatInfoSelector_YCbCr10p_CbYCr	YCbCr 4:4:4 10-bit packed
PixelFormatInfoSelector_YCbCr12_CbYCr	YCbCr 4:4:4 12-bit unpacked
PixelFormatInfoSelector_YCbCr12p_CbYCr	YCbCr 4:4:4 12-bit packed
PixelFormatInfoSelector_YCbCr411_8	YCbCr 4:1:1 8-bit
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY	YCbCr 4:1:1 8-bit

Enumerator

PixelFormatInfoSelector_YCbCr422_8	YCbCr 4:2:2 8-bit
PixelFormatInfoSelector_YCbCr422_8_CbYCrY	YCbCr 4:2:2 8-bit
PixelFormatInfoSelector_YCbCr422_10	YCbCr 4:2:2 10-bit unpacked
PixelFormatInfoSelector_YCbCr422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked
PixelFormatInfoSelector_YCbCr422_10p	YCbCr 4:2:2 10-bit packed
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed
PixelFormatInfoSelector_YCbCr422_12	YCbCr 4:2:2 12-bit unpacked
PixelFormatInfoSelector_YCbCr422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked
PixelFormatInfoSelector_YCbCr422_12p	YCbCr 4:2:2 12-bit packed
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed
PixelFormatInfoSelector_YCbCr601_8_CbYCr	YCbCr 4:4:4 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_411_8_CbYY↔CrYY	YCbCr 4:1:1 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_8	YCbCr 4:2:2 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_10	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_10_CbY↔CrY	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_10p	YCbCr 4:2:2 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_10p_Cb↔YCrY	YCbCr 4:2:2 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_12	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_12_CbY↔CrY	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_12p	YCbCr 4:2:2 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_12p_Cb↔YCrY	YCbCr 4:2:2 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr709_8_CbYCr	YCbCr 4:4:4 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_411_8_CbYY↔CrYY	YCbCr 4:1:1 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_422_8	YCbCr 4:2:2 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_422_10	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_10_CbY↔CrY	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_10p	YCbCr 4:2:2 10-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_422_10p_Cb↔YCrY	YCbCr 4:2:2 10-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_422_12	YCbCr 4:2:2 12-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_12_CbY↔CrY	YCbCr 4:2:2 12-bit unpacked BT.709

Enumerator

PixelFormatInfoSelector_YCbCr709_422_12p	YCbCr 4:2:2 12-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_422_12p_CbCrY	YCbCr 4:2:2 12-bit packed BT.709
PixelFormatInfoSelector_YUV8_UYV	YUV 4:4:4 8-bit
PixelFormatInfoSelector_YUV411_8_UYYVYY	YUV 4:1:1 8-bit
PixelFormatInfoSelector_YUV422_8	YUV 4:2:2 8-bit
PixelFormatInfoSelector_YUV422_8_UYVY	YUV 4:2:2 8-bit
PixelFormatInfoSelector_Polarized8	Monochrome Polarized 8-bit
PixelFormatInfoSelector_Polarized10p	Monochrome Polarized 10-bit packed
PixelFormatInfoSelector_Polarized12p	Monochrome Polarized 12-bit packed
PixelFormatInfoSelector_Polarized16	Monochrome Polarized 16-bit
PixelFormatInfoSelector_BayerRGPolarized8	Polarized Bayer Red Green filter 8-bit
PixelFormatInfoSelector_BayerRGPolarized10p	Polarized Bayer Red Green filter 10-bit packed
PixelFormatInfoSelector_BayerRGPolarized12p	Polarized Bayer Red Green filter 12-bit packed
PixelFormatInfoSelector_BayerRGPolarized16	Polarized Bayer Red Green filter 16-bit
PixelFormatInfoSelector_LLCMono8	Lossless Compression Monochrome 8-bit
PixelFormatInfoSelector_LLCBayerRG8	Lossless Compression Bayer Red Green filter 8-bit
PixelFormatInfoSelector_JPEGMono8	JPEG Monochrome 8-bit
PixelFormatInfoSelector_JPEGColor8	JPEG Color 8-bit
NUM_PIXELFORMATINFOSELECTOR	

13.6.2.129 PixelSizeEnums

```
enum PixelSizeEnums
```

< Total size in bits of a pixel of the image.

Enumerator

PixelSize_Bpp1	1 bit per pixel.
PixelSize_Bpp2	2 bits per pixel.
PixelSize_Bpp4	4 bits per pixel.
PixelSize_Bpp8	8 bits per pixel.
PixelSize_Bpp10	10 bits per pixel.
PixelSize_Bpp12	12 bits per pixel.
PixelSize_Bpp14	14 bits per pixel.
PixelSize_Bpp16	16 bits per pixel.
PixelSize_Bpp20	20 bits per pixel.
PixelSize_Bpp24	24 bits per pixel.
PixelSize_Bpp30	30 bits per pixel.
PixelSize_Bpp32	32 bits per pixel.
PixelSize_Bpp36	36 bits per pixel.
PixelSize_Bpp48	48 bits per pixel.
PixelSize_Bpp64	64 bits per pixel.
PixelSize_Bpp96	96 bits per pixel.
NUM_PIXELSIZE	

13.6.2.130 RegionDestinationEnums

enum [RegionDestinationEnums](#)

< Control the destination of the selected region.

Enumerator

RegionDestination_Stream0	The destination of the region is the data stream 0.
RegionDestination_Stream1	The destination of the region is the data stream 1.
RegionDestination_Stream2	The destination of the region is the data stream 2.
NUM_REGIONDESTINATION	

13.6.2.131 RegionModeEnums

enum [RegionModeEnums](#)

< Controls if the selected Region of interest is active and streaming.

Enumerator

RegionMode_Off	Disable the usage of the Region.
RegionMode_On	Enable the usage of the Region.
NUM_REGIONMODE	

13.6.2.132 RegionSelectorEnums

enum [RegionSelectorEnums](#)

< Selects the Region of interest to control. The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently.

Enumerator

RegionSelector_Region0	Selected feature will control the region 0.
RegionSelector_Region1	Selected feature will control the region 1.
RegionSelector_Region2	Selected feature will control the region 2.
RegionSelector_All	Selected features will control all the regions at the same time.
NUM_REGIONSELECTOR	

13.6.2.133 RgbTransformLightSourceEnums

enum [RgbTransformLightSourceEnums](#)

< Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources. Selecting a value also sets the white balance ratios (BalanceRatioRed and BalanceRatioBlue), but those can be overwritten through manual or auto white balance.

Enumerator

RgbTransformLightSource_General	Uses a matrix calibrated for a wide range of light sources.
RgbTransformLightSource_Tungsten2800K	Uses a matrix optimized for tungsten/incandescent light with color temperature 2800K.
RgbTransformLightSource_WarmFluorescent3000K	Uses a matrix optimized for a typical warm fluorescecent light with color temperature 3000K.
RgbTransformLightSource_CoolFluorescent4000K	Uses a matrix optimized for a typical cool fluorescecent light with color temperature 4000K.
RgbTransformLightSource_Daylight5000K	Uses a matrix optimized for noon Daylight with color temperature 5000K.
RgbTransformLightSource_Cloudy6500K	Uses a matrix optimized for a cloudy sky with color temperature 6500K.
RgbTransformLightSource_Shade8000K	Uses a matrix optimized for shade with color temperature 8000K.
RgbTransformLightSource_Custom	Uses a custom matrix set by the user through the ColorTransformationValueSelector and ColorTransformationValue controls.
NUM_RGBTRANSFORMLIGHTSOURCE	

13.6.2.134 Scan3dCoordinateReferenceSelectorEnums

enum [Scan3dCoordinateReferenceSelectorEnums](#)

< Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

Enumerator

Scan3dCoordinateReferenceSelector_RotationX	Rotation around X axis.
Scan3dCoordinateReferenceSelector_RotationY	Rotation around Y axis.
Scan3dCoordinateReferenceSelector_RotationZ	Rotation around Z axis.
Scan3dCoordinateReferenceSelector_TranslationX	X axis translation.
Scan3dCoordinateReferenceSelector_TranslationY	Y axis translation.
Scan3dCoordinateReferenceSelector_TranslationZ	Z axis translation.
NUM_SCAN3DCOORDINATEREFERENCESELECTOR	

13.6.2.135 Scan3dCoordinateSelectorEnums

```
enum Scan3dCoordinateSelectorEnums
```

< Selects the individual coordinates in the vectors for 3D information/transformation.

Enumerator

Scan3dCoordinateSelector_CoordinateA	The first (X or Theta) coordinate
Scan3dCoordinateSelector_CoordinateB	The second (Y or Phi) coordinate
Scan3dCoordinateSelector_CoordinateC	The third (Z or Rho) coordinate.
NUM_SCAN3DCOORDINATESELECTOR	

13.6.2.136 Scan3dCoordinateSystemEnums

```
enum Scan3dCoordinateSystemEnums
```

< Specifies the Coordinate system to use for the device.

Enumerator

Scan3dCoordinateSystem_Cartesian	Default value. 3-axis orthogonal, right-hand X-Y-Z.
Scan3dCoordinateSystem_Spherical	A Theta-Phi-Rho coordinate system.
Scan3dCoordinateSystem_Cylindrical	A Theta-Y-Rho coordinate system.
NUM_SCAN3DCOORDINATESYSTEM	

13.6.2.137 Scan3dCoordinateSystemReferenceEnums

```
enum Scan3dCoordinateSystemReferenceEnums
```

< Defines coordinate system reference location.

Enumerator

Scan3dCoordinateSystemReference_Anchor	Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.
Scan3dCoordinateSystemReference_Transformed	Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices.
NUM_SCAN3DCOORDINATESYSTEMREFERENCE	

13.6.2.138 Scan3dCoordinateTransformSelectorEnums

enum [Scan3dCoordinateTransformSelectorEnums](#)

< Sets the index to read/write a coordinate transform value.

Enumerator

Scan3dCoordinateTransformSelector_RotationX	Rotation around X axis.
Scan3dCoordinateTransformSelector_RotationY	Rotation around Y axis.
Scan3dCoordinateTransformSelector_RotationZ	Rotation around Z axis.
Scan3dCoordinateTransformSelector_TranslationX	Translation along X axis.
Scan3dCoordinateTransformSelector_TranslationY	Translation along Y axis.
Scan3dCoordinateTransformSelector_TranslationZ	Translation along Z axis.
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR	

13.6.2.139 Scan3dDistanceUnitEnums

enum [Scan3dDistanceUnitEnums](#)

< Specifies the unit used when delivering calibrated distance data.

Enumerator

Scan3dDistanceUnit_Millimeter	Distance values are in millimeter units (default).
Scan3dDistanceUnit_Inch	Distance values are in inch units.
NUM_SCAN3DDISTANCEUNIT	

13.6.2.140 Scan3dOutputModeEnums

enum [Scan3dOutputModeEnums](#)

< Controls the Calibration and data organization of the device, naming the coordinates transmitted.

Enumerator

Scan3dOutputMode_UncalibratedC	Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.
Scan3dOutputMode_CalibratedABC_Grid	3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.
Scan3dOutputMode_CalibratedABC_PointCloud	3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.

Enumerator

Scan3dOutputMode_CalibratedAC	2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.
Scan3dOutputMode_CalibratedAC_Linescan	2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.
Scan3dOutputMode_CalibratedC	Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.
Scan3dOutputMode_CalibratedC_Linescan	Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.
Scan3dOutputMode_RectifiedC	Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats.
Scan3dOutputMode_RectifiedC_Linescan	Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D_C pixels. The B (Y) axis comes from the encoder chunk value.
Scan3dOutputMode_DisparityC	Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.
Scan3dOutputMode_DisparityC_Linescan	Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value.
NUM_SCAN3DOUTPUTMODE	

13.6.2.141 SensorDigitizationTapsEnums

```
enum SensorDigitizationTapsEnums
```

< Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

Enumerator

SensorDigitizationTaps_One	1 tap.
SensorDigitizationTaps_Two	2 taps.
SensorDigitizationTaps_Three	3 taps.
SensorDigitizationTaps_Four	4 taps.
SensorDigitizationTaps_Eight	8 taps.
SensorDigitizationTaps_Ten	10 taps.
NUM_SENSORDIGITIZATIONTAPS	

13.6.2.142 SensorShutterModeEnums

enum [SensorShutterModeEnums](#)

< Sets the shutter mode of the device.

Enumerator

SensorShutterMode_Global	The shutter opens and closes at the same time for all pixels. All the pixels are exposed for the same length of time at the same time.
SensorShutterMode_Rolling	The shutter opens and closes sequentially for groups (typically lines) of pixels. All the pixels are exposed for the same length of time but not at the same time.
SensorShutterMode_GlobalReset	The shutter opens at the same time for all pixels but ends in a sequential manner. The pixels are exposed for different lengths of time.
NUM_SENSORSHUTTERMODE	

13.6.2.143 SensorTapsEnums

enum [SensorTapsEnums](#)

< Number of taps of the camera sensor.

Enumerator

SensorTaps_One	1 tap.
SensorTaps_Two	2 taps.
SensorTaps_Three	3 taps.
SensorTaps_Four	4 taps.
SensorTaps_Eight	8 taps.
SensorTaps_Ten	10 taps.
NUM_SENSORTAPS	

13.6.2.144 SequencerConfigurationModeEnums

enum [SequencerConfigurationModeEnums](#)

< Controls whether or not a sequencer is in configuration mode.

Enumerator

SequencerConfigurationMode_Off	
SequencerConfigurationMode_On	
NUM_SEQUENCERCONFIGURATIONMODE	

13.6.2.145 SequencerConfigurationValidEnums

enum [SequencerConfigurationValidEnums](#)

< Display whether the current sequencer configuration is valid to run.

Enumerator

SequencerConfigurationValid_No	
SequencerConfigurationValid_Yes	
NUM_SEQUENCERCONFIGURATIONVALID	

13.6.2.146 SequencerModeEnums

enum [SequencerModeEnums](#)

< Controls whether or not a sequencer is active.

Enumerator

SequencerMode_Off	
SequencerMode_On	
NUM_SEQUENCERMODE	

13.6.2.147 SequencerSetValidEnums

enum [SequencerSetValidEnums](#)

< Displays whether the currently selected sequencer set's register contents are valid to use.

Enumerator

SequencerSetValid_No	
SequencerSetValid_Yes	
NUM_SEQUENCERSETVALID	

13.6.2.148 SequencerTriggerActivationEnums

enum [SequencerTriggerActivationEnums](#)

< Specifies the activation mode of the sequencer trigger.

Enumerator

SequencerTriggerActivation_RisingEdge	
SequencerTriggerActivation_FallingEdge	
SequencerTriggerActivation_AnyEdge	
SequencerTriggerActivation_LevelHigh	
SequencerTriggerActivation_LevelLow	
NUM_SEQUENCERTRIGGERACTIVATION	

13.6.2.149 SequencerTriggerSourceEnums

enum [SequencerTriggerSourceEnums](#)

< Specifies the internal signal or physical input line to use as the sequencer trigger source.

Enumerator

SequencerTriggerSource_Off	
SequencerTriggerSource_FrameStart	
NUM_SEQUENCERTRIGGERSOURCE	

13.6.2.150 SerialPortBaudRateEnums

enum [SerialPortBaudRateEnums](#)

< This feature controls the baud rate used by the selected serial port.

Enumerator

SerialPortBaudRate_Baud300	
SerialPortBaudRate_Baud600	
SerialPortBaudRate_Baud1200	
SerialPortBaudRate_Baud2400	
SerialPortBaudRate_Baud4800	
SerialPortBaudRate_Baud9600	
SerialPortBaudRate_Baud14400	
SerialPortBaudRate_Baud19200	
SerialPortBaudRate_Baud38400	
SerialPortBaudRate_Baud57600	
SerialPortBaudRate_Baud115200	
SerialPortBaudRate_Baud230400	
SerialPortBaudRate_Baud460800	
SerialPortBaudRate_Baud921600	
NUM_SERIALPORTBAUDRATE	

13.6.2.151 SerialPortParityEnums

enum [SerialPortParityEnums](#)

< This feature controls the parity used by the selected serial port.

Enumerator

SerialPortParity_None	
SerialPortParity_Odd	
SerialPortParity_Even	
SerialPortParity_Mark	
SerialPortParity_Space	
NUM_SERIALPORTPARITY	

13.6.2.152 SerialPortSelectorEnums

enum [SerialPortSelectorEnums](#)

< Selects which serial port of the device to control.

Enumerator

SerialPortSelector_SerialPort0	
NUM_SERIALPORTSELECTOR	

13.6.2.153 SerialPortSourceEnums

enum [SerialPortSourceEnums](#)

< Specifies the physical input Line on which to receive serial data.

Enumerator

SerialPortSource_Line0	
SerialPortSource_Line1	
SerialPortSource_Line2	
SerialPortSource_Line3	
SerialPortSource_Off	
NUM_SERIALPORTSOURCE	

13.6.2.154 SerialPortStopBitsEnums

enum `SerialPortStopBitsEnums`

< This feature controls the number of stop bits used by the selected serial port.

Enumerator

SerialPortStopBits_Bits1	
SerialPortStopBits_Bits1AndAHalf	
SerialPortStopBits_Bits2	
NUM_SERIALPORTSTOPBITS	

13.6.2.155 SoftwareSignalSelectorEnums

enum `SoftwareSignalSelectorEnums`

< Selects which Software Signal features to control.

Enumerator

SoftwareSignalSelector_SoftwareSignal0	Selects the software generated signal to control.
SoftwareSignalSelector_SoftwareSignal1	Selects the software generated signal to control.
SoftwareSignalSelector_SoftwareSignal2	Selects the software generated signal to control.
NUM_SOFTWARESIGNALSELECTOR	

13.6.2.156 SourceSelectorEnums

enum `SourceSelectorEnums`

< Selects the source to control.

Enumerator

SourceSelector_Source0	Selects the data source 0.
SourceSelector_Source1	Selects the data source 1.
SourceSelector_Source2	Selects the data source 2.
SourceSelector_All	Selects all the data sources.
NUM_SOURCESELECTOR	

13.6.2.157 TestPatternEnums

enum `TestPatternEnums`

< Selects the type of test pattern that is generated by the device as image source.

Enumerator

TestPattern_Off	Test pattern is disabled.
TestPattern_Increment	Pixel value increments by 1 for each pixel.
TestPattern_SensorTestPattern	A test pattern generated by the image sensor. The pattern varies for different sensor models.
NUM_TESTPATTERN	

13.6.2.158 TestPatternGeneratorSelectorEnums

enum `TestPatternGeneratorSelectorEnums`

< Selects which test pattern generator is controlled by the TestPattern feature.

Enumerator

TestPatternGeneratorSelector_Sensor	TestPattern feature controls the sensor's test pattern generator.
TestPatternGeneratorSelector_PipelineStart	TestPattern feature controls the test pattern inserted at the start of the image pipeline.
NUM_TESTPATTERNGENERATORSELECTOR	

13.6.2.159 TimerSelectorEnums

enum `TimerSelectorEnums`

< Selects which Timer to configure.

Enumerator

TimerSelector_Timer0	Selects the Timer 0.
TimerSelector_Timer1	Selects the Timer 1.
TimerSelector_Timer2	Selects the Timer 2.
NUM_TIMERSELECTOR	

13.6.2.160 TimerStatusEnums

```
enum TimerStatusEnums
```

< Returns the current status of the Timer.

Enumerator

TimerStatus_TimerIdle	The Timer is idle.
TimerStatus_TimerTriggerWait	The Timer is waiting for a start trigger.
TimerStatus_TimerActive	The Timer is counting for the specified duration.
TimerStatus_TimerCompleted	The Timer reached the TimerDuration count.
NUM_TIMERSTATUS	

13.6.2.161 TimerTriggerActivationEnums

```
enum TimerTriggerActivationEnums
```

< Selects the activation mode of the trigger to start the Timer.

Enumerator

TimerTriggerActivation_RisingEdge	Starts counting on the Rising Edge of the selected trigger signal.
TimerTriggerActivation_FallingEdge	Starts counting on the Falling Edge of the selected trigger signal.
TimerTriggerActivation_AnyEdge	Starts counting on the Falling or Rising Edge of the selected trigger signal.
TimerTriggerActivation_LevelHigh	Counts as long as the selected trigger signal level is High.
TimerTriggerActivation_LevelLow	Counts as long as the selected trigger signal level is Low.
NUM_TIMERTRIGGERACTIVATION	

13.6.2.162 TimerTriggerSourceEnums

```
enum TimerTriggerSourceEnums
```

< Selects the source of the trigger to start the Timer.

Enumerator

TimerTriggerSource_Off	Disables the Timer trigger.
TimerTriggerSource_AcquisitionTrigger	Starts with the reception of the Acquisition Trigger.
TimerTriggerSource_AcquisitionStart	Starts with the reception of the Acquisition Start.
TimerTriggerSource_AcquisitionEnd	Starts with the reception of the Acquisition End.
TimerTriggerSource_FrameTrigger	Starts with the reception of the Frame Start Trigger.
TimerTriggerSource_FrameStart	Starts with the reception of the Frame Start.

Enumerator

TimerTriggerSource_FrameEnd	Starts with the reception of the Frame End.
TimerTriggerSource_FrameBurstStart	Starts with the reception of the Frame Burst Start.
TimerTriggerSource_FrameBurstEnd	Starts with the reception of the Frame Burst End.
TimerTriggerSource_LineTrigger	Starts with the reception of the Line Start Trigger.
TimerTriggerSource_LineStart	Starts with the reception of the Line Start.
TimerTriggerSource_LineEnd	Starts with the reception of the Line End.
TimerTriggerSource_ExposureStart	Starts with the reception of the Exposure Start.
TimerTriggerSource_ExposureEnd	Starts with the reception of the Exposure End.
TimerTriggerSource_Line0	Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_Line1	Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_Line2	Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_UserOutput0	Specifies which User Output bit signal to use as internal source for the trigger.
TimerTriggerSource_UserOutput1	Specifies which User Output bit signal to use as internal source for the trigger.
TimerTriggerSource_UserOutput2	Specifies which User Output bit signal to use as internal source for the trigger.
TimerTriggerSource_Counter0Start	Starts with the reception of the Counter Start.
TimerTriggerSource_Counter1Start	Starts with the reception of the Counter Start.
TimerTriggerSource_Counter2Start	Starts with the reception of the Counter Start.
TimerTriggerSource_Counter0End	Starts with the reception of the Counter End.
TimerTriggerSource_Counter1End	Starts with the reception of the Counter End.
TimerTriggerSource_Counter2End	Starts with the reception of the Counter End.
TimerTriggerSource_Timer0Start	Starts with the reception of the Timer Start.
TimerTriggerSource_Timer1Start	Starts with the reception of the Timer Start.
TimerTriggerSource_Timer2Start	Starts with the reception of the Timer Start.
TimerTriggerSource_Timer0End	Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.
TimerTriggerSource_Timer1End	Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.
TimerTriggerSource_Timer2End	Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.
TimerTriggerSource_Encoder0	Starts with the reception of the Encoder output signal.
TimerTriggerSource_Encoder1	Starts with the reception of the Encoder output signal.
TimerTriggerSource_Encoder2	Starts with the reception of the Encoder output signal.
TimerTriggerSource_SoftwareSignal0	Starts on the reception of the Software Signal.
TimerTriggerSource_SoftwareSignal1	Starts on the reception of the Software Signal.
TimerTriggerSource_SoftwareSignal2	Starts on the reception of the Software Signal.
TimerTriggerSource_Action0	Starts with the assertion of the chosen action signal.
TimerTriggerSource_Action1	Starts with the assertion of the chosen action signal.
TimerTriggerSource_Action2	Starts with the assertion of the chosen action signal.
TimerTriggerSource_LinkTrigger0	Starts with the reception of the chosen Link Trigger.
TimerTriggerSource_LinkTrigger1	Starts with the reception of the chosen Link Trigger.
TimerTriggerSource_LinkTrigger2	Starts with the reception of the chosen Link Trigger.
NUM_TIMERTRIGGERSOURCE	

13.6.2.163 TransferComponentSelectorEnums

```
enum TransferComponentSelectorEnums
```

< Selects the color component for the control of the TransferStreamChannel feature.

Enumerator

TransferComponentSelector_Red	The TransferStreamChannel feature controls the index of the stream channel for the streaming of the red plane of the planar pixel formats.
TransferComponentSelector_Green	The TransferStreamChannel feature controls the index of the stream channel for the streaming of the green plane of the planar pixel formats.
TransferComponentSelector_Blue	The TransferStreamChannel feature controls the index of the stream channel for the streaming of blue plane of the planar pixel formats.
TransferComponentSelector_All	The TransferStreamChannel feature controls the index of the stream channel for the streaming of all the planes of the planar pixel formats simultaneously or non planar pixel formats.
NUM_TRANSFERCOMPONENTSELECTOR	

13.6.2.164 TransferControlModeEnums

```
enum TransferControlModeEnums
```

< Selects the control method for the transfers. Basic and Automatic start transmitting data as soon as there is enough data to fill a link layer packet. User Controlled allows you to directly control the transfer of blocks.

Enumerator

TransferControlMode_Basic	Basic
TransferControlMode_Automatic	Automatic
TransferControlMode_UserControlled	User Controlled
NUM_TRANSFERCONTROLMODE	

13.6.2.165 TransferOperationModeEnums

```
enum TransferOperationModeEnums
```

< Selects the operation mode of the transfer. Continuous is similar to Basic/Automatic but you can start/stop the transfer while acquisition runs independently. Multi Block transmits a specified number of blocks and then stops.

Enumerator

TransferOperationMode_Continuous	Continuous
TransferOperationMode_MultiBlock	Multi Block
NUM_TRANSFEROPERATIONMODE	

13.6.2.166 TransferQueueModeEnums

enum [TransferQueueModeEnums](#)

< Specifies the operation mode of the transfer queue.

Enumerator

TransferQueueMode_FirstInFirstOut	Blocks first In are transferred Out first.
NUM_TRANSFERQUEUEMODE	

13.6.2.167 TransferSelectorEnums

enum [TransferSelectorEnums](#)

< Selects which stream transfers are currently controlled by the selected Transfer features.

Enumerator

TransferSelector_Stream0	The transfer features control the data stream 0.
TransferSelector_Stream1	The transfer features control the data stream 1.
TransferSelector_Stream2	The transfer features control the data stream 2.
TransferSelector_All	The transfer features control all the data streams simulateneously.
NUM_TRANSFERSELECTOR	

13.6.2.168 TransferStatusSelectorEnums

enum [TransferStatusSelectorEnums](#)

< Selects which status of the transfer module to read.

Enumerator

TransferStatusSelector_Streaming	Data blocks are transmitted when enough data is available.
TransferStatusSelector_Paused	Data blocks transmission is suspended immediately.

Enumerator

TransferStatusSelector_Stopping	Data blocks transmission is stopping. The current block transmission will be completed and the transfer state will stop.
TransferStatusSelector_Stopped	Data blocks transmission is stopped.
TransferStatusSelector_QueueOverflow	Data blocks queue is in overflow state.
NUM_TRANSFERSTATUSSELECTION	

13.6.2.169 TransferTriggerActivationEnums

enum [TransferTriggerActivationEnums](#)

< Specifies the activation mode of the transfer control trigger.

Enumerator

TransferTriggerActivation_RisingEdge	Specifies that the trigger is considered valid on the rising edge of the source signal.
TransferTriggerActivation_FallingEdge	Specifies that the trigger is considered valid on the falling edge of the source signal.
TransferTriggerActivation_AnyEdge	Specifies that the trigger is considered valid on the falling or rising edge of the source signal.
TransferTriggerActivation_LevelHigh	Specifies that the trigger is considered valid as long as the level of the source signal is high. This can apply to TransferActive and TransferPause trigger.
TransferTriggerActivation_LevelLow	Specifies that the trigger is considered valid as long as the level of the source signal is low. This can apply to TransferActive and TransferPause trigger.
NUM_TRANSFERTRIGGERACTIVATION	

13.6.2.170 TransferTriggerModeEnums

enum [TransferTriggerModeEnums](#)

< Controls if the selected trigger is active.

Enumerator

TransferTriggerMode_Off	Disables the selected trigger.
TransferTriggerMode_On	Enable the selected trigger.
NUM_TRANSFERTRIGGERMODE	

13.6.2.171 TransferTriggerSelectorEnums

```
enum TransferTriggerSelectorEnums
```

< Selects the type of transfer trigger to configure.

Enumerator

TransferTriggerSelector_TransferStart	Selects a trigger to start the transfers.
TransferTriggerSelector_TransferStop	Selects a trigger to stop the transfers.
TransferTriggerSelector_TransferAbort	Selects a trigger to abort the transfers.
TransferTriggerSelector_TransferPause	Selects a trigger to pause the transfers.
TransferTriggerSelector_TransferResume	Selects a trigger to Resume the transfers.
TransferTriggerSelector_TransferActive	Selects a trigger to Activate the transfers. This trigger type is used when TriggerActivation is set LevelHigh or levelLow.
TransferTriggerSelector_TransferBurstStart	Selects a trigger to start the transfer of a burst of frames specified by TransferBurstCount.
TransferTriggerSelector_TransferBurstStop	Selects a trigger to end the transfer of a burst of frames.
NUM_TRANSFERTRIGGERSELECTOR	

13.6.2.172 TransferTriggerSourceEnums

```
enum TransferTriggerSourceEnums
```

< Specifies the signal to use as the trigger source for transfers.

Enumerator

TransferTriggerSource_Line0	Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.
TransferTriggerSource_Line1	Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.
TransferTriggerSource_Line2	Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.
TransferTriggerSource_Counter0Start	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter1Start	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter2Start	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter0End	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter1End	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter2End	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer0Start	Specifies which Timer signal to use as internal source for the transfer control trigger signal.

Enumerator

TransferTriggerSource_Timer1Start	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer2Start	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer0End	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer1End	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer2End	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_SoftwareSignal0	Specifies which Software Signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_SoftwareSignal1	Specifies which Software Signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_SoftwareSignal2	Specifies which Software Signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Action0	Specifies which Action command to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Action1	Specifies which Action command to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Action2	Specifies which Action command to use as internal source for the transfer control trigger signal.
NUM_TRANSFERTRIGGERSOURCE	

13.6.2.173 TriggerActivationEnums

```
enum TriggerActivationEnums
```

< Specifies the activation mode of the trigger.

Enumerator

TriggerActivation_LevelLow	
TriggerActivation_LevelHigh	
TriggerActivation_FallingEdge	
TriggerActivation_RisingEdge	
TriggerActivation_AnyEdge	
NUM_TRIGGERACTIVATION	

13.6.2.174 TriggerModeEnums

```
enum TriggerModeEnums
```

< Controls whether or not trigger is active.

Enumerator

TriggerMode_Off	
TriggerMode_On	
NUM_TRIGGERMODE	

13.6.2.175 TriggerOverlapEnums

```
enum TriggerOverlapEnums
```

< Specifies the overlap mode of the trigger.

Enumerator

TriggerOverlap_Off	
TriggerOverlap_ReadOut	
TriggerOverlap_PreviousFrame	
NUM_TRIGGEROVERLAP	

13.6.2.176 TriggerSelectorEnums

```
enum TriggerSelectorEnums
```

< Selects the type of trigger to configure.

Enumerator

TriggerSelector_AcquisitionStart	
TriggerSelector_FrameStart	
TriggerSelector_FrameBurstStart	
NUM_TRIGGERSELECTOR	

13.6.2.177 TriggerSourceEnums

```
enum TriggerSourceEnums
```

< Specifies the internal signal or physical input line to use as the trigger source.

Enumerator

TriggerSource_Software	

Enumerator

TriggerSource_Line0	
TriggerSource_Line1	
TriggerSource_Line2	
TriggerSource_Line3	
TriggerSource_UserOutput0	
TriggerSource_UserOutput1	
TriggerSource_UserOutput2	
TriggerSource_UserOutput3	
TriggerSource_Counter0Start	
TriggerSource_Counter1Start	
TriggerSource_Counter0End	
TriggerSource_Counter1End	
TriggerSource_LogicBlock0	
TriggerSource_LogicBlock1	
TriggerSource_Action0	
NUM_TRIGGERSOURCE	

13.6.2.178 UserOutputSelectorEnums

enum `UserOutputSelectorEnums`

< Selects which bit of the User Output register is set by UserOutputValue.

Enumerator

UserOutputSelector_UserOutput0	
UserOutputSelector_UserOutput1	
UserOutputSelector_UserOutput2	
UserOutputSelector_UserOutput3	
NUM_USEROUTPUTSELECTOR	

13.6.2.179 UserSetDefaultEnums

enum `UserSetDefaultEnums`

< Selects the feature User Set to load and make active by default when the device is restarted.

Enumerator

UserSetDefault_Default	Factory default set.
UserSetDefault_UserSet0	User configurable set 0.
UserSetDefault_UserSet1	User configurable set 1.
NUM_USERSETDEFAULT	

13.6.2.180 UserSetSelectorEnums

```
enum UserSetSelectorEnums
```

< Selects the feature User Set to load, save or configure.

Enumerator

UserSetSelector_Default	Factory default set.
UserSetSelector_UserSet0	User configurable set 0.
UserSetSelector_UserSet1	User configurable set 1.
NUM_USERSETSELECTOR	

13.6.2.181 WhiteClipSelectorEnums

```
enum WhiteClipSelectorEnums
```

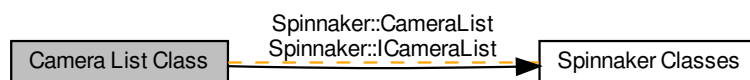
< Selects which White Clip to control.

Enumerator

WhiteClipSelector_All	White Clip will be applied to all channels or taps.
WhiteClipSelector_Red	White Clip will be applied to the red channel.
WhiteClipSelector_Green	White Clip will be applied to the green channel.
WhiteClipSelector_Blue	White Clip will be applied to the blue channel.
WhiteClipSelector_Y	White Clip will be applied to Y channel.
WhiteClipSelector_U	White Clip will be applied to U channel.
WhiteClipSelector_V	White Clip will be applied to V channel.
WhiteClipSelector_Tap1	White Clip will be applied to Tap 1.
WhiteClipSelector_Tap2	White Clip will be applied to Tap 2.
NUM_WHITECLIPSELECTOR	

13.7 Camera List Class

Collaboration diagram for Camera List Class:



Classes

- class [CameraList](#)
Used to hold a list of camera objects.
- class [ICameraList](#)
Used to hold a list of camera objects.

13.7.1 Detailed Description

13.8 CameraPtr Class

Collaboration diagram for CameraPtr Class:



Classes

- class [CameraPtr](#)
A reference tracked pointer to a camera object.

Functions

- [CameraPtr](#) () throw ()
Default constructor.
- [CameraPtr](#) (const int) throw ()
Default constructor.
- [CameraPtr](#) (const long) throw ()
Default constructor with argument.
- [CameraPtr](#) (const std::nullptr_t) throw ()

13.8.1 Detailed Description

13.8.2 Function Documentation

13.8.2.1 CameraPtr() [1/4]

```
CameraPtr ( ) throw )    [inline]
```

Default constructor.

13.8.2.2 CameraPtr() [2/4]

```
CameraPtr (
    const int ) throw )    [inline]
```

Default constructor.

13.8.2.3 CameraPtr() [3/4]

```
CameraPtr (
    const long ) throw )    [inline]
```

Default constructor with argument.

13.8.2.4 CameraPtr() [4/4]

```
CameraPtr (
    const std::nullptr_t ) throw )    [inline]
```

13.9 ChunkData Class

Collaboration diagram for ChunkData Class:



Classes

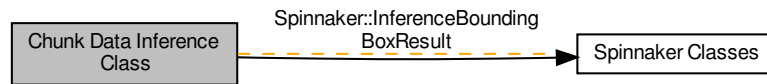
- class [ChunkData](#)

The chunk data which contains additional information about an image.

13.9.1 Detailed Description

13.10 Chunk Data Inference Class

Collaboration diagram for Chunk Data Inference Class:



Classes

- class [InferenceBoundingBoxResult](#)
An inference bounding boxes object which holds information about the detected bounding boxes.

Functions

- [InferenceBoundingBoxResult](#) ()
Default Constructor.
- [~InferenceBoundingBoxResult](#) ()
Destructor.
- [InferenceBoundingBoxResult](#) (const uint8_t *data, const int64_t lengthInBytes)
Default Constructor with arguments.
- [InferenceBoundingBoxResult](#) (const [InferenceBoundingBoxResult](#) &other)
Copy Constructor.
- [InferenceBoundingBoxResult](#) & operator= (const [InferenceBoundingBoxResult](#) &rhs)
Assignment Operator.
- int8_t [GetVersion](#) () const
Returns the bounding box format version number.
- int16_t [GetBoxCount](#) () const
Returns the number of bounding boxes.
- int8_t [GetBoxSize](#) () const
Returns the number of bytes allocated for one bounding box.
- [InferenceBoundingBox](#) [GetBoxAt](#) (const uint16_t index) const
Returns the bounding box at specified index.

Variables

- int16_t [topLeftXCoord](#)
- int16_t [topLeftYCoord](#)
- int16_t [bottomRightXCoord](#)
- int16_t [bottomRightYCoord](#)
- int16_t [centerXCoord](#)
- int16_t [centerYCoord](#)
- int16_t [radius](#)
- int16_t [topLeftXCoord](#)

- `int16_t` `topLeftYCoord`
- `int16_t` `bottomRightXCoord`
- `int16_t` `bottomRightYCoord`
- `short` `rotationAngle`
- `InferenceBoxType` `boxType`
- `int16_t` `classId`
- `float32_t` `confidence`
- `InferenceBoxRect` `rect`
- `InferenceBoxCircle` `circle`
- `InferenceBoxRotatedRect` `rotatedRect`

13.10.1 Detailed Description

13.10.2 Function Documentation

13.10.2.1 `GetBoxAt()`

```
InferenceBoundingBox GetBoxAt (
    const uint16_t index ) const
```

Returns the bounding box at specified index.

Parameters

<i>index</i>	Index of the bounding box to return.

13.10.2.2 `GetBoxCount()`

```
int16_t GetBoxCount ( ) const
```

Returns the number of bounding boxes.

13.10.2.3 `GetBoxSize()`

```
int8_t GetBoxSize ( ) const
```

Returns the number of bytes allocated for one bounding box.

13.10.2.4 GetVersion()

```
int8_t GetVersion ( ) const
```

Returns the bounding box format version number.

13.10.2.5 InferenceBoundingBoxResult() [1/3]

```
InferenceBoundingBoxResult ( )
```

Default Constructor.

13.10.2.6 InferenceBoundingBoxResult() [2/3]

```
InferenceBoundingBoxResult (
    const uint8_t * data,
    const int64_t lengthInBytes )
```

Default Constructor with arguments.

Parameters

<i>data</i>	The bounding box binary data from chunk data.
<i>lengthInBytes</i>	The length of bounding box binary data in bytes.

13.10.2.7 InferenceBoundingBoxResult() [3/3]

```
InferenceBoundingBoxResult (
    const InferenceBoundingBoxResult & other )
```

Copy Constructor.

13.10.2.8 operator=()

```
InferenceBoundingBoxResult& operator= (
    const InferenceBoundingBoxResult & rhs )
```

Assignment Operator.

13.10.2.9 `~InferenceBoundingBoxResult()``~InferenceBoundingBoxResult ()`

Destructor.

13.10.3 Variable Documentation**13.10.3.1** `bottomRightXCoord` [1/2]`int16_t bottomRightXCoord`**13.10.3.2** `bottomRightXCoord` [2/2]`int16_t bottomRightXCoord`**13.10.3.3** `bottomRightYCoord` [1/2]`int16_t bottomRightYCoord`**13.10.3.4** `bottomRightYCoord` [2/2]`int16_t bottomRightYCoord`**13.10.3.5** `boxType``InferenceBoxType boxType`**13.10.3.6** `centerXCoord``int16_t centerXCoord`

13.10.3.7 centerYCoord

```
int16_t centerYCoord
```

13.10.3.8 circle

```
InferenceBoxCircle circle
```

13.10.3.9 classId

```
int16_t classId
```

13.10.3.10 confidence

```
float32_t confidence
```

13.10.3.11 radius

```
int16_t radius
```

13.10.3.12 rect

```
InferenceBoxRect rect
```

13.10.3.13 rotatedRect

```
InferenceBoxRotatedRect rotatedRect
```

13.10.3.14 rotationAngle

```
short rotationAngle
```


13.10.3.15 `topLeftXCoord` [1/2]

```
int16_t topLeftXCoord
```

13.10.3.16 `topLeftXCoord` [2/2]

```
int16_t topLeftXCoord
```

13.10.3.17 `topLeftYCoord` [1/2]

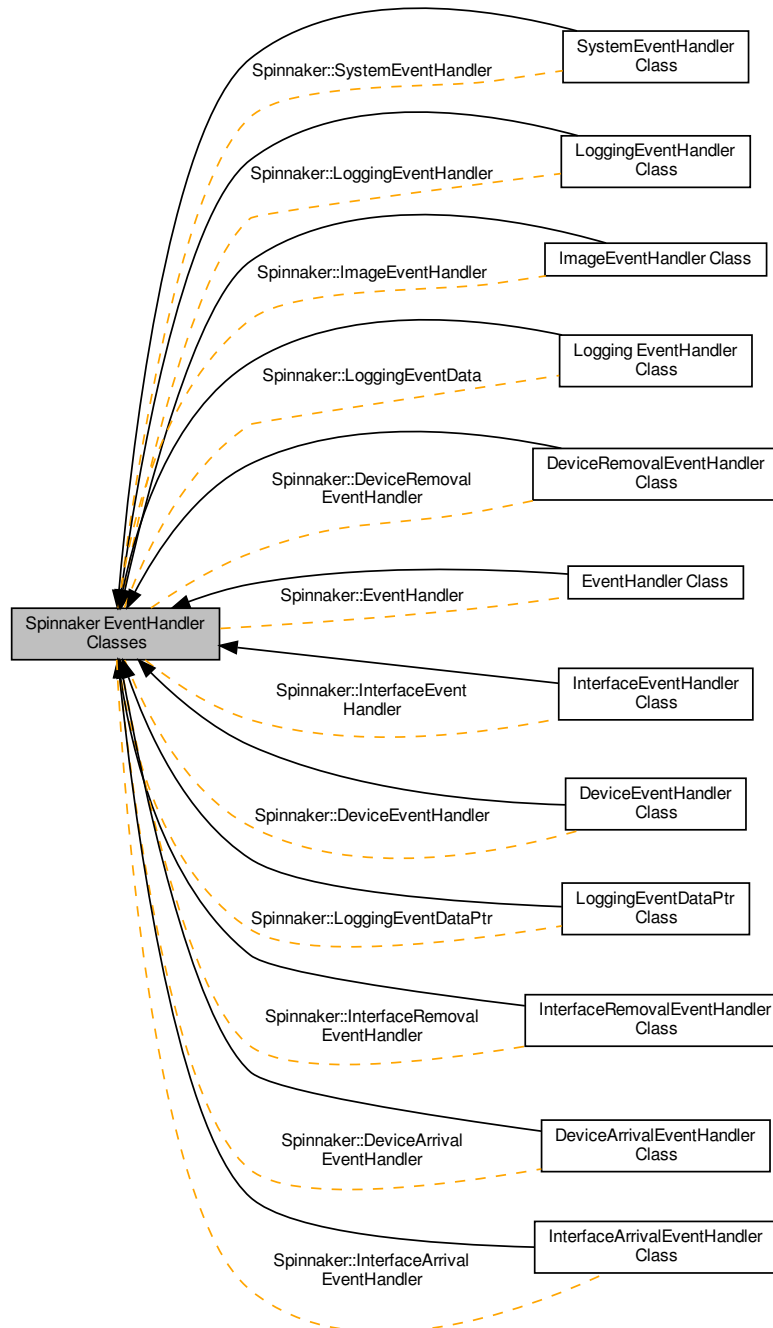
```
int16_t topLeftYCoord
```

13.10.3.18 `topLeftYCoord` [2/2]

```
int16_t topLeftYCoord
```

13.11 Spinnaker EventHandler Classes

Collaboration diagram for Spinnaker EventHandler Classes:



Modules

- [DeviceArrivalEventHandler Class](#)
- [DeviceEventHandler Class](#)

- [DeviceRemovalEventHandler Class](#)
- [EventHandler Class](#)
- [ImageEventHandler Class](#)
- [InterfaceArrivalEventHandler Class](#)
- [InterfaceEventHandler Class](#)
- [InterfaceRemovalEventHandler Class](#)
- [Logging EventHandler Class](#)
- [LoggingEventDataPtr Class](#)
- [LoggingEventHandler Class](#)
- [SystemEventHandler Class](#)

Classes

- class [DeviceArrivalEventHandler](#)
An event handler for capturing the device arrival event.
- class [DeviceEventHandler](#)
A handler to device events.
- class [DeviceRemovalEventHandler](#)
An event handler for capturing the device removal event.
- class [EventHandler](#)
The base class for all event handler types.
- class [ImageEventHandler](#)
A handler for capturing image arrival events.
- class [InterfaceArrivalEventHandler](#)
An event handler for capturing the interface arrival event.
- class [InterfaceEventHandler](#)
A handler to device arrival and removal events on all interfaces.
- class [InterfaceRemovalEventHandler](#)
An event handler for capturing the interface removal event.
- class [LoggingEventData](#)
The [LoggingEventData](#) object.
- class [LoggingEventDataPtr](#)
A reference tracked pointer to the [LoggingEvent](#) object.
- class [LoggingEventHandler](#)
An event handler for capturing the device logging event.
- class [SystemEventHandler](#)
A handler to interface arrival and removal events on the system.

13.11.1 Detailed Description

13.12 DeviceArrivalEventHandler Class

Collaboration diagram for DeviceArrivalEventHandler Class:



Classes

- class [DeviceArrivalEventHandler](#)
An event handler for capturing the device arrival event.

13.12.1 Detailed Description

13.13 DeviceEventHandler Class

Collaboration diagram for DeviceEventHandler Class:



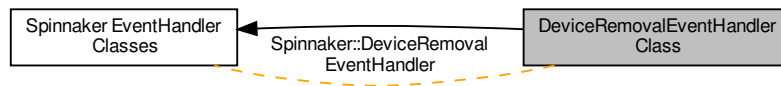
Classes

- class [DeviceEventHandler](#)
A handler to device events.

13.13.1 Detailed Description

13.14 DeviceRemovalEventHandler Class

Collaboration diagram for DeviceRemovalEventHandler Class:



Classes

- class [DeviceRemovalEventHandler](#)
An event handler for capturing the device removal event.

13.14.1 Detailed Description

13.15 EventHandler Class

Collaboration diagram for EventHandler Class:



Classes

- class [EventHandler](#)
The base class for all event handler types.

13.15.1 Detailed Description

13.16 Exception Class

Collaboration diagram for Exception Class:



Classes

- class [Exception](#)

The [Exception](#) object represents an error that is returned from the library.

13.16.1 Detailed Description

13.17 Image Class

Collaboration diagram for Image Class:



Classes

- class [Image](#)
The image object class.

13.17.1 Detailed Description

13.18 ImageEventHandler Class

Collaboration diagram for ImageEventHandler Class:



Classes

- class [ImageEventHandler](#)
A handler for capturing image arrival events.

13.18.1 Detailed Description

13.19 ImagePtr Class

Collaboration diagram for ImagePtr Class:



Classes

- class `ImagePtr`
A reference tracked pointer to an image object.

13.19.1 Detailed Description

13.20 ImageStatistics Class

Collaboration diagram for ImageStatistics Class:



Classes

- class [ImageStatistics](#)
Represents image statistics for an image.

13.20.1 Detailed Description

13.21 Image Utility Class

Collaboration diagram for Image Utility Class:



Classes

- class [ImageUtility](#)

Static helper functions for the image object class.

13.21.1 Detailed Description

13.22 Image Utility CCM Class

Collaboration diagram for Image Utility CCM Class:



Classes

- class [ImageUtilityCCM](#)

Static function to create color corrected images from an image object.

13.22.1 Detailed Description

13.23 Image Utility Heatmap Class

Collaboration diagram for Image Utility Heatmap Class:



Classes

- class [ImageUtilityHeatmap](#)

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

13.23.1 Detailed Description

13.24 Image Utility Polarization Class

Collaboration diagram for Image Utility Polarization Class:



Classes

- class [ImageUtilityPolarization](#)

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

13.24.1 Detailed Description

13.25 Interface Class

Collaboration diagram for Interface Class:



Classes

- class [Interface](#)

An interface object which holds a list of cameras.

13.25.1 Detailed Description

13.26 InterfaceArrivalEventHandler Class

Collaboration diagram for InterfaceArrivalEventHandler Class:



Classes

- class [InterfaceArrivalEventHandler](#)
An event handler for capturing the interface arrival event.

13.26.1 Detailed Description

13.27 InterfaceEventHandler Class

Collaboration diagram for InterfaceEventHandler Class:



Classes

- class [InterfaceEventHandler](#)

A handler to device arrival and removal events on all interfaces.

13.27.1 Detailed Description

13.28 InterfaceList Class

Collaboration diagram for InterfaceList Class:



Classes

- class [InterfaceList](#)
A list of the available interfaces on the system.

13.28.1 Detailed Description

13.29 InterfacePtr Class

Collaboration diagram for InterfacePtr Class:



Classes

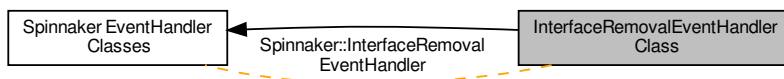
- class [InterfacePtr](#)

A reference tracked pointer to the interface object.

13.29.1 Detailed Description

13.30 InterfaceRemovalEventHandler Class

Collaboration diagram for InterfaceRemovalEventHandler Class:



Classes

- class [InterfaceRemovalEventHandler](#)
An event handler for capturing the interface removal event.

13.30.1 Detailed Description

13.31 Logging EventHandler Class

Collaboration diagram for Logging EventHandler Class:



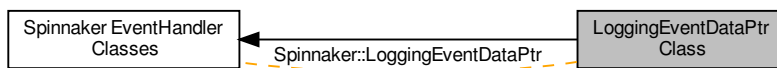
Classes

- class [LoggingEventData](#)
The [LoggingEventData](#) object.

13.31.1 Detailed Description

13.32 LoggingEventDataPtr Class

Collaboration diagram for LoggingEventDataPtr Class:



Classes

- class [LoggingEventDataPtr](#)
A reference tracked pointer to the LoggingEvent object.

13.32.1 Detailed Description

13.33 LoggingEventHandler Class

Collaboration diagram for LoggingEventHandler Class:



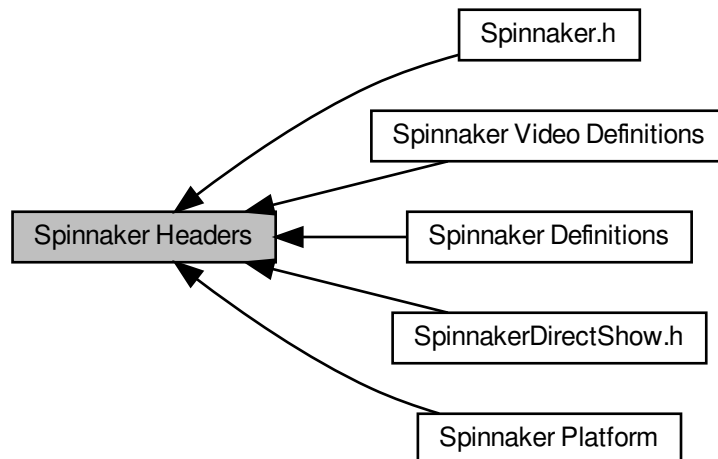
Classes

- class [LoggingEventHandler](#)
An event handler for capturing the device logging event.

13.33.1 Detailed Description

13.34 Spinnaker Headers

Collaboration diagram for Spinnaker Headers:



Modules

- [Spinnaker.h](#)
Global header file for [Spinnaker](#).
- [Spinnaker Definitions](#)
Definitions file for [Spinnaker](#).
- [SpinnakerDirectShow.h](#)
Global header file for [Spinnaker](#) DirectShow library.
- [Spinnaker Platform](#)
Platform-specific header file for [Spinnaker](#).
- [Spinnaker Video Definitions](#)
Definitions file for [Spinnaker](#) video recorder.

Classes

- struct [MJPGOption](#)
Options for saving MJPG files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [AVIOption](#)
Options for saving AVI files.

Variables

- `const uint64_t EVENT_TIMEOUT_NONE = 0`
Timeout values for getting next image, device, or interface event.
- `const uint64_t EVENT_TIMEOUT_INFINITE = 0xFFFFFFFFFFFFFFFF`

Spinnaker GenAPI Functions

These functions deal with [Spinnaker](#) GenAPI NodeMap and Node Accesses for setting camera properties.

- `STDMETHOD() NodeMapGetNumNodes (size_t *numNodes)=0`
Retrieves the number of nodes available in the node map for the currently selected camera.
- `STDMETHOD() NodeMapGetNodeAtIndex (size_t index, char *nodeName, size_t bufferSize)=0`
Retrieves the string representation of the node at the specified index.
- `STDMETHOD() NodesImplemented (const char *nodeName, bool *isImplemented)=0`
Checks if a node is implemented.
- `STDMETHOD() NodesAvailable (const char *nodeName, bool *isAvailable)=0`
Checks if a node is available.
- `STDMETHOD() NodesReadable (const char *nodeName, bool *isReadable)=0`
Checks if a node is readable.
- `STDMETHOD() NodesWritable (const char *nodeName, bool *isWritable)=0`
Checks if a node is writable.
- `STDMETHOD() NodeGetType (const char *nodeName, char *typeName, size_t bufferSize)=0`
Gets the type of the node.
- `STDMETHOD() NodeToString (const char *nodeName, char *valueAsString, size_t bufferSize)=0`
Gets content of the node as string.
- `STDMETHOD() NodeGetDisplayName (const char *nodeName, char *displayName, size_t bufferSize)=0`
Gets a name string for display.

Integer Node Functions

These functions deal with [Spinnaker](#) GenAPI Integer Node Accesses

- `STDMETHOD() IntegerGetValue (const char *nodeName, int64_t *value)=0`
Gets integer node value for the specified feature.
- `STDMETHOD() IntegerSetValue (const char *nodeName, int64_t value)=0`
Sets integer node value for the specified feature.
- `STDMETHOD() IntegerGetMax (const char *nodeName, int64_t *maxValue)=0`
Gets maximum integer value allowed for the specified feature.
- `STDMETHOD() IntegerGetMin (const char *nodeName, int64_t *minValue)=0`
Gets minimum integer value allowed for the specified feature.
- `STDMETHOD() IntegerGetIncMode (const char *nodeName, Spinnaker::GenApi::EIncMode *incMode)=0`
Gets integer increment mode for the specified feature.
- `STDMETHOD() IntegerGetInc (const char *nodeName, int64_t *increment)=0`
Gets integer step increment for the specified feature.

Float Node Functions

These functions deal with [Spinnaker](#) GenAPI Float Node Accesses

- `STDMETHOD() FloatGetValue` (const char *nodeName, double *value)=0
Gets float node value for the specified feature.
- `STDMETHOD() FloatSetValue` (const char *nodeName, double value)=0
Sets float node value for the specified feature.
- `STDMETHOD() FloatGetMax` (const char *nodeName, double *floatMax)=0
Gets maximum float value allowed for the specified feature.
- `STDMETHOD() FloatGetMin` (const char *nodeName, double *floatMin)=0
Gets minimum float value allowed for the specified feature.
- `STDMETHOD() FloatGetIncMode` (const char *nodeName, [Spinnaker::GenApi::EIncMode](#) *incMode)=0
Gets float increment mode for the specified feature.
- `STDMETHOD() FloatGetInc` (const char *nodeName, double *increment)=0
Gets float step increment for the specified feature.

Boolean Node Functions

These functions deal with [Spinnaker](#) GenAPI Boolean Node Accesses

- `STDMETHOD() BooleanGetValue` (const char *nodeName, bool *value)=0
Gets boolean node value for the specified feature.
- `STDMETHOD() BooleanSetValue` (const char *nodeName, bool value)=0
Sets boolean node value for the specified feature.

String Node Functions

These functions deal with [Spinnaker](#) GenAPI String Node Accesses

- `STDMETHOD() StringGetValue` (const char *nodeName, char *value, size_t bufferSize)=0
Gets string node value for the specified feature.
- `STDMETHOD() StringSetValue` (const char *nodeName, const char *value)=0
Sets string node value for the specified feature.

Enumeration Node Functions

These functions deal with [Spinnaker](#) GenAPI Enumeration and EnumeratEntry Node Accesses

- const char * [enumerationName](#)
Checks if an enumeration entry exists for the specified enumeration feature.
- const char const char * [enumerationEntryName](#)
- const char const char bool * [entryExist](#) = 0
- const char unsigned int [entryIndex](#)
- const char unsigned int char size_t [bufferSize](#) = 0
- `STDMETHOD() EnumerationGetEntry` (const char *nodeName, char *value, size_t bufferSize)=0
Gets enumeration entry string for the specified enumeration feature.
- `STDMETHOD() EnumerationSetEntry` (const char *nodeName, const char *value)=0
Sets enumeration entry for the specified enumeration feature.
- `STDMETHOD() EnumerationGetNumEntries` (const char *[enumerationName](#), size_t *numEntries)=0
Retrieves the number of enumeration entry nodes available for the specified enumeration feature.

Command Node Functions

These functions deal with [Spinnaker](#) GenAPI Command Node Accesses

- `STDMETHOD() CommandExecute (const char *nodeName)=0`
Executes the command for the specified feature.

13.34.1 Detailed Description

13.34.2 Function Documentation

13.34.2.1 BooleanGetValue()

```
STDMETHOD() BooleanGetValue (  
    const char * nodeName,  
    bool * value ) [pure virtual]
```

Gets boolean node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.2 BooleanSetValue()

```
STDMETHOD() BooleanSetValue (  
    const char * nodeName,  
    bool value ) [pure virtual]
```

Sets boolean node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.3 CommandExecute()

```
STDMETHOD() CommandExecute (
    const char * nodeName ) [pure virtual]
```

Executes the command for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name

Returns

An HRESULT error code indicating the success or failure of the function.

13.34.2.4 EnumerationGetEntry()

```
STDMETHOD() EnumerationGetEntry (
    const char * nodeName,
    char * value,
    size_t bufferSize ) [pure virtual]
```

Gets enumeration entry string for the specified enumeration feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The enumeration entry string read from the device
<i>bufferSize</i>	Size of the provided value buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.5 EnumerationGetNumEntries()

```
STDMETHOD() EnumerationGetNumEntries (
    const char * enumerationName,
    size_t * numEntries ) [pure virtual]
```

Retrieves the number of enumeration entry nodes available for the specified enumeration feature.

Parameters

<i>enumerationName</i>	GenICam device enumeration feature name
<i>numEntries</i>	Number of nodes available for the specified enumeration feature

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.6 EnumerationSetEntry()

```
STDMETHOD() EnumerationSetEntry (  
    const char * nodeName,  
    const char * value ) [pure virtual]
```

Sets enumeration entry for the specified enumeration feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The enumeration entry string to set to device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.7 FloatGetInc()

```
STDMETHOD() FloatGetInc (  
    const char * nodeName,  
    double * increment ) [pure virtual]
```

Gets float step increment for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>increment</i>	The step increment value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.8 FloatGetIncMode()

```
STDMETHOD() FloatGetIncMode (
    const char * nodeName,
    Spinnaker::GenApi::EIncMode * incMode ) [pure virtual]
```

Gets float increment mode for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>incMode</i>	Spinnaker GenAPI increment mode for the specified feature

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.9 FloatGetMax()

```
STDMETHOD() FloatGetMax (
    const char * nodeName,
    double * floatMax ) [pure virtual]
```

Gets maximum float value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>floatMax</i>	The maximum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.10 FloatGetMin()

```
STDMETHOD() FloatGetMin (
    const char * nodeName,
    double * floatMin ) [pure virtual]
```

Gets minimum float value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>floatMin</i>	The minimum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.11 FloatGetValue()

```
STDMETHOD() FloatGetValue (
    const char * nodeName,
    double * value ) [pure virtual]
```

Gets float node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.12 FloatSetValue()

```
STDMETHOD() FloatSetValue (
    const char * nodeName,
    double value ) [pure virtual]
```

Sets float node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.13 IntegerGetInc()

```
STDMETHOD() IntegerGetInc (
    const char * nodeName,
    int64_t * increment ) [pure virtual]
```

Gets integer step increment for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>increment</i>	The step increment value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.14 IntegerGetIncMode()

```
STDMETHOD() IntegerGetIncMode (
    const char * nodeName,
    Spinnaker::GenApi::EIncMode * incMode ) [pure virtual]
```

Gets integer increment mode for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>incMode</i>	Spinnaker GenAPI increment mode for the specified feature

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.15 IntegerGetMax()

```
STDMETHOD() IntegerGetMax (
    const char * nodeName,
    int64_t * maxValue ) [pure virtual]
```

Gets maximum integer value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>maxValue</i>	The maximum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.16 IntegerGetMin()

```
STDMETHOD() IntegerGetMin (
    const char * nodeName,
    int64_t * minValue ) [pure virtual]
```

Gets minimum integer value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>minValue</i>	The minimum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.17 IntegerGetValue()

```
STDMETHOD() IntegerGetValue (
    const char * nodeName,
    int64_t * value ) [pure virtual]
```

Gets integer node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.18 IntegerSetValue()

```
STDMETHOD() IntegerSetValue (
    const char * nodeName,
    int64_t value ) [pure virtual]
```

Sets integer node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.19 NodeGetDisplayName()

```
STDMETHOD() NodeGetDisplayName (
    const char * nodeName,
    char * displayName,
    size_t bufferSize ) [pure virtual]
```

Gets a name string for display.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>displayName</i>	Display name of the device feature
<i>bufferSize</i>	Size of the provided displayName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.20 NodeGetType()

```
STDMETHOD() NodeGetType (
    const char * nodeName,
    char * typeName,
    size_t bufferSize ) [pure virtual]
```

Gets the type of the node.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>typeName</i>	The type of the node. Available type names are: Integer, Float, Boolean, String, Enumeration, Entry, Command and Other
<i>bufferSize</i>	Size of the provided typeName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.21 NodeIsAvailable()

```
STDMETHOD() NodeIsAvailable (
    const char * nodeName,
    bool * isAvailable ) [pure virtual]
```

Checks if a node is available.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isAvailable</i>	Flag indicating whether node is available or not

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.22 NodeIsImplemented()

```
STDMETHOD() NodeIsImplemented (
    const char * nodeName,
    bool * isImplemented ) [pure virtual]
```

Checks if a node is implemented.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isImplemented</i>	Flag indicating whether node is implemented or not

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.23 NodeIsReadable()

```
STDMETHOD() NodeIsReadable (
    const char * nodeName,
    bool * isReadable ) [pure virtual]
```

Checks if a node is readable.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isReadable</i>	Flag indicating whether node is readable or not

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.24 NodeIsWritable()

```
STDMETHOD() NodeIsWritable (
    const char * nodeName,
    bool * isWritable ) [pure virtual]
```

Checks if a node is writable.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isWritable</i>	Flag indicating whether node is writable or not

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.25 NodeMapGetNodeAtIndex()

```
STDMETHOD() NodeMapGetNodeAtIndex (
    size_t index,
    char * nodeName,
    size_t bufferSize ) [pure virtual]
```

Retrieves the string representation of the node at the specified index.

Parameters

<i>index</i>	Node index in the NodeMap
<i>nodeName</i>	GenICam node name at specified index to be populated
<i>bufferSize</i>	Size of the provided nodeName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.26 NodeMapGetNumNodes()

```
STDMETHOD() NodeMapGetNumNodes (
    size_t * numNodes ) [pure virtual]
```

Retrieves the number of nodes available in the node map for the currently selected camera.

Parameters

<i>numNodes</i>	Number of nodes available for the currently selected device

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.27 NodeToString()

```
STDMETHOD() NodeToString (
    const char * nodeName,
    char * valueAsString,
    size_t bufferSize ) [pure virtual]
```

Gets content of the node as string.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>valueAsString</i>	String representation of the node value
<i>bufferSize</i>	Size of the provided valueAsString buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.28 StringGetValue()

```
STDMETHOD() StringGetValue (
    const char * nodeName,
    char * value,
    size_t bufferSize ) [pure virtual]
```

Gets string node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device
<i>bufferSize</i>	Size of the provided value buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.34.2.29 StringSetValue()

```
STDMETHOD() StringSetValue (  
    const char * nodeName,  
    const char * value ) [pure virtual]
```

Sets string node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

13.34.3 Variable Documentation**13.34.3.1 bufferSize**

```
const char unsigned int char size_t bufferSize = 0
```

13.34.3.2 entryExist

```
const char const char bool* entryExist = 0
```

13.34.3.3 entryIndex

```
const char unsigned int entryIndex
```

13.34.3.4 enumerationEntryName

```
const char unsigned int char * enumerationEntryName
```

13.34.3.5 enumerationName

```
const char * enumerationName
```

Checks if an enumeration entry exists for the specified enumeration feature.

Gets enumeration entry string for the specified enumeration feature at the specified index.

Parameters

<i>enumerationName</i>	GenICam device enumeration feature name
<i>enumerationEntryName</i>	GenICam device enumeration entry name to check
<i>entryExist</i>	Flag indicating whether the specified entry node exists for the enumeration node

Returns

An HRESULT indicating the success or failure of the function.

Parameters

<i>enumerationName</i>	GenICam device enumeration feature name
<i>entryIndex</i>	Enumeration entry index
<i>enumerationEntryName</i>	GenICam device enumeration entry name at specified entryIndex to be populated
<i>bufferSize</i>	Size of the provided enumerationEntryName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.34.3.6 EVENT_TIMEOUT_INFINITE

```
const uint64_t EVENT_TIMEOUT_INFINITE = 0xFFFFFFFFFFFFFFFF
```

13.34.3.7 EVENT_TIMEOUT_NONE

```
const uint64_t EVENT_TIMEOUT_NONE = 0
```

Timeout values for getting next image, device, or interface event.

13.35 Spinnaker.h

Global header file for [Spinnaker](#).

Collaboration diagram for Spinnaker.h:



Global header file for [Spinnaker](#).

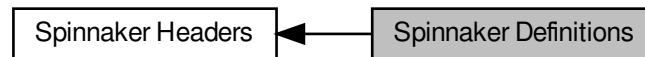
By including this file, all required header files for full [Spinnaker](#) operation will be included automatically. It is recommended that this file be used instead of manually including individual header files.

We welcome your bug reports, suggestions, and comments: <https://www.flir.com/support-center/rma/iis-support>

13.36 Spinnaker Definitions

Definitions file for [Spinnaker](#).

Collaboration diagram for Spinnaker Definitions:



Classes

- struct [PNGOption](#)
Options for saving PNG images.
- struct [PPMOption](#)
Options for saving PPM images.
- struct [PGMOption](#)
Options for saving PGM images.
- struct [TIFFOption](#)
Options for saving TIFF images.
- struct [JPEGOption](#)
Options for saving JPEG image.
- struct [JPG2Option](#)
Options for saving JPEG2000 image.
- struct [BMPOption](#)
Options for saving Bitmap image.
- struct [LibraryVersion](#)
Provides easier access to the current version of [Spinnaker](#).
- struct [ActionCommandResult](#)
Action Command Result.
- struct [CCMSettings](#)
- struct [DeviceEventInferenceData](#)
Data Fields for Device Event payload for EventInference.
- struct [DeviceEventExposureEndData](#)
Data Fields for Device Event payload for EventExposureEnd.

Enumerations

- enum [Error](#) {
[SPINNAKER_ERR_SUCCESS](#) = 0,
[SPINNAKER_ERR_ERROR](#) = -1001,
[SPINNAKER_ERR_NOT_INITIALIZED](#) = -1002,
[SPINNAKER_ERR_NOT_IMPLEMENTED](#) = -1003,
[SPINNAKER_ERR_RESOURCE_IN_USE](#) = -1004,
[SPINNAKER_ERR_ACCESS_DENIED](#) = -1005,

```

SPINNAKER_ERR_INVALID_HANDLE = -1006,
SPINNAKER_ERR_INVALID_ID = -1007,
SPINNAKER_ERR_NO_DATA = -1008,
SPINNAKER_ERR_INVALID_PARAMETER = -1009,
SPINNAKER_ERR_IO = -1010,
SPINNAKER_ERR_TIMEOUT = -1011,
SPINNAKER_ERR_ABORT = -1012,
SPINNAKER_ERR_INVALID_BUFFER = -1013,
SPINNAKER_ERR_NOT_AVAILABLE = -1014,
SPINNAKER_ERR_INVALID_ADDRESS = -1015,
SPINNAKER_ERR_BUFFER_TOO_SMALL = -1016,
SPINNAKER_ERR_INVALID_INDEX = -1017,
SPINNAKER_ERR_PARSING_CHUNK_DATA = -1018,
SPINNAKER_ERR_INVALID_VALUE = -1019,
SPINNAKER_ERR_RESOURCE_EXHAUSTED = -1020,
SPINNAKER_ERR_OUT_OF_MEMORY = -1021,
SPINNAKER_ERR_BUSY = -1022,
GENICAM_ERR_INVALID_ARGUMENT = -2001,
GENICAM_ERR_OUT_OF_RANGE = -2002,
GENICAM_ERR_PROPERTY = -2003,
GENICAM_ERR_RUN_TIME = -2004,
GENICAM_ERR_LOGICAL = -2005,
GENICAM_ERR_ACCESS = -2006,
GENICAM_ERR_TIMEOUT = -2007,
GENICAM_ERR_DYNAMIC_CAST = -2008,
GENICAM_ERR_GENERIC = -2009,
GENICAM_ERR_BAD_ALLOCATION = -2010,
SPINNAKER_ERR_IM_CONVERT = -3001,
SPINNAKER_ERR_IM_COPY = -3002,
SPINNAKER_ERR_IM_MALLOC = -3003,
SPINNAKER_ERR_IM_NOT_SUPPORTED = -3004,
SPINNAKER_ERR_IM_HISTOGRAM_RANGE = -3005,
SPINNAKER_ERR_IM_HISTOGRAM_MEAN = -3006,
SPINNAKER_ERR_IM_MIN_MAX = -3007,
SPINNAKER_ERR_IM_COLOR_CONVERSION = -3008,
SPINNAKER_ERR_IM_DECOMPRESSION = -3009,
SPINNAKER_ERR_CUSTOM_ID = -10000 }

```

Spinnaker enum definitions.

- enum EventType {


```

SPINNAKER_EVENT_ARRIVAL_REMOVAL,
SPINNAKER_EVENT_DEVICE,
SPINNAKER_EVENT_DEVICE_SPECIFIC,
SPINNAKER_EVENT_NEW_BUFFER,
SPINNAKER_EVENT_LOGGING_EVENT,
SPINNAKER_EVENT_UNKNOWN,
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL }

```

Event types in Spinnaker.

- enum PixelFormatNamespaceID {


```

SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }

```

This enum represents the namespace in which the TL specific pixel format resides.

- enum ColorProcessingAlgorithm {


```

DEFAULT,
NO_COLOR_PROCESSING,

```

```

NEAREST_NEIGHBOR,
NEAREST_NEIGHBOR_AVG,
BILINEAR,
EDGE_SENSING,
HQ_LINEAR,
IPP,
DIRECTIONAL_FILTER,
RIGOROUS,
WEIGHTED_DIRECTIONAL_FILTER }

```

Color processing algorithms.

- enum `ImageFileFormat` {
`FROM_FILE_EXT` = -1,
`PGM`,
`PPM`,
`BMP`,
`JPEG`,
`JPEG2000`,
`TIFF`,
`PNG`,
`RAW`,
`JPEG12_C`,
`IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

File formats to be used for saving images to disk.

- enum `ImageStatus` {
`IMAGE_UNKNOWN_ERROR` = -1,
`IMAGE_NO_ERROR` = 0,
`IMAGE_CRC_CHECK_FAILED` = 1,
`IMAGE_DATA_OVERFLOW` = 2,
`IMAGE_MISSING_PACKETS`,
`IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT`,
`IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT`,
`IMAGE_PACKETID_INCONSISTENT`,
`IMAGE_MISSING_LEADER` = 7,
`IMAGE_MISSING_TRAILER`,
`IMAGE_DATA_INCOMPLETE`,
`IMAGE_INFO_INCONSISTENT`,
`IMAGE_CHUNK_DATA_INVALID` = 11,
`IMAGE_NO_SYSTEM_RESOURCES` = 12 }

Status of images returned from `GetNextImage()` call.

- enum `StatisticsChannel` {
`GREY`,
`RED`,
`GREEN`,
`BLUE`,
`HUE`,
`SATURATION`,
`LIGHTNESS`,
`NUM_STATISTICS_CHANNELS` }

Channels that allow statistics to be calculated.

- enum `SpinnakerLogLevel` {
`LOG_LEVEL_OFF` = -1,
`LOG_LEVEL_FATAL` = 0,
`LOG_LEVEL_ALERT` = 100,
`LOG_LEVEL_CRIT` = 200,
`LOG_LEVEL_ERROR` = 300,
`LOG_LEVEL_WARN` = 400,
`LOG_LEVEL_NOTICE` = 500,

```
LOG_LEVEL_INFO = 600,
LOG_LEVEL_DEBUG = 700,
LOG_LEVEL_NOTSET = 800 }
```

log levels

- enum `PayloadTypeInfoIds` {


```
PAYLOAD_TYPE_UNKNOWN = 0,
PAYLOAD_TYPE_IMAGE = 1,
PAYLOAD_TYPE_RAW_DATA = 2,
PAYLOAD_TYPE_FILE = 3,
PAYLOAD_TYPE_CHUNK_DATA = 4,
PAYLOAD_TYPE_JPEG = 5,
PAYLOAD_TYPE_JPEG2000 = 6,
PAYLOAD_TYPE_H264 = 7,
PAYLOAD_TYPE_CHUNK_ONLY = 8,
PAYLOAD_TYPE_DEVICE_SPECIFIC = 9,
PAYLOAD_TYPE_MULTI_PART = 10,
PAYLOAD_TYPE_CUSTOM_ID = 1000,
PAYLOAD_TYPE_EXTENDED_CHUNK = 1001,
PAYLOAD_TYPE_LOSSLESS_COMPRESSED = 1002,
PAYLOAD_TYPE_LOSSY_COMPRESSED = 1003,
PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED = 1004,
PAYLOAD_TYPE_CHUNK_DATA_LOSSLESS_COMPRESSED = 1005,
PAYLOAD_TYPE_CHUNK_DATA_LOSSY_COMPRESSED = 1006 }
```
- enum `ActionCommandStatus` {


```
ACTION_COMMAND_STATUS_OK = 0,
ACTION_COMMAND_STATUS_NO_REF_TIME,
ACTION_COMMAND_STATUS_OVERFLOW = 0x8015,
ACTION_COMMAND_STATUS_ACTION_LATE,
ACTION_COMMAND_STATUS_ERROR }
```

Possible Status Codes Returned from Action Command.

- enum `PixelFormatIntType` {


```
IntType_UINT8,
IntType_INT8,
IntType_UINT10,
IntType_UINT10p,
IntType_UINT10P,
IntType_UINT12,
IntType_UINT12p,
IntType_UINT12P,
IntType_UINT14,
IntType_UINT16,
IntType_INT16,
IntType_FLOAT32,
IntType_UNKNOWN }
```

Possible integer types and packing used in a pixel format.

- enum `BufferOwnership` {


```
BUFFER_OWNERSHIP_SYSTEM,
BUFFER_OWNERSHIP_USER }
```
- enum `CCMColorTemperature` {


```
TUNGSTEN_2800K,
WARM_FLUORESCENT_3000K,
COOL_FLUORESCENT_4000K,
SUNNY_5000K,
CLOUDY_6500K,
SHADE_8000K }
```
- enum `CCMType` {


```
LINEAR_3X3,
POLYNOMIAL_9X3 }
```

- enum [CCMSensor](#) { [IMX250](#) }

13.36.1 Detailed Description

Definitions file for [Spinnaker](#).

13.36.2 Enumeration Type Documentation

13.36.2.1 ActionCommandStatus

enum [ActionCommandStatus](#)

Possible Status Codes Returned from Action Command.

Enumerator

ACTION_COMMAND_STATUS_OK	
ACTION_COMMAND_STATUS_NO_REF_TIME	
ACTION_COMMAND_STATUS_OVERFLOW	
ACTION_COMMAND_STATUS_ACTION_LATE	
ACTION_COMMAND_STATUS_ERROR	

13.36.2.2 BufferOwnership

enum [BufferOwnership](#)

Enumerator

BUFFER_OWNERSHIP_SYSTEM	
BUFFER_OWNERSHIP_USER	

13.36.2.3 CCMColorTemperature

enum [CCMColorTemperature](#)

Enumerator

TUNGSTEN_2800K	
WARM_FLUORESCENT_3000K	

Enumerator

COOL_FLUORESCENT_4000K	
SUNNY_5000K	
CLOUDY_6500K	
SHADE_8000K	

13.36.2.4 CCMSensor

enum `CCMSensor`

Enumerator

IMX250	

13.36.2.5 CCMTType

enum `CCMTType`

Enumerator

LINEAR_3X3	
POLYNOMIAL_9X3	

13.36.2.6 ColorProcessingAlgorithm

enum `ColorProcessingAlgorithm`

Color processing algorithms.

Please refer to our knowledge base at article at <https://www.flir.com/support-center/iis/machine-vision/kn> for complete details for each algorithm.

Enumerator

DEFAULT	Default method.
NO_COLOR_PROCESSING	No color processing.
NEAREST_NEIGHBOR	Fastest but lowest quality. Equivalent to FLYCAPTURE_NEAREST_NEIGHBOR_FAST in FlyCapture.
NEAREST_NEIGHBOR_AVG	Nearest Neighbor with averaged green pixels. Higher quality but slower compared to nearest neighbor without averaging.
BILINEAR	Weighted average of surrounding 4 pixels in a 2x2 neighborhood.

Enumerator

EDGE_SENSING	Weights surrounding pixels based on localized edge orientation.
HQ_LINEAR	Well-balanced speed and quality.
IPP	Multi-threaded with similar results to edge sensing.
DIRECTIONAL_FILTER	Best quality but much faster than rigorous. More memory intensive than other color processing algorithms.
RIGOROUS	Slowest but produces good results.
WEIGHTED_DIRECTIONAL_FILTER	Weighted pixel average from different directions.

13.36.2.7 Error

enum [Error](#)

[Spinnaker](#) enum definitions.

The error codes used in [Spinnaker](#). These codes are returned as part of [Spinnaker::Exception](#). The error codes in the range of -1000 to -1999 are reserved for exceptions that map directly to GenTL values. The error codes in the range of -2000 to -2999 are reserved for [GenICam](#) related errors. The error codes in the range of -3000 to -3999 are reserved for image processing related errors.

Enumerator

SPINNAKER_ERR_SUCCESS	
SPINNAKER_ERR_ERROR	
SPINNAKER_ERR_NOT_INITIALIZED	
SPINNAKER_ERR_NOT_IMPLEMENTED	
SPINNAKER_ERR_RESOURCE_IN_USE	
SPINNAKER_ERR_ACCESS_DENIED	
SPINNAKER_ERR_INVALID_HANDLE	
SPINNAKER_ERR_INVALID_ID	
SPINNAKER_ERR_NO_DATA	
SPINNAKER_ERR_INVALID_PARAMETER	
SPINNAKER_ERR_IO	
SPINNAKER_ERR_TIMEOUT	
SPINNAKER_ERR_ABORT	
SPINNAKER_ERR_INVALID_BUFFER	
SPINNAKER_ERR_NOT_AVAILABLE	
SPINNAKER_ERR_INVALID_ADDRESS	
SPINNAKER_ERR_BUFFER_TOO_SMALL	
SPINNAKER_ERR_INVALID_INDEX	
SPINNAKER_ERR_PARSING_CHUNK_DATA	
SPINNAKER_ERR_INVALID_VALUE	
SPINNAKER_ERR_RESOURCE_EXHAUSTED	
SPINNAKER_ERR_OUT_OF_MEMORY	
SPINNAKER_ERR_BUSY	
GENICAM_ERR_INVALID_ARGUMENT	
GENICAM_ERR_OUT_OF_RANGE	
GENICAM_ERR_PROPERTY	

Enumerator

GENICAM_ERR_RUN_TIME	
GENICAM_ERR_LOGICAL	
GENICAM_ERR_ACCESS	
GENICAM_ERR_TIMEOUT	
GENICAM_ERR_DYNAMIC_CAST	
GENICAM_ERR_GENERIC	
GENICAM_ERR_BAD_ALLOCATION	
SPINNAKER_ERR_IM_CONVERT	
SPINNAKER_ERR_IM_COPY	
SPINNAKER_ERR_IM_MALLOC	
SPINNAKER_ERR_IM_NOT_SUPPORTED	
SPINNAKER_ERR_IM_HISTOGRAM_RANGE	
SPINNAKER_ERR_IM_HISTOGRAM_MEAN	
SPINNAKER_ERR_IM_MIN_MAX	
SPINNAKER_ERR_IM_COLOR_CONVERSION	
SPINNAKER_ERR_IM_DECOMPRESSION	
SPINNAKER_ERR_CUSTOM_ID	

13.36.2.8 EventType

enum [EventType](#)

Event types in [Spinnaker](#).

See also

[EventHandler::GetEventType\(\)](#)

Enumerator

SPINNAKER_EVENT_ARRIVAL_REMOVAL	
SPINNAKER_EVENT_DEVICE	
SPINNAKER_EVENT_DEVICE_SPECIFIC	
SPINNAKER_EVENT_NEW_BUFFER	
SPINNAKER_EVENT_LOGGING_EVENT	
SPINNAKER_EVENT_UNKNOWN	
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL	

13.36.2.9 ImageFileFormat

enum [ImageFileFormat](#)

File formats to be used for saving images to disk.

Enumerator

FROM_FILE_EXT	Determine file format from file extension.
PGM	Portable gray map.
PPM	Portable pixmap.
BMP	Bitmap.
JPEG	JPEG.
JPEG2000	JPEG 2000.
TIFF	Tagged image file format.
PNG	Portable network graphics.
RAW	Raw data.
JPEG12_C	12 bit compressed JPEG data.
IMAGE_FILE_FORMAT_FORCE_32BITS	

13.36.2.10 ImageStatus

enum `ImageStatus`

Status of images returned from `GetNextImage()` call.

Enumerator

IMAGE_UNKNOWN_ERROR	<code>Image</code> has an unknown error.
IMAGE_NO_ERROR	<code>Image</code> is returned from <code>GetNextImage()</code> call without any errors.
IMAGE_CRC_CHECK_FAILED	<code>Image</code> failed CRC check.
IMAGE_DATA_OVERFLOW	Received more data than the size of the image.
IMAGE_MISSING_PACKETS	<code>Image</code> has missing packets. Potential fixes include enabling jumbo packets and adjusting packet size/delay. For more information see https://www.flir.com/support-center/iis/machine-vision/application
IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT	<code>Image</code> leader is incomplete. Could be caused by missing packet(s). See link above.
IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT	<code>Image</code> trailer is incomplete. Could be caused by missing packet(s). See link above.
IMAGE_PACKETID_INCONSISTENT	<code>Image</code> has an inconsistent packet id. Could be caused by missing packet(s). See link above.
IMAGE_MISSING_LEADER	<code>Image</code> leader is missing. Could be caused by missing packet(s). See link above.
IMAGE_MISSING_TRAILER	<code>Image</code> trailer is missing. Could be caused by missing packet(s). See link above.
IMAGE_DATA_INCOMPLETE	<code>Image</code> data is incomplete. Could be caused by missing packet(s). See link above.
IMAGE_INFO_INCONSISTENT	<code>Image</code> info is corrupted. Could be caused by missing packet(s). See link above.
IMAGE_CHUNK_DATA_INVALID	<code>Image</code> chunk data is invalid.
IMAGE_NO_SYSTEM_RESOURCES	<code>Image</code> cannot be processed due to lack of system resources.

13.36.2.11 PayloadTypeInfoIDs

enum `PayloadTypeInfoIDs`

Enumerator

PAYLOAD_TYPE_UNKNOWN	
PAYLOAD_TYPE_IMAGE	
PAYLOAD_TYPE_RAW_DATA	
PAYLOAD_TYPE_FILE	
PAYLOAD_TYPE_CHUNK_DATA	
PAYLOAD_TYPE_JPEG	
PAYLOAD_TYPE_JPEG2000	
PAYLOAD_TYPE_H264	
PAYLOAD_TYPE_CHUNK_ONLY	
PAYLOAD_TYPE_DEVICE_SPECIFIC	
PAYLOAD_TYPE_MULTI_PART	
PAYLOAD_TYPE_CUSTOM_ID	
PAYLOAD_TYPE_EXTENDED_CHUNK	
PAYLOAD_TYPE_LOSSLESS_COMPRESSED	
PAYLOAD_TYPE_LOSSY_COMPRESSED	
PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED	
PAYLOAD_TYPE_CHUNK_DATA_LOSSLESS_COMPRESSED	
PAYLOAD_TYPE_CHUNK_DATA_LOSSY_COMPRESSED	

13.36.2.12 PixelFormatIntType

enum `PixelFormatIntType`

Possible integer types and packing used in a pixel format.

Enumerator

IntType_UINT8	
IntType_INT8	
IntType_UINT10	
IntType_UINT10p	
IntType_UINT10P	
IntType_UINT12	
IntType_UINT12p	
IntType_UINT12P	
IntType_UINT14	
IntType_UINT16	
IntType_INT16	
IntType_FLOAT32	
IntType_UNKNOWN	

13.36.2.13 PixelFormatNamespaceID

enum [PixelFormatNamespaceID](#)

This enum represents the namespace in which the TL specific pixel format resides.

This enum is returned from a captured image when calling [Image::GetTLPixelFormatNamespace\(\)](#). It can be used to interpret the raw pixel format returned from [Image::GetTLPixelFormat\(\)](#).

See also

[Image::GetTLPixelFormat\(\)](#)

[Image::GetTLPixelFormatNamespace\(\)](#)

Enumerator

SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN	
SPINNAKER_PIXELFORMAT_NAMESPACE_GEV	
SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC	
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT	
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT	
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID	

13.36.2.14 SpinnakerLogLevel

enum [SpinnakerLogLevel](#)

log levels

Enumerator

LOG_LEVEL_OFF	
LOG_LEVEL_FATAL	
LOG_LEVEL_ALERT	
LOG_LEVEL_CRIT	
LOG_LEVEL_ERROR	
LOG_LEVEL_WARN	
LOG_LEVEL_NOTICE	
LOG_LEVEL_INFO	
LOG_LEVEL_DEBUG	
LOG_LEVEL_NOTSET	

13.36.2.15 StatisticsChannel

```
enum StatisticsChannel
```

Channels that allow statistics to be calculated.

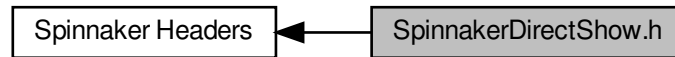
Enumerator

GREY	
RED	
GREEN	
BLUE	
HUE	
SATURATION	
LIGHTNESS	
NUM_STATISTICS_CHANNELS	

13.37 SpinnakerDirectShow.h

Global header file for [Spinnaker](#) DirectShow library.

Collaboration diagram for SpinnakerDirectShow.h:



Functions

- `STDMETHOD() GetSelectedCameraIndex (unsigned int *selectedIndex)=0`
Retrieves an integer index to the currently selected camera.
- `STDMETHOD() SetSelectedCameraIndex (unsigned int index, bool needsRelease=true)=0`
Sets the currently selected camera to the index specified.
- `STDMETHOD() GetCameraInfo (unsigned int index, char *model, char *type, char *serial, size_t buffer↔
Size)=0`
Retrieves general information about the camera at the specified index.
- `STDMETHOD() IsStreaming (bool *isStreaming)=0`
Checks if the selected camera is currently streaming.
- `STDMETHOD() BeginAcquisition ()=0`
Starts the image acquisition engine for the currently selected camera.
- `STDMETHOD() EndAcquisition ()=0`
Stops the image acquisition engine for the currently selected camera.

Variables

- static const GUID [IID_ISpinnakerInterface](#)
This is the Interface that allows users to get and set device node properties on the camera.

13.37.1 Detailed Description

Global header file for [Spinnaker](#) DirectShow library.

By including this file, all required header files for full [Spinnaker](#) DirectShow operation will be included automatically. It is recommended that this file be used instead of manually including individual header files.

We welcome your bug reports, suggestions, and comments: <https://www.flir.com/support-center/rma/iis-support>

13.37.2 Function Documentation

13.37.2.1 BeginAcquisition()

```
STDMETHOD() BeginAcquisition ( ) [pure virtual]
```

Starts the image acquisition engine for the currently selected camera.

Nothing occurs if the camera is already streaming.

Returns

An HRESULT indicating the success or failure of the function.

13.37.2.2 EndAcquisition()

```
STDMETHOD() EndAcquisition ( ) [pure virtual]
```

Stops the image acquisition engine for the currently selected camera.

Nothing occurs if there were no prior call to [BeginAcquisition\(\)](#). Note that [EndAcquisition\(\)](#) needs to be called before selecting a new camera.

Returns

An HRESULT indicating the success or failure of the function.

13.37.2.3 GetCameraInfo()

```
STDMETHOD() GetCameraInfo (
    unsigned int index,
    char * model,
    char * type,
    char * serial,
    size_t bufferSize ) [pure virtual]
```

Retrieves general information about the camera at the specified index.

Parameters

<i>index</i>	The index at which to retrieve the camera information
<i>model</i>	Model description of the camera
<i>type</i>	Transport layer description of the camera
<i>serial</i>	Serial number of the camera
<i>bufferSize</i>	Size of the individual provided model, sensor, serial buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

13.37.2.4 GetSelectedCameraIndex()

```
STDMETHOD() GetSelectedCameraIndex (
    unsigned int * selectedIndex ) [pure virtual]
```

Retrieves an integer index to the currently selected camera.

Parameters

<i>selectedIndex</i>	An integer that represents the index of the selected camera

Returns

An HRESULT indicating the success or failure of the function.

13.37.2.5 IsStreaming()

```
STDMETHOD() IsStreaming (
    bool * isStreaming ) [pure virtual]
```

Checks if the selected camera is currently streaming.

Parameters

<i>isStreaming</i>	Flag indicating whether camera is streaming or not

Returns

An HRESULT indicating the success or failure of the function.

13.37.2.6 SetSelectedCameraIndex()

```
STDMETHOD() SetSelectedCameraIndex (
    unsigned int index,
    bool needsRelease = true ) [pure virtual]
```

Sets the currently selected camera to the index specified.

This function will do nothing if the selected index is out of range.

Parameters

<i>index</i>	The index at which to retrieve the camera object
<i>needsRelease</i>	Specifies whether to release the previously selected camera object

Returns

An HRESULT indicating the success or failure of the function.

13.37.3 Variable Documentation**13.37.3.1 IID_ISpinnakerInterface**

```
const GUID IID_ISpinnakerInterface [static]
```

Initial value:

```
= {0x5143ad36, 0x51d1, 0x422d, 0x9b, 0x17, 0x81, 0xdd, 0x3d, 0xb0, 0x60, 0xe}}
```

This is the Interface that allows users to get and set device node properties on the camera.

You can query the SpinnakerDirectShow Capture Filter for IID_ISpinnakerInterface and it will return a pointer to the ISpinnakerInterface interface. The GUID is equivalent to {5143AD36-51D1-422D-9B17-81DD3DB0600E}.

13.38 Spinnaker Platform

Platform-specific header file for [Spinnaker](#).

Collaboration diagram for Spinnaker Platform:



Macros

- `#define SPINNAKER_API_ABSTRACT /*nothing*/`
- `#define SPINNAKER_API __attribute__((visibility("default")))`
- `#define SPINNAKER_LOCAL __attribute__((visibility("hidden")))`

13.38.1 Detailed Description

Platform-specific header file for [Spinnaker](#).

All the platform-specific code that is required by individual compilers to produce the appropriate code for each platform.

13.38.2 Macro Definition Documentation

13.38.2.1 SPINNAKER_API

```
#define SPINNAKER_API __attribute__((visibility("default")))
```

13.38.2.2 SPINNAKER_API_ABSTRACT

```
#define SPINNAKER_API_ABSTRACT /*nothing*/
```

13.38.2.3 SPINNAKER_LOCAL

```
#define SPINNAKER_LOCAL __attribute__((visibility("hidden")))
```

13.39 Spinnaker Video Class

Collaboration diagram for Spinnaker Video Class:



Classes

- class [SpinVideo](#)

Provides the functionality for the user to record images to an AVI/MP4 file.

13.39.1 Detailed Description

13.40 Spinnaker Video Definitions

Definitions file for [Spinnaker](#) video recorder.

Collaboration diagram for Spinnaker Video Definitions:



Definitions file for [Spinnaker](#) video recorder.

13.41 System Class

Collaboration diagram for System Class:



Classes

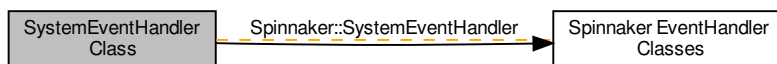
- class [System](#)

The system object is used to retrieve the list of interfaces and cameras available.

13.41.1 Detailed Description

13.42 SystemEventHandler Class

Collaboration diagram for SystemEventHandler Class:



Classes

- class [SystemEventHandler](#)

A handler to interface arrival and removal events on the system.

13.42.1 Detailed Description

13.43 SystemPtr Class

Collaboration diagram for SystemPtr Class:



Classes

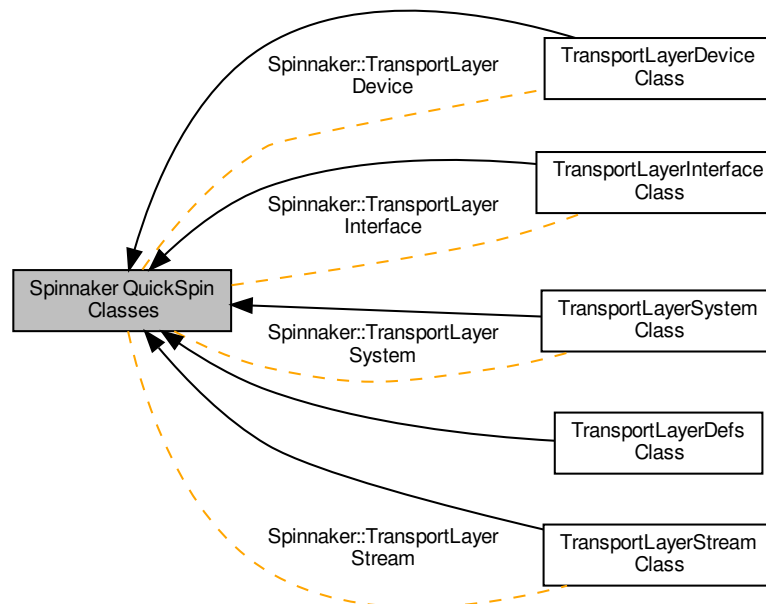
- class `SystemPtr`

A reference tracked pointer to a system object.

13.43.1 Detailed Description

13.44 Spinnaker QuickSpin Classes

Collaboration diagram for Spinnaker QuickSpin Classes:



Modules

- [TransportLayerDefs Class](#)
- [TransportLayerDevice Class](#)
- [TransportLayerInterface Class](#)
- [TransportLayerStream Class](#)
- [TransportLayerSystem Class](#)

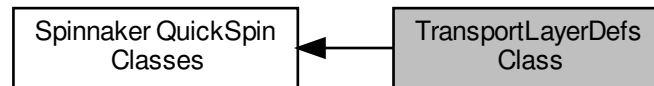
Classes

- class [TransportLayerDevice](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerInterface](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerStream](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerSystem](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

13.44.1 Detailed Description

13.45 TransportLayerDefs Class

Collaboration diagram for TransportLayerDefs Class:



Enumerations

- enum `StreamTypeEnum` {
`StreamType_GigEVision`,
`StreamType_CameraLink`,
`StreamType_CameraLinkHS`,
`StreamType_CoaXPress`,
`StreamType_USB3Vision`,
`StreamType_Custom`,
`NUMSTREAMTYPE` }

The enum definitions for TL Device nodes from the transport layer .xml files.

- enum `StreamBufferCountModeEnum` {
`StreamBufferCountMode_Manual`,
`StreamBufferCountMode_Auto`,
`NUMSTREAMBUFFERCOUNTMODE` }
- enum `StreamBufferHandlingModeEnum` {
`StreamBufferHandlingMode_OldestFirst`,
`StreamBufferHandlingMode_OldestFirstOverwrite`,
`StreamBufferHandlingMode_NewestOnly`,
`StreamBufferHandlingMode_NewestFirst`,
`NUMSTREAMBUFFERHANDLINGMODE` }
- enum `DeviceTypeEnum` {
`DeviceType_GigEVision`,
`DeviceType_CameraLink`,
`DeviceType_CameraLinkHS`,
`DeviceType_CoaXPress`,
`DeviceType_USB3Vision`,
`DeviceType_Custom`,
`NUMDEVICETYPE` }
- enum `DeviceAccessStatusEnum` {
`DeviceAccessStatus_Unknown`,
`DeviceAccessStatus_ReadWrite`,
`DeviceAccessStatus_ReadOnly`,
`DeviceAccessStatus_NoAccess`,
`DeviceAccessStatus_Busy`,
`DeviceAccessStatus_OpenReadWrite`,
`DeviceAccessStatus_OpenReadOnly`,
`NUMDEVICEACCESSSTATUS` }

- enum `GevCCPEnum` {
`GevCCP_EnumEntry_GevCCP_OpenAccess`,
`GevCCP_EnumEntry_GevCCP_ExclusiveAccess`,
`GevCCP_EnumEntry_GevCCP_ControlAccess`,
`NUMGEVCCP` }
- enum `GUIXMLLocationEnum` {
`GUIXMLLocation_Device`,
`GUIXMLLocation_Host`,
`NUMGUIXMLLOCATION` }
- enum `GenICamXMLLocationEnum` {
`GenICamXMLLocation_Device`,
`GenICamXMLLocation_Host`,
`NUMGENICAMXMLLOCATION` }
- enum `DeviceEndiannessMechanismEnum` {
`DeviceEndiannessMechanism_Legacy`,
`DeviceEndiannessMechanism_Standard`,
`NUMDEVICEENDIANESSMECHANISM` }
- enum `DeviceCurrentSpeedEnum` {
`DeviceCurrentSpeed_UnknownSpeed`,
`DeviceCurrentSpeed_LowSpeed`,
`DeviceCurrentSpeed_FullSpeed`,
`DeviceCurrentSpeed_HighSpeed`,
`DeviceCurrentSpeed_SuperSpeed`,
`NUMDEVICECURRENTSPEED` }
- enum `InterfaceTypeEnum` {
`InterfaceType_GigEVision`,
`InterfaceType_CameraLink`,
`InterfaceType_CameraLinkHS`,
`InterfaceType_CoaXPress`,
`InterfaceType_USB3Vision`,
`InterfaceType_Custom`,
`NUMINTERFACETYPE` }
- enum `POEStatusEnum` {
`POEStatus_NotSupported`,
`POEStatus_PowerOff`,
`POEStatus_PowerOn`,
`NUMPOESTATUS` }
- enum `FilterDriverStatusEnum` {
`FilterDriverStatus_NotSupported`,
`FilterDriverStatus_Disabled`,
`FilterDriverStatus_Enabled`,
`NUMFILTERDRIVERSTATUS` }
- enum `TLTypeEnum` {
`TLType_GigEVision`,
`TLType_CameraLink`,
`TLType_CameraLinkHS`,
`TLType_CoaXPress`,
`TLType_USB3Vision`,
`TLType_Mixed`,
`TLType_Custom`,
`NUMTLTYPE` }

13.45.1 Detailed Description

13.45.2 Enumeration Type Documentation

13.45.2.1 DeviceAccessStatusEnum

enum [DeviceAccessStatusEnum](#)

< Gets the access status the transport layer Producer has on the device.

Enumerator

DeviceAccessStatus_Unknown	Not known to producer.
DeviceAccessStatus_ReadWrite	Full access
DeviceAccessStatus_ReadOnly	Read-only access
DeviceAccessStatus_NoAccess	Not available to connect
DeviceAccessStatus_Busy	The device is already opened by another entity
DeviceAccessStatus_OpenReadWrite	Open in Read/Write mode by this GenTL host
DeviceAccessStatus_OpenReadOnly	Open in Read access mode by this GenTL host
NUMDEVICEACCESSSTATUS	

13.45.2.2 DeviceCurrentSpeedEnum

enum [DeviceCurrentSpeedEnum](#)

< The USB Speed that the device is currently operating at.

Enumerator

DeviceCurrentSpeed_UnknownSpeed	Unknown-Speed.
DeviceCurrentSpeed_LowSpeed	Low-Speed.
DeviceCurrentSpeed_FullSpeed	Full-Speed.
DeviceCurrentSpeed_HighSpeed	High-Speed.
DeviceCurrentSpeed_SuperSpeed	Super-Speed.
NUMDEVICECURRENTSPEED	

13.45.2.3 DeviceEndiannessMechanismEnum

enum [DeviceEndiannessMechanismEnum](#)

< Identifies the endianness handling mode.

Enumerator

DeviceEndiannessMechanism_Legacy	Handling the device endianness according to GenICam Schema 1.0
DeviceEndiannessMechanism_Standard	Handling the device endianness according to GenICam Schema 1.1 and later
NUMDEVICEENDIANESSMECHANISM	

13.45.2.4 DeviceTypeEnum

enum [DeviceTypeEnum](#)

< Transport layer type of the device.

Enumerator

DeviceType_GigEVision	GigE Vision
DeviceType_CameraLink	Camera Link
DeviceType_CameraLinkHS	Camera Link High Speed
DeviceType_CoaXPRESS	CoaXPRESS
DeviceType_USB3Vision	USB3 Vision
DeviceType_Custom	Custom transport layer
NUMDEVICETYPE	

13.45.2.5 FilterDriverStatusEnum

enum [FilterDriverStatusEnum](#)

< Reports whether FLIR Light Weight Filter Driver is enabled or not.

Enumerator

FilterDriverStatus_NotSupported	Not Supported
FilterDriverStatus_Disabled	FLIR Light Weight Filter Driver is disabled
FilterDriverStatus_Enabled	FLIR Light Weight Filter Driver is enabled
NUMFILTERDRIVERSTATUS	

13.45.2.6 GenICamXMLLocationEnum

enum [GenICamXMLLocationEnum](#)

< Sets the location to load [GenICam](#) XML.

Enumerator

GenICamXMLLocation_Device	Load GenICam XML from device
GenICamXMLLocation_Host	Load GenICam XML from host
NUMGENICAMXMLLOCATION	

13.45.2.7 GevCCPEnum

enum [GevCCPEnum](#)

< Controls the device access privilege of an application.

Enumerator

GevCCP_EnumEntry_GevCCP_OpenAccess	Open access privilege.
GevCCP_EnumEntry_GevCCP_ExclusiveAccess	Exclusive access privilege.
GevCCP_EnumEntry_GevCCP_ControlAccess	Control access privilege.
NUMGEVCCP	

13.45.2.8 GUIXMLLocationEnum

enum [GUIXMLLocationEnum](#)

< Sets the location to load GUI XML.

Enumerator

GUIXMLLocation_Device	Load XML from device
GUIXMLLocation_Host	Load XML from host
NUMGUIXMLLOCATION	

13.45.2.9 InterfaceTypeEnum

enum [InterfaceTypeEnum](#)

< Transport layer type of the interface.

Enumerator

InterfaceType_GigEVision	GigE Vision
InterfaceType_CameraLink	Camera Link
InterfaceType_CameraLinkHS	Camera Link High Speed
InterfaceType_CoaXPress	CoaXPress
InterfaceType_USB3Vision	USB3 Vision
InterfaceType_Custom	Custom transport layer
NUMINTERFACETYPE	

13.45.2.10 POEStatusEnum

enum [POEStatusEnum](#)

< Reports and controls the interface's power over Ethernet status.

Enumerator

POEStatus_NotSupported	Not Supported
POEStatus_PowerOff	Power is Off
POEStatus_PowerOn	Power is On
NUMPOESTATUS	

13.45.2.11 StreamBufferCountModeEnum

enum [StreamBufferCountModeEnum](#)

< Controls access to setting the number of buffers used for the stream.

Enumerator

StreamBufferCountMode_Manual	The number of buffers used for the stream are set by the user.
StreamBufferCountMode_Auto	DEPRECATED. The number of buffers used for the stream is automatically calculated based on the device frame rate.
NUMSTREAMBUFFERCOUNTMODE	

13.45.2.12 StreamBufferHandlingModeEnum

enum [StreamBufferHandlingModeEnum](#)

< Available buffer handling modes of this data stream:

Enumerator

StreamBufferHandlingMode_OldestFirst	The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.
StreamBufferHandlingMode_OldestFirstOverwrite	The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires. If a new buffer arrives it will overwrite the existing buffer from the head of the queue (behaves like a circular buffer).

Enumerator

StreamBufferHandlingMode_NewestOnly	The application always gets the latest completed buffer (the newest one). If the Output Buffer Queue is empty, the application waits for a newly acquired buffer until the timeout expires. This buffer handling mode is typically used in a live display GUI where it is important that there is no lag between camera and display.
StreamBufferHandlingMode_NewestFirst	The application always gets the buffer from the tail of the output buffer queue (thus, the newest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.
NUMSTREAMBUFFERHANDLINGMODE	

13.45.2.13 StreamTypeEnum

enum [StreamTypeEnum](#)

The enum definitions for TL Device nodes from the transport layer .xml files.

< Stream type of the device.

Enumerator

StreamType_GigEVision	GigE Vision
StreamType_CameraLink	Camera Link
StreamType_CameraLinkHS	Camera Link High Speed
StreamType_CoaXPRESS	CoaXPRESS
StreamType_USB3Vision	USB3 Vision
StreamType_Custom	Custom transport layer
NUMSTREAMTYPE	

13.45.2.14 TLTypeEnum

enum [TLTypeEnum](#)

< Transport layer type of the GenTL Producer implementation.

Enumerator

TLType_GigEVision	GigE Vision
TLType_CameraLink	Camera Link
TLType_CameraLinkHS	Camera Link High Speed
TLType_CoaXPRESS	CoaXPRESS

Enumerator

TLType_USB3Vision	USB3 Vision
TLType_Mixed	Different Interface modules of the GenTL Producer are of different types
TLType_Custom	Custom transport layer
NUMTLTYPE	

13.46 TransportLayerDevice Class

Collaboration diagram for TransportLayerDevice Class:



Classes

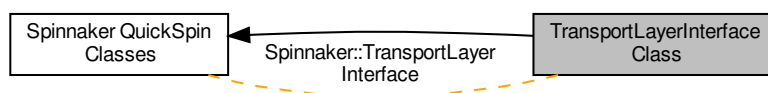
- class [TransportLayerDevice](#)

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

13.46.1 Detailed Description

13.47 TransportLayerInterface Class

Collaboration diagram for TransportLayerInterface Class:



Classes

- class [TransportLayerInterface](#)

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

13.47.1 Detailed Description

13.48 TransportLayerStream Class

Collaboration diagram for TransportLayerStream Class:



Classes

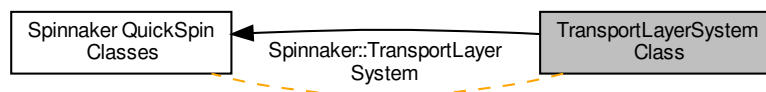
- class [TransportLayerStream](#)

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

13.48.1 Detailed Description

13.49 TransportLayerSystem Class

Collaboration diagram for TransportLayerSystem Class:



Classes

- class [TransportLayerSystem](#)

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

13.49.1 Detailed Description

13.50 Camera Base Interface Class

Collaboration diagram for Camera Base Interface Class:



Classes

- class [ICameraBase](#)

The interface file for base class for the camera object.

13.50.1 Detailed Description

13.51 IChunkData Class

Collaboration diagram for IChunkData Class:



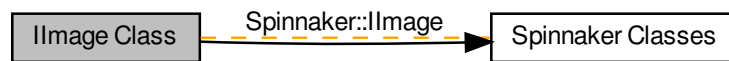
Classes

- class [IChunkData](#)
The [Interface](#) file for [ChunkData](#).

13.51.1 Detailed Description

13.52 IImage Class

Collaboration diagram for IImage Class:



Classes

- class [IImage](#)
The interface file for [Image](#).

13.52.1 Detailed Description

13.53 IImageStatistics Class

Collaboration diagram for IImageStatistics Class:



Classes

- class [IImageStatistics](#)

The interface file for image statistics.

13.53.1 Detailed Description

13.54 IInterface Class

Collaboration diagram for IInterface Class:



Classes

- class [IInterface](#)

The interface file for [IInterface](#).

13.54.1 Detailed Description

13.55 InterfaceList Class

Collaboration diagram for InterfaceList Class:



Classes

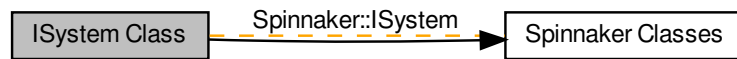
- class [InterfaceList](#)

The interface file for [InterfaceList](#) class.

13.55.1 Detailed Description

13.56 ISystem Class

Collaboration diagram for ISystem Class:



Classes

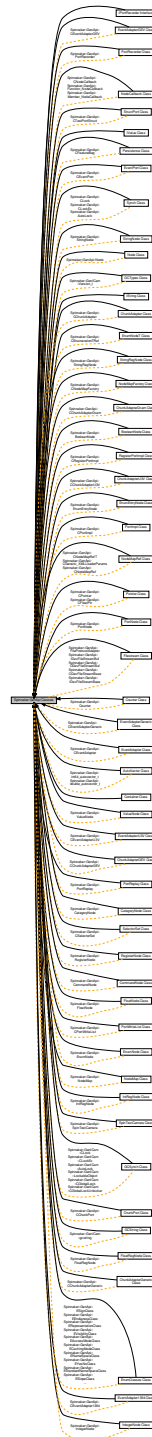
- class [ISystem](#)

The interface file for [System](#).

13.56.1 Detailed Description

13.57 Spinnaker GenApi Classes

Collaboration diagram for Spinnaker GenApi Classes:



Modules

- [AutoVector Class](#)
- [BooleanNode Class](#)

- [CategoryNode Class](#)
- [ChunkAdapter Class](#)
- [ChunkAdapterDcam Class](#)
- [ChunkAdapterGeneric Class](#)
- [ChunkAdapterGEV Class](#)
- [ChunkPort Class](#)
- [CommandNode Class](#)
- [Container Class](#)
- [Counter Class](#)
- [EnumClasses Class](#)
- [EnumEntryNode Class](#)
- [EnumNode Class](#)
- [EnumNodeT Class](#)
- [EventAdapter Class](#)
- [EventAdapter1394 Class](#)
- [EventAdapterGeneric Class](#)
- [EventAdapterGEV Class](#)
- [EventAdapterU3V Class](#)
- [EventPort Class](#)
- [Filestream Class](#)
- [FloatNode Class](#)
- [FloatRegNode Class](#)
- [GCString Class](#)
- [GCSynch Class](#)
- [GCTypes Class](#)
- [IntegerNode Class](#)
- [IntRegNode Class](#)
- [IString Class](#)
- [IValue Class](#)
- [Node Class](#)
- [NodeCallback Class](#)
- [NodeMap Class](#)
- [NodeMapFactory Class](#)
- [NodeMapRef Class](#)
- [Persistence Class](#)
- [Pointer Class](#)
- [PortImpl Class](#)
- [PortNode Class](#)
- [PortRecorder Class](#)
- [PortReplay Class](#)
- [PortWriteList Class](#)
- [RegisterNode Class](#)
- [RegisterPortImpl Class](#)
- [SelectorSet Class](#)
- [SpinTestCamera Class](#)
- [StringNode Class](#)
- [StringRegNode Class](#)
- [StructPort Class](#)
- [Synch Class](#)
- [ValueNode Class](#)
- [ChunkAdapterU3V Class](#)
- [IPortRecorder Interface](#)

Classes

- class [int64_autovector_t](#)
Vector of integers with reference counting.
- class [double_autovector_t](#)
Vector of doubles with reference counting.
- class [BooleanNode](#)
Interface for string properties.
- class [CategoryNode](#)
Interface for string properties.
- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.
- class [CChunkAdapterDcam](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterGeneric](#)
- class [CChunkAdapterGEV](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.
- class [CChunkPort](#)
Port attachable to a chunk in a buffer.
- class [CommandNode](#)
Interface for string properties.
- class [Counter](#)
Definition of a simple [Counter](#) class.
- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [EEndianessClass](#)
Holds conversion methods for the endianess enumeration.
- class [ERepresentationClass](#)
Holds conversion methods for the representation enumeration.
- class [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- class [EAccessModeClass](#)
Holds conversion methods for the access mode enumeration.
- class [ECachingModeClass](#)
Holds conversion methods for the caching mode enumeration.
- class [ENamespaceClass](#)
Holds conversion methods for the namespace enumeration.
- class [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [ESlopeClass](#)
Holds conversion methods for the converter formulas.
- class [EDisplayNotationClass](#)
Holds conversion methods for the notation type of floats.
- class [EInputDirectionClass](#)
Holds conversion methods for the notation type of floats.
- class [EGenApiSchemaVersionClass](#)
helper class converting EGenApiSchemaVersion from and to string

- class [EnumEntryNode](#)
Interface for string properties.
- class [EnumNode](#)
Interface for string properties.
- class [CEnumerationTRef< EnumT >](#)
Interface for string properties.
- class [CEventAdapter](#)
Delivers Events to ports.
- class [CEventAdapter1394](#)
Distribute the events to the node map.
- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.
- class [CEventAdapterGEV](#)
Connects a GigE Event to a node map.
- class [CEventAdapterU3V](#)
Connects a U3V Event to a node map.
- class [CEventPort](#)
Port attachable to an event.
- class [FileProtocolAdapter](#)
Adapter between the std::iostreambuf and the SFNC Features representing the device file system.
- class [IDevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBase< CharType, Traits >](#)
- class [IDevFileStreamBase< CharType, Traits >](#)
- class [FloatNode](#)
Interface for string properties.
- class [FloatRegNode](#)
Interface for string properties.
- class [gcstring](#)
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)
- class [LockableObject< Object >](#)
Instance-Lock for an object.
- class [CGlobalLock](#)
Named global lock which can be used over process boundaries.
- class [CGlobalLockUnlocker](#)
Unlocks the global lock object on destruction.
- struct [Version_t](#)
Version.
- class [IntegerNode](#)
Interface for string properties.
- class [IntRegNode](#)
Interface for string properties.
- class [Node](#)
class common to all nodes
- class [CNodeCallback](#)
callback body instance for INode pointers
- class [Function_NodeCallback< Function >](#)

- Container for a function pointer.*

 - class [Member_NodeCallback](#)< Client, Member >
- Container for a member function pointer.*

 - class [NodeMap](#)
- Smart pointer template for NodeMaps with create function.*

 - class [CNodeMapFactory](#)
- The node map factory is used for creating node maps from camera description files.*

 - class [CNodeMapRefT](#)< TCameraParams >
- Smartpointer template for NodeMaps with create function.*

 - class [CGeneric_XMLLoaderParams](#)
- Empty base class used by class [CNodeMapRef](#) as generic template argument.*

 - class [CNodeMapRef](#)
- Smartpointer for NodeMaps with create function.*

 - class [CFeatureBag](#)
- Bag holding streamable features of a nodetree.*

 - class [CPointer](#)< T, B >
- Encapsulates a [GenApi](#) pointer dealing with the dynamic_cast automatically.*

 - class [CFloatPtr](#)
- SmartPointer for IFloat interface pointer.*

 - class [CPortImpl](#)
- Standard implementation for a port.*

 - class [PortNode](#)
- [Interface](#) for value properties.*

 - class [PortRecorder](#)
- [Interface](#) for recording write commands on a port.*

 - class [PortReplay](#)
- [Interface](#) for replaying write commands on a port.*

 - class [CPortWriteList](#)
- Container holding a list of port write commands.*

 - class [RegisterNode](#)
- [Interface](#) for string properties.*

 - class [CRegisterPortImpl](#)
- Standard implementation for a port using a register based transport layer.*

 - class [CSelectorSet](#)
- The set of selectors selecting a given node.*

 - class [SpinTestCamera](#)
- [Interface](#) for string properties.*

 - class [StringNode](#)
- [Interface](#) for string properties.*

 - class [StringRegNode](#)
- [Interface](#) for string properties.*

 - class [CTestPortStruct](#)< CDataStruct >
- Implements a register spaces based on a C++ struct.*

 - class [CLock](#)
- A lock class.*

 - class [CLockEx](#)
- This class is for testing purposes only.*

 - class [AutoLock](#)
- [Interface](#) for value properties.*

 - class [ValueNode](#)

Typedefs

- typedef [Node](#) [CNodeRef](#)
- typedef [Node](#) [CSelectorRef](#)
- typedef [NodeMap](#) [CNodeMapRef](#)

Functions

- [SPINNAKER_API](#) [IDestroy](#) * [CastToIDestroy](#) ([INodeMap](#) *pNodeMap)
makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)
- template<class [TCameraParams](#) >
void [_LoadXMLFromFile](#) (const [GenICam::gcstring](#) &FileName)
- template<class [TCameraParams](#) >
void [_LoadXMLFromZIPFile](#) (const [GenICam::gcstring](#) &ZipFileName)
- template<class [TCameraParams](#) >
void [_LoadXMLFromFileInject](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectFileName)
- template<class [TCameraParams](#) >
void [_LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)
- template<class [TCameraParams](#) >
void [_LoadXMLFromZIPData](#) (const void *zipData, size_t zipSize)
- template<class [TCameraParams](#) >
void [_LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLData, const [GenICam::gcstring](#) &InjectXMLData)
- template<class [TCameraParams](#) >
void [_GetSupportedSchemaVersions](#) ([GenICam::gcstring_vector](#) &SchemaVersions)
- template<class [TCameraParams](#) >
[GenICam::gcstring](#) [_GetDeviceName](#) ()
- template<class [TCameraParams](#) >
void [_Poll](#) (int64_t ElapsedTime)
- template<class [TCameraParams](#) >
void [_GetNodes](#) ([NodeList_t](#) &Nodes)
- template<class [TCameraParams](#) >
[INode](#) * [_GetNode](#) (const [GenICam::gcstring](#) &key)
- template<class [TCameraParams](#) >
void [_InvalidateNodes](#) ()
- template<class [TCameraParams](#) >
bool [_Connect](#) ([IPort](#) *pPort, const [GenICam::gcstring](#) &PortName)
- template<class [TCameraParams](#) >
bool [_Connect](#) ([IPort](#) *pPort)
- template<class [TCameraParams](#) >
bool [_ClearXMLCache](#) ()
- [SPINNAKER_API](#) std::istream & [EatComments](#) (std::istream &is)
Helper function ignoring lines starting with comment character '#'.
- [SPINNAKER_API](#) std::istream & [operator>>](#) (std::istream &is, [CFeatureBag](#) &FeatureBag)
Reads in persistent data from a stream.
- [SPINNAKER_API](#) std::ostream & [operator<<](#) (std::ostream &os, const [CFeatureBag](#) &FeatureBag)
writes out persistent data to a stream
- [CNodeMapRefT](#) (const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- [CNodeMapRefT](#) ([INodeMap](#) *pNodeMap, const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- [CNodeMapRefT](#) (const [CNodeMapRefT](#) &Them)
Copy constructor.

- `CNodeMapRefT` & `operator=` (`INodeMap` *pNodeMap)
Assignment of an INodeMap.*
- `CNodeMapRefT` & `operator=` (const `CNodeMapRefT` &Them)
Assignment.
- virtual `~CNodeMapRefT` ()
Destructor.
- void `_Destroy` ()
Destroys the node map.

13.57.1 Detailed Description

13.57.2 Typedef Documentation

13.57.2.1 CNodeMapRef

```
typedef NodeMap CNodeMapRef
```

13.57.2.2 CNodeRef

```
typedef Node CNodeRef
```

13.57.2.3 CSelectorRef

```
typedef Node CSelectorRef
```

13.57.3 Function Documentation

13.57.3.1 _ClearXMLCache()

```
bool Spinnaker::GenApi::_ClearXMLCache ( ) [inline]
```

13.57.3.2 `_Connect()` [1/2]

```
bool Spinnaker::GenApi::_Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) [inline]
```

13.57.3.3 `_Connect()` [2/2]

```
bool Spinnaker::GenApi::_Connect (
    IPort * pPort ) [inline]
```

13.57.3.4 `_Destroy()`

```
void _Destroy ( ) [inline]
```

Destroys the node map.

13.57.3.5 `_GetDeviceName()`

```
GenICam::gcstring Spinnaker::GenApi::_GetDeviceName ( ) [inline]
```

13.57.3.6 `_GetNode()`

```
INode* Spinnaker::GenApi::_GetNode (
    const GenICam::gcstring & key ) [inline]
```

13.57.3.7 `_GetNodes()`

```
void Spinnaker::GenApi::_GetNodes (
    NodeList_t & Nodes ) [inline]
```

13.57.3.8 `_GetSupportedSchemaVersions()`

```
void Spinnaker::GenApi::_GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) [inline]
```

13.57.3.9 _InvalidateNodes()

```
void Spinnaker::GenApi::_InvalidateNodes ( ) [inline]
```

13.57.3.10 _LoadXMLFromFile()

```
void Spinnaker::GenApi::_LoadXMLFromFile (
    const GenICam::gcstring & FileName ) [inline]
```

13.57.3.11 _LoadXMLFromFileInject()

```
void Spinnaker::GenApi::_LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName ) [inline]
```

13.57.3.12 _LoadXMLFromString()

```
void Spinnaker::GenApi::_LoadXMLFromString (
    const GenICam::gcstring & XMLData ) [inline]
```

13.57.3.13 _LoadXMLFromStringInject()

```
void Spinnaker::GenApi::_LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData,
    const GenICam::gcstring & InjectXMLData ) [inline]
```

13.57.3.14 _LoadXMLFromZIPData()

```
void Spinnaker::GenApi::_LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize ) [inline]
```

13.57.3.15 _LoadXMLFromZIPFile()

```
void Spinnaker::GenApi::_LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName ) [inline]
```

13.57.3.16 _Poll()

```
void Spinnaker::GenApi::_Poll (
    int64_t ElapsedTime ) [inline]
```

13.57.3.17 CastToIDestroy()

```
SPINNAKER_API IDestroy* Spinnaker::GenApi::CastToIDestroy (
    INodeMap * pNodeMap )
```

makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)

13.57.3.18 CNodeMapRefT() [1/3]

```
CNodeMapRefT (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

13.57.3.19 CNodeMapRefT() [2/3]

```
CNodeMapRefT (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

13.57.3.20 CNodeMapRefT() [3/3]

```
CNodeMapRefT (
    const CNodeMapRefT< TCameraParams > & Them )
```

Copy constructor.

13.57.3.21 EatComments()

```
SPINNAKER_API std::istream& Spinnaker::GenApi::EatComments (
    std::istream & is )
```

Helper function ignoring lines starting with comment character '#'.

13.57.3.22 operator<<()

```
SPINNAKER_API std::ostream& Spinnaker::GenApi::operator<< (
    std::ostream & os,
    const CFeatureBag & FeatureBag )
```

writes out persistent data to a stream

13.57.3.23 operator=() [1/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    const CNodeMapRefT< TCameraParams > & Them )
```

Assignment.

13.57.3.24 operator=() [2/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    INodeMap * pNodeMap )
```

Assignment of an INodeMap*.

13.57.3.25 operator>>()

```
SPINNAKER_API std::istream& Spinnaker::GenApi::operator>> (
    std::istream & is,
    CFeatureBag & FeatureBag )
```

Reads in persistent data from a stream.

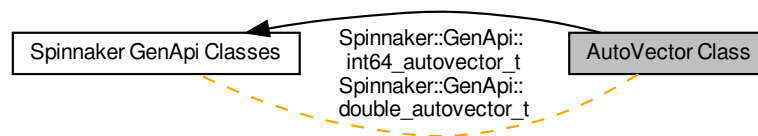
13.57.3.26 ~CNodeMapRefT()

```
~CNodeMapRefT ( ) [inline], [virtual]
```

Destructor.

13.58 AutoVector Class

Collaboration diagram for AutoVector Class:



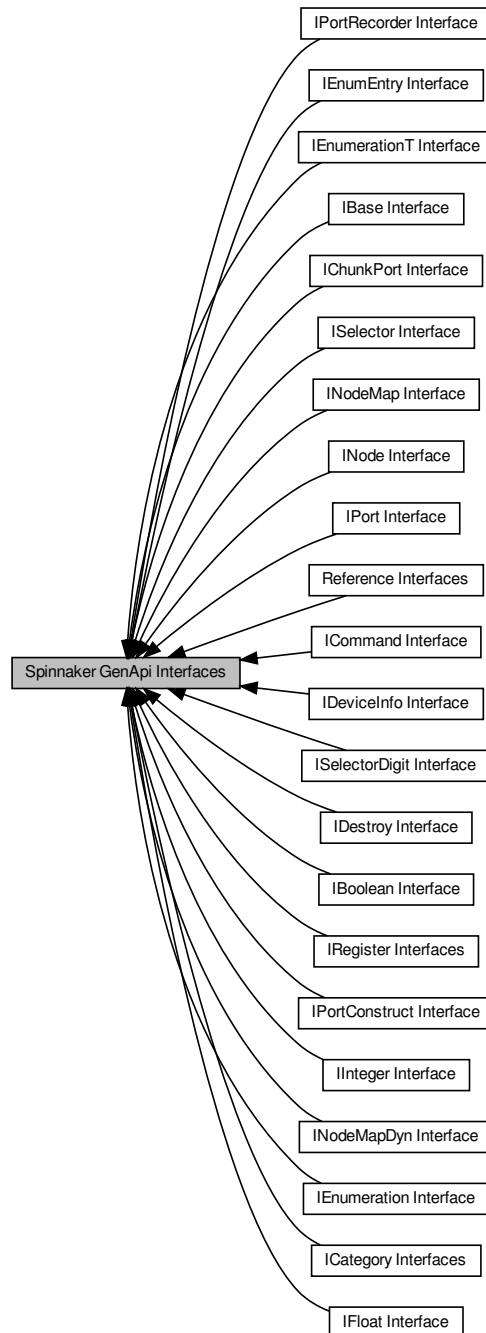
Classes

- class [int64_autovector_t](#)
Vector of integers with reference counting.
- class [double_autovector_t](#)
Vector of doubles with reference counting.

13.58.1 Detailed Description

13.59 Spinnaker GenApi Interfaces

Collaboration diagram for Spinnaker GenApi Interfaces:



Modules

- [IBase Interface](#)
- [IBoolean Interface](#)

- [ICategory Interfaces](#)
- [IChunkPort Interface](#)
- [ICommand Interface](#)
- [IDestroy Interface](#)
- [IDeviceInfo Interface](#)
- [IEnumEntry Interface](#)
- [IEnumeration Interface](#)
- [IEnumerationT Interface](#)
- [IFloat Interface](#)
- [IInteger Interface](#)
- [INode Interface](#)
- [INodeMap Interface](#)
- [INodeMapDyn Interface](#)
- [IPort Interface](#)
- [IPortConstruct Interface](#)
- [IPortRecorder Interface](#)
- [IRegister Interfaces](#)
- [ISelector Interface](#)
- [ISelectorDigit Interface](#)
- [Reference Interfaces](#)

Typedefs

- typedef node_vector [NodeList_t](#)
a list of node references
- typedef intptr_t [CallbackHandleType](#)
the callback handle for nodes

13.59.1 Detailed Description

13.59.2 Typedef Documentation

13.59.2.1 CallbackHandleType

```
typedef intptr_t CallbackHandleType
```

the callback handle for nodes

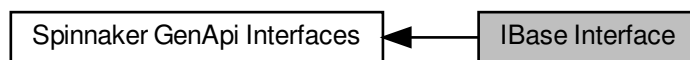
13.59.2.2 NodeList_t

```
typedef node_vector NodeList\_t
```

a list of node references

13.60 IBase Interface

Collaboration diagram for IBase Interface:



Variables

- [interface SPINNAKER_API_ABSTRACT IBase](#)
Base interface common to all nodes.

13.60.1 Detailed Description

13.60.2 Variable Documentation

13.60.2.1 IBase

```
interface SPINNAKER_API_ABSTRACT IBase
```

Initial value:

```
{  
    virtual EAccessMode GetAccessMode() const = 0  
}
```

Base interface common to all nodes.

13.61 BooleanNode Class

Collaboration diagram for BooleanNode Class:



Classes

- class [BooleanNode](#)
Interface for string properties.

Typedefs

- typedef [BooleanNode](#) [CBooleanRef](#)

13.61.1 Detailed Description

13.61.2 Typedef Documentation

13.61.2.1 CBooleanRef

```
typedef BooleanNode CBooleanRef
```

13.62 CategoryNode Class

Collaboration diagram for CategoryNode Class:



Classes

- class [CategoryNode](#)
Interface for string properties.

Typedefs

- typedef [CategoryNode](#) [CCategoryRef](#)

13.62.1 Detailed Description

13.62.2 Typedef Documentation

13.62.2.1 CCategoryRef

typedef [CategoryNode](#) [CCategoryRef](#)

13.63 ChunkAdapter Class

Collaboration diagram for ChunkAdapter Class:



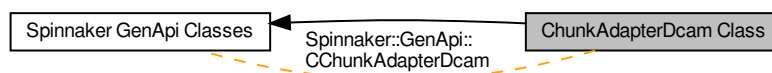
Classes

- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.

13.63.1 Detailed Description

13.64 ChunkAdapterDcam Class

Collaboration diagram for ChunkAdapterDcam Class:



Classes

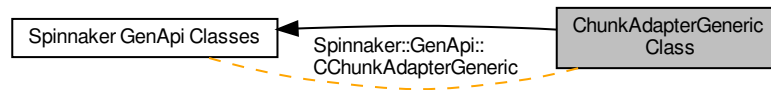
- class [CChunkAdapterDcam](#)

Connects a chunked DCAM buffer to a node map.

13.64.1 Detailed Description

13.65 ChunkAdapterGeneric Class

Collaboration diagram for ChunkAdapterGeneric Class:



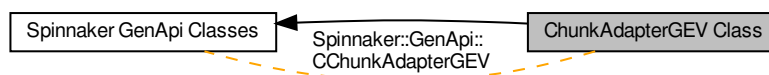
Classes

- class [CChunkAdapterGeneric](#)

13.65.1 Detailed Description

13.66 ChunkAdapterGEV Class

Collaboration diagram for ChunkAdapterGEV Class:



Classes

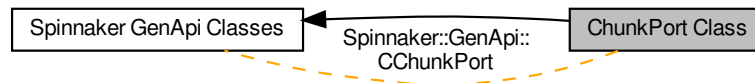
- class [CChunkAdapterGEV](#)

Connects a chunked DCAM buffer to a node map.

13.66.1 Detailed Description

13.67 ChunkPort Class

Collaboration diagram for ChunkPort Class:



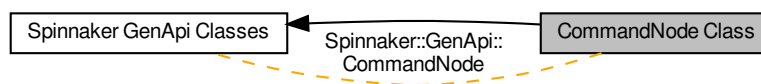
Classes

- class [CChunkPort](#)
Port attachable to a chunk in a buffer.

13.67.1 Detailed Description

13.68 CommandNode Class

Collaboration diagram for CommandNode Class:



Classes

- class [CommandNode](#)
Interface for string properties.

Typedefs

- typedef [CommandNode](#) [CCommandRef](#)

13.68.1 Detailed Description

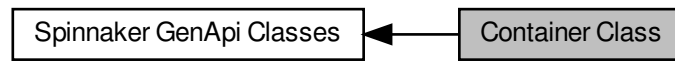
13.68.2 Typedef Documentation

13.68.2.1 CCommandRef

typedef [CommandNode](#) [CCommandRef](#)

13.69 Container Class

Collaboration diagram for Container Class:



13.70 Counter Class

Collaboration diagram for Counter Class:



Classes

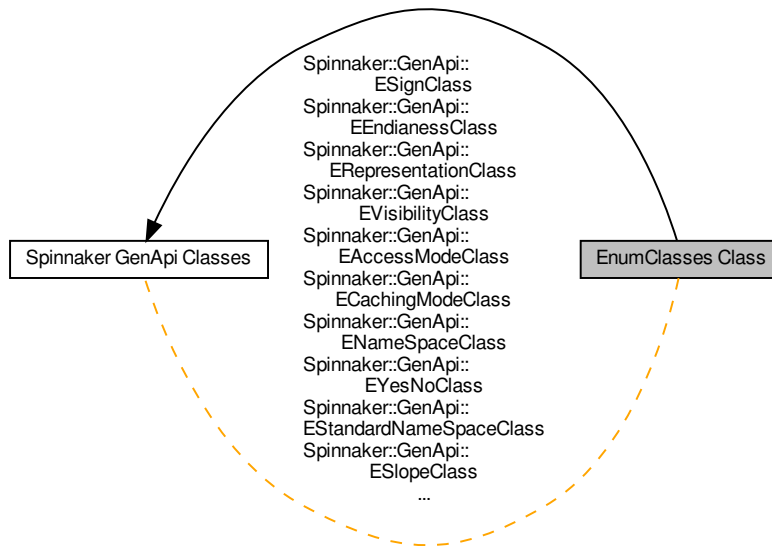
- class [Counter](#)

Definition of a simple [Counter](#) class.

13.70.1 Detailed Description

13.71 EnumClasses Class

Collaboration diagram for EnumClasses Class:



Classes

- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [EEndiannessClass](#)
Holds conversion methods for the endianness enumeration.
- class [ERepresentationClass](#)
Holds conversion methods for the representation enumeration.
- class [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- class [EAccessModeClass](#)
Holds conversion methods for the access mode enumeration.
- class [ECachingModeClass](#)
Holds conversion methods for the caching mode enumeration.
- class [ENamespaceClass](#)
Holds conversion methods for the namespace enumeration.
- class [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [ESlopeClass](#)
Holds conversion methods for the converter formulas.
- class [EDisplayNotationClass](#)
Holds conversion methods for the notation type of floats.
- class [EInputDirectionClass](#)
Holds conversion methods for the notation type of floats.
- class [EGenApiSchemaVersionClass](#)
helper class converting EGenApiSchemaVersion from and to string

13.71.1 Detailed Description

13.72 EnumEntryNode Class

Collaboration diagram for EnumEntryNode Class:



Classes

- class [EnumEntryNode](#)
Interface for string properties.

Typedefs

- typedef [EnumEntryNode](#) [CEnumEntryRef](#)

13.72.1 Detailed Description

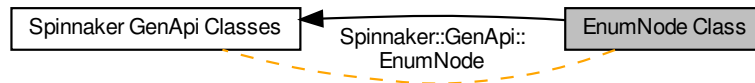
13.72.2 Typedef Documentation

13.72.2.1 CEnumEntryRef

```
typedef EnumEntryNode CEnumEntryRef
```

13.73 EnumNode Class

Collaboration diagram for EnumNode Class:



Classes

- class [EnumNode](#)
Interface for string properties.

Typedefs

- typedef [EnumNode](#) [CEnumerationRef](#)

13.73.1 Detailed Description

13.73.2 Typedef Documentation

13.73.2.1 CEnumerationRef

```
typedef EnumNode CEnumerationRef
```

13.74 EnumNodeT Class

Collaboration diagram for EnumNodeT Class:



Classes

- class [CEnumerationTRef< EnumT >](#)
Interface for string properties.

13.74.1 Detailed Description

13.75 EventAdapter Class

Collaboration diagram for EventAdapter Class:



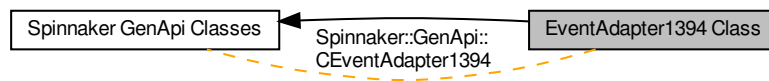
Classes

- class [CEventAdapter](#)
Delivers Events to ports.

13.75.1 Detailed Description

13.76 EventAdapter1394 Class

Collaboration diagram for EventAdapter1394 Class:



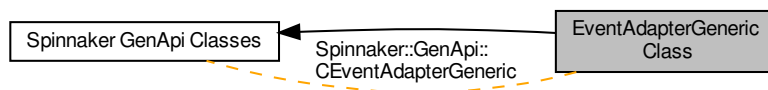
Classes

- class [CEventAdapter1394](#)
Distribute the events to the node map.

13.76.1 Detailed Description

13.77 EventAdapterGeneric Class

Collaboration diagram for EventAdapterGeneric Class:



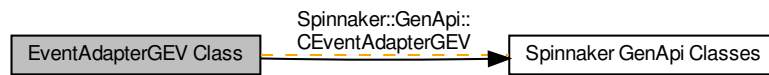
Classes

- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.

13.77.1 Detailed Description

13.78 EventAdapterGEV Class

Collaboration diagram for EventAdapterGEV Class:



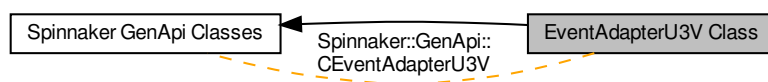
Classes

- class [CEventAdapterGEV](#)
Connects a GigE Event to a node map.

13.78.1 Detailed Description

13.79 EventAdapterU3V Class

Collaboration diagram for EventAdapterU3V Class:



Classes

- class [CEventAdapterU3V](#)
Connects a U3V Event to a node map.

13.79.1 Detailed Description

13.80 EventPort Class

Collaboration diagram for EventPort Class:



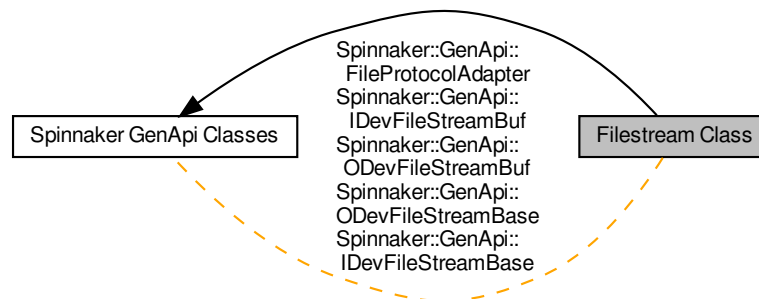
Classes

- class [CEventPort](#)
Port attachable to an event.

13.80.1 Detailed Description

13.81 Filestream Class

Collaboration diagram for Filestream Class:



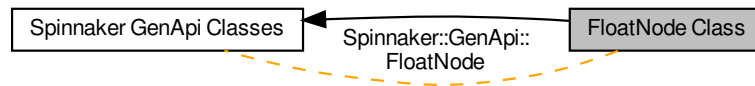
Classes

- class [FileProtocolAdapter](#)
Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.
- class [IDevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBase< CharType, Traits >](#)
- class [IDevFileStreamBase< CharType, Traits >](#)

13.81.1 Detailed Description

13.82 FloatNode Class

Collaboration diagram for FloatNode Class:



Classes

- class [FloatNode](#)
Interface for string properties.

Typedefs

- typedef [FloatNode](#) [CFloatRef](#)

13.82.1 Detailed Description

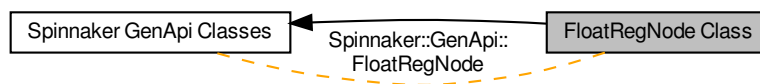
13.82.2 Typedef Documentation

13.82.2.1 CFloatRef

```
typedef FloatNode CFloatRef
```

13.83 FloatRegNode Class

Collaboration diagram for FloatRegNode Class:



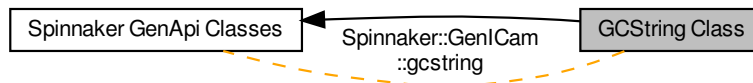
Classes

- class [FloatRegNode](#)
Interface for string properties.

13.83.1 Detailed Description

13.84 GCString Class

Collaboration diagram for GCString Class:



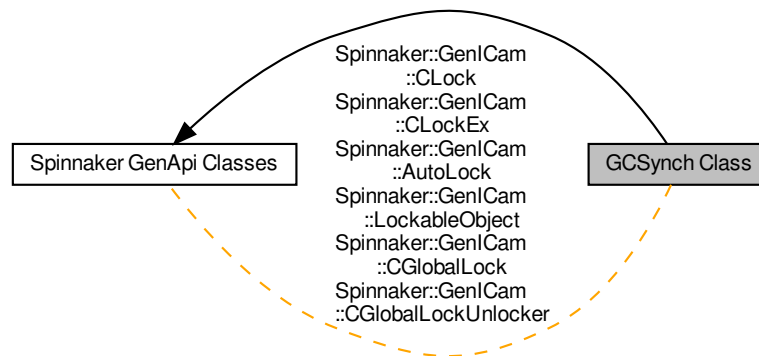
Classes

- class [gcstring](#)

13.84.1 Detailed Description

13.85 GCSynch Class

Collaboration diagram for GCSynch Class:



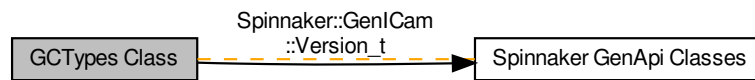
Classes

- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)
- class [LockableObject< Object >](#)
Instance-Lock for an object.
- class [CGlobalLock](#)
Named global lock which can be used over process boundaries.
- class [CGlobalLockUnlocker](#)
Unlocks the global lock object on destruction.

13.85.1 Detailed Description

13.86 GTypes Class

Collaboration diagram for GTypes Class:



Classes

- struct [Version_t](#)
Version.

Typedefs

- typedef float [float32_t](#)
32 bit floating point
- typedef double [float64_t](#)
64 bit floating point

13.86.1 Detailed Description

13.86.2 Typedef Documentation

13.86.2.1 float32_t

```
typedef float float32\_t
```

32 bit floating point

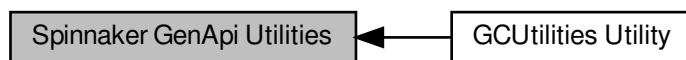
13.86.2.2 float64_t

```
typedef double float64\_t
```

64 bit floating point

13.87 Spinnaker GenApi Utilities

Collaboration diagram for Spinnaker GenApi Utilities:



Modules

- [GCUtilities Utility](#)

13.87.1 Detailed Description

13.88 GCUtilities Utility

Collaboration diagram for GCUtilities Utility:



Functions

- `template<typename Td , typename Ts >`
`Td INTEGRAL_CAST2 (Ts s)`
This verifies at runtime if there was no loss of data if an type Ts (e.g.
- `template<typename T >`
`T INTEGRAL_CAST (int64_t ll)`
This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.
- `SPINNAKER_API bool DoesEnvironmentVariableExist (const Spinnaker::GenICam::gcstring &VariableName)`
Returns true if an environment variable exists.
- `SPINNAKER_API gcstring GetValueOfEnvironmentVariable (const gcstring &VariableName)`
Retrieve the value of an environment variable.
- `SPINNAKER_API bool GetValueOfEnvironmentVariable (const gcstring &VariableName, gcstring &VariableContent)`
Retrieve the value of an environment variable.
- `SPINNAKER_API gcstring UrlEncode (const gcstring &Input)`
Converts \ to / and replaces all unsafe characters by their xx equivalent.
- `SPINNAKER_API gcstring UrlDecode (const gcstring &Input)`
Replaces xx escapes by their char equivalent.
- `SPINNAKER_API void ReplaceEnvironmentVariables (gcstring &Buffer, bool ReplaceBlankBy20=false)`
Replaces in a string and replace ' ' with %20.
- `SPINNAKER_API gcstring GetGenICamCacheFolder (void)`
Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling SetGenICamCacheFolder().
- `SPINNAKER_API gcstring GetGenICamLogConfig (void)`
Retrieve the path of the GenICam logging properties file.
- `SPINNAKER_API gcstring GetGenICamCLProtocolFolder (void)`
Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling SetGenICamCLProtocolFolder().
- `SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)`
Stores the path of the GenICam cache folder.
- `SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)`
Stores the path of the GenICam logging properties file.
- `SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)`
Stores the path of the CLProtocol folder.
- `SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")`

splits str input string into a list of tokens using the delimiter

- **SPINNAKER_API** void **GetFiles** (const **gcstring** &FileTemplate, **gcstring_vector** &FileNames, const bool DirectoriesOnly=false)

Gets a list of files or directories matching a given FileTemplate.

- **SPINNAKER_API** **gcstring** **GetModulePathFromFunction** (void *pFunction)

Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

13.88.1 Detailed Description

13.88.2 Function Documentation

13.88.2.1 DoesEnvironmentVariableExist()

```
SPINNAKER_API bool Spinnaker::GenICam::DoesEnvironmentVariableExist (
    const Spinnaker::GenICam::gcstring & VariableName )
```

Returns true if an environment variable exists.

13.88.2.2 GetFiles()

```
SPINNAKER_API void Spinnaker::GenICam::GetFiles (
    const gcstring & FileTemplate,
    gcstring_vector & FileNames,
    const bool DirectoriesOnly = false )
```

Gets a list of files or directories matching a given FileTemplate.

Parameters

<i>FileNames</i>	The file template. Can contain environment variables.
<i>DirectoriesOnly</i>	A list of files matching the file template

13.88.2.3 GetGenICamCacheFolder()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamCacheFolder (
    void )
```

Retrieve the path of the [GenICam](#) cache folder The path to the cache folder can be stored by calling [SetGenICamCacheFolder\(\)](#).

If [GetGenICamCacheFolder\(\)](#) is called before [SetGenICamCacheFolder\(\)](#), it will return the value of environment variable `GENICAM_CACHE_Vx_y`. If this environment variable does not exist, an exception will be thrown.

13.88.2.4 GetGenICamCLProtocolFolder()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamCLProtocolFolder (
    void )
```

Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling [SetGenICamCLProtocolFolder\(\)](#).

If [GetGenICamCLProtocolFolder\(\)](#) is called before [SetGenICamCLProtocolFolder\(\)](#), it will return the value of environment variable `GENICAM_CLPROTOCOL`. If this environment variable does not exist, an exception will be thrown.

13.88.2.5 GetGenICamLogConfig()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamLogConfig (
    void )
```

Retrieve the path of the [GenICam](#) logging properties file.

The path to the logging properties file can be stored by calling [SetGenICamLogConfig\(\)](#). If [GetGenICamLogConfig\(\)](#) is called before [SetGenICamLogConfig\(\)](#), it will return the value of environment variable `GENICAM_LOG_CONFIG_Vx_y`. If this environment variable does not exist, an exception will be thrown.

13.88.2.6 GetModulePathFromFunction()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetModulePathFromFunction (
    void * pFunction )
```

Gets the full path to the module (DLL/SO) containing the given *pFunction*; empty string if not found.

true = only subdirectories (ex . and ..) are retrieved; false = only files are retrieved

13.88.2.7 GetValueOfEnvironmentVariable() [1/2]

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetValueOfEnvironmentVariable (
    const gcstring & VariableName )
```

Retrieve the value of an environment variable.

Exceptions

<i>runtime_exception</i>	if not found

13.88.2.8 GetValueOfEnvironmentVariable() [2/2]

```
SPINNAKER_API bool Spinnaker::GenICam::GetValueOfEnvironmentVariable (
    const gcstring & VariableName,
    gcstring & VariableContent )
```

Retrieve the value of an environment variable.

Returns

true if environment variable was found, otherwise false

13.88.2.9 INTEGRAL_CAST()

```
T Spinnaker::GenICam::INTEGRAL_CAST (
    int64_t II ) [inline]
```

This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.

int32_t)

13.88.2.10 INTEGRAL_CAST2()

```
Td Spinnaker::GenICam::INTEGRAL_CAST2 (
    Ts s ) [inline]
```

This verifies at runtime if there was no loss of data if an type Ts (e.g.

int64t) was downcast to type Td (e.g. int32_t)

13.88.2.11 ReplaceEnvironmentVariables()

```
SPINNAKER_API void Spinnaker::GenICam::ReplaceEnvironmentVariables (
    gcstring & Buffer,
    bool ReplaceBlankBy20 = false )
```

Replaces in a string and replace ' ' with %20.

13.88.2.12 SetGenICamCacheFolder()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamCacheFolder (
    const gcstring & path )
```

Stores the path of the **GenICam** cache folder.

13.88.2.13 SetGenICamCLProtocolFolder()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamCLProtocolFolder (
    const gcstring & path )
```

Stores the path of the CLProtocol folder.

13.88.2.14 SetGenICamLogConfig()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamLogConfig (
    const gcstring & path )
```

Stores the path of the **GenICam** logging properties file.

13.88.2.15 Tokenize()

```
SPINNAKER_API void Spinnaker::GenICam::Tokenize (
    const gcstring & str,
    gcstring_vector & tokens,
    const gcstring & delimiters = " " )
```

splits str input string into a list of tokens using the delimiter

Parameters

<i>str</i>	string to be split
<i>tokens</i>	result of the splitting operation
<i>delimiters</i>	delimiters for the splitting

13.88.2.16 UrlDecode()

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlDecode (
    const gcstring & Input )
```

Replaces xx escapes by their char equivalent.

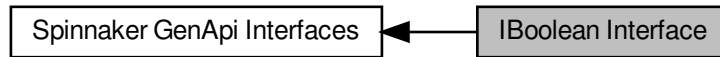
13.88.2.17 UrlEncode()

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlEncode (
    const gcstring & Input )
```

Converts \ to / and replaces all unsafe characters by their xx equivalent.

13.89 IBoolean Interface

Collaboration diagram for IBoolean Interface:



Functions

- virtual void `operator=` (bool Value)
Set node value.
- virtual bool `GetValue` (bool `Verify`=false, bool `IgnoreCache`=false) const =0
Get node value.
- virtual bool `operator()` () const
Get node value.

Variables

- `interface SPINNAKER_API_ABSTRACT IBoolean`
Interface for Boolean properties.
- `interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0`

13.89.1 Detailed Description

13.89.2 Function Documentation

13.89.2.1 `GetValue()`

```

GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) const [pure virtual]
  
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

13.89.2.2 operator()

```
GenICam::gcstring operator() ( ) const [virtual]
```

Get node value.

Execute the command.

13.89.2.3 operator=()

```
virtual void Spinnaker::GenApi::operator= (
    bool Value ) [virtual]
```

Set node value.

13.89.3 Variable Documentation

13.89.3.1 IBoolean

```
interface SPINNAKER_API_ABSTRACT IBoolean
```

[Interface](#) for Boolean properties.

13.89.3.2 Verify

```
interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0
```


13.90 ICategory Interfaces

Collaboration diagram for ICategory Interfaces:



Variables

- [interface SPINNAKER_API_ABSTRACT ICategory](#)
Gives access to a category node.

13.90.1 Detailed Description

13.90.2 Variable Documentation

13.90.2.1 ICategory

```
interface SPINNAKER_API_ABSTRACT ICategory
```

Gives access to a category node.

13.91 IChunkPort Interface

Collaboration diagram for IChunkPort Interface:



Macros

- `#define` [CHUNK_BASE_ADDRESS_REGISTER](#) [GC_INT64_MAX](#)
Address of a `int64_t` pseudo register containing the base address of the chunk (`MAX_INT64`)
- `#define` [CHUNK_BASE_ADDRESS_REGISTER_LEN](#) 8
Length of the `CHUNK_BASE_ADDRESS_REGISTER` pseudo register.
- `#define` [CHUNK_LENGTH_REGISTER](#) ([GC_INT64_MAX](#) - 15)
Address of a `int64_t` pseudo register containing the length of the chunk.
- `#define` [CHUNK_LENGTH_REGISTER_LEN](#) 8
Length of the `CHUNK_LENGTH_REGISTER` pseudo register.

Functions

- virtual [EYesNo CacheChunkData](#) () const =0
Indicates if the chunk a adapter must hold a cached version of the chunk data.

Variables

- [interface SPINNAKER_API_ABSTRACT IChunkPort](#)
Interface for ports attached to a chunk.

13.91.1 Detailed Description

13.91.2 Macro Definition Documentation

13.91.2.1 `CHUNK_BASE_ADDRESS_REGISTER`

```
#define CHUNK_BASE_ADDRESS_REGISTER GC_INT64_MAX
```

Address of a `int64_t` pseudo register containing the base address of the chunk (`MAX_INT64`)

13.91.2.2 CHUNK_BASE_ADDRESS_REGISTER_LEN

```
#define CHUNK_BASE_ADDRESS_REGISTER_LEN 8
```

Length of the CHUNK_BASE_ADDRESS_REGISTER pseudo register.

13.91.2.3 CHUNK_LENGTH_REGISTER

```
#define CHUNK_LENGTH_REGISTER (GC_INT64_MAX - 15)
```

Address of a int64_t pseudo register containing the length of the chunk.

13.91.2.4 CHUNK_LENGTH_REGISTER_LEN

```
#define CHUNK_LENGTH_REGISTER_LEN 8
```

Length of the CHUNK_LENGTH_REGISTER pseudo register.

13.91.3 Function Documentation

13.91.3.1 CacheChunkData()

```
virtual EYesNo Spinnaker::GenApi::CacheChunkData ( ) const [pure virtual]
```

Indicates if the chunk a adapter must hold a cached version of the chunk data.

13.91.4 Variable Documentation

13.91.4.1 IChunkPort

```
interface SPINNAKER_API_ABSTRACT IChunkPort
```

[Interface](#) for ports attached to a chunk.

13.92 ICommand Interface

Collaboration diagram for ICommand Interface:



Functions

- virtual bool [IsDone](#) (bool [Verify](#)=true)=0
Query whether the command is executed.

Variables

- [interface SPINNAKER_API_ABSTRACT ICommand](#)
Interface for command like properties.

13.92.1 Detailed Description

13.92.2 Function Documentation

13.92.2.1 IsDone()

```
virtual bool Spinnaker::GenApi::IsDone (
    bool Verify = true ) [pure virtual]
```

Query whether the command is executed.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked

Returns

True if the Execute command has finished; false otherwise

13.92.3 Variable Documentation

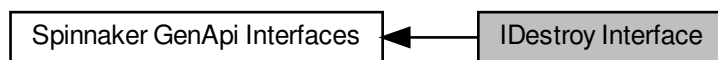
13.92.3.1 ICommand

```
interface SPINNAKER_API_ABSTRACT ICommand
```

Interface for command like properties.

13.93 IDestroy Interface

Collaboration diagram for IDestroy Interface:



Variables

- `interface SPINNAKER_API_ABSTRACT IDestroy`
Interface to destroy an object.

13.93.1 Detailed Description

13.93.2 Variable Documentation

13.93.2.1 IDestroy

```
interface SPINNAKER_API_ABSTRACT IDestroy
```

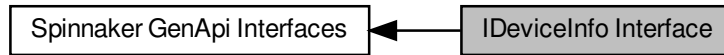
Initial value:

```
{  
    virtual void Destroy() = 0
```

Interface to destroy an object.

13.94 IDeviceInfo Interface

Collaboration diagram for IDeviceInfo Interface:



Functions

- virtual `GenICam::gcstring GetVendorName ()=0`
Get the vendor name.
- virtual `GenICam::gcstring GetToolTip ()=0`
Get tool tip.
- virtual `GenICam::gcstring GetStandardNameSpace ()=0`
Get the standard name space.
- virtual void `GetGenApiVersion (GenICam::Version_t &Version, uint16_t &Build)=0`
Get the version of the DLL's GenApi implementation.
- virtual void `GetSchemaVersion (GenICam::Version_t &Version)=0`
Get the schema version number.
- virtual void `GetDeviceVersion (GenICam::Version_t &Version)=0`
Get the version of the device description file.
- virtual `GenICam::gcstring GetProductGuid ()=0`
Get the Guid describing the product.
- virtual `GenICam::gcstring GetVersionGuid ()=0`
Get the Guid describing the product version.

Variables

- `interface SPINNAKER_API_ABSTRACT IDeviceInfo`
Interface to get information about the device (= nodemap)

13.94.1 Detailed Description

13.94.2 Function Documentation

13.94.2.1 GetDeviceVersion()

```
virtual void Spinnaker::GenApi::GetDeviceVersion (
    GenICam::Version_t & Version ) [pure virtual]
```

Get the version of the device description file.

13.94.2.2 GetGenApiVersion()

```
virtual void Spinnaker::GenApi::GetGenApiVersion (
    GenICam::Version_t & Version,
    uint16_t & Build ) [pure virtual]
```

Get the version of the DLL's [GenApi](#) implementation.

13.94.2.3 GetProductGuid()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetProductGuid ( ) [pure virtual]
```

Get the Guid describing the product.

13.94.2.4 GetSchemaVersion()

```
virtual void Spinnaker::GenApi::GetSchemaVersion (
    GenICam::Version_t & Version ) [pure virtual]
```

Get the schema version number.

13.94.2.5 GetStandardNameSpace()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetStandardNameSpace ( ) [pure virtual]
```

Get the standard name space.

13.94.2.6 GetToolTip()

```
GenICam::gcstring GetToolTip ( ) [pure virtual]
```

Get tool tip.

Get a short description of the node.

13.94.2.7 GetVendorName()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetVendorName ( ) [pure virtual]
```

Get the vendor name.

13.94.2.8 GetVersionGuid()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetVersionGuid ( ) [pure virtual]
```

Get the Guid describing the product version.

13.94.3 Variable Documentation

13.94.3.1 IDeviceInfo

```
interface SPINNAKER_API_ABSTRACT IDeviceInfo
```

Initial value:

```
{  
    virtual GenICam::gcstring GetModelName() = 0
```

[Interface](#) to get information about the device (= nodemap)

13.95 IEnumEntry Interface

Collaboration diagram for IEnumEntry Interface:



Functions

- virtual [GenICam::gcstring GetSymbolic](#) () const =0
Get symbolic enum value.
- virtual double [GetNumericValue](#) ()=0
Get double number associated with the entry.
- virtual bool [IsSelfClearing](#) ()=0
Indicates if the corresponding EnumEntry is self clearing.

Variables

- [interface SPINNAKER_API_ABSTRACT IEnumEntry](#)
Interface of single enum value.

13.95.1 Detailed Description

13.95.2 Function Documentation

13.95.2.1 GetNumericValue()

```
virtual double Spinnaker::GenApi::GetNumericValue ( ) [pure virtual]
```

Get double number associated with the entry.

13.95.2.2 GetSymbolic()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetSymbolic ( ) const [pure virtual]
```

Get symbolic enum value.

13.95.2.3 IsSelfClearing()

```
virtual bool Spinnaker::GenApi::IsSelfClearing ( ) [pure virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

13.95.3 Variable Documentation

13.95.3.1 IEnumEntry

```
interface SPINNAKER_API_ABSTRACT IEnumEntry
```

Interface of single enum value.

Maps of Enum Values to symbolic values

13.96 IEnumeration Interface

Collaboration diagram for IEnumeration Interface:



Functions

- virtual void [GetEntries](#) ([NodeList_t](#) &Entries)=0
Get list of entry nodes.
- virtual void [SetIntValue](#) (int64_t Value, bool [Verify](#)=true)=0
Set integer node value.
- virtual [GenICam::gcstring operator*](#) ()=0
Get string node value.
- virtual int64_t [GetIntValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
Get integer node value.
- virtual [IEnumEntry](#) * [GetEntryByName](#) (const [GenICam::gcstring](#) &Symbolic)=0
Get an entry node by name.
- virtual [IEnumEntry](#) * [GetEntry](#) (const int64_t IntValue)=0
Get an entry node by its IntValue.
- virtual [IEnumEntry](#) * [GetCurrentEntry](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
Get the current entry.

Variables

- [interface SPINNAKER_API_ABSTRACT IEnumeration](#)
Interface for enumeration properties.

13.96.1 Detailed Description

13.96.2 Function Documentation

13.96.2.1 GetCurrentEntry()

```

IEnumEntry * GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
  
```

Get the current entry.

13.96.2.2 GetEntries()

```
virtual void Spinnaker::GenApi::GetEntries (
    NodeList_t & Entries ) [pure virtual]
```

Get list of entry nodes.

13.96.2.3 GetEntry()

```
IEnumEntry * GetEntry (
    const int64_t IntValue ) [pure virtual]
```

Get an entry node by its IntValue.

13.96.2.4 GetEntryByName()

```
virtual IEnumEntry* Spinnaker::GenApi::GetEntryByName (
    const GenICam::gcstring & Symbolic ) [pure virtual]
```

Get an entry node by name.

13.96.2.5 GetIntValue()

```
virtual int64_t Spinnaker::GenApi::GetIntValue (
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
```

Get integer node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

13.96.2.6 operator*()

```
GenICam::gcstring operator* ( ) [pure virtual]
```

Get string node value.

Get node value.

13.96.2.7 SetIntValue()

```
virtual void Spinnaker::GenApi::SetIntValue (
    int64_t Value,
    bool Verify = true ) [pure virtual]
```

Set integer node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

13.96.3 Variable Documentation

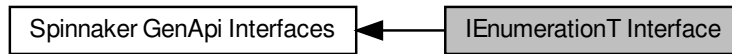
13.96.3.1 IEnumeration

```
interface SPINNAKER_API_ABSTRACT IEnumeration
```

[Interface](#) for enumeration properties.

13.97 IEnumerationT Interface

Collaboration diagram for IEnumerationT Interface:



Functions

- virtual [IEnumeration](#) & [operator=](#) (EnumT Value)=0
Set node value.
- virtual [IEnumEntry](#) * [GetEntry](#) (const EnumT Value)=0
returns the EnumEntry object belonging to the Value
- virtual [IEnumeration](#) & [operator=](#) (const [GenICam::gcstring](#) &ValueStr)=0
Set string node value.

Variables

- template<typename EnumT >
[interface SPINNAKER_API_ABSTRACT IEnumerationT](#)
Interface for enumeration properties.
- template<typename EnumT >
[interface SPINNAKER_API_ABSTRACT](#) virtual public [IEnumReference](#)
Interface to construct an enum reference.

13.97.1 Detailed Description

13.97.2 Function Documentation

13.97.2.1 GetEntry()

```
virtual IEnumEntry* Spinnaker::GenApi::GetEntry (
    const EnumT Value ) [pure virtual]
```

returns the EnumEntry object belonging to the Value

13.97.2.2 operator=() [1/2]

```
virtual IEnumeration& Spinnaker::GenApi::operator= (
    EnumT Value ) [pure virtual]
```

Set node value.

13.97.2.3 operator=() [2/2]

```
IString & operator= (
    const GenICam::gcstring & ValueStr ) [pure virtual]
```

Set string node value.

Set node value.

Note : the operator= is not inherited thus the operator= versions from IEnumeration must be implemented again

13.97.3 Variable Documentation**13.97.3.1 IEnumerationT**

```
interface SPINNAKER_API_ABSTRACT IEnumerationT
```

Interface for enumeration properties.

13.97.3.2 IEnumReference

```
interface SPINNAKER_API_ABSTRACT IEnumReference
```

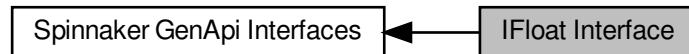
Initial value:

```
{
    virtual void SetValue(EnumT Value, bool Verify = true) = 0
```

Interface to construct an enum reference.

13.98 IFloat Interface

Collaboration diagram for IFloat Interface:



Functions

- virtual [IFloat & operator=](#) (double Value)=0
Set node value.
- virtual double [GetMin](#) ()=0
Get minimum value allowed.
- virtual double [GetMax](#) ()=0
Get maximum value allowed.
- virtual bool [HasInc](#) ()=0
True if the float has a constant increment.
- virtual [EIncMode GetIncMode](#) ()=0
Get increment mode.
- virtual double [GetInc](#) ()=0
Get the constant increment if there is any.
- virtual [double_autovector_t GetListOfValidValues](#) (bool bounded=true)=0
Get list of valid value.
- virtual [ERepresentation GetRepresentation](#) ()=0
Get recommended representation.
- virtual [GenICam::gcstring GetUnit](#) () const =0
Get the physical unit name.
- virtual [EDisplayNotation GetDisplayNotation](#) () const =0
Get the way the float should be converted to a string.
- virtual [int64_t GetDisplayPrecision](#) () const =0
Get the precision to be used when converting the float to a string.
- virtual void [ImposeMin](#) (double Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)=0
Restrict maximum value.

Variables

- [interface SPINNAKER_API_ABSTRACT IFloat](#)
Interface for float properties.

13.98.1 Detailed Description

13.98.2 Function Documentation

13.98.2.1 GetDisplayNotation()

```
virtual EDisplayNotation Spinnaker::GenApi::GetDisplayNotation ( ) const [pure virtual]
```

Get the way the float should be converted to a string.

13.98.2.2 GetDisplayPrecision()

```
virtual int64_t Spinnaker::GenApi::GetDisplayPrecision ( ) const [pure virtual]
```

Get the precision to be used when converting the float to a string.

13.98.2.3 GetInc()

```
int64_t GetInc ( ) [pure virtual]
```

Get the constant increment if there is any.

Get increment.

13.98.2.4 GetIncMode()

```
EIncMode GetIncMode ( ) [pure virtual]
```

Get increment mode.

13.98.2.5 GetListOfValidValues()

```
int64\_autovector\_t GetListOfValidValues (
    bool bounded = true ) [pure virtual]
```

Get list of valid value.

13.98.2.6 GetMax()

```
int64_t GetMax ( ) [pure virtual]
```

Get maximum value allowed.

13.98.2.7 GetMin()

```
int64_t GetMin ( ) [pure virtual]
```

Get minimum value allowed.

13.98.2.8 GetRepresentation()

```
ERepresentation GetRepresentation ( ) [pure virtual]
```

Get recommended representation.

13.98.2.9 GetUnit()

```
GenICam::gcstring GetUnit ( ) const [pure virtual]
```

Get the physical unit name.

13.98.2.10 HasInc()

```
virtual bool Spinnaker::GenApi::HasInc ( ) [pure virtual]
```

True if the float has a constant increment.

13.98.2.11 ImposeMax()

```
virtual void Spinnaker::GenApi::ImposeMax (
    double Value ) [pure virtual]
```

Restrict maximum value.

13.98.2.12 `ImposeMin()`

```
virtual void Spinnaker::GenApi::ImposeMin (  
    double Value ) [pure virtual]
```

Restrict minimum value.

13.98.2.13 `operator=()`

```
virtual IFloat& Spinnaker::GenApi::operator= (  
    double Value ) [pure virtual]
```

Set node value.

13.98.3 Variable Documentation

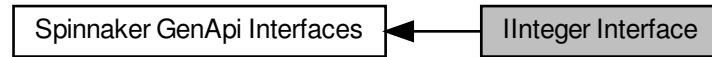
13.98.3.1 `IFloat`

```
interface SPINNAKER_API_ABSTRACT IFloat
```

Interface for float properties.

13.99 Integer Interface

Collaboration diagram for Integer Interface:



Functions

- virtual [Integer](#) & [operator=](#) (int64_t Value)=0
Set node value.
- virtual void [ImposeMin](#) (int64_t Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (int64_t Value)=0
Restrict maximum value.

Variables

- [interface SPINNAKER_API_ABSTRACT Integer](#)
Interface for integer properties.

13.99.1 Detailed Description

13.99.2 Function Documentation

13.99.2.1 ImposeMax()

```
virtual void Spinnaker::GenApi::ImposeMax (
    int64_t Value ) [pure virtual]
```

Restrict maximum value.

13.99.2.2 ImposeMin()

```
virtual void Spinnaker::GenApi::ImposeMin (
    int64_t Value ) [pure virtual]
```

Restrict minimum value.

13.99.2.3 operator=()

```
virtual IInteger& Spinnaker::GenApi::operator= (  
    int64_t Value ) [pure virtual]
```

Set node value.

13.99.3 Variable Documentation

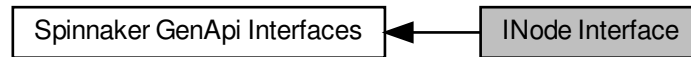
13.99.3.1 IInteger

```
interface SPINNAKER_API_ABSTRACT IInteger
```

[Interface](#) for integer properties.

13.100 INode Interface

Collaboration diagram for INode Interface:



Functions

- virtual [GenApi::ENamespace GetNamespace \(\)](#) const =0
Get name space.
- virtual [EVisibility GetVisibility \(\)](#) const =0
Get the recommended visibility of the node.
- virtual void [InvalidateNode \(\)](#)=0
Indicates that the node's value may have changed.
- virtual bool [IsCachable \(\)](#) const =0
Is the node value cacheable.
- virtual [EYesNo IsAccessModeCacheable \(\)](#) const =0
True if the AccessMode can be cached.
- virtual [ECachingMode GetCachingMode \(\)](#) const =0
Get Caching Mode.
- virtual [int64_t GetPollingTime \(\)](#) const =0
recommended polling time (for non-cacheable nodes)
- virtual [GenICam::gcstring GetDescription \(\)](#) const =0
Get a long description of the node.
- virtual [GenICam::gcstring GetDisplayName \(\)](#) const =0
Get a name string for display.
- virtual void [GetChildren \(GenApi::NodeList_t &Children, ELinkType LinkType=ctReadingChildren\)](#) const =0
Get all nodes this node directly depends on.
- virtual void [GetParents \(GenApi::NodeList_t &Parents\)](#) const =0
Gets all nodes this node is directly depending on.
- virtual [CallbackHandleType RegisterCallback \(CNodeCallback *pCallback\)](#)=0
Register change callback Takes ownership of the CNodeCallback object.
- virtual bool [DeregisterCallback \(CallbackHandleType hCallback\)](#)=0
De register change callback Destroys CNodeCallback object.
- virtual [INodeMap * GetNodeMap \(\)](#) const =0
Retrieves the central node map.
- virtual [GenICam::gcstring GetEventID \(\)](#) const =0
Get the EventId of the node.
- virtual bool [IsStreamable \(\)](#) const =0
True if the node is streamable.
- virtual void [GetPropertyNames \(GenICam::gcstring_vector &PropertyNames\)](#) const =0
Returns a list of the names all properties set during initialization.

- virtual bool [GetProperty](#) (const [GenlCam::gcstring](#) &PropertyName, [GenlCam::gcstring](#) &ValueStr, [GenlCam::gcstring](#) &AttributeStr)=0
Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.
- virtual void [ImposeAccessMode](#) ([EAccessMode](#) ImposedAccessMode)=0
Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) ([EVisibility](#) ImposedVisibility)=0
Imposes a visibility to the natural visibility of the node.
- virtual [INode](#) * [GetAlias](#) () const =0
Retrieves the a node which describes the same feature in a different way.
- virtual [INode](#) * [GetCastAlias](#) () const =0
Retrieves the a node which describes the same feature so that it can be casted.
- virtual [GenlCam::gcstring](#) [GetDocuURL](#) () const =0
Gets a URL pointing to the documentation of that feature.
- virtual bool [IsDeprecated](#) () const =0
True if the node should not be used any more.
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const =0
Get the type of the main interface of a node.
- virtual bool [IsFeature](#) () const =0
True if the node can be reached via category nodes from a category node named "Root".
- virtual bool [operator==](#) (int nullPtr) const =0
- virtual bool [operator!=](#) (int nullPtr) const =0
- bool [IsReadable](#) ([EAccessMode](#) AccessMode)
Tests if readable.
- bool [IsReadable](#) (const [IBase](#) *p)
Checks if a node is readable.
- bool [IsReadable](#) (const [IBase](#) &r)
Checks if a node is readable.
- bool [IsWritable](#) ([EAccessMode](#) AccessMode)
Tests if writable.
- bool [IsWritable](#) (const [IBase](#) *p)
Checks if a node is writable.
- bool [IsWritable](#) (const [IBase](#) &r)
Checks if a node is writable.
- bool [IsImplemented](#) ([EAccessMode](#) AccessMode)
Tests if implemented.
- bool [IsImplemented](#) (const [IBase](#) *p)
Checks if a node is implemented.
- bool [IsImplemented](#) (const [IBase](#) &r)
Checks if a node is implemented.
- bool [IsAvailable](#) ([EAccessMode](#) AccessMode)
Tests if available.
- bool [IsAvailable](#) (const [IBase](#) *p)
Checks if a node is available.
- bool [IsAvailable](#) (const [IBase](#) &r)
Checks if a node is available.
- [EAccessMode](#) [Combine](#) ([EAccessMode](#) Peter, [EAccessMode](#) Paul)
Computes which access mode the two guards allow together.
- bool [IsVisible](#) ([EVisibility](#) Visibility, [EVisibility](#) MaxVisibility)
Tests Visibility CAVE : this relies on the EVisibility enum's coding.
- [EVisibility](#) [Combine](#) ([EVisibility](#) Peter, [EVisibility](#) Paul)

Computes which visibility the two guards allow together.

- bool `IsCacheable` (`ECachingMode` `CachingMode`)
Tests Cacheability.
- `ECachingMode` `Combine` (`ECachingMode` `Peter`, `ECachingMode` `Paul`)
Computes which CachingMode results from a combination.

Variables

- interface `SPINNAKER_API_ABSTRACT` `INode`
Interface common to all nodes.
- interface `SPINNAKER_API_ABSTRACT` virtual public `IReference`
Interface to construct a reference.

13.100.1 Detailed Description

13.100.2 Function Documentation

13.100.2.1 `Combine()` [1/3]

```
EAccessMode Spinnaker::GenApi::Combine (
    EAccessMode Peter,
    EAccessMode Paul ) [inline]
```

Computes which access mode the two guards allow together.

13.100.2.2 `Combine()` [2/3]

```
EVisibility Spinnaker::GenApi::Combine (
    EVisibility Peter,
    EVisibility Paul ) [inline]
```

Computes which visibility the two guards allow together.

13.100.2.3 `Combine()` [3/3]

```
ECachingMode Spinnaker::GenApi::Combine (
    ECachingMode Peter,
    ECachingMode Paul ) [inline]
```

Computes which CachingMode results from a combination.

13.100.2.4 DeregisterCallback()

```
virtual bool Spinnaker::GenApi::DeregisterCallback (
    CallbackHandleType hCallback ) [pure virtual]
```

De register change callback Destroys [CNodeCallback](#) object.

Returns

true if the callback handle was valid

13.100.2.5 GetAlias()

```
virtual INode* Spinnaker::GenApi::GetAlias ( ) const [pure virtual]
```

Retrieves the a node which describes the same feature in a different way.

13.100.2.6 GetCachingMode()

```
virtual ECachingMode Spinnaker::GenApi::GetCachingMode ( ) const [pure virtual]
```

Get Caching Mode.

13.100.2.7 GetCastAlias()

```
virtual INode* Spinnaker::GenApi::GetCastAlias ( ) const [pure virtual]
```

Retrieves the a node which describes the same feature so that it can be casted.

13.100.2.8 GetChildren()

```
virtual void Spinnaker::GenApi::GetChildren (
    GenApi::NodeList_t & Children,
    ELinkType LinkType = ctReadingChildren ) const [pure virtual]
```

Get all nodes this node directly depends on.

Parameters

out	<i>Children</i>	List of children nodes
	<i>LinkType</i>	The link type

13.100.2.9 GetDescription()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDescription ( ) const [pure virtual]
```

Get a long description of the node.

13.100.2.10 GetDisplayName()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDisplayName ( ) const [pure virtual]
```

Get a name string for display.

13.100.2.11 GetDocuURL()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDocuURL ( ) const [pure virtual]
```

Gets a URL pointing to the documentation of that feature.

13.100.2.12 GetEventID()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetEventID ( ) const [pure virtual]
```

Get the EventId of the node.

13.100.2.13 GetNameSpace()

```
virtual GenApi::ENamespace Spinnaker::GenApi::GetNameSpace ( ) const [pure virtual]
```

Get name space.

13.100.2.14 GetNodeMap()

```
virtual INodeMap* Spinnaker::GenApi::GetNodeMap ( ) const [pure virtual]
```

Retrieves the central node map.

13.100.2.15 GetParents()

```
virtual void Spinnaker::GenApi::GetParents (
    GenApi::NodeList_t & Parents ) const [pure virtual]
```

Gets all nodes this node is directly depending on.

Parameters

out	<i>Parents</i>	List of parent nodes

13.100.2.16 GetPollingTime()

```
virtual int64_t Spinnaker::GenApi::GetPollingTime ( ) const [pure virtual]
```

recommended polling time (for non-cacheable nodes)

13.100.2.17 GetPrincipalInterfaceType()

```
virtual EInterfaceType Spinnaker::GenApi::GetPrincipalInterfaceType ( ) const [pure virtual]
```

Get the type of the main interface of a node.

13.100.2.18 GetProperty()

```
virtual bool Spinnaker::GenApi::GetProperty (
    const GenICam::gcstring & PropertyName,
    GenICam::gcstring & ValueStr,
    GenICam::gcstring & AttributeStr ) [pure virtual]
```

Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.

13.100.2.19 GetPropertyNames()

```
virtual void Spinnaker::GenApi::GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [pure virtual]
```

Returns a list of the names all properties set during initialization.

13.100.2.20 GetVisibility()

```
virtual EVisibility Spinnaker::GenApi::GetVisibility ( ) const [pure virtual]
```

Get the recommended visibility of the node.

13.100.2.21 ImposeAccessMode()

```
virtual void Spinnaker::GenApi::ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [pure virtual]
```

Imposes an access mode to the natural access mode of the node.

13.100.2.22 ImposeVisibility()

```
virtual void Spinnaker::GenApi::ImposeVisibility (
    EVisibility ImposedVisibility ) [pure virtual]
```

Imposes a visibility to the natural visibility of the node.

13.100.2.23 InvalidateNode()

```
virtual void Spinnaker::GenApi::InvalidateNode ( ) [pure virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

13.100.2.24 IsAccessModeCacheable()

```
virtual EYesNo Spinnaker::GenApi::IsAccessModeCacheable ( ) const [pure virtual]
```

True if the AccessMode can be cached.

13.100.2.25 IsAvailable() [1/3]

```
bool Spinnaker::GenApi::IsAvailable (
    EAccessMode AccessMode ) [inline]
```

Tests if available.

13.100.2.26 IsAvailable() [2/3]

```
bool Spinnaker::GenApi::IsAvailable (
    const IBase * p ) [inline]
```

Checks if a node is available.

13.100.2.27 IsAvailable() [3/3]

```
bool Spinnaker::GenApi::IsAvailable (
    const IBase & r ) [inline]
```

Checks if a node is available.

13.100.2.28 IsCacheable()

```
virtual bool Spinnaker::GenApi::IsCacheable ( ) const [pure virtual]
```

Is the node value cacheable.

13.100.2.29 IsCacheable()

```
bool Spinnaker::GenApi::IsCacheable (
    ECachingMode CachingMode ) [inline]
```

Tests Cacheability.

13.100.2.30 IsDeprecated()

```
virtual bool Spinnaker::GenApi::IsDeprecated ( ) const [pure virtual]
```

True if the node should not be used any more.

13.100.2.31 IsFeature()

```
virtual bool Spinnaker::GenApi::IsFeature ( ) const [pure virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

13.100.2.32 IsImplemented() [1/3]

```
bool Spinnaker::GenApi::IsImplemented (
    EAccessMode AccessMode ) [inline]
```

Tests if implemented.

13.100.2.33 IsImplemented() [2/3]

```
bool Spinnaker::GenApi::IsImplemented (
    const IBase * p ) [inline]
```

Checks if a node is implemented.

13.100.2.34 IsImplemented() [3/3]

```
bool Spinnaker::GenApi::IsImplemented (
    const IBase & r ) [inline]
```

Checks if a node is implemented.

13.100.2.35 IsReadable() [1/3]

```
bool Spinnaker::GenApi::IsReadable (
    EAccessMode AccessMode ) [inline]
```

Tests if readable.

13.100.2.36 IsReadable() [2/3]

```
bool Spinnaker::GenApi::IsReadable (
    const IBase * p ) [inline]
```

Checks if a node is readable.

13.100.2.37 IsReadable() [3/3]

```
bool Spinnaker::GenApi::IsReadable (
    const IBase & r ) [inline]
```

Checks if a node is readable.

13.100.2.38 IsStreamable()

```
virtual bool Spinnaker::GenApi::IsStreamable ( ) const [pure virtual]
```

True if the node is streamable.

13.100.2.39 isVisible()

```
bool Spinnaker::GenApi::isVisible (
    EVisibility Visibility,
    EVisibility MaxVisiblity ) [inline]
```

Tests Visibility CAVE : this relies on the EVisibility enum's coding.

13.100.2.40 IsWritable() [1/3]

```
bool Spinnaker::GenApi::IsWritable (
    EAccessMode AccessMode ) [inline]
```

Tests if writable.

13.100.2.41 IsWritable() [2/3]

```
bool Spinnaker::GenApi::IsWritable (
    const IBase * p ) [inline]
```

Checks if a node is writable.

13.100.2.42 IsWritable() [3/3]

```
bool Spinnaker::GenApi::IsWritable (
    const IBase & r ) [inline]
```

Checks if a node is writable.

13.100.2.43 operator!=(())

```
virtual bool Spinnaker::GenApi::operator!= (
    int nullptr ) const [pure virtual]
```

13.100.2.44 operator==(())

```
virtual bool Spinnaker::GenApi::operator== (
    int nullptr ) const [pure virtual]
```


13.100.2.45 RegisterCallback()

```
virtual CallbackHandleType Spinnaker::GenApi::RegisterCallback (
    CNodeCallback * pCallback ) [pure virtual]
```

Register change callback Takes ownership of the [CNodeCallback](#) object.

13.100.3 Variable Documentation

13.100.3.1 INode

```
interface SPINNAKER_API_ABSTRACT INode
```

[Interface](#) common to all nodes.

13.100.3.2 IReference

```
interface SPINNAKER_API_ABSTRACT IReference
```

Initial value:

```
{
    virtual GenICam::gcstring GetName(bool FullQualified = false) const = 0
```

[Interface](#) to construct a reference.

13.101 INodeMap Interface

Collaboration diagram for INodeMap Interface:



Functions

- virtual [INode](#) * [GetNode](#) (const [GenICam::gcstring](#) &Name) const =0
Retrieves the node from the central map by Name.
- virtual void [InvalidateNodes](#) () const =0
Invalidates all nodes.
- virtual bool [Connect](#) ([IPort](#) *pPort, const [GenICam::gcstring](#) &PortName) const =0
Connects a port to a port node with given name.
- virtual bool [Connect](#) ([IPort](#) *pPort) const =0
Connects a port to the standard port "Device".
- virtual void [Poll](#) (int64_t ElapsedTime)=0
Fires nodes which have a polling time.
- virtual [CLock](#) & [GetLock](#) () const =0
Returns the lock which guards the node map.
- virtual uint64_t [GetNumNodes](#) () const =0
Get the number of nodes in the map.
- virtual [GenICam::gcstring](#) [GetDeviceName](#) () const =0
Get a name of the device.

Variables

- [interface SPINNAKER_API_ABSTRACT INodeMap](#)
Interface to access the node map.

13.101.1 Detailed Description

13.101.2 Function Documentation

13.101.2.1 Connect() [1/2]

```
virtual bool Spinnaker::GenApi::Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) const [pure virtual]
```

Connects a port to a port node with given name.

13.101.2.2 Connect() [2/2]

```
virtual bool Spinnaker::GenApi::Connect (
    IPort * pPort ) const [pure virtual]
```

Connects a port to the standard port "Device".

13.101.2.3 GetDeviceName()

```
GenICam::gcstring GetDeviceName ( ) [pure virtual]
```

Get a name of the device.

Get device name The device name identifies a device instance, e.g.

for debugging purposes. The default is "Device".

13.101.2.4 GetLock()

```
virtual CLock& Spinnaker::GenApi::GetLock ( ) const [pure virtual]
```

Returns the lock which guards the node map.

13.101.2.5 GetNode()

```
virtual INode* Spinnaker::GenApi::GetNode (
    const GenICam::gcstring & Name ) const [pure virtual]
```

Retrieves the node from the central map by Name.

13.101.2.6 GetNumNodes()

```
virtual uint64_t Spinnaker::GenApi::GetNumNodes ( ) const [pure virtual]
```

Get the number of nodes in the map.

13.101.2.7 InvalidateNodes()

```
virtual void Spinnaker::GenApi::InvalidateNodes ( ) const [pure virtual]
```

Invalidates all nodes.

13.101.2.8 Poll()

```
virtual void Spinnaker::GenApi::Poll (
    int64_t ElapsedTime ) [pure virtual]
```

Fires nodes which have a polling time.

13.101.3 Variable Documentation

13.101.3.1 INodeMap

```
interface SPINNAKER_API_ABSTRACT INodeMap
```

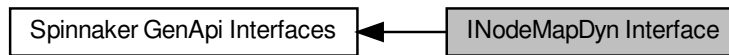
Initial value:

```
{
    virtual void GetNodes(NodeList_t & Nodes) const = 0
}
```

[Interface](#) to access the node map.

13.102 INodeMapDyn Interface

Collaboration diagram for INodeMapDyn Interface:



Functions

- virtual void [LoadXMLFromFile](#) (const [GenICam::gcstring](#) &FileName)=0
Loads an XML from a file.
- virtual void [LoadXMLFromFileInject](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectFileName)=0
Loads an XML from a file with injection.
- virtual void [LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)=0
Loads an XML from a string.
- virtual void [LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLData, const [GenICam::gcstring](#) &InjectXMLData)=0
Loads an XML from a string with injection.
- virtual void [PreprocessXMLFromFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32_t XMLValidation=[xv↔Default](#))=0
Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void [MergeXMLFiles](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectedFileName, const [GenICam::gcstring](#) &OutputFileName)=0
Injects an XML file into a target file.
- virtual void [ExtractIndependentSubtree](#) (const [GenICam::gcstring](#) &XMLData, const [GenICam::gcstring](#) &InjectXMLData, const [GenICam::gcstring](#) &SubTreeRootNodeName, [GenICam::gcstring](#) &Extracted↔Subtree)=0
Extract independent subtree.
- virtual void [GetSupportedSchemaVersions](#) ([GenICam::gcstring_vector](#) &SchemaVersions)=0
Gets a list of supported schema versions.
- virtual void [LoadXMLFromZIPFile](#) (const [GenICam::gcstring](#) &ZipFileName)=0
Loads an XML from a ZIP file.
- virtual void [LoadXMLFromZIPData](#) (const void *zipData, size_t zipSize)=0
Loads an XML from a ZIP data buffer.
- virtual void [PreprocessXMLFromZIPFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32_t XMLValidation=[xv↔Default](#))=0
Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

Variables

- [interface SPINNAKER_API_ABSTRACT INodeMapDyn](#)
Interface to access the node map.

13.102.1 Detailed Description

13.102.2 Function Documentation

13.102.2.1 ExtractIndependentSubtree()

```
virtual void Spinnaker::GenApi::ExtractIndependentSubtree (
    const GenICam::gcstring & XMLData,
    const GenICam::gcstring & InjectXMLData,
    const GenICam::gcstring & SubTreeRootNodeName,
    GenICam::gcstring & ExtractedSubtree ) [pure virtual]
```

Extract independent subtree.

Parameters

<i>InjectXMLData</i>	The XML data the subtree is extracted from.
<i>SubTreeRootNodeName</i>	Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed.
<i>ExtractedSubtree</i>	The name of the node that represents the root of the subtree that shall be extracted.> The returned extracted subtree as string.

13.102.2.2 GetSupportedSchemaVersions()

```
virtual void Spinnaker::GenApi::GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) [pure virtual]
```

Gets a list of supported schema versions.

Each list entry is a string with the format "<Major>.<Minor>" where <Major> and <Minor> are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

13.102.2.3 LoadXMLFromFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromFile (
    const GenICam::gcstring & FileName ) [pure virtual]
```

Loads an XML from a file.

13.102.2.4 LoadXMLFromFileInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName ) [pure virtual]
```

Loads an XML from a file with injection.

13.102.2.5 LoadXMLFromString()

```
virtual void Spinnaker::GenApi::LoadXMLFromString (
    const GenICam::gcstring & XMLData ) [pure virtual]
```

Loads an XML from a string.

13.102.2.6 LoadXMLFromStringInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData,
    const GenICam::gcstring & InjectXMLData ) [pure virtual]
```

Loads an XML from a string with injection.

13.102.2.7 LoadXMLFromZIPData()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize ) [pure virtual]
```

Loads an XML from a ZIP data buffer.

13.102.2.8 LoadXMLFromZIPFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName ) [pure virtual]
```

Loads an XML from a ZIP file.

13.102.2.9 MergeXMLFiles()

```
virtual void Spinnaker::GenApi::MergeXMLFiles (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectedFileName,
    const GenICam::gcstring & OutputFileName ) [pure virtual]
```

Injects an XML file into a target file.

Parameters

<i>InjectedFileName</i>	Name of the target XML file to process
<i>OutputFileName</i>	Name of the Injected XML file to process> Name of the output file

13.102.2.10 PreprocessXMLFromFile()

```
virtual void Spinnaker::GenApi::PreprocessXMLFromFile (
    const GenICam::gcstring & XMLFileName,
    const GenICam::gcstring & StyleSheetFileName,
    const GenICam::gcstring & OutputFileName,
    const uint32_t XMLValidation = xvDefault ) [pure virtual]
```

Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

Parameters

<i>StyleSheetFileName</i>	The name of the XML file to process
<i>OutputFileName</i>	Optional name of a style sheet which is applied after the pre-processor (can be empty string)> This has no effect if the OutputFileName is an empty string
<i>XMLValidation</i>	Optional name of an output file into which the processed data is written (can be empty string)> Optional bit mask formed from EXMLValidation enumeration indicating which tests should be performed on the XML file

13.102.2.11 PreprocessXMLFromZIPFile()

```
virtual void Spinnaker::GenApi::PreprocessXMLFromZIPFile (
    const GenICam::gcstring & XMLFileName,
    const GenICam::gcstring & StyleSheetFileName,
    const GenICam::gcstring & OutputFileName,
    const uint32_t XMLValidation = xvDefault ) [pure virtual]
```

Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

Parameters

<i>StyleSheetFileName</i>	The name of the XML file to process
<i>OutputFileName</i>	Optional name of a style sheet which is applied after the pre-processor (can be empty string)> This has no effect if the OutputFileName is an empty string
<i>XML Validation</i>	Optional name of an output file into which the processed data is written (can be empty string)> Optional bit mask formed from EXMLValidation enumeration indicating which tests should be performed on the XML file

13.102.3 Variable Documentation

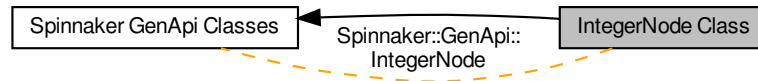
13.102.3.1 INodeMapDyn

```
interface SPINNAKER_API_ABSTRACT INodeMapDyn
```

Interface to access the node map.

13.103 IntegerNode Class

Collaboration diagram for IntegerNode Class:



Classes

- class [IntegerNode](#)
Interface for string properties.

Typedefs

- typedef [IntegerNode CIntegerRef](#)

13.103.1 Detailed Description

13.103.2 Typedef Documentation

13.103.2.1 CIntegerRef

typedef [IntegerNode CIntegerRef](#)

13.104 IntRegNode Class

Collaboration diagram for IntRegNode Class:



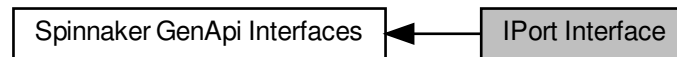
Classes

- class [IntRegNode](#)
Interface for string properties.

13.104.1 Detailed Description

13.105 IPort Interface

Collaboration diagram for IPort Interface:



Functions

- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))=0
Writes a chunk of bytes to the port.

Variables

- [interface SPINNAKER_API_ABSTRACT IPort](#)
Interface for ports.
- [interface SPINNAKER_API_ABSTRACT](#) int64_t [Address](#)
- [interface SPINNAKER_API_ABSTRACT](#) int64_t int64_t [Length](#) = 0

13.105.1 Detailed Description

13.105.2 Function Documentation

13.105.2.1 Write()

```
virtual void Spinnaker::GenApi::Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes a chunk of bytes to the port.

13.105.3 Variable Documentation

13.105.3.1 Address

```
interface SPINNAKER_API_ABSTRACT int64_t Address
```

13.105.3.2 IPort

```
interface SPINNAKER_API_ABSTRACT IPort
```

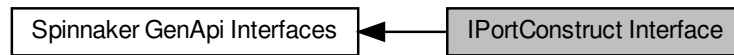
Interface for ports.

13.105.3.3 Length

```
interface SPINNAKER_API_ABSTRACT int64_t Length = 0
```

13.106 IPortConstruct Interface

Collaboration diagram for IPortConstruct Interface:



Functions

- virtual [EYesNo GetSwapEndianness \(\)=0](#)
Determines if the port adapter must perform an endianness swap.

Variables

- [interface SPINNAKER_API IPortConstruct](#)
Interface for ports.

13.106.1 Detailed Description

13.106.2 Function Documentation

13.106.2.1 GetSwapEndianness()

```
virtual EYesNo Spinnaker::GenApi::GetSwapEndianness ( ) [pure virtual]
```

Determines if the port adapter must perform an endianness swap.

13.106.3 Variable Documentation

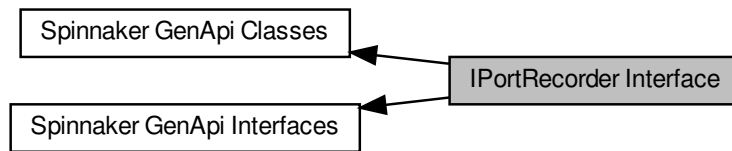
13.106.3.1 IPortConstruct

```
interface SPINNAKER\_API IPortConstruct
```

[Interface](#) for ports.

13.107 IPortRecorder Interface

Collaboration diagram for IPortRecorder Interface:



Functions

- virtual void [Replay](#) (IPort *pPort)=0
Replays the write command to the given port interface.
- virtual void [SetCookie](#) (const int64_t Value)=0
Sets a cookie in case the port implementation want to cache a command list.
- virtual int64_t [GetCookie](#) ()=0
Gets the cookie a port implementation may have set for caching a command list.
- virtual void [StopRecording](#) ()=0
Stops recording.

Variables

- [interface SPINNAKER_API_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER_API_ABSTRACT IPortReplay](#)
Interface for replaying write commands on a port.
- [interface SPINNAKER_API_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER_API_ABSTRACT IPortRecorder](#)
Interface for recording write commands on a port.

13.107.1 Detailed Description

13.107.2 Function Documentation

13.107.2.1 GetCookie()

```
virtual int64_t Spinnaker::GenApi::GetCookie ( ) [pure virtual]
```

Gets the cookie a port implementation may have set for caching a command list.

13.107.2.2 Replay()

```
virtual void Spinnaker::GenApi::Replay (
    IPort * pPort ) [pure virtual]
```

Replays the write command to the given port interface.

13.107.2.3 SetCookie()

```
virtual void Spinnaker::GenApi::SetCookie (
    const int64_t Value ) [pure virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

13.107.2.4 StopRecording()

```
virtual void Spinnaker::GenApi::StopRecording ( ) [pure virtual]
```

Stops recording.

13.107.3 Variable Documentation

13.107.3.1 Invalidate

```
interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0
```

13.107.3.2 IPortRecorder

```
interface SPINNAKER_API_ABSTRACT IPortRecorder
```

Interface for recording write commands on a port.

13.107.3.3 IPortReplay

```
interface SPINNAKER_API_ABSTRACT IPortReplay
```

Interface for replaying write commands on a port.

13.107.3.4 IPortWriteList

```
interface SPINNAKER_API_ABSTRACT IPortWriteList
```

Initial value:

```
{
    virtual void Write(const void* pBuffer, int64_t Address, int64_t
        Length) = 0
```


13.108 IRegister Interfaces

Collaboration diagram for IRegister Interfaces:



Functions

- virtual void [Get](#) (uint8_t *pBuffer, int64_t [Length](#), bool [Verify](#)=false, bool IgnoreCache=false)=0
Fills a buffer with the register's contents.
- virtual int64_t [GetLength](#) ()=0
Retrieves the Length of the register [Bytes].
- virtual int64_t [GetAddress](#) ()=0
Retrieves the Address of the register.

Variables

- [interface SPINNAKER_API_ABSTRACT IRegister](#)
Interface for registers.

13.108.1 Detailed Description

13.108.2 Function Documentation

13.108.2.1 Get()

```

virtual void Spinnaker::GenApi::Get (
    uint8_t * pBuffer,
    int64_t Length,
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
  
```

Fills a buffer with the register's contents.

Parameters

<i>pBuffer</i>	The buffer receiving the data to read
<i>Length</i>	The number of bytes to retrieve
<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

13.108.2.2 GetAddress()

```
virtual int64_t Spinnaker::GenApi::GetAddress ( ) [pure virtual]
```

Retrieves the Address of the register.

13.108.2.3 GetLength()

```
virtual int64_t Spinnaker::GenApi::GetLength ( ) [pure virtual]
```

Retrieves the Length of the register [Bytes].

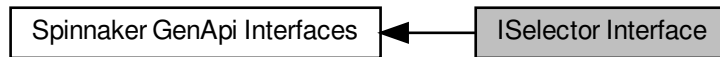
13.108.3 Variable Documentation**13.108.3.1 IRegister**

```
interface SPINNAKER_API_ABSTRACT IRegister
```

[Interface](#) for registers.

13.109 ISelector Interface

Collaboration diagram for ISelector Interface:



Functions

- virtual void [GetSelectedFeatures](#) (FeatureList_t &) const =0
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) (FeatureList_t &) const =0
retrieve the group of features selecting this node

Variables

- [interface SPINNAKER_API_ABSTRACT ISelector](#)
Interface for groups of features selected by a single one.

13.109.1 Detailed Description

13.109.2 Function Documentation

13.109.2.1 GetSelectedFeatures()

```
virtual void Spinnaker::GenApi::GetSelectedFeatures (
    FeatureList_t & ) const [pure virtual]
```

retrieve the group of selected features

13.109.2.2 GetSelectingFeatures()

```
virtual void Spinnaker::GenApi::GetSelectingFeatures (
    FeatureList_t & ) const [pure virtual]
```

retrieve the group of features selecting this node

13.109.3 Variable Documentation

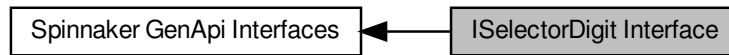
13.109.3.1 ISelector

```
interface SPINNAKER_API_ABSTRACT ISelector
```

[Interface](#) for groups of features selected by a single one.

13.110 ISelectorDigit Interface

Collaboration diagram for ISelectorDigit Interface:



Functions

- virtual bool [SetNext](#) (bool Tick=true)=0
Sets digit to next value.
- virtual void [Restore](#) ()=0
Restores the selectors' values found at creation.
- virtual [GenICam::gcstring ToString](#) ()=0
Returns a string representation of the digit.
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)=0
Retrieves an ordered list of selectors.

Variables

- [interface SPINNAKER_API_ABSTRACT ISelectorDigit](#)
Interface of a "digit" of the "counter" formed by the selector set.

13.110.1 Detailed Description

13.110.2 Function Documentation

13.110.2.1 GetSelectorList()

```
virtual void Spinnaker::GenApi::GetSelectorList (
    FeatureList_t & SelectorList,
    bool Incremental = false ) [pure virtual]
```

Retrieves an ordered list of selectors.

Parameters

<i>Incremental</i>	List to contain the selector pointer> if true only seletor changed since the last GetNext are contained

13.110.2.2 Restore()

```
virtual void Spinnaker::GenApi::Restore ( ) [pure virtual]
```

Restores the selectors' values found at creation.

13.110.2.3 SetNext()

```
virtual bool Spinnaker::GenApi::SetNext (
    bool Tick = true ) [pure virtual]
```

Sets digit to next value.

Parameters

<i>Tick</i>	if false the counter does not tick (but realize it could have)

Returns

true if the resulting value is valid

13.110.2.4 ToString()

```
virtual GenICam::gcstring Spinnaker::GenApi::ToString ( ) [pure virtual]
```

Returns a string representation of the digit.

13.110.3 Variable Documentation

13.110.3.1 ISelectorDigit

```
interface SPINNAKER_API_ABSTRACT ISelectorDigit
```

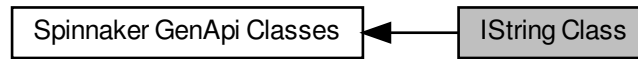
Initial value:

```
{  
  
    virtual bool SetFirst() = 0
```

[Interface](#) of a "digit" of the "counter" formed by the selector set.

13.111 IString Class

Collaboration diagram for IString Class:



Functions

- virtual int64_t [GetMaxLength](#) ()=0
Retrieves the maximum length of the string in bytes.

Variables

- [interface SPINNAKER_API_ABSTRACT IString](#)
Interface for string properties.

13.111.1 Detailed Description

13.111.2 Function Documentation

13.111.2.1 GetMaxLength()

```
virtual int64_t Spinnaker::GenApi::GetMaxLength ( ) [pure virtual]
```

Retrieves the maximum length of the string in bytes.

13.111.3 Variable Documentation

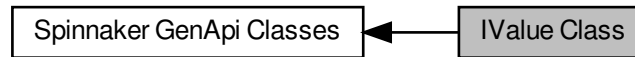
13.111.3.1 IString

```
interface SPINNAKER_API_ABSTRACT IString
```

[Interface](#) for string properties.

13.112 IValue Class

Collaboration diagram for IValue Class:



Functions

- virtual [GenICam::gcstring ToString](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
Get content of the node as string.
- virtual void [FromString](#) (const [GenICam::gcstring](#) &ValueStr, bool [Verify](#)=true)=0
Set content of the node as string.
- virtual bool [IsValueCacheValid](#) () const =0
Checks if the value comes from cache or is requested from another node.

Variables

- [interface SPINNAKER_API_ABSTRACT IValue](#)
Interface for value properties.

13.112.1 Detailed Description

13.112.2 Function Documentation

13.112.2.1 FromString()

```
virtual void Spinnaker::GenApi::FromString (
    const GenICam::gcstring & ValueStr,
    bool Verify = true ) [pure virtual]
```

Set content of the node as string.

Parameters

<i>ValueStr</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

13.112.2.2 IsValueCacheValid()

```
virtual bool Spinnaker::GenApi::IsValueCacheValid ( ) const [pure virtual]
```

Checks if the value comes from cache or is requested from another node.

13.112.2.3 ToString()

```
virtual GenICam::gcstring Spinnaker::GenApi::ToString (
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
```

Get content of the node as string.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

13.112.3 Variable Documentation

13.112.3.1 IValue

```
interface SPINNAKER_API_ABSTRACT IValue
```

Interface for value properties.

13.113 Node Class

Collaboration diagram for Node Class:



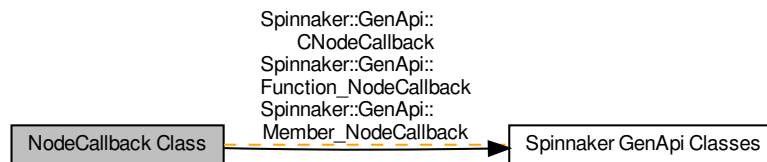
Classes

- class [Node](#)
class common to all nodes

13.113.1 Detailed Description

13.114 NodeCallback Class

Collaboration diagram for NodeCallback Class:



Classes

- class [CNodeCallback](#)
callback body instance for INode pointers
- class [Function_NodeCallback](#)< [Function](#) >
Container for a function pointer.
- class [Member_NodeCallback](#)< [Client](#), [Member](#) >
Container for a member function pointer.

Enumerations

- enum [ECallbackType](#) {
 [cbPostInsideLock](#) = 1,
 [cbPostOutsideLock](#) = 2 }
the type of callback

Functions

- template<class [Function](#) >
 [CNodeCallback](#) * [make_NodeCallback](#) ([INode](#) *pNode, [Function](#) function, [ECallbackType](#) CallbackType)
 make a new callback object for C functions
- template<class [Function](#) >
 intptr_t [Register](#) ([INode](#) *pNode, [Function](#) f, [ECallbackType](#) CallbackType=[cbPostInsideLock](#))
 Register a C-function as a callback.
- template<class [Client](#) , class [Member](#) >
 [CNodeCallback](#) * [make_NodeCallback](#) ([INode](#) *pNode, [Client](#) &client, [Member](#) member, [ECallbackType](#) CallbackType)
 make a new callback object for member functions
- template<class [Client](#) , class [Member](#) >
 intptr_t [Register](#) ([INode](#) *pNode, [Client](#) &c, [Member](#) m, [ECallbackType](#) CallbackType=[cbPostInsideLock](#))
 Register a C++-member function a callback.
- [SPINNAKER_API](#) void [Deregister](#) ([GenApi::CallbackHandleType](#) pCallbackInfo)
 Unregistering callback by handle.

13.114.1 Detailed Description

13.114.2 Enumeration Type Documentation

13.114.2.1 ECallbackType

enum [ECallbackType](#)

the type of callback

Enumerator

cbPostInsideLock	
cbPostOutsideLock	callback is fired on leaving the tree inside the lock-guarded area

13.114.3 Function Documentation

13.114.3.1 Deregister()

```
SPINNAKER_API void Spinnaker::GenApi::Deregister (  
    GenApi::CallbackHandleType pCallbackInfo )
```

Unregistering callback by handle.

13.114.3.2 make_NodeCallback() [1/2]

```
CNodeCallback* Spinnaker::GenApi::make_NodeCallback (  
    INode * pNode,  
    Function function,  
    ECallbackType CallbackType )
```

make a new callback object for C functions

13.114.3.3 make_NodeCallback() [2/2]

```
CNodeCallback* Spinnaker::GenApi::make_NodeCallback (
    INode * pNode,
    Client & client,
    Member member,
    ECallbackType CallbackType )
```

make a new callback object for member functions

13.114.3.4 Register() [1/2]

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Function f,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C-function as a callback.

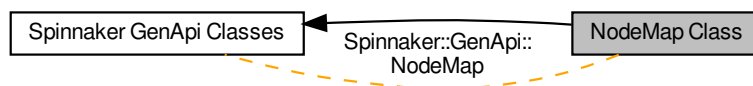
13.114.3.5 Register() [2/2]

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Client & c,
    Member m,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C++-member function a callback.

13.115 NodeMap Class

Collaboration diagram for NodeMap Class:



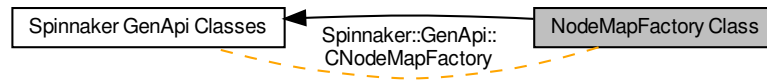
Classes

- class [NodeMap](#)
Smart pointer template for NodeMaps with create function.

13.115.1 Detailed Description

13.116 NodeMapFactory Class

Collaboration diagram for NodeMapFactory Class:



Classes

- class [CNodeMapFactory](#)

The node map factory is used for creating node maps from camera description files.

Enumerations

- enum [ECacheUsage_t](#) {
[CacheUsage_Automatic](#),
[CacheUsage_ForceWrite](#),
[CacheUsage_ForceRead](#),
[CacheUsage_Ignore](#) }

Lists the cache usage strategies.

- enum [EContentType_t](#) {
[ContentType_Xml](#),
[ContentType_ZippedXml](#) }

Lists the processable file types.

13.116.1 Detailed Description

13.116.2 Enumeration Type Documentation

13.116.2.1 ECacheUsage_t

enum [ECacheUsage_t](#)

Lists the cache usage strategies.

The cache stores preprocessed camera description xml files providing faster access or smaller footprint. note The environment variable GENICAM_CACHE_VERSION, e.g. GENICAM_CACHE_V3_0, must contain the path to cache directory for using the cache.

Enumerator

CacheUsage_Automatic	The use of cache files is determined automatically.
CacheUsage_ForceWrite	Forces the loading and preprocessing of the camera description xml file. If a cache directory is available the result of preprocessing is written to the cache.
CacheUsage_ForceRead	Suppresses loading and preprocessing of the camera description xml file and. forces reading a cache file from cache directory. Fails if no matching cache file is available.
CacheUsage_Ignore	Forces the loading and preprocessing of the camera description xml file. No cache file is written.

13.116.2.2 EContentType_t

enum [EContentType_t](#)

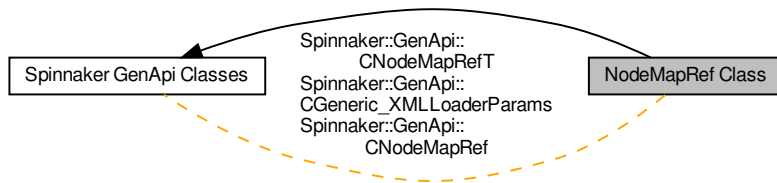
Lists the processable file types.

Enumerator

ContentType_Xml	XML camera description file text.
ContentType_ZippedXml	Zipped XML camera description file text.

13.117 NodeMapRef Class

Collaboration diagram for NodeMapRef Class:



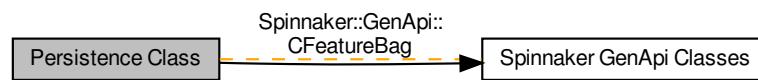
Classes

- class [CNodeMapRefT< TCameraParams >](#)
Smartpointer template for NodeMaps with create function.
- class [CGeneric_XMLLoaderParams](#)
Empty base class used by class [CNodeMapRef](#) as generic template argument.
- class [CNodeMapRef](#)
Smartpointer for NodeMaps with create function.

13.117.1 Detailed Description

13.118 Persistence Class

Collaboration diagram for Persistence Class:



Classes

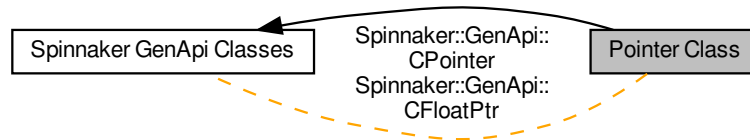
- class [CFeatureBag](#)

Bag holding streamable features of a nodetree.

13.118.1 Detailed Description

13.119 Pointer Class

Collaboration diagram for Pointer Class:



Classes

- class `CPointer< T, B >`
Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.
- class `CFloatPtr`
SmartPointer for IFloat interface pointer.

Typedefs

- typedef `CPointer< IBase > CBasePtr`
SmartPointer for IBase interface pointer.
- typedef `CPointer< INode, IBase > CNodePtr`
SmartPointer for INode interface pointer.
- typedef `CPointer< IValue > CValuePtr`
SmartPointer for IValue interface pointer.
- typedef `CPointer< ICategory > CCategoryPtr`
SmartPointer for ICategory interface pointer.
- typedef `CPointer< IBoolean > CBooleanPtr`
SmartPointer for IBoolean interface pointer.
- typedef `CPointer< IInteger > CIntegerPtr`
SmartPointer for IInteger interface pointer.
- typedef `CPointer< IString > CStringPtr`
SmartPointer for IString interface pointer.
- typedef `CPointer< IRegister > CRegisterPtr`
SmartPointer for IRegister interface pointer.
- typedef `CPointer< IEnumeration > CEnumerationPtr`
SmartPointer for IEnumeration interface pointer.
- typedef `CPointer< IEnumEntry > CEnumEntryPtr`
SmartPointer for IEnumEntry interface pointer.
- typedef `CPointer< IPort > CPortPtr`
SmartPointer for IPort interface pointer.
- typedef `CPointer< IPortReplay > CPortReplayPtr`
SmartPointer for IPortReplay interface pointer.
- typedef `CPointer< IPortRecorder > CPortRecorderPtr`

- SmartPointer for IPortRecorder interface pointer.
 • typedef [CPointer](#)< [IPortWriteList](#), [IPortWriteList](#) > [CPortWriteListPtr](#)
 SmartPointer for IPortWriteList interface pointer.
- typedef [CPointer](#)< [IChunkPort](#) > [CChunkPortPtr](#)
 SmartPointer for IChunkPort interface pointer.
- typedef [CPointer](#)< [INodeMap](#), [INodeMap](#) > [CNodeMapPtr](#)
 SmartPointer for INodeMap interface pointer.
- typedef [CPointer](#)< [INodeMapDyn](#), [INodeMap](#) > [CNodeMapDynPtr](#)
 SmartPointer for INodeMapDyn interface pointer.
- typedef [CPointer](#)< [IDeviceInfo](#), [INodeMap](#) > [CDeviceInfoPtr](#)
 SmartPointer for IDeviceInfo interface pointer.
- typedef [CPointer](#)< [ISelector](#) > [CSelectorPtr](#)
 SmartPointer for ISelector interface pointer.
- typedef [CPointer](#)< [ICommand](#) > [CCommandPtr](#)
 SmartPointer for ICommand interface pointer.
- typedef [CPointer](#)< [IPortConstruct](#) > [CPortConstructPtr](#)
 SmartPointer for IPortConstruct interface pointer.

Functions

- template<class T , class B >
 bool [IsReadable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
 Checks if a node is readable.
- template<class T , class B >
 bool [IsWritable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
 Checks if a node is Writable.
- template<class T , class B >
 bool [IsImplemented](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
 Checks if a node is Implemented.
- template<class T , class B >
 bool [IsAvailable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
 Checks if a node is Available.
- [GenICam::gcstring GetInterfaceName](#) (IBase *pBase)
 Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

13.119.1 Detailed Description

13.119.2 Typedef Documentation

13.119.2.1 CBasePtr

```
typedef CPointer<IBase> CBasePtr
```

SmartPointer for IBase interface pointer.

13.119.2.2 CBooleanPtr

```
typedef CPointer<IBoolean> CBooleanPtr
```

SmartPointer for IBoolean interface pointer.

13.119.2.3 CCategoryPtr

```
typedef CPointer<ICategory> CCategoryPtr
```

SmartPointer for ICategory interface pointer.

13.119.2.4 CChunkPortPtr

```
typedef CPointer<IChunkPort> CChunkPortPtr
```

SmartPointer for IChunkPort interface pointer.

13.119.2.5 CCommandPtr

```
typedef CPointer<ICommand> CCommandPtr
```

SmartPointer for ICommand interface pointer.

13.119.2.6 CDeviceInfoPtr

```
typedef CPointer<IDeviceInfo, INodeMap> CDeviceInfoPtr
```

SmartPointer for IDeviceInfo interface pointer.

13.119.2.7 CEnumEntryPtr

```
typedef CPointer<IEnumEntry> CEnumEntryPtr
```

SmartPointer for IEnumEntry interface pointer.

13.119.2.8 CEnumerationPtr

```
typedef CPointer<IEnumeration> CEnumerationPtr
```

SmartPointer for IEnumeration interface pointer.

13.119.2.9 CIntegerPtr

```
typedef CPointer<IInteger> CIntegerPtr
```

SmartPointer for IInteger interface pointer.

13.119.2.10 CNodeMapDynPtr

```
typedef CPointer<INodeMapDyn, INodeMap> CNodeMapDynPtr
```

SmartPointer for INodeMapDyn interface pointer.

13.119.2.11 CNodeMapPtr

```
typedef CPointer<INodeMap, INodeMap> CNodeMapPtr
```

SmartPointer for INodeMap interface pointer.

13.119.2.12 CNodePtr

```
typedef CPointer<INode, IBase> CNodePtr
```

SmartPointer for INode interface pointer.

13.119.2.13 CPortConstructPtr

```
typedef CPointer<IPortConstruct> CPortConstructPtr
```

SmartPointer for IPortConstruct interface pointer.

13.119.2.14 CPortPtr

```
typedef CPointer<IPort> CPortPtr
```

SmartPointer for IPort interface pointer.

13.119.2.15 CPortRecorderPtr

```
typedef CPointer<IPortRecorder> CPortRecorderPtr
```

SmartPointer for IPortRecorder interface pointer.

13.119.2.16 CPortReplayPtr

```
typedef CPointer<IPortReplay> CPortReplayPtr
```

SmartPointer for IPortReplay interface pointer.

13.119.2.17 CPortWriteListPtr

```
typedef CPointer<IPortWriteList, IPortWriteList> CPortWriteListPtr
```

SmartPointer for IPortWriteList interface pointer.

13.119.2.18 CRegisterPtr

```
typedef CPointer<IRegister> CRegisterPtr
```

SmartPointer for IRegister interface pointer.

13.119.2.19 CSelectorPtr

```
typedef CPointer<ISelector> CSelectorPtr
```

SmartPointer for ISelector interface pointer.

13.119.2.20 CStringPtr

```
typedef CPointer<IString> CStringPtr
```

SmartPointer for IString interface pointer.

13.119.2.21 CValuePtr

```
typedef CPointer<IValue> CValuePtr
```

SmartPointer for IValue interface pointer.

13.119.3 Function Documentation**13.119.3.1 GetInterfaceName()**

```
GenICam::gcstring Spinnaker::GenApi::GetInterfaceName (
    IBase * pBase ) [inline]
```

Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

13.119.3.2 IsAvailable()

```
bool Spinnaker::GenApi::IsAvailable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Available.

13.119.3.3 IsImplemented()

```
bool Spinnaker::GenApi::IsImplemented (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Implemented.

13.119.3.4 IsReadable()

```
bool Spinnaker::GenApi::IsReadable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is readable.

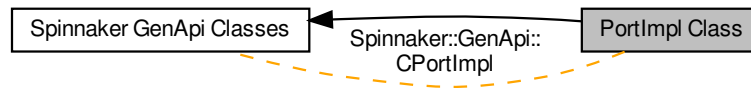
13.119.3.5 IsWritable()

```
bool Spinnaker::GenApi::IsWritable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Writable.

13.120 PortImpl Class

Collaboration diagram for PortImpl Class:



Classes

- class [CPortImpl](#)
Standard implementation for a port.

13.120.1 Detailed Description

13.121 PortNode Class

Collaboration diagram for PortNode Class:



Classes

- class [PortNode](#)
[Interface](#) for value properties.

Typedefs

- typedef [PortNode](#) [CPortRef](#)

13.121.1 Detailed Description

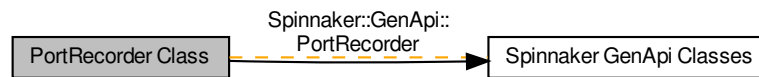
13.121.2 Typedef Documentation

13.121.2.1 CPortRef

```
typedef PortNode CPortRef
```

13.122 PortRecorder Class

Collaboration diagram for PortRecorder Class:



Classes

- class [PortRecorder](#)
Interface for recording write commands on a port.

Typedefs

- typedef [PortRecorder](#) [CPortRecorderRef](#)
Reference to an IPortRecorder pointer.

13.122.1 Detailed Description

13.122.2 Typedef Documentation

13.122.2.1 CPortRecorderRef

typedef [PortRecorder](#) [CPortRecorderRef](#)

Reference to an IPortRecorder pointer.

13.123 PortReplay Class

Collaboration diagram for PortReplay Class:



Classes

- class [PortReplay](#)
[Interface](#) for replaying write commands on a port.

13.123.1 Detailed Description

13.124 PortWriteList Class

Collaboration diagram for PortWriteList Class:



Classes

- class [CPortWriteList](#)
Container holding a list of port write commands.

13.124.1 Detailed Description

13.125 Reference Interfaces

Collaboration diagram for Reference Interfaces:



Functions

- virtual void [SetNumEnums](#) (int NumEnums)=0
sets the number of enum values

13.125.1 Detailed Description

13.125.2 Function Documentation

13.125.2.1 SetNumEnums()

```
virtual void Spinnaker::GenApi::SetNumEnums (  
    int NumEnums ) [pure virtual]
```

sets the number of enum values

13.126 RegisterNode Class

Collaboration diagram for RegisterNode Class:



Classes

- class [RegisterNode](#)
Interface for string properties.

Typedefs

- typedef [RegisterNode](#) [CRegisterRef](#)

13.126.1 Detailed Description

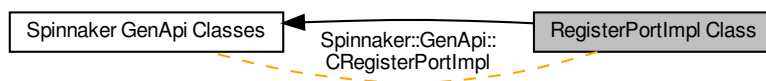
13.126.2 Typedef Documentation

13.126.2.1 CRegisterRef

```
typedef RegisterNode CRegisterRef
```

13.127 RegisterPortImpl Class

Collaboration diagram for RegisterPortImpl Class:



Classes

- class [CRegisterPortImpl](#)

Standard implementation for a port using a register based transport layer.

13.127.1 Detailed Description

13.128 SelectorSet Class

Collaboration diagram for SelectorSet Class:



Classes

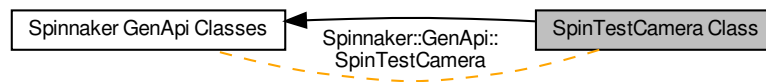
- class [CSelectorSet](#)

The set of selectors selecting a given node.

13.128.1 Detailed Description

13.129 SpinTestCamera Class

Collaboration diagram for SpinTestCamera Class:



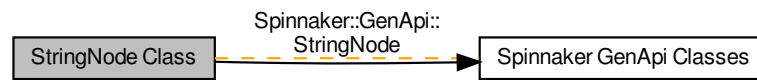
Classes

- class [SpinTestCamera](#)

13.129.1 Detailed Description

13.130 StringNode Class

Collaboration diagram for StringNode Class:



Classes

- class [StringNode](#)
Interface for string properties.

Typedefs

- typedef [StringNode](#) [CStringRef](#)

13.130.1 Detailed Description

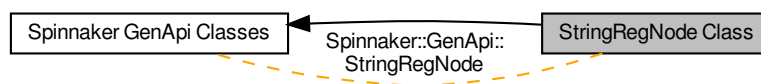
13.130.2 Typedef Documentation

13.130.2.1 CStringRef

```
typedef StringNode CStringRef
```

13.131 StringRegNode Class

Collaboration diagram for StringRegNode Class:



Classes

- class [StringRegNode](#)
Interface for string properties.

13.131.1 Detailed Description

13.132 StructPort Class

Collaboration diagram for StructPort Class:



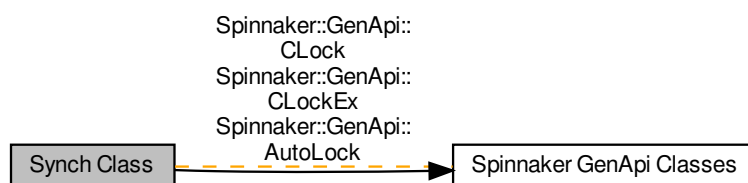
Classes

- class [CTestPortStruct< CDataStruct >](#)
Implements a register spaces based on a C++ struct.

13.132.1 Detailed Description

13.133 Synch Class

Collaboration diagram for Synch Class:



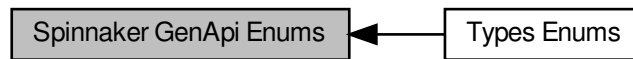
Classes

- class [CLOCK](#)
A lock class.
- class [CLOCKEx](#)
This class is for testing purposes only.
- class [AutoLock](#)

13.133.1 Detailed Description

13.134 Spinnaker GenApi Enums

Collaboration diagram for Spinnaker GenApi Enums:



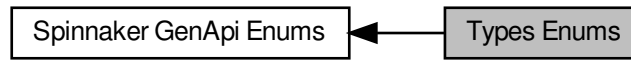
Modules

- [Types Enums](#)

13.134.1 Detailed Description

13.135 Types Enums

Collaboration diagram for Types Enums:



Macros

- `#define \_UndefinedRepresentation _UndefinedRepresentation`

Typedefs

- `typedef GenICam::gcstring_vector StringList\_t`
A list of strings.

Enumerations

- `enum ESign {`
`Signed,`
`Unsigned,`
`\_UndefinedSign }`
signed or unsigned integers
- `enum EAccessMode {`
`NI,`
`NA,`
`WO,`
`RO,`
`RW,`
`\_UndefinedAccesMode,`
`\_CycleDetectAccesMode }`
access mode of a node
- `enum EVisibility {`
`Beginner = 0,`
`Expert = 1,`
`Guru = 2,`
`Invisible = 3,`
`\_UndefinedVisibility = 99 }`
recommended visibility of a node
- `enum ECachingMode {`
`NoCache,`
`WriteThrough,`
`WriteAround,`
`\_UndefinedCachingMode }`
caching mode of a register

- enum `ERepresentation` {
`Linear`,
`Logarithmic`,
`Boolean`,
`PureNumber`,
`HexNumber`,
`IPV4Address`,
`MACAddress`,
`_UndefinedRepresentation` }
recommended representation of a node value
- enum `EEndianess` {
`BigEndian`,
`LittleEndian`,
`_UndefinedEndian` }
Endianess of a value in a register.
- enum `ENameSpace` {
`Custom`,
`Standard`,
`_UndefinedNameSpace` }
Defines if a node name is standard or custom.
- enum `EStandardNameSpace` {
`None`,
`GEV`,
`IIDC`,
`CL`,
`USB`,
`_UndefinedStandardNameSpace` }
Defines from which standard namespace a node name comes from.
- enum `EYesNo` {
`Yes` = 1,
`No` = 0,
`_UndefinedYesNo` = 2 }
Defines the choices of a Yes/No alternative.
- enum `ESlope` {
`Increasing`,
`Decreasing`,
`Varying`,
`Automatic`,
`_UndefinedESlope` }
typedef for formula type
- enum `EXMLValidation` {
`xvLoad` = 0x00000001L,
`xvCycles` = 0x00000002L,
`xvSFNC` = 0x00000004L,
`xvDefault` = 0x00000000L,
`xvAll` = 0xffffffffL,
`_UndefinedEXMLValidation` = 0x80000000L }
typedef describing the different validity checks which can be performed on an XML file
- enum `EDisplayNotation` {
`fnAutomatic`,
`fnFixed`,
`fnScientific`,
`_UndefinedEDisplayNotation` }
typedef for float notation
- enum `EInterfaceType` {
`intfIValue`,

```

intfIBase,
intfInteger,
intfBoolean,
intfCommand,
intfFloat,
intfString,
intfRegister,
intfCategory,
intfEnumeration,
intfEnumEntry,
intfIPort }

```

typedef for interface type

- enum [ELinkType](#) {
[ctParentNodes](#),
[ctReadingChildren](#),
[ctWritingChildren](#),
[ctInvalidatingChildren](#),
[ctDependingNodes](#),
[ctTerminalNodes](#) }

typedef for link type

- enum [EIncMode](#) {
[noIncrement](#),
[fixedIncrement](#),
[listIncrement](#) }

typedef for increment mode

- enum [EInputDirection](#) {
[idFrom](#),
[idTo](#),
[idNone](#) }

typedef for link type

- enum [EGenApiSchemaVersion](#) {
[v1_0](#) = 1,
[v1_1](#) = 2,
[_Undefined](#) = -1 }

GenApi schema version.

13.135.1 Detailed Description

13.135.2 Macro Definition Documentation

13.135.2.1 [_UndefinedRepresentation](#)

```
#define \_UndefinedRepresentation \_UndefinedRepresentation
```

13.135.3 Typedef Documentation

13.135.3.1 StringList_t

```
typedef GenICam::gcstring_vector StringList_t
```

A list of strings.

13.135.4 Enumeration Type Documentation

13.135.4.1 EAccessMode

```
enum EAccessMode
```

access mode of a node

Enumerator

NI	
NA	Not implemented.
WO	Not available.
RO	Write Only.
RW	Read Only.
_UndefinedAccesMode	Read and Write.
_CycleDetectAccesMode	Object is not yet initialized. used internally for AccessMode cycle detection

13.135.4.2 ECachingMode

```
enum ECachingMode
```

caching mode of a register

Enumerator

NoCache	
WriteThrough	Do not use cache.
WriteAround	Write to cache and register.
_UndefinedCachingMode	Write to register, write to cache on read. Not yet initialized

13.135.4.3 EDisplayNotation

```
enum EDisplayNotation
```

typedef for float notation

Enumerator

fnAutomatic	
fnFixed	the notation if either scientific or fixed depending on what is shorter
fnScientific	the notation is fixed, e.g. 123.4
_UndefinedEDisplayNotation	the notation is scientific, e.g. 1.234e2 Object is not yet initialized

13.135.4.4 EEndianess

enum [EEndianess](#)

Endianess of a value in a register.

Enumerator

BigEndian	
LittleEndian	Register is big endian.
_UndefinedEndian	Register is little endian. Object is not yet initialized

13.135.4.5 EGenApiSchemaVersion

enum [EGenApiSchemaVersion](#)

[GenApi](#) schema version.

Enumerator

v1_0	
v1_1	
_Undefined	

13.135.4.6 EIncMode

enum [EIncMode](#)

typedef for increment mode

Enumerator

noIncrement	
fixedIncrement	The feature has no increment.
listIncrement	The feature has a fix increment.

13.135.4.7 EInputDirection

enum [EInputDirection](#)

typedef for link type

Enumerator

idFrom	
idTo	Indicates a swiss knife that it is used as worker for a converter computing FROM.
idNone	Indicates a swiss knife that it is used as worker for a converter computing TO. SwissKnife is not used within a converter

13.135.4.8 EInterfaceType

enum [EInterfaceType](#)

typedef for interface type

Enumerator

intfIValue	
intfIBase	IValue interface.
intfIInteger	IBase interface.
intfIBoolean	IInteger interface.
intfICommand	IBoolean interface.
intfIFloat	ICommand interface.
intfIString	IFloat interface.
intfIRegister	IString interface.
intfICategory	IRegister interface.
intfIEnumeration	ICategory interface.
intfIEnumEntry	IEnumeration interface.
intfIPort	IEnumEntry interface. IPort interface

13.135.4.9 ELinkType

enum [ELinkType](#)

typedef for link type

Enumerator

ctParentNodes	
ctReadingChildren	All nodes for which this node is at least an invalidating child.
ctWritingChildren	All nodes which can be read from.
ctInvalidatingChildren	All nodes which can write a value further down the node stack.
ctDependingNodes	All directly connected nodes which invalidate this node.
ctTerminalNodes	All directly or indirectly connected nodes which are invalidated by this nodes (i.e. which are dependent on this node) All indirectly connected terminal nodes

13.135.4.10 ENameSpace

enum [ENameSpace](#)

Defines if a node name is standard or custom.

Enumerator

Custom	
Standard	name resides in custom namespace
_UndefinedNameSpace	name resides in one of the standard namespaces Object is not yet initialized

13.135.4.11 ERepresentation

enum [ERepresentation](#)

recommended representation of a node value

Enumerator

Linear	
Logarithmic	Slider with linear behavior.
Boolean	Slider with logarithmic behavior.
PureNumber	Check box.
HexNumber	Decimal number in an edit control.
IPV4Address	Hex number in an edit control.
MACAddress	IP-Address.
_UndefinedRepresentation	MAC-Address.

13.135.4.12 ESign

enum [ESign](#)

signed or unsigned integers

Enumerator

Signed	
Unsigned	Integer is signed.
_UndefinedSign	Integer is unsigned. Object is not yet initialized

13.135.4.13 ESlope

enum [ESlope](#)

typedef for formula type

Enumerator

Increasing	
Decreasing	strictly monotonous increasing
Varying	strictly monotonous decreasing
Automatic	slope changes, e.g. at run-time
_UndefinedESlope	slope is determined automatically by probing the function Object is not yet initialized

13.135.4.14 EStandardNameSpace

enum [EStandardNameSpace](#)

Defines from which standard namespace a node name comes from.

Enumerator

None	
GEV	name resides in custom namespace
IIDC	name resides in GigE Vision namespace
CL	name resides in 1394 IIDC namespace
USB	name resides in camera link namespace
_UndefinedStandardNameSpace	name resides in USB namespace Object is not yet initialized

13.135.4.15 EVisibility

enum [EVisibility](#)

recommended visibility of a node

Enumerator

Beginner	
Expert	Always visible.
Guru	Visible for experts or Gurus.
Invisible	Visible for Gurus.
_UndefinedVisibility	Not Visible.

13.135.4.16 EXMLValidation

enum [EXMLValidation](#)

typedef describing the different validity checks which can be performed on an XML file

The enum values for a bit field of length uint32_t

Enumerator

xvLoad	
xvCycles	Creates a dummy node map.
xvSFNC	checks for write and dependency cycles (implies xvLoad)
xvDefault	checks for conformance with the standard feature naming convention (SFNC)
xvAll	checks performed if nothing else is said
_UndefinedEXMLValidation	all possible checks

13.135.4.17 EYesNo

enum [EYesNo](#)

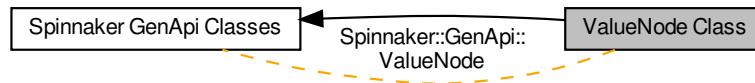
Defines the choices of a Yes/No alternative.

Enumerator

Yes	
No	yes
_UndefinedYesNo	no

13.136 ValueNode Class

Collaboration diagram for ValueNode Class:



Classes

- class [ValueNode](#)
Interface for value properties.

Typedefs

- typedef [ValueNode](#) [CValueRef](#)

13.136.1 Detailed Description

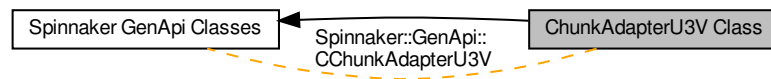
13.136.2 Typedef Documentation

13.136.2.1 CValueRef

```
typedef ValueNode CValueRef
```

13.137 ChunkAdapterU3V Class

Collaboration diagram for ChunkAdapterU3V Class:



Classes

- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.

13.137.1 Detailed Description

Chapter 14

Namespace Documentation

14.1 AdapterConfig Namespace Reference

Classes

- struct [AdapterInfo](#)
- struct [IpInfo](#)

Enumerations

- enum [AdapterConfigErr](#) {
 [IP_ADDRESS_INVALID](#),
 [IP_ADDRESS_IS_NOT_V4](#),
 [IP_ADDRESS_TOO_LARGE](#),
 [IP_ADDRESS_TOO_SMALL](#),
 [HOST_ADDRESS_ZERO](#),
 [SUBNET_MASK_INVALID](#),
 [VALID_SUBNET_NOT_FOUND](#) }

Functions

- [ADAPTERCONFIG_API](#) std::vector< [AdapterInfo](#) > [RetrieveAllAdapters](#) ()
- [ADAPTERCONFIG_API](#) void [AutoPopulateAdapterInfo](#) (std::vector< [AdapterInfo](#) > &adaptersToConfigure, const std::vector< [AdapterInfo](#) > &allAdapters)
- [ADAPTERCONFIG_API](#) void [AutoPopulateAdvancedProperties](#) (std::vector< [AdapterInfo](#) > &adaptersToConfigure)
- [ADAPTERCONFIG_API](#) void [PopulateAdapterIpInfo](#) ([IpInfo](#) startingIpInfo, std::vector< [AdapterInfo](#) > &adaptersToConfigure, const std::vector< [AdapterInfo](#) > &allAdapters, bool skipIfIPValid=false)
- [ADAPTERCONFIG_API](#) void [ValidateIpAddress](#) (const std::string &ipAddr, unsigned int subnetMaskLength)
- [ADAPTERCONFIG_API](#) bool [IsValidIpAddress](#) (const std::string &ipAddr)
- [ADAPTERCONFIG_API](#) bool [IsValidSubnetMask](#) (const std::string &subnetMask)
- [ADAPTERCONFIG_API](#) bool [IsOnSameSubnet](#) (const std::string &ipAddrStr1, const std::string &ipAddrStr2, const unsigned int subnetMaskLength)
- [ADAPTERCONFIG_API](#) unsigned int [GetSubnetMaskLength](#) (const std::string &subnetMask)
- [ADAPTERCONFIG_API](#) std::string [GetEnumerationLogFileName](#) ()
- [ADAPTERCONFIG_API](#) std::string [GetConfigLogFileName](#) ()

- [ADAPTERCONFIG_API](#) void [ConfigureAdapter](#) ([AdapterInfo](#) &adapter, bool configureIP, bool configureAdvancedProperties)
- [ADAPTERCONFIG_API](#) unsigned int [GetAutoSubnetMaskLength](#) ()
- [ADAPTERCONFIG_API](#) std::string [GetAutoSubnetMask](#) ()
- [ADAPTERCONFIG_API](#) std::string [GetMaxIpAddress](#) ()
- [ADAPTERCONFIG_API](#) std::string [GetMinIpAddress](#) ()
- [ADAPTERCONFIG_API](#) std::string [GetAutoGigabitDesc](#) ()
- [ADAPTERCONFIG_API](#) std::string [GetAuto10GDesc](#) ()
- [ADAPTERCONFIG_API](#) std::string [GetAutoStartIp](#) ()

14.1.1 Enumeration Type Documentation

14.1.1.1 AdapterConfigErr

```
enum AdapterConfigErr
```

Enumerator

IP_ADDRESS_INVALID	
IP_ADDRESS_IS_NOT_V4	
IP_ADDRESS_TOO_LARGE	
IP_ADDRESS_TOO_SMALL	
HOST_ADDRESS_ZERO	
SUBNET_MASK_INVALID	
VALID_SUBNET_NOT_FOUND	

14.1.2 Function Documentation

14.1.2.1 AutoPopulateAdapterInfo()

```
ADAPTERCONFIG\_API void AdapterConfig::AutoPopulateAdapterInfo (
    std::vector< AdapterInfo > & adaptersToConfigure,
    const std::vector< AdapterInfo > & allAdapters )
```

14.1.2.2 AutoPopulateAdvancedProperties()

```
ADAPTERCONFIG\_API void AdapterConfig::AutoPopulateAdvancedProperties (
    std::vector< AdapterInfo > & adaptersToConfigure )
```

14.1.2.3 ConfigureAdapter()

```
ADAPTERCONFIG_API void AdapterConfig::ConfigureAdapter (
    AdapterInfo & adapter,
    bool configureIP,
    bool configureAdvancedProperties )
```

14.1.2.4 GetAuto10GDesc()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAuto10GDesc ( )
```

14.1.2.5 GetAutoGigabitDesc()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoGigabitDesc ( )
```

14.1.2.6 GetAutoStartIp()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoStartIp ( )
```

14.1.2.7 GetAutoSubnetMask()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoSubnetMask ( )
```

14.1.2.8 GetAutoSubnetMaskLength()

```
ADAPTERCONFIG_API unsigned int AdapterConfig::GetAutoSubnetMaskLength ( )
```

14.1.2.9 GetConfigLogFileName()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetConfigLogFileName ( )
```

14.1.2.10 GetEnumerationLogFileName()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetEnumerationLogFileName ( )
```

14.1.2.11 GetMaxIpAddress()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetMaxIpAddress ( )
```

14.1.2.12 GetMinIpAddress()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetMinIpAddress ( )
```

14.1.2.13 GetSubnetMaskLength()

```
ADAPTERCONFIG_API unsigned int AdapterConfig::GetSubnetMaskLength (
    const std::string & subnetMask )
```

14.1.2.14 IsOnSameSubnet()

```
ADAPTERCONFIG_API bool AdapterConfig::IsOnSameSubnet (
    const std::string & ipAddrStr1,
    const std::string & ipAddrStr2,
    const unsigned int subnetMaskLength )
```

14.1.2.15 IsValidIpAddress()

```
ADAPTERCONFIG_API bool AdapterConfig::IsValidIpAddress (
    const std::string & ipAddr )
```

14.1.2.16 IsValidSubnetMask()

```
ADAPTERCONFIG_API bool AdapterConfig::IsValidSubnetMask (
    const std::string & subnetMask )
```


14.1.2.17 PopulateAdapterIpInfo()

```
ADAPTERCONFIG_API void AdapterConfig::PopulateAdapterIpInfo (
    IpInfo startingIpInfo,
    std::vector< AdapterInfo > & adaptersToConfigure,
    const std::vector< AdapterInfo > & allAdapters,
    bool skipIfIPValid = false )
```

14.1.2.18 RetrieveAllAdapters()

```
ADAPTERCONFIG_API std::vector<AdapterInfo> AdapterConfig::RetrieveAllAdapters ( )
```

14.1.2.19 ValidateIpAddress()

```
ADAPTERCONFIG_API void AdapterConfig::ValidateIpAddress (
    const std::string & ipAddr,
    unsigned int subnetMaskLength )
```

14.2 Conversion Namespace Reference

Functions

- string [NumToCString](#) (int number)
- string [NumToCString](#) (double number)
- string [NumToCString](#) (float number)

14.2.1 Function Documentation

14.2.1.1 NumToCString() [1/3]

```
string NumToCString (
    int number )
```

14.2.1.2 NumToCString() [2/3]

```
string NumToCString (
    double number )
```

14.2.1.3 NumToCString() [3/3]

```
string Conversion::NumToCString (
    float number )
```

14.3 CpuUtil Namespace Reference

Classes

- struct [CpuUsageInfo](#)

Functions

- bool [StartCpuTracing](#) ([CpuUsageInfo](#) *cpuUsage)
- bool [StopCpuTracing](#) ([CpuUsageInfo](#) *cpuUsage)
- std::string [GetCpuStats](#) ([CpuUsageInfo](#) *cpuUsage)

14.3.1 Function Documentation

14.3.1.1 GetCpuStats()

```
std::string GetCpuStats (
    CpuUsageInfo * cpuUsage )
```

14.3.1.2 StartCpuTracing()

```
bool StartCpuTracing (
    CpuUsageInfo * cpuUsage )
```

14.3.1.3 StopCpuTracing()

```
bool StopCpuTracing (
    CpuUsageInfo * cpuUsage )
```

14.4 PerformanceCounter Namespace Reference

Functions

- void [StartPerformanceCounter](#) ()
- double [GetPerformanceCounter](#) ()

Variables

- double [PCFreq](#)
- `__int64` [CounterStart](#)

14.4.1 Function Documentation

14.4.1.1 GetPerformanceCounter()

```
double GetPerformanceCounter ( )
```

14.4.1.2 StartPerformanceCounter()

```
void StartPerformanceCounter ( )
```

14.4.2 Variable Documentation

14.4.2.1 CounterStart

```
__int64 CounterStart
```

14.4.2.2 PCFreq

```
double PCFreq
```

14.5 SecondsCounter Namespace Reference

Functions

- void [StartSecondsCounter](#) ()
- int [GetSecondsCounter](#) ()

Variables

- `time_t` [startTime](#)
- `time_t` [endTime](#)
- `double` [timeDiff](#)

14.5.1 Function Documentation

14.5.1.1 GetSecondsCounter()

```
int GetSecondsCounter ( )
```

14.5.1.2 StartSecondsCounter()

```
void StartSecondsCounter ( )
```

14.5.2 Variable Documentation

14.5.2.1 endTime

```
time_t endTime
```

14.5.2.2 startTime

```
time_t startTime
```

14.5.2.3 timeDiff

```
double timeDiff
```

14.6 Spinnaker Namespace Reference

Namespaces

- [GenApi](#)
- [GenICam](#)
- [Video](#)

Classes

- struct [ActionCommandResult](#)
Action Command Result.
- class [BasePtr](#)
The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.
- struct [BMPOption](#)
Options for saving Bitmap image.
- class [Camera](#)
The camera object class.
- class [CameraBase](#)
The base class for the camera object.
- class [CameraList](#)
Used to hold a list of camera objects.
- class [CameraPtr](#)
A reference tracked pointer to a camera object.
- struct [CCMSettings](#)
- class [ChunkData](#)
The chunk data which contains additional information about an image.
- class [DeviceArrivalEventHandler](#)
An event handler for capturing the device arrival event.
- struct [DeviceEventExposureEndData](#)
Data Fields for Device Event payload for EventExposureEnd.
- class [DeviceEventHandler](#)
A handler to device events.
- struct [DeviceEventInferenceData](#)
Data Fields for Device Event payload for EventInference.
- class [DeviceEventUtility](#)
- class [DeviceRemovalEventHandler](#)
An event handler for capturing the device removal event.
- class [EventHandler](#)
The base class for all event handler types.
- class [Exception](#)
The [Exception](#) object represents an error that is returned from the library.
- class [ICameraBase](#)
The interface file for base class for the camera object.
- class [ICameraList](#)
Used to hold a list of camera objects.
- class [IChunkData](#)
The [Interface](#) file for [ChunkData](#).
- class [IDataStream](#)
- class [IDeviceArrivalEventHandler](#)

- class [IDeviceEventHandler](#)
- class [IDeviceRemovalEventHandler](#)
- class [IImage](#)
 - The interface file for [Image](#).*
- class [IImageEventHandler](#)
- class [IImageStatistics](#)
 - The interface file for image statistics.*
- class [IInterface](#)
 - The interface file for [Interface](#).*
- class [IInterfaceArrivalEventHandler](#)
- class [IInterfaceEventHandler](#)
- class [IInterfaceList](#)
 - The interface file for [InterfaceList](#) class.*
- class [IInterfaceRemovalEventHandler](#)
- class [ILoggingEventHandler](#)
- class [Image](#)
 - The image object class.*
- class [ImageEventHandler](#)
 - A handler for capturing image arrival events.*
- class [ImagePtr](#)
 - A reference tracked pointer to an image object.*
- class [ImageStatistics](#)
 - Represents image statistics for an image.*
- class [ImageUtility](#)
 - Static helper functions for the image object class.*
- class [ImageUtilityCCM](#)
 - Static function to create color corrected images from an image object.*
- class [ImageUtilityHeatmap](#)
 - Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.*
- class [ImageUtilityPolarization](#)
 - Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.*
- class [InferenceBoundingBoxResult](#)
 - An inference bounding boxes object which holds information about the detected bounding boxes.*
- struct [InferenceBoxCircle](#)
- struct [InferenceBoxRect](#)
 - Inference Bounding Box Type Data Structures.*
- struct [InferenceBoxRotatedRect](#)
- class [Interface](#)
 - An interface object which holds a list of cameras.*
- class [InterfaceArrivalEventHandler](#)
 - An event handler for capturing the interface arrival event.*
- class [InterfaceEventHandler](#)
 - A handler to device arrival and removal events on all interfaces.*
- class [InterfaceList](#)
 - A list of the available interfaces on the system.*
- class [InterfacePtr](#)
 - A reference tracked pointer to the interface object.*
- class [InterfaceRemovalEventHandler](#)
 - An event handler for capturing the interface removal event.*
- class [ISystem](#)
 - The interface file for [System](#).*

- class [ISystemEventHandler](#)
- struct [JPEGOption](#)
Options for saving JPEG image.
- struct [JPG2Option](#)
Options for saving JPEG2000 image.
- struct [LibraryVersion](#)
Provides easier access to the current version of [Spinnaker](#).
- class [LoggingEventData](#)
The [LoggingEventData](#) object.
- class [LoggingEventDataPtr](#)
A reference tracked pointer to the [LoggingEvent](#) object.
- class [LoggingEventHandler](#)
An event handler for capturing the device logging event.
- struct [PGMOption](#)
Options for saving PGM images.
- struct [PNGOption](#)
Options for saving PNG images.
- struct [PPMOption](#)
Options for saving PPM images.
- class [System](#)
The system object is used to retrieve the list of interfaces and cameras available.
- class [SystemEventHandler](#)
A handler to interface arrival and removal events on the system.
- class [SystemPtr](#)
A reference tracked pointer to a system object.
- struct [TIFFOption](#)
Options for saving TIFF images.
- class [TransportLayerDevice](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerInterface](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerStream](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerSystem](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Enumerations

- enum [LUTSelectorEnums](#) {
 [LUTSelector_LUT1](#),
 [NUM_LUTSELECTOR](#) }
- The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.*
- enum [ExposureModeEnums](#) {
 [ExposureMode_Timed](#),
 [ExposureMode_TriggerWidth](#),
 [NUM_EXPOSUREMODE](#) }
- enum [AcquisitionModeEnums](#) {
 [AcquisitionMode_Continuous](#),
 [AcquisitionMode_SingleFrame](#),
 [AcquisitionMode_MultiFrame](#),
 [NUM_ACQUISITIONMODE](#) }

- enum [TriggerSourceEnums](#) {
 [TriggerSource_Software](#),
 [TriggerSource_Line0](#),
 [TriggerSource_Line1](#),
 [TriggerSource_Line2](#),
 [TriggerSource_Line3](#),
 [TriggerSource_UserOutput0](#),
 [TriggerSource_UserOutput1](#),
 [TriggerSource_UserOutput2](#),
 [TriggerSource_UserOutput3](#),
 [TriggerSource_Counter0Start](#),
 [TriggerSource_Counter1Start](#),
 [TriggerSource_Counter0End](#),
 [TriggerSource_Counter1End](#),
 [TriggerSource_LogicBlock0](#),
 [TriggerSource_LogicBlock1](#),
 [TriggerSource_Action0](#),
 [NUM_TRIGGERSOURCE](#) }
- enum [TriggerActivationEnums](#) {
 [TriggerActivation_LevelLow](#),
 [TriggerActivation_LevelHigh](#),
 [TriggerActivation_FallingEdge](#),
 [TriggerActivation_RisingEdge](#),
 [TriggerActivation_AnyEdge](#),
 [NUM_TRIGGERACTIVATION](#) }
- enum [SensorShutterModeEnums](#) {
 [SensorShutterMode_Global](#),
 [SensorShutterMode_Rolling](#),
 [SensorShutterMode_GlobalReset](#),
 [NUM_SENSORSHUTTERMODE](#) }
- enum [TriggerModeEnums](#) {
 [TriggerMode_Off](#),
 [TriggerMode_On](#),
 [NUM_TRIGGERMODE](#) }
- enum [TriggerOverlapEnums](#) {
 [TriggerOverlap_Off](#),
 [TriggerOverlap_ReadOut](#),
 [TriggerOverlap_PreviousFrame](#),
 [NUM_TRIGGEROVERLAP](#) }
- enum [TriggerSelectorEnums](#) {
 [TriggerSelector_AcquisitionStart](#),
 [TriggerSelector_FrameStart](#),
 [TriggerSelector_FrameBurstStart](#),
 [NUM_TRIGGERSELECTOR](#) }
- enum [ExposureAutoEnums](#) {
 [ExposureAuto_Off](#),
 [ExposureAuto_Once](#),
 [ExposureAuto_Continuous](#),
 [NUM_EXPOSUREAUTO](#) }
- enum [EventSelectorEnums](#) {
 [EventSelector_Error](#),
 [EventSelector_ExposureEnd](#),
 [EventSelector_SerialPortReceive](#),
 [NUM_EVENTSELECTOR](#) }
- enum [EventNotificationEnums](#) {
 [EventNotification_On](#),
 [EventNotification_Off](#),
 [NUM_EVENTNOTIFICATION](#) }

- enum [LogicBlockSelectorEnums](#) {
 [LogicBlockSelector_LogicBlock0](#),
 [LogicBlockSelector_LogicBlock1](#),
 [NUM_LOGICBLOCKSELECTOR](#) }
- enum [LogicBlockLUTInputActivationEnums](#) {
 [LogicBlockLUTInputActivation_LevelLow](#),
 [LogicBlockLUTInputActivation_LevelHigh](#),
 [LogicBlockLUTInputActivation_FallingEdge](#),
 [LogicBlockLUTInputActivation_RisingEdge](#),
 [LogicBlockLUTInputActivation_AnyEdge](#),
 [NUM_LOGICBLOCKLUTINPUTACTIVATION](#) }
- enum [LogicBlockLUTInputSelectorEnums](#) {
 [LogicBlockLUTInputSelector_Input0](#),
 [LogicBlockLUTInputSelector_Input1](#),
 [LogicBlockLUTInputSelector_Input2](#),
 [LogicBlockLUTInputSelector_Input3](#),
 [NUM_LOGICBLOCKLUTINPUTSELECTOR](#) }
- enum [LogicBlockLUTInputSourceEnums](#) {
 [LogicBlockLUTInputSource_Zero](#),
 [LogicBlockLUTInputSource_Line0](#),
 [LogicBlockLUTInputSource_Line1](#),
 [LogicBlockLUTInputSource_Line2](#),
 [LogicBlockLUTInputSource_Line3](#),
 [LogicBlockLUTInputSource_UserOutput0](#),
 [LogicBlockLUTInputSource_UserOutput1](#),
 [LogicBlockLUTInputSource_UserOutput2](#),
 [LogicBlockLUTInputSource_UserOutput3](#),
 [LogicBlockLUTInputSource_Counter0Start](#),
 [LogicBlockLUTInputSource_Counter1Start](#),
 [LogicBlockLUTInputSource_Counter0End](#),
 [LogicBlockLUTInputSource_Counter1End](#),
 [LogicBlockLUTInputSource_LogicBlock0](#),
 [LogicBlockLUTInputSource_LogicBlock1](#),
 [LogicBlockLUTInputSource_ExposureStart](#),
 [LogicBlockLUTInputSource_ExposureEnd](#),
 [LogicBlockLUTInputSource_FrameTriggerWait](#),
 [LogicBlockLUTInputSource_AcquisitionActive](#),
 [NUM_LOGICBLOCKLUTINPUTSOURCE](#) }
- enum [LogicBlockLUTSelectorEnums](#) {
 [LogicBlockLUTSelector_Value](#),
 [LogicBlockLUTSelector_Enable](#),
 [NUM_LOGICBLOCKLUTSELECTOR](#) }
- enum [ColorTransformationSelectorEnums](#) {
 [ColorTransformationSelector_RGBtoRGB](#),
 [ColorTransformationSelector_RGBtoYUV](#),
 [NUM_COLORTRANSFORMATIONSELECTOR](#) }
- enum [RgbTransformLightSourceEnums](#) {
 [RgbTransformLightSource_General](#),
 [RgbTransformLightSource_Tungsten2800K](#),
 [RgbTransformLightSource_WarmFluorescent3000K](#),
 [RgbTransformLightSource_CoolFluorescent4000K](#),
 [RgbTransformLightSource_Daylight5000K](#),
 [RgbTransformLightSource_Cloudy6500K](#),
 [RgbTransformLightSource_Shade8000K](#),
 [RgbTransformLightSource_Custom](#),
 [NUM_RGBTRANSFORMLIGHTSOURCE](#) }
- enum [ColorTransformationValueSelectorEnums](#) {
 [ColorTransformationValueSelector_Gain00](#),

```

ColorTransformationValueSelector_Gain01,
ColorTransformationValueSelector_Gain02,
ColorTransformationValueSelector_Gain10,
ColorTransformationValueSelector_Gain11,
ColorTransformationValueSelector_Gain12,
ColorTransformationValueSelector_Gain20,
ColorTransformationValueSelector_Gain21,
ColorTransformationValueSelector_Gain22,
ColorTransformationValueSelector_Offset0,
ColorTransformationValueSelector_Offset1,
ColorTransformationValueSelector_Offset2,
NUM_COLORTRANSFORMATIONVALUESELECTOR }

• enum DeviceRegistersEndiannessEnums {
    DeviceRegistersEndianness_Little,
    DeviceRegistersEndianness_Big,
    NUM_DEVICEREGISTERSENDIANNESSENUMS }

• enum DeviceScanTypeEnum {
    DeviceScanType_Areascan,
    NUM_DEVICESCANTYPE }

• enum DeviceCharacterSetEnums {
    DeviceCharacterSet_UTF8,
    DeviceCharacterSet_ASCII,
    NUM_DEVICECHARACTERSET }

• enum DeviceTLTypeEnum {
    DeviceTLType_GigEVision,
    DeviceTLType_CameraLink,
    DeviceTLType_CameraLinkHS,
    DeviceTLType_CoaXPress,
    DeviceTLType_USB3Vision,
    DeviceTLType_Custom,
    NUM_DEVICETLTYPE }

• enum DevicePowerSupplySelectorEnums {
    DevicePowerSupplySelector_External,
    NUM_DEVICEPOWERSUPPLYSELECTOR }

• enum DeviceTemperatureSelectorEnums {
    DeviceTemperatureSelector_Sensor,
    NUM_DEVICETEMPERATURESELECTOR }

• enum DeviceIndicatorModeEnums {
    DeviceIndicatorMode_Inactive,
    DeviceIndicatorMode_Active,
    DeviceIndicatorMode_ErrorStatus,
    NUM_DEVICEINDICATORMODE }

• enum AutoExposureControlPriorityEnums {
    AutoExposureControlPriority_Gain,
    AutoExposureControlPriority_ExposureTime,
    NUM_AUTOEXPOSURECONTROLPRIORITY }

• enum AutoExposureMeteringModeEnums {
    AutoExposureMeteringMode_Average,
    AutoExposureMeteringMode_Spot,
    AutoExposureMeteringMode_Partial,
    AutoExposureMeteringMode_CenterWeighted,
    AutoExposureMeteringMode_HistogramPeak,
    NUM_AUTOEXPOSUREMETERINGMODE }

• enum BalanceWhiteAutoProfileEnums {
    BalanceWhiteAutoProfile_Indoor,
    BalanceWhiteAutoProfile_Outdoor,
    NUM_BALANCEWHITEAUTOPROFILE }

```

- enum [AutoAlgorithmSelectorEnums](#) {
[AutoAlgorithmSelector_Awb](#),
[AutoAlgorithmSelector_Ae](#),
[NUM_AUTOALGORITHMSELECTOR](#) }
- enum [AutoExposureTargetGreyValueAutoEnums](#) {
[AutoExposureTargetGreyValueAuto_Off](#),
[AutoExposureTargetGreyValueAuto_Continuous](#),
[NUM_AUTOEXPOSURETARGETGREYVALUEAUTO](#) }
- enum [AutoExposureLightingModeEnums](#) {
[AutoExposureLightingMode_AutoDetect](#),
[AutoExposureLightingMode_Backlight](#),
[AutoExposureLightingMode_Frontlight](#),
[AutoExposureLightingMode_Normal](#),
[NUM_AUTOEXPOSURELIGHTINGMODE](#) }
- enum [GevIEEE1588StatusEnums](#) {
[GevIEEE1588Status_Initializing](#),
[GevIEEE1588Status_Faulty](#),
[GevIEEE1588Status_Disabled](#),
[GevIEEE1588Status_Listening](#),
[GevIEEE1588Status_PreMaster](#),
[GevIEEE1588Status_Master](#),
[GevIEEE1588Status_Passive](#),
[GevIEEE1588Status_Uncalibrated](#),
[GevIEEE1588Status_Slave](#),
[NUM_GEVIEEE1588STATUS](#) }
- enum [GevIEEE1588ModeEnums](#) {
[GevIEEE1588Mode_Auto](#),
[GevIEEE1588Mode_SlaveOnly](#),
[NUM_GEVIEEE1588MODE](#) }
- enum [GevIEEE1588ClockAccuracyEnums](#) {
[GevIEEE1588ClockAccuracy_Unknown](#),
[NUM_GEVIEEE1588CLOCKACCURACY](#) }
- enum [GevCCPEnums](#) {
[GevCCP_OpenAccess](#),
[GevCCP_ExclusiveAccess](#),
[GevCCP_ControlAccess](#),
[NUM_GEVCCP](#) }
- enum [GevSupportedOptionSelectorEnums](#) {
[GevSupportedOptionSelector_UserDefinedName](#),
[GevSupportedOptionSelector_SerialNumber](#),
[GevSupportedOptionSelector_HeartbeatDisable](#),
[GevSupportedOptionSelector_LinkSpeed](#),
[GevSupportedOptionSelector_CCPApplicationSocket](#),
[GevSupportedOptionSelector_ManifestTable](#),
[GevSupportedOptionSelector_TestData](#),
[GevSupportedOptionSelector_DiscoveryAckDelay](#),
[GevSupportedOptionSelector_DiscoveryAckDelayWritable](#),
[GevSupportedOptionSelector_ExtendedStatusCodes](#),
[GevSupportedOptionSelector_Action](#),
[GevSupportedOptionSelector_PendingAck](#),
[GevSupportedOptionSelector_EventData](#),
[GevSupportedOptionSelector_Event](#),
[GevSupportedOptionSelector_PacketResend](#),
[GevSupportedOptionSelector_WriteMem](#),
[GevSupportedOptionSelector_CommandsConcatenation](#),
[GevSupportedOptionSelector_IPConfigurationLLA](#),
[GevSupportedOptionSelector_IPConfigurationDHCP](#),
[GevSupportedOptionSelector_IPConfigurationPersistentIP](#),

```
GevSupportedOptionSelector_StreamChannelSourceSocket,  
GevSupportedOptionSelector_MessageChannelSourceSocket,  
NUM_GEVSUPPORTEDOPTIONSELECTOR }  
  
• enum BlackLevelSelectorEnums {  
    BlackLevelSelector_All,  
    BlackLevelSelector_Analog,  
    BlackLevelSelector_Digital,  
    NUM_BLACKLEVELSELECTOR }  
  
• enum BalanceWhiteAutoEnums {  
    BalanceWhiteAuto_Off,  
    BalanceWhiteAuto_Once,  
    BalanceWhiteAuto_Continuous,  
    NUM_BALANCEWHITEAUTO }  
  
• enum GainAutoEnums {  
    GainAuto_Off,  
    GainAuto_Once,  
    GainAuto_Continuous,  
    NUM_GAINAUTO }  
  
• enum BalanceRatioSelectorEnums {  
    BalanceRatioSelector_Red,  
    BalanceRatioSelector_Blue,  
    NUM_BALANCERATIOSELECTOR }  
  
• enum GainSelectorEnums {  
    GainSelector_All,  
    NUM_GAINSELECTOR }  
  
• enum DefectCorrectionModeEnums {  
    DefectCorrectionMode_Average,  
    DefectCorrectionMode_Highlight,  
    DefectCorrectionMode_Zero,  
    NUM_DEFECTCORRECTIONMODE }  
  
• enum UserSetSelectorEnums {  
    UserSetSelector_Default,  
    UserSetSelector_UserSet0,  
    UserSetSelector_UserSet1,  
    NUM_USERSETSELECTOR }  
  
• enum UserSetDefaultEnums {  
    UserSetDefault_Default,  
    UserSetDefault_UserSet0,  
    UserSetDefault_UserSet1,  
    NUM_USERSETDEFAULT }  
  
• enum SerialPortBaudRateEnums {  
    SerialPortBaudRate_Baud300,  
    SerialPortBaudRate_Baud600,  
    SerialPortBaudRate_Baud1200,  
    SerialPortBaudRate_Baud2400,  
    SerialPortBaudRate_Baud4800,  
    SerialPortBaudRate_Baud9600,  
    SerialPortBaudRate_Baud14400,  
    SerialPortBaudRate_Baud19200,  
    SerialPortBaudRate_Baud38400,  
    SerialPortBaudRate_Baud57600,  
    SerialPortBaudRate_Baud115200,  
    SerialPortBaudRate_Baud230400,  
    SerialPortBaudRate_Baud460800,  
    SerialPortBaudRate_Baud921600,  
    NUM_SERIALPORTBAUDRATE }  
  
• enum SerialPortParityEnums {  
    SerialPortParity_None,
```

```
SerialPortParity_Odd,  
SerialPortParity_Even,  
SerialPortParity_Mark,  
SerialPortParity_Space,  
NUM_SERIALPORTPARITY }  
• enum SerialPortSelectorEnums {  
    SerialPortSelector_SerialPort0,  
    NUM_SERIALPORTSELECTOR }  
• enum SerialPortStopBitsEnums {  
    SerialPortStopBits_Bits1,  
    SerialPortStopBits_Bits1AndAHalf,  
    SerialPortStopBits_Bits2,  
    NUM_SERIALPORTSTOPBITS }  
• enum SerialPortSourceEnums {  
    SerialPortSource_Line0,  
    SerialPortSource_Line1,  
    SerialPortSource_Line2,  
    SerialPortSource_Line3,  
    SerialPortSource_Off,  
    NUM_SERIALPORTSOURCE }  
• enum SequencerModeEnums {  
    SequencerMode_Off,  
    SequencerMode_On,  
    NUM_SEQUENCERMODE }  
• enum SequencerConfigurationValidEnums {  
    SequencerConfigurationValid_No,  
    SequencerConfigurationValid_Yes,  
    NUM_SEQUENCERCONFIGURATIONVALID }  
• enum SequencerSetValidEnums {  
    SequencerSetValid_No,  
    SequencerSetValid_Yes,  
    NUM_SEQUENCERSETVALID }  
• enum SequencerTriggerActivationEnums {  
    SequencerTriggerActivation_RisingEdge,  
    SequencerTriggerActivation_FallingEdge,  
    SequencerTriggerActivation_AnyEdge,  
    SequencerTriggerActivation_LevelHigh,  
    SequencerTriggerActivation_LevelLow,  
    NUM_SEQUENCERTRIGGERACTIVATION }  
• enum SequencerConfigurationModeEnums {  
    SequencerConfigurationMode_Off,  
    SequencerConfigurationMode_On,  
    NUM_SEQUENCERCONFIGURATIONMODE }  
• enum SequencerTriggerSourceEnums {  
    SequencerTriggerSource_Off,  
    SequencerTriggerSource_FrameStart,  
    NUM_SEQUENCERTRIGGERSOURCE }  
• enum TransferQueueModeEnums {  
    TransferQueueMode_FirstInFirstOut,  
    NUM_TRANSFERQUEUEMODE }  
• enum TransferOperationModeEnums {  
    TransferOperationMode_Continuous,  
    TransferOperationMode_MultiBlock,  
    NUM_TRANSFEROPERATIONMODE }  
• enum TransferControlModeEnums {  
    TransferControlMode_Basic,  
    TransferControlMode_Automatic,
```

```

TransferControlMode_UserControlled,
NUM_TRANSFERCONTROLMODE }

• enum ChunkGainSelectorEnums {
    ChunkGainSelector_All,
    ChunkGainSelector_Red,
    ChunkGainSelector_Green,
    ChunkGainSelector_Blue,
    NUM_CHUNKGAINSELECTOR }

• enum ChunkSelectorEnums {
    ChunkSelector_Image,
    ChunkSelector_CRC,
    ChunkSelector_FrameID,
    ChunkSelector_OffsetX,
    ChunkSelector_OffsetY,
    ChunkSelector_Width,
    ChunkSelector_Height,
    ChunkSelector_ExposureTime,
    ChunkSelector_Gain,
    ChunkSelector_BlackLevel,
    ChunkSelector_PixelFormat,
    ChunkSelector_Timestamp,
    ChunkSelector_SequencerSetActive,
    ChunkSelector_SerialData,
    ChunkSelector_ExposureEndLineStatusAll,
    NUM_CHUNKSELECTOR }

• enum ChunkBlackLevelSelectorEnums {
    ChunkBlackLevelSelector_All,
    NUM_CHUNKBLACKLEVELSELECTOR }

• enum ChunkPixelFormatEnums {
    ChunkPixelFormat_Mono8,
    ChunkPixelFormat_Mono12Packed,
    ChunkPixelFormat_Mono16,
    ChunkPixelFormat_RGB8Packed,
    ChunkPixelFormat_YUV422Packed,
    ChunkPixelFormat_BayerGR8,
    ChunkPixelFormat_BayerRG8,
    ChunkPixelFormat_BayerGB8,
    ChunkPixelFormat_BayerBG8,
    ChunkPixelFormat_YCbCr601_422_8_CbYCrY,
    NUM_CHUNKPIXELFORMAT }

• enum FileOperationStatusEnums {
    FileOperationStatus_Success,
    FileOperationStatus_Failure,
    FileOperationStatus_Overflow,
    NUM_FILEOPERATIONSTATUS }

• enum FileOpenModeEnums {
    FileOpenMode_Read,
    FileOpenMode_Write,
    FileOpenMode_ReadWrite,
    NUM_FILEOPENMODE }

• enum FileOperationSelectorEnums {
    FileOperationSelector_Open,
    FileOperationSelector_Close,
    FileOperationSelector_Read,
    FileOperationSelector_Write,
    FileOperationSelector_Delete,
    NUM_FILEOPERATIONSELECTOR }

```

- enum FileSelectorEnums {
FileSelector_UserSetDefault,
FileSelector_UserSet0,
FileSelector_UserSet1,
FileSelector_UserFile1,
FileSelector_SerialPort0,
NUM_FILESELECTOR }
- enum BinningSelectorEnums {
BinningSelector_All,
BinningSelector_Sensor,
BinningSelector_ISP,
NUM_BINNINGSELECTOR }
- enum TestPatternGeneratorSelectorEnums {
TestPatternGeneratorSelector_Sensor,
TestPatternGeneratorSelector_PipelineStart,
NUM_TESTPATTERNGENERATORSELECTOR }
- enum CompressionSaturationPriorityEnums {
CompressionSaturationPriority_DropFrame,
CompressionSaturationPriority_ReduceFrameRate,
NUM_COMPRESSIONSATURATIONPRIORITY }
- enum TestPatternEnums {
TestPattern_Off,
TestPattern_Increment,
TestPattern_SensorTestPattern,
NUM_TESTPATTERN }
- enum PixelColorFilterEnums {
PixelColorFilter_None,
PixelColorFilter_BayerRG,
PixelColorFilter_BayerGB,
PixelColorFilter_BayerGR,
PixelColorFilter_BayerBG,
NUM_PIXELCOLORFILTER }
- enum AdcBitDepthEnums {
AdcBitDepth_Bit8,
AdcBitDepth_Bit10,
AdcBitDepth_Bit12,
AdcBitDepth_Bit14,
NUM_ADCBITDEPTH }
- enum DecimationHorizontalModeEnums {
DecimationHorizontalMode_Discard,
NUM_DECIMATIONHORIZONTALMODE }
- enum BinningVerticalModeEnums {
BinningVerticalMode_Sum,
BinningVerticalMode_Average,
NUM_BINNINGVERTICALMODE }
- enum PixelSizeEnums {
PixelSize_Bpp1,
PixelSize_Bpp2,
PixelSize_Bpp4,
PixelSize_Bpp8,
PixelSize_Bpp10,
PixelSize_Bpp12,
PixelSize_Bpp14,
PixelSize_Bpp16,
PixelSize_Bpp20,
PixelSize_Bpp24,
PixelSize_Bpp30,
PixelSize_Bpp32,

```
PixelSize_Bpp36,  
PixelSize_Bpp48,  
PixelSize_Bpp64,  
PixelSize_Bpp96,  
NUM_PIXELSIZE }  
  
• enum DecimationSelectorEnums {  
    DecimationSelector_All,  
    DecimationSelector_Sensor,  
    NUM_DECIMATIONSELECTOR }  
  
• enum ImageCompressionModeEnums {  
    ImageCompressionMode_Off,  
    ImageCompressionMode_Lossless,  
    NUM_IMAGECOMPRESSIONMODE }  
  
• enum BinningHorizontalModeEnums {  
    BinningHorizontalMode_Sum,  
    BinningHorizontalMode_Average,  
    NUM_BINNINGHORIZONTALMODE }  
  
• enum PixelFormatEnums {  
    PixelFormat_Mono8,  
    PixelFormat_Mono16,  
    PixelFormat_RGB8Packed,  
    PixelFormat_BayerGR8,  
    PixelFormat_BayerRG8,  
    PixelFormat_BayerGB8,  
    PixelFormat_BayerBG8,  
    PixelFormat_BayerGR16,  
    PixelFormat_BayerRG16,  
    PixelFormat_BayerGB16,  
    PixelFormat_BayerBG16,  
    PixelFormat_Mono12Packed,  
    PixelFormat_BayerGR12Packed,  
    PixelFormat_BayerRG12Packed,  
    PixelFormat_BayerGB12Packed,  
    PixelFormat_BayerBG12Packed,  
    PixelFormat_YUV411Packed,  
    PixelFormat_YUV422Packed,  
    PixelFormat_YUV444Packed,  
    PixelFormat_Mono12p,  
    PixelFormat_BayerGR12p,  
    PixelFormat_BayerRG12p,  
    PixelFormat_BayerGB12p,  
    PixelFormat_BayerBG12p,  
    PixelFormat_YCbCr8,  
    PixelFormat_YCbCr422_8,  
    PixelFormat_YCbCr411_8,  
    PixelFormat_BGR8,  
    PixelFormat_BGRa8,  
    PixelFormat_Mono10Packed,  
    PixelFormat_BayerGR10Packed,  
    PixelFormat_BayerRG10Packed,  
    PixelFormat_BayerGB10Packed,  
    PixelFormat_BayerBG10Packed,  
    PixelFormat_Mono10p,  
    PixelFormat_BayerGR10p,  
    PixelFormat_BayerRG10p,  
    PixelFormat_BayerGB10p,  
    PixelFormat_BayerBG10p,  
    PixelFormat_Mono1p,
```


[PixelFormat_Mono2p,](#)
[PixelFormat_Mono4p,](#)
[PixelFormat_Mono8s,](#)
[PixelFormat_Mono10,](#)
[PixelFormat_Mono12,](#)
[PixelFormat_Mono14,](#)
[PixelFormat_Mono16s,](#)
[PixelFormat_Mono32f,](#)
[PixelFormat_BayerBG10,](#)
[PixelFormat_BayerBG12,](#)
[PixelFormat_BayerGB10,](#)
[PixelFormat_BayerGB12,](#)
[PixelFormat_BayerGR10,](#)
[PixelFormat_BayerGR12,](#)
[PixelFormat_BayerRG10,](#)
[PixelFormat_BayerRG12,](#)
[PixelFormat_RGBa8,](#)
[PixelFormat_RGBa10,](#)
[PixelFormat_RGBa10p,](#)
[PixelFormat_RGBa12,](#)
[PixelFormat_RGBa12p,](#)
[PixelFormat_RGBa14,](#)
[PixelFormat_RGBa16,](#)
[PixelFormat_RGB8,](#)
[PixelFormat_RGB8_Planar,](#)
[PixelFormat_RGB10,](#)
[PixelFormat_RGB10_Planar,](#)
[PixelFormat_RGB10p,](#)
[PixelFormat_RGB10p32,](#)
[PixelFormat_RGB12,](#)
[PixelFormat_RGB12_Planar,](#)
[PixelFormat_RGB12p,](#)
[PixelFormat_RGB14,](#)
[PixelFormat_RGB16,](#)
[PixelFormat_RGB16s,](#)
[PixelFormat_RGB32f,](#)
[PixelFormat_RGB16_Planar,](#)
[PixelFormat_RGB565p,](#)
[PixelFormat_BGRa10,](#)
[PixelFormat_BGRa10p,](#)
[PixelFormat_BGRa12,](#)
[PixelFormat_BGRa12p,](#)
[PixelFormat_BGRa14,](#)
[PixelFormat_BGRa16,](#)
[PixelFormat_RGBa32f,](#)
[PixelFormat_BGR10,](#)
[PixelFormat_BGR10p,](#)
[PixelFormat_BGR12,](#)
[PixelFormat_BGR12p,](#)
[PixelFormat_BGR14,](#)
[PixelFormat_BGR16,](#)
[PixelFormat_BGR565p,](#)
[PixelFormat_R8,](#)
[PixelFormat_R10,](#)
[PixelFormat_R12,](#)
[PixelFormat_R16,](#)
[PixelFormat_G8,](#)
[PixelFormat_G10,](#)

[PixelFormat_G12,](#)
[PixelFormat_G16,](#)
[PixelFormat_B8,](#)
[PixelFormat_B10,](#)
[PixelFormat_B12,](#)
[PixelFormat_B16,](#)
[PixelFormat_Coord3D_ABC8,](#)
[PixelFormat_Coord3D_ABC8_Planar,](#)
[PixelFormat_Coord3D_ABC10p,](#)
[PixelFormat_Coord3D_ABC10p_Planar,](#)
[PixelFormat_Coord3D_ABC12p,](#)
[PixelFormat_Coord3D_ABC12p_Planar,](#)
[PixelFormat_Coord3D_ABC16,](#)
[PixelFormat_Coord3D_ABC16_Planar,](#)
[PixelFormat_Coord3D_ABC32f,](#)
[PixelFormat_Coord3D_ABC32f_Planar,](#)
[PixelFormat_Coord3D_AC8,](#)
[PixelFormat_Coord3D_AC8_Planar,](#)
[PixelFormat_Coord3D_AC10p,](#)
[PixelFormat_Coord3D_AC10p_Planar,](#)
[PixelFormat_Coord3D_AC12p,](#)
[PixelFormat_Coord3D_AC12p_Planar,](#)
[PixelFormat_Coord3D_AC16,](#)
[PixelFormat_Coord3D_AC16_Planar,](#)
[PixelFormat_Coord3D_AC32f,](#)
[PixelFormat_Coord3D_AC32f_Planar,](#)
[PixelFormat_Coord3D_A8,](#)
[PixelFormat_Coord3D_A10p,](#)
[PixelFormat_Coord3D_A12p,](#)
[PixelFormat_Coord3D_A16,](#)
[PixelFormat_Coord3D_A32f,](#)
[PixelFormat_Coord3D_B8,](#)
[PixelFormat_Coord3D_B10p,](#)
[PixelFormat_Coord3D_B12p,](#)
[PixelFormat_Coord3D_B16,](#)
[PixelFormat_Coord3D_B32f,](#)
[PixelFormat_Coord3D_C8,](#)
[PixelFormat_Coord3D_C10p,](#)
[PixelFormat_Coord3D_C12p,](#)
[PixelFormat_Coord3D_C16,](#)
[PixelFormat_Coord3D_C32f,](#)
[PixelFormat_Confidence1,](#)
[PixelFormat_Confidence1p,](#)
[PixelFormat_Confidence8,](#)
[PixelFormat_Confidence16,](#)
[PixelFormat_Confidence32f,](#)
[PixelFormat_BiColorBGRG8,](#)
[PixelFormat_BiColorBGRG10,](#)
[PixelFormat_BiColorBGRG10p,](#)
[PixelFormat_BiColorBGRG12,](#)
[PixelFormat_BiColorBGRG12p,](#)
[PixelFormat_BiColorRGBG8,](#)
[PixelFormat_BiColorRGBG10,](#)
[PixelFormat_BiColorRGBG10p,](#)
[PixelFormat_BiColorRGBG12,](#)
[PixelFormat_BiColorRGBG12p,](#)
[PixelFormat_SCF1WBWG8,](#)
[PixelFormat_SCF1WBWG10,](#)

[PixelFormat_SCF1WBWG10p,](#)
[PixelFormat_SCF1WBWG12,](#)
[PixelFormat_SCF1WBWG12p,](#)
[PixelFormat_SCF1WBWG14,](#)
[PixelFormat_SCF1WBWG16,](#)
[PixelFormat_SCF1WGWB8,](#)
[PixelFormat_SCF1WGWB10,](#)
[PixelFormat_SCF1WGWB10p,](#)
[PixelFormat_SCF1WGWB12,](#)
[PixelFormat_SCF1WGWB12p,](#)
[PixelFormat_SCF1WGWB14,](#)
[PixelFormat_SCF1WGWB16,](#)
[PixelFormat_SCF1WGWR8,](#)
[PixelFormat_SCF1WGWR10,](#)
[PixelFormat_SCF1WGWR10p,](#)
[PixelFormat_SCF1WGWR12,](#)
[PixelFormat_SCF1WGWR12p,](#)
[PixelFormat_SCF1WGWR14,](#)
[PixelFormat_SCF1WGWR16,](#)
[PixelFormat_SCF1WRWG8,](#)
[PixelFormat_SCF1WRWG10,](#)
[PixelFormat_SCF1WRWG10p,](#)
[PixelFormat_SCF1WRWG12,](#)
[PixelFormat_SCF1WRWG12p,](#)
[PixelFormat_SCF1WRWG14,](#)
[PixelFormat_SCF1WRWG16,](#)
[PixelFormat_YCbCr8_CbYCr,](#)
[PixelFormat_YCbCr10_CbYCr,](#)
[PixelFormat_YCbCr10p_CbYCr,](#)
[PixelFormat_YCbCr12_CbYCr,](#)
[PixelFormat_YCbCr12p_CbYCr,](#)
[PixelFormat_YCbCr411_8_CbYYCrYY,](#)
[PixelFormat_YCbCr422_8_CbYCrY,](#)
[PixelFormat_YCbCr422_10,](#)
[PixelFormat_YCbCr422_10_CbYCrY,](#)
[PixelFormat_YCbCr422_10p,](#)
[PixelFormat_YCbCr422_10p_CbYCrY,](#)
[PixelFormat_YCbCr422_12,](#)
[PixelFormat_YCbCr422_12_CbYCrY,](#)
[PixelFormat_YCbCr422_12p,](#)
[PixelFormat_YCbCr422_12p_CbYCrY,](#)
[PixelFormat_YCbCr601_8_CbYCr,](#)
[PixelFormat_YCbCr601_10_CbYCr,](#)
[PixelFormat_YCbCr601_10p_CbYCr,](#)
[PixelFormat_YCbCr601_12_CbYCr,](#)
[PixelFormat_YCbCr601_12p_CbYCr,](#)
[PixelFormat_YCbCr601_411_8_CbYYCrYY,](#)
[PixelFormat_YCbCr601_422_8,](#)
[PixelFormat_YCbCr601_422_8_CbYCrY,](#)
[PixelFormat_YCbCr601_422_10,](#)
[PixelFormat_YCbCr601_422_10_CbYCrY,](#)
[PixelFormat_YCbCr601_422_10p,](#)
[PixelFormat_YCbCr601_422_10p_CbYCrY,](#)
[PixelFormat_YCbCr601_422_12,](#)
[PixelFormat_YCbCr601_422_12_CbYCrY,](#)
[PixelFormat_YCbCr601_422_12p,](#)
[PixelFormat_YCbCr601_422_12p_CbYCrY,](#)
[PixelFormat_YCbCr709_8_CbYCr,](#)

```

PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

• enum DecimationVerticalModeEnums {
    DecimationVerticalMode_Discard,
    NUM_DECIMATIONVERTICALMODE }

• enum LineModeEnums {
    LineMode_Input,
    LineMode_Output,
    NUM_LINEMODE }

• enum LineSourceEnums {
    LineSource_Off,
    LineSource_Line0,
    LineSource_Line1,
    LineSource_Line2,
    LineSource_Line3,
    LineSource_UserOutput0,
    LineSource_UserOutput1,
    LineSource_UserOutput2,
    LineSource_UserOutput3,
    LineSource_Counter0Active,
    LineSource_Counter1Active,

```

```
LineSource_LogicBlock0,  
LineSource_LogicBlock1,  
LineSource_ExposureActive,  
LineSource_FrameTriggerWait,  
LineSource_SerialPort0,  
LineSource_PPSSignal,  
LineSource_AllPixel,  
LineSource_AnyPixel,  
NUM_LINESOURCE }  
  
• enum LineInputFilterSelectorEnums {  
    LineInputFilterSelector_Deglintch,  
    LineInputFilterSelector_Debounce,  
    NUM_LINEINPUTFILTERSELECTOR }  
  
• enum UserOutputSelectorEnums {  
    UserOutputSelector_UserOutput0,  
    UserOutputSelector_UserOutput1,  
    UserOutputSelector_UserOutput2,  
    UserOutputSelector_UserOutput3,  
    NUM_USEROUTPUTSELECTOR }  
  
• enum LineFormatEnums {  
    LineFormat_NoConnect,  
    LineFormat_TriState,  
    LineFormat_TTL,  
    LineFormat_LVDS,  
    LineFormat_RS422,  
    LineFormat_OptoCoupled,  
    LineFormat_OpenDrain,  
    NUM_LINEFORMAT }  
  
• enum LineSelectorEnums {  
    LineSelector_Line0,  
    LineSelector_Line1,  
    LineSelector_Line2,  
    LineSelector_Line3,  
    NUM_LINESELECTOR }  
  
• enum ExposureActiveModeEnums {  
    ExposureActiveMode_Line1,  
    ExposureActiveMode_AnyPixels,  
    ExposureActiveMode_AllPixels,  
    NUM_EXPOSUREACTIVEMODE }  
  
• enum CounterTriggerActivationEnums {  
    CounterTriggerActivation_LevelLow,  
    CounterTriggerActivation_LevelHigh,  
    CounterTriggerActivation_FallingEdge,  
    CounterTriggerActivation_RisingEdge,  
    CounterTriggerActivation_AnyEdge,  
    NUM_COUNTERTRIGGERACTIVATION }  
  
• enum CounterSelectorEnums {  
    CounterSelector_Counter0,  
    CounterSelector_Counter1,  
    NUM_COUNTERSELECTOR }  
  
• enum CounterStatusEnums {  
    CounterStatus_CounterIdle,  
    CounterStatus_CounterTriggerWait,  
    CounterStatus_CounterActive,  
    CounterStatus_CounterCompleted,  
    CounterStatus_CounterOverflow,  
    NUM_COUNTERSTATUS }
```

- `enum CounterTriggerSourceEnums {`
 `CounterTriggerSource_Off,`
 `CounterTriggerSource_Line0,`
 `CounterTriggerSource_Line1,`
 `CounterTriggerSource_Line2,`
 `CounterTriggerSource_Line3,`
 `CounterTriggerSource_UserOutput0,`
 `CounterTriggerSource_UserOutput1,`
 `CounterTriggerSource_UserOutput2,`
 `CounterTriggerSource_UserOutput3,`
 `CounterTriggerSource_Counter0Start,`
 `CounterTriggerSource_Counter1Start,`
 `CounterTriggerSource_Counter0End,`
 `CounterTriggerSource_Counter1End,`
 `CounterTriggerSource_LogicBlock0,`
 `CounterTriggerSource_LogicBlock1,`
 `CounterTriggerSource_ExposureStart,`
 `CounterTriggerSource_ExposureEnd,`
 `CounterTriggerSource_FrameTriggerWait,`
 `NUM_COUNTERTRIGGERSOURCE }`
- `enum CounterResetSourceEnums {`
 `CounterResetSource_Off,`
 `CounterResetSource_Line0,`
 `CounterResetSource_Line1,`
 `CounterResetSource_Line2,`
 `CounterResetSource_Line3,`
 `CounterResetSource_UserOutput0,`
 `CounterResetSource_UserOutput1,`
 `CounterResetSource_UserOutput2,`
 `CounterResetSource_UserOutput3,`
 `CounterResetSource_Counter0Start,`
 `CounterResetSource_Counter1Start,`
 `CounterResetSource_Counter0End,`
 `CounterResetSource_Counter1End,`
 `CounterResetSource_LogicBlock0,`
 `CounterResetSource_LogicBlock1,`
 `CounterResetSource_ExposureStart,`
 `CounterResetSource_ExposureEnd,`
 `CounterResetSource_FrameTriggerWait,`
 `NUM_COUNTERRESETSOURCE }`
- `enum CounterEventSourceEnums {`
 `CounterEventSource_Off,`
 `CounterEventSource_MHzTick,`
 `CounterEventSource_Line0,`
 `CounterEventSource_Line1,`
 `CounterEventSource_Line2,`
 `CounterEventSource_Line3,`
 `CounterEventSource_UserOutput0,`
 `CounterEventSource_UserOutput1,`
 `CounterEventSource_UserOutput2,`
 `CounterEventSource_UserOutput3,`
 `CounterEventSource_Counter0Start,`
 `CounterEventSource_Counter1Start,`
 `CounterEventSource_Counter0End,`
 `CounterEventSource_Counter1End,`
 `CounterEventSource_LogicBlock0,`
 `CounterEventSource_LogicBlock1,`
 `CounterEventSource_ExposureStart,`

- CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }
- enum CounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }
- enum CounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }
- enum DeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICETYPE }
- enum DeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }
- enum DeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }
- enum DeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }
- enum DeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }
- enum DeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,
DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }
- enum DeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }
- enum DeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICESERIALPORTSELECTOR }
- enum DeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,
DeviceSerialPortBaudRate_Baud19200,
DeviceSerialPortBaudRate_Baud38400,
DeviceSerialPortBaudRate_Baud57600,
DeviceSerialPortBaudRate_Baud115200,
DeviceSerialPortBaudRate_Baud230400,

```

DeviceSerialPortBaudRate_Baud460800,
DeviceSerialPortBaudRate_Baud921600,
NUM_DEVICESERIALPORTBAUDRATE }

• enum SensorTapsEnums {
    SensorTaps_One,
    SensorTaps_Two,
    SensorTaps_Three,
    SensorTaps_Four,
    SensorTaps_Eight,
    SensorTaps_Ten,
    NUM_SENSORTAPS }

• enum SensorDigitizationTapsEnums {
    SensorDigitizationTaps_One,
    SensorDigitizationTaps_Two,
    SensorDigitizationTaps_Three,
    SensorDigitizationTaps_Four,
    SensorDigitizationTaps_Eight,
    SensorDigitizationTaps_Ten,
    NUM_SENSORDIGITIZATIONTAPS }

• enum RegionSelectorEnums {
    RegionSelector_Region0,
    RegionSelector_Region1,
    RegionSelector_Region2,
    RegionSelector_All,
    NUM_REGIONSELECTOR }

• enum RegionModeEnums {
    RegionMode_Off,
    RegionMode_On,
    NUM_REGIONMODE }

• enum RegionDestinationEnums {
    RegionDestination_Stream0,
    RegionDestination_Stream1,
    RegionDestination_Stream2,
    NUM_REGIONDESTINATION }

• enum ImageComponentSelectorEnums {
    ImageComponentSelector_Intensity,
    ImageComponentSelector_Color,
    ImageComponentSelector_Infrared,
    ImageComponentSelector_Ultraviolet,
    ImageComponentSelector_Range,
    ImageComponentSelector_Disparity,
    ImageComponentSelector_Confidence,
    ImageComponentSelector_Scatter,
    NUM_IMAGECOMPONENTSELECTOR }

• enum PixelFormatInfoSelectorEnums {
    PixelFormatInfoSelector_Mono1p,
    PixelFormatInfoSelector_Mono2p,
    PixelFormatInfoSelector_Mono4p,
    PixelFormatInfoSelector_Mono8,
    PixelFormatInfoSelector_Mono8s,
    PixelFormatInfoSelector_Mono10,
    PixelFormatInfoSelector_Mono10p,
    PixelFormatInfoSelector_Mono12,
    PixelFormatInfoSelector_Mono12p,
    PixelFormatInfoSelector_Mono14,
    PixelFormatInfoSelector_Mono16,
    PixelFormatInfoSelector_Mono16s,
    PixelFormatInfoSelector_Mono32f,

```


[PixelFormatInfoSelector_BayerBG8,](#)
[PixelFormatInfoSelector_BayerBG10,](#)
[PixelFormatInfoSelector_BayerBG10p,](#)
[PixelFormatInfoSelector_BayerBG12,](#)
[PixelFormatInfoSelector_BayerBG12p,](#)
[PixelFormatInfoSelector_BayerBG16,](#)
[PixelFormatInfoSelector_BayerGB8,](#)
[PixelFormatInfoSelector_BayerGB10,](#)
[PixelFormatInfoSelector_BayerGB10p,](#)
[PixelFormatInfoSelector_BayerGB12,](#)
[PixelFormatInfoSelector_BayerGB12p,](#)
[PixelFormatInfoSelector_BayerGB16,](#)
[PixelFormatInfoSelector_BayerGR8,](#)
[PixelFormatInfoSelector_BayerGR10,](#)
[PixelFormatInfoSelector_BayerGR10p,](#)
[PixelFormatInfoSelector_BayerGR12,](#)
[PixelFormatInfoSelector_BayerGR12p,](#)
[PixelFormatInfoSelector_BayerGR16,](#)
[PixelFormatInfoSelector_BayerRG8,](#)
[PixelFormatInfoSelector_BayerRG10,](#)
[PixelFormatInfoSelector_BayerRG10p,](#)
[PixelFormatInfoSelector_BayerRG12,](#)
[PixelFormatInfoSelector_BayerRG12p,](#)
[PixelFormatInfoSelector_BayerRG16,](#)
[PixelFormatInfoSelector_RGBa8,](#)
[PixelFormatInfoSelector_RGBa10,](#)
[PixelFormatInfoSelector_RGBa10p,](#)
[PixelFormatInfoSelector_RGBa12,](#)
[PixelFormatInfoSelector_RGBa12p,](#)
[PixelFormatInfoSelector_RGBa14,](#)
[PixelFormatInfoSelector_RGBa16,](#)
[PixelFormatInfoSelector_RGB8,](#)
[PixelFormatInfoSelector_RGB8_Planar,](#)
[PixelFormatInfoSelector_RGB10,](#)
[PixelFormatInfoSelector_RGB10_Planar,](#)
[PixelFormatInfoSelector_RGB10p,](#)
[PixelFormatInfoSelector_RGB10p32,](#)
[PixelFormatInfoSelector_RGB12,](#)
[PixelFormatInfoSelector_RGB12_Planar,](#)
[PixelFormatInfoSelector_RGB12p,](#)
[PixelFormatInfoSelector_RGB14,](#)
[PixelFormatInfoSelector_RGB16,](#)
[PixelFormatInfoSelector_RGB16s,](#)
[PixelFormatInfoSelector_RGB32f,](#)
[PixelFormatInfoSelector_RGB16_Planar,](#)
[PixelFormatInfoSelector_RGB565p,](#)
[PixelFormatInfoSelector_BGRa8,](#)
[PixelFormatInfoSelector_BGRa10,](#)
[PixelFormatInfoSelector_BGRa10p,](#)
[PixelFormatInfoSelector_BGRa12,](#)
[PixelFormatInfoSelector_BGRa12p,](#)
[PixelFormatInfoSelector_BGRa14,](#)
[PixelFormatInfoSelector_BGRa16,](#)
[PixelFormatInfoSelector_RGBa32f,](#)
[PixelFormatInfoSelector_BGR8,](#)
[PixelFormatInfoSelector_BGR10,](#)
[PixelFormatInfoSelector_BGR10p,](#)
[PixelFormatInfoSelector_BGR12,](#)

[PixelFormatInfoSelector_BGR12p,](#)
[PixelFormatInfoSelector_BGR14,](#)
[PixelFormatInfoSelector_BGR16,](#)
[PixelFormatInfoSelector_BGR565p,](#)
[PixelFormatInfoSelector_R8,](#)
[PixelFormatInfoSelector_R10,](#)
[PixelFormatInfoSelector_R12,](#)
[PixelFormatInfoSelector_R16,](#)
[PixelFormatInfoSelector_G8,](#)
[PixelFormatInfoSelector_G10,](#)
[PixelFormatInfoSelector_G12,](#)
[PixelFormatInfoSelector_G16,](#)
[PixelFormatInfoSelector_B8,](#)
[PixelFormatInfoSelector_B10,](#)
[PixelFormatInfoSelector_B12,](#)
[PixelFormatInfoSelector_B16,](#)
[PixelFormatInfoSelector_Coord3D_ABC8,](#)
[PixelFormatInfoSelector_Coord3D_ABC8_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC10p,](#)
[PixelFormatInfoSelector_Coord3D_ABC10p_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC12p,](#)
[PixelFormatInfoSelector_Coord3D_ABC12p_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC16,](#)
[PixelFormatInfoSelector_Coord3D_ABC16_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC32f,](#)
[PixelFormatInfoSelector_Coord3D_ABC32f_Planar,](#)
[PixelFormatInfoSelector_Coord3D_AC8,](#)
[PixelFormatInfoSelector_Coord3D_AC8_Planar,](#)
[PixelFormatInfoSelector_Coord3D_AC10p,](#)
[PixelFormatInfoSelector_Coord3D_AC10p_Planar,](#)
[PixelFormatInfoSelector_Coord3D_AC12p,](#)
[PixelFormatInfoSelector_Coord3D_AC12p_Planar,](#)
[PixelFormatInfoSelector_Coord3D_AC16,](#)
[PixelFormatInfoSelector_Coord3D_AC16_Planar,](#)
[PixelFormatInfoSelector_Coord3D_AC32f,](#)
[PixelFormatInfoSelector_Coord3D_AC32f_Planar,](#)
[PixelFormatInfoSelector_Coord3D_A8,](#)
[PixelFormatInfoSelector_Coord3D_A10p,](#)
[PixelFormatInfoSelector_Coord3D_A12p,](#)
[PixelFormatInfoSelector_Coord3D_A16,](#)
[PixelFormatInfoSelector_Coord3D_A32f,](#)
[PixelFormatInfoSelector_Coord3D_B8,](#)
[PixelFormatInfoSelector_Coord3D_B10p,](#)
[PixelFormatInfoSelector_Coord3D_B12p,](#)
[PixelFormatInfoSelector_Coord3D_B16,](#)
[PixelFormatInfoSelector_Coord3D_B32f,](#)
[PixelFormatInfoSelector_Coord3D_C8,](#)
[PixelFormatInfoSelector_Coord3D_C10p,](#)
[PixelFormatInfoSelector_Coord3D_C12p,](#)
[PixelFormatInfoSelector_Coord3D_C16,](#)
[PixelFormatInfoSelector_Coord3D_C32f,](#)
[PixelFormatInfoSelector_Confidence1,](#)
[PixelFormatInfoSelector_Confidence1p,](#)
[PixelFormatInfoSelector_Confidence8,](#)
[PixelFormatInfoSelector_Confidence16,](#)
[PixelFormatInfoSelector_Confidence32f,](#)
[PixelFormatInfoSelector_BiColorBGRG8,](#)
[PixelFormatInfoSelector_BiColorBGRG10,](#)

[PixelFormatInfoSelector_BiColorBGRG10p,](#)
[PixelFormatInfoSelector_BiColorBGRG12,](#)
[PixelFormatInfoSelector_BiColorBGRG12p,](#)
[PixelFormatInfoSelector_BiColorRGBG8,](#)
[PixelFormatInfoSelector_BiColorRGBG10,](#)
[PixelFormatInfoSelector_BiColorRGBG10p,](#)
[PixelFormatInfoSelector_BiColorRGBG12,](#)
[PixelFormatInfoSelector_BiColorRGBG12p,](#)
[PixelFormatInfoSelector_SCF1WBWG8,](#)
[PixelFormatInfoSelector_SCF1WBWG10,](#)
[PixelFormatInfoSelector_SCF1WBWG10p,](#)
[PixelFormatInfoSelector_SCF1WBWG12,](#)
[PixelFormatInfoSelector_SCF1WBWG12p,](#)
[PixelFormatInfoSelector_SCF1WBWG14,](#)
[PixelFormatInfoSelector_SCF1WBWG16,](#)
[PixelFormatInfoSelector_SCF1WGWB8,](#)
[PixelFormatInfoSelector_SCF1WGWB10,](#)
[PixelFormatInfoSelector_SCF1WGWB10p,](#)
[PixelFormatInfoSelector_SCF1WGWB12,](#)
[PixelFormatInfoSelector_SCF1WGWB12p,](#)
[PixelFormatInfoSelector_SCF1WGWB14,](#)
[PixelFormatInfoSelector_SCF1WGWB16,](#)
[PixelFormatInfoSelector_SCF1WGWR8,](#)
[PixelFormatInfoSelector_SCF1WGWR10,](#)
[PixelFormatInfoSelector_SCF1WGWR10p,](#)
[PixelFormatInfoSelector_SCF1WGWR12,](#)
[PixelFormatInfoSelector_SCF1WGWR12p,](#)
[PixelFormatInfoSelector_SCF1WGWR14,](#)
[PixelFormatInfoSelector_SCF1WGWR16,](#)
[PixelFormatInfoSelector_SCF1WRWG8,](#)
[PixelFormatInfoSelector_SCF1WRWG10,](#)
[PixelFormatInfoSelector_SCF1WRWG10p,](#)
[PixelFormatInfoSelector_SCF1WRWG12,](#)
[PixelFormatInfoSelector_SCF1WRWG12p,](#)
[PixelFormatInfoSelector_SCF1WRWG14,](#)
[PixelFormatInfoSelector_SCF1WRWG16,](#)
[PixelFormatInfoSelector_YCbCr8,](#)
[PixelFormatInfoSelector_YCbCr8_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr10_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr10p_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr12_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr12p_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr411_8,](#)
[PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,](#)
[PixelFormatInfoSelector_YCbCr422_8,](#)
[PixelFormatInfoSelector_YCbCr422_8_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_10,](#)
[PixelFormatInfoSelector_YCbCr422_10_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_10p,](#)
[PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_12,](#)
[PixelFormatInfoSelector_YCbCr422_12_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_12p,](#)
[PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr601_8_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr601_10_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr601_10p_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr601_12_CbYCr,](#)

```

PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

• enum DeinterlacingEnums {
    Deinterlacing_Off,
    Deinterlacing_LineDuplication,
    Deinterlacing_Weave,
    NUM_DEINTERLACING }

• enum ImageCompressionRateOptionEnums {
    ImageCompressionRateOption_FixBitrate,
    ImageCompressionRateOption_FixQuality,
    NUM_IMAGECOMPRESSIONRATEOPTION }

• enum ImageCompressionJPEGFormatOptionEnums {
    ImageCompressionJPEGFormatOption_Lossless,
    ImageCompressionJPEGFormatOption_BaselineStandard,
    ImageCompressionJPEGFormatOption_BaselineOptimized,

```

```
ImageCompressionJPEGFormatOption_Progressive,  
NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }  
• enum AcquisitionStatusSelectorEnums {  
    AcquisitionStatusSelector_AcquisitionTriggerWait,  
    AcquisitionStatusSelector_AcquisitionActive,  
    AcquisitionStatusSelector_AcquisitionTransfer,  
    AcquisitionStatusSelector_FrameTriggerWait,  
    AcquisitionStatusSelector_FrameActive,  
    AcquisitionStatusSelector_ExposureActive,  
    NUM_ACQUISITIONSTATUSSELECTOR }  
• enum ExposureTimeModeEnums {  
    ExposureTimeMode_Common,  
    ExposureTimeMode_Individual,  
    NUM_EXPOSURETIMEMODE }  
• enum ExposureTimeSelectorEnums {  
    ExposureTimeSelector_Common,  
    ExposureTimeSelector_Red,  
    ExposureTimeSelector_Green,  
    ExposureTimeSelector_Blue,  
    ExposureTimeSelector_Cyan,  
    ExposureTimeSelector_Magenta,  
    ExposureTimeSelector_Yellow,  
    ExposureTimeSelector_Infrared,  
    ExposureTimeSelector_Ultraviolet,  
    ExposureTimeSelector_Stage1,  
    ExposureTimeSelector_Stage2,  
    NUM_EXPOSURETIMESELECTOR }  
• enum GainAutoBalanceEnums {  
    GainAutoBalance_Off,  
    GainAutoBalance_Once,  
    GainAutoBalance_Continuous,  
    NUM_GAINAUTOBALANCE }  
• enum BlackLevelAutoEnums {  
    BlackLevelAuto_Off,  
    BlackLevelAuto_Once,  
    BlackLevelAuto_Continuous,  
    NUM_BLACKLEVELAUTO }  
• enum BlackLevelAutoBalanceEnums {  
    BlackLevelAutoBalance_Off,  
    BlackLevelAutoBalance_Once,  
    BlackLevelAutoBalance_Continuous,  
    NUM_BLACKLEVELAUTOBALANCE }  
• enum WhiteClipSelectorEnums {  
    WhiteClipSelector_All,  
    WhiteClipSelector_Red,  
    WhiteClipSelector_Green,  
    WhiteClipSelector_Blue,  
    WhiteClipSelector_Y,  
    WhiteClipSelector_U,  
    WhiteClipSelector_V,  
    WhiteClipSelector_Tap1,  
    WhiteClipSelector_Tap2,  
    NUM_WHITECLIPSELECTOR }  
• enum TimerSelectorEnums {  
    TimerSelector_Timer0,  
    TimerSelector_Timer1,  
    TimerSelector_Timer2,  
    NUM_TIMERSELECTOR }
```

- enum TimerStatusEnums {
TimerStatus_TimerIdle,
TimerStatus_TimerTriggerWait,
TimerStatus_TimerActive,
TimerStatus_TimerCompleted,
NUM_TIMERSTATUS }
- enum TimerTriggerSourceEnums {
TimerTriggerSource_Off,
TimerTriggerSource_AcquisitionTrigger,
TimerTriggerSource_AcquisitionStart,
TimerTriggerSource_AcquisitionEnd,
TimerTriggerSource_FrameTrigger,
TimerTriggerSource_FrameStart,
TimerTriggerSource_FrameEnd,
TimerTriggerSource_FrameBurstStart,
TimerTriggerSource_FrameBurstEnd,
TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }
- enum TimerTriggerActivationEnums {
TimerTriggerActivation_RisingEdge,
TimerTriggerActivation_FallingEdge,
TimerTriggerActivation_AnyEdge,
TimerTriggerActivation_LevelHigh,
TimerTriggerActivation_LevelLow,

```
NUM_TIMERTRIGGERACTIVATION }  
  
• enum EncoderSelectorEnums {  
    EncoderSelector_Encoder0,  
    EncoderSelector_Encoder1,  
    EncoderSelector_Encoder2,  
    NUM_ENCODERSELECTOR }  
  
• enum EncoderSourceAEnums {  
    EncoderSourceA_Off,  
    EncoderSourceA_Line0,  
    EncoderSourceA_Line1,  
    EncoderSourceA_Line2,  
    NUM_ENCODERSOURCEA }  
  
• enum EncoderSourceBEnums {  
    EncoderSourceB_Off,  
    EncoderSourceB_Line0,  
    EncoderSourceB_Line1,  
    EncoderSourceB_Line2,  
    NUM_ENCODERSOURCEB }  
  
• enum EncoderModeEnums {  
    EncoderMode_FourPhase,  
    EncoderMode_HighResolution,  
    NUM_ENCODERMODE }  
  
• enum EncoderOutputModeEnums {  
    EncoderOutputMode_Off,  
    EncoderOutputMode_PositionUp,  
    EncoderOutputMode_PositionDown,  
    EncoderOutputMode_DirectionUp,  
    EncoderOutputMode_DirectionDown,  
    EncoderOutputMode_Motion,  
    NUM_ENCODEROUTPUTMODE }  
  
• enum EncoderStatusEnums {  
    EncoderStatus_EncoderUp,  
    EncoderStatus_EncoderDown,  
    EncoderStatus_EncoderIdle,  
    EncoderStatus_EncoderStatic,  
    NUM_ENCODERSTATUS }  
  
• enum EncoderResetSourceEnums {  
    EncoderResetSource_Off,  
    EncoderResetSource_AcquisitionTrigger,  
    EncoderResetSource_AcquisitionStart,  
    EncoderResetSource_AcquisitionEnd,  
    EncoderResetSource_FrameTrigger,  
    EncoderResetSource_FrameStart,  
    EncoderResetSource_FrameEnd,  
    EncoderResetSource_ExposureStart,  
    EncoderResetSource_ExposureEnd,  
    EncoderResetSource_Line0,  
    EncoderResetSource_Line1,  
    EncoderResetSource_Line2,  
    EncoderResetSource_Counter0Start,  
    EncoderResetSource_Counter1Start,  
    EncoderResetSource_Counter2Start,  
    EncoderResetSource_Counter0End,  
    EncoderResetSource_Counter1End,  
    EncoderResetSource_Counter2End,  
    EncoderResetSource_Timer0Start,  
    EncoderResetSource_Timer1Start,  
    EncoderResetSource_Timer2Start,
```

```

EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }
• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }
• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }
• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }
• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,
NUM_SOURCESELECTOR }
• enum TransferSelectorEnums {
TransferSelector_Stream0,
TransferSelector_Stream1,
TransferSelector_Stream2,
TransferSelector_All,
NUM_TRANSFERSELECTOR }
• enum TransferTriggerSelectorEnums {
TransferTriggerSelector_TransferStart,
TransferTriggerSelector_TransferStop,
TransferTriggerSelector_TransferAbort,
TransferTriggerSelector_TransferPause,
TransferTriggerSelector_TransferResume,
TransferTriggerSelector_TransferActive,
TransferTriggerSelector_TransferBurstStart,
TransferTriggerSelector_TransferBurstStop,
NUM_TRANSFERTRIGGERSELECTOR }
• enum TransferTriggerModeEnums {
TransferTriggerMode_Off,
TransferTriggerMode_On,
NUM_TRANSFERTRIGGERMODE }

```


- enum [TransferTriggerSourceEnums](#) {
 [TransferTriggerSource_Line0](#),
 [TransferTriggerSource_Line1](#),
 [TransferTriggerSource_Line2](#),
 [TransferTriggerSource_Counter0Start](#),
 [TransferTriggerSource_Counter1Start](#),
 [TransferTriggerSource_Counter2Start](#),
 [TransferTriggerSource_Counter0End](#),
 [TransferTriggerSource_Counter1End](#),
 [TransferTriggerSource_Counter2End](#),
 [TransferTriggerSource_Timer0Start](#),
 [TransferTriggerSource_Timer1Start](#),
 [TransferTriggerSource_Timer2Start](#),
 [TransferTriggerSource_Timer0End](#),
 [TransferTriggerSource_Timer1End](#),
 [TransferTriggerSource_Timer2End](#),
 [TransferTriggerSource_SoftwareSignal0](#),
 [TransferTriggerSource_SoftwareSignal1](#),
 [TransferTriggerSource_SoftwareSignal2](#),
 [TransferTriggerSource_Action0](#),
 [TransferTriggerSource_Action1](#),
 [TransferTriggerSource_Action2](#),
 [NUM_TRANSFERTRIGGERSOURCE](#) }
- enum [TransferTriggerActivationEnums](#) {
 [TransferTriggerActivation_RisingEdge](#),
 [TransferTriggerActivation_FallingEdge](#),
 [TransferTriggerActivation_AnyEdge](#),
 [TransferTriggerActivation_LevelHigh](#),
 [TransferTriggerActivation_LevelLow](#),
 [NUM_TRANSFERTRIGGERACTIVATION](#) }
- enum [TransferStatusSelectorEnums](#) {
 [TransferStatusSelector_Streaming](#),
 [TransferStatusSelector_Paused](#),
 [TransferStatusSelector_Stopping](#),
 [TransferStatusSelector_Stopped](#),
 [TransferStatusSelector_QueueOverflow](#),
 [NUM_TRANSFERSTATUSSELECTOR](#) }
- enum [TransferComponentSelectorEnums](#) {
 [TransferComponentSelector_Red](#),
 [TransferComponentSelector_Green](#),
 [TransferComponentSelector_Blue](#),
 [TransferComponentSelector_All](#),
 [NUM_TRANSFERCOMPONENTSELECTOR](#) }
- enum [Scan3dDistanceUnitEnums](#) {
 [Scan3dDistanceUnit_Millimeter](#),
 [Scan3dDistanceUnit_Inch](#),
 [NUM_SCAN3DDISTANCEUNIT](#) }
- enum [Scan3dCoordinateSystemEnums](#) {
 [Scan3dCoordinateSystem_Cartesian](#),
 [Scan3dCoordinateSystem_Spherical](#),
 [Scan3dCoordinateSystem_Cylindrical](#),
 [NUM_SCAN3DCOORDINATESYSTEM](#) }
- enum [Scan3dOutputModeEnums](#) {
 [Scan3dOutputMode_UncalibratedC](#),
 [Scan3dOutputMode_CalibratedABC_Grid](#),
 [Scan3dOutputMode_CalibratedABC_PointCloud](#),
 [Scan3dOutputMode_CalibratedAC](#),
 [Scan3dOutputMode_CalibratedAC_Linescan](#),

```

Scan3dOutputMode_CalibratedC,
Scan3dOutputMode_CalibratedC_Linescan,
Scan3dOutputMode_RectifiedC,
Scan3dOutputMode_RectifiedC_Linescan,
Scan3dOutputMode_DisparityC,
Scan3dOutputMode_DisparityC_Linescan,
NUM_SCAN3DOUTPUTMODE }

• enum Scan3dCoordinateSystemReferenceEnums {
    Scan3dCoordinateSystemReference_Anchor,
    Scan3dCoordinateSystemReference_Transformed,
    NUM_SCAN3DCOORDINATESYSTEMREFERENCE }

• enum Scan3dCoordinateSelectorEnums {
    Scan3dCoordinateSelector_CoordinateA,
    Scan3dCoordinateSelector_CoordinateB,
    Scan3dCoordinateSelector_CoordinateC,
    NUM_SCAN3DCOORDINATESELECTOR }

• enum Scan3dCoordinateTransformSelectorEnums {
    Scan3dCoordinateTransformSelector_RotationX,
    Scan3dCoordinateTransformSelector_RotationY,
    Scan3dCoordinateTransformSelector_RotationZ,
    Scan3dCoordinateTransformSelector_TranslationX,
    Scan3dCoordinateTransformSelector_TranslationY,
    Scan3dCoordinateTransformSelector_TranslationZ,
    NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }

• enum Scan3dCoordinateReferenceSelectorEnums {
    Scan3dCoordinateReferenceSelector_RotationX,
    Scan3dCoordinateReferenceSelector_RotationY,
    Scan3dCoordinateReferenceSelector_RotationZ,
    Scan3dCoordinateReferenceSelector_TranslationX,
    Scan3dCoordinateReferenceSelector_TranslationY,
    Scan3dCoordinateReferenceSelector_TranslationZ,
    NUM_SCAN3DCOORDINATEREFERENCESELECTOR }

• enum ChunkImageComponentEnums {
    ChunkImageComponent_Intensity,
    ChunkImageComponent_Color,
    ChunkImageComponent_Infrared,
    ChunkImageComponent_Ultraviolet,
    ChunkImageComponent_Range,
    ChunkImageComponent_Disparity,
    ChunkImageComponent_Confidence,
    ChunkImageComponent_Scatter,
    NUM_CHUNKIMAGECOMPONENT }

• enum ChunkCounterSelectorEnums {
    ChunkCounterSelector_Counter0,
    ChunkCounterSelector_Counter1,
    ChunkCounterSelector_Counter2,
    NUM_CHUNKCOUNTERSELECTOR }

• enum ChunkTimerSelectorEnums {
    ChunkTimerSelector_Timer0,
    ChunkTimerSelector_Timer1,
    ChunkTimerSelector_Timer2,
    NUM_CHUNKTIMERSELECTOR }

• enum ChunkEncoderSelectorEnums {
    ChunkEncoderSelector_Encoder0,
    ChunkEncoderSelector_Encoder1,
    ChunkEncoderSelector_Encoder2,
    NUM_CHUNKENCODERSELECTOR }

```

- enum [ChunkEncoderStatusEnums](#) {
[ChunkEncoderStatus_EncoderUp](#),
[ChunkEncoderStatus_EncoderDown](#),
[ChunkEncoderStatus_EncoderIdle](#),
[ChunkEncoderStatus_EncoderStatic](#),
[NUM_CHUNKENCODERSTATUS](#) }
- enum [ChunkExposureTimeSelectorEnums](#) {
[ChunkExposureTimeSelector_Common](#),
[ChunkExposureTimeSelector_Red](#),
[ChunkExposureTimeSelector_Green](#),
[ChunkExposureTimeSelector_Blue](#),
[ChunkExposureTimeSelector_Cyan](#),
[ChunkExposureTimeSelector_Magenta](#),
[ChunkExposureTimeSelector_Yellow](#),
[ChunkExposureTimeSelector_Infrared](#),
[ChunkExposureTimeSelector_Ultraviolet](#),
[ChunkExposureTimeSelector_Stage1](#),
[ChunkExposureTimeSelector_Stage2](#),
[NUM_CHUNKEXPOSURETIMESELECTOR](#) }
- enum [ChunkSourceIDEnums](#) {
[ChunkSourceID_Source0](#),
[ChunkSourceID_Source1](#),
[ChunkSourceID_Source2](#),
[NUM_CHUNKSOURCEID](#) }
- enum [ChunkRegionIDEnums](#) {
[ChunkRegionID_Region0](#),
[ChunkRegionID_Region1](#),
[ChunkRegionID_Region2](#),
[NUM_CHUNKREGIONID](#) }
- enum [ChunkTransferStreamIDEnums](#) {
[ChunkTransferStreamID_Stream0](#),
[ChunkTransferStreamID_Stream1](#),
[ChunkTransferStreamID_Stream2](#),
[ChunkTransferStreamID_Stream3](#),
[NUM_CHUNKTRANSFERSTREAMID](#) }
- enum [ChunkScan3dDistanceUnitEnums](#) {
[ChunkScan3dDistanceUnit_Millimeter](#),
[ChunkScan3dDistanceUnit_Inch](#),
[NUM_CHUNKSCAN3DDISTANCEUNIT](#) }
- enum [ChunkScan3dOutputModeEnums](#) {
[ChunkScan3dOutputMode_UncalibratedC](#),
[ChunkScan3dOutputMode_CalibratedABC_Grid](#),
[ChunkScan3dOutputMode_CalibratedABC_PointCloud](#),
[ChunkScan3dOutputMode_CalibratedAC](#),
[ChunkScan3dOutputMode_CalibratedAC_Linescan](#),
[ChunkScan3dOutputMode_CalibratedC](#),
[ChunkScan3dOutputMode_CalibratedC_Linescan](#),
[ChunkScan3dOutputMode_RectifiedC](#),
[ChunkScan3dOutputMode_RectifiedC_Linescan](#),
[ChunkScan3dOutputMode_DisparityC](#),
[ChunkScan3dOutputMode_DisparityC_Linescan](#),
[NUM_CHUNKSCAN3DOUTPUTMODE](#) }
- enum [ChunkScan3dCoordinateSystemEnums](#) {
[ChunkScan3dCoordinateSystem_Cartesian](#),
[ChunkScan3dCoordinateSystem_Spherical](#),
[ChunkScan3dCoordinateSystem_Cylindrical](#),
[NUM_CHUNKSCAN3DCOORDINATESYSTEM](#) }

- `enum ChunkScan3dCoordinateSystemReferenceEnums {
 ChunkScan3dCoordinateSystemReference_Anchor,
 ChunkScan3dCoordinateSystemReference_Transformed,
 NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }`
- `enum ChunkScan3dCoordinateSelectorEnums {
 ChunkScan3dCoordinateSelector_CoordinateA,
 ChunkScan3dCoordinateSelector_CoordinateB,
 ChunkScan3dCoordinateSelector_CoordinateC,
 NUM_CHUNKSCAN3DCOORDINATESELECTOR }`
- `enum ChunkScan3dCoordinateTransformSelectorEnums {
 ChunkScan3dCoordinateTransformSelector_RotationX,
 ChunkScan3dCoordinateTransformSelector_RotationY,
 ChunkScan3dCoordinateTransformSelector_RotationZ,
 ChunkScan3dCoordinateTransformSelector_TranslationX,
 ChunkScan3dCoordinateTransformSelector_TranslationY,
 ChunkScan3dCoordinateTransformSelector_TranslationZ,
 NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }`
- `enum ChunkScan3dCoordinateReferenceSelectorEnums {
 ChunkScan3dCoordinateReferenceSelector_RotationX,
 ChunkScan3dCoordinateReferenceSelector_RotationY,
 ChunkScan3dCoordinateReferenceSelector_RotationZ,
 ChunkScan3dCoordinateReferenceSelector_TranslationX,
 ChunkScan3dCoordinateReferenceSelector_TranslationY,
 ChunkScan3dCoordinateReferenceSelector_TranslationZ,
 NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }`
- `enum DeviceTapGeometryEnums {
 DeviceTapGeometry_Geometry_1X_1Y,
 DeviceTapGeometry_Geometry_1X2_1Y,
 DeviceTapGeometry_Geometry_1X2_1Y2,
 DeviceTapGeometry_Geometry_2X_1Y,
 DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
 DeviceTapGeometry_Geometry_2XE_1Y2,
 DeviceTapGeometry_Geometry_2XM_1Y,
 DeviceTapGeometry_Geometry_2XM_1Y2,
 DeviceTapGeometry_Geometry_1X_1Y2,
 DeviceTapGeometry_Geometry_1X_2YE,
 DeviceTapGeometry_Geometry_1X3_1Y,
 DeviceTapGeometry_Geometry_3X_1Y,
 DeviceTapGeometry_Geometry_1X,
 DeviceTapGeometry_Geometry_1X2,
 DeviceTapGeometry_Geometry_2X,
 DeviceTapGeometry_Geometry_2XE,
 DeviceTapGeometry_Geometry_2XM,
 DeviceTapGeometry_Geometry_1X3,
 DeviceTapGeometry_Geometry_3X,
 DeviceTapGeometry_Geometry_1X4_1Y,
 DeviceTapGeometry_Geometry_4X_1Y,
 DeviceTapGeometry_Geometry_2X2_1Y,
 DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
 DeviceTapGeometry_Geometry_1X2_2YE,
 DeviceTapGeometry_Geometry_2X_2YE,
 DeviceTapGeometry_Geometry_2XE_2YE,
 DeviceTapGeometry_Geometry_2XM_2YE,
 DeviceTapGeometry_Geometry_1X4,
 DeviceTapGeometry_Geometry_4X,
 DeviceTapGeometry_Geometry_2X2,
 DeviceTapGeometry_Geometry_2X2E,
 DeviceTapGeometry_Geometry_2X2M,`

```

DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

• enum GevPhysicalLinkConfigurationEnums {
    GevPhysicalLinkConfiguration_SingleLink,
    GevPhysicalLinkConfiguration_MultiLink,
    GevPhysicalLinkConfiguration_StaticLAG,
    GevPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVPHYSICALLINKCONFIGURATION }

• enum GevCurrentPhysicalLinkConfigurationEnums {
    GevCurrentPhysicalLinkConfiguration_SingleLink,
    GevCurrentPhysicalLinkConfiguration_MultiLink,
    GevCurrentPhysicalLinkConfiguration_StaticLAG,
    GevCurrentPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

• enum GevIPConfigurationStatusEnums {
    GevIPConfigurationStatus_None,
    GevIPConfigurationStatus_PersistentIP,
    GevIPConfigurationStatus_DHCP,
    GevIPConfigurationStatus_LLA,
    GevIPConfigurationStatus_ForceIP,
    NUM_GEVIPCONFIGURATIONSTATUS }

• enum GevGVCPEExtendedStatusCodesSelectorEnums {
    GevGVCPEExtendedStatusCodesSelector_Version1_1,
    GevGVCPEExtendedStatusCodesSelector_Version2_0,
    NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }

• enum GevGVSPExtendedIDModeEnums {
    GevGVSPExtendedIDMode_Off,
    GevGVSPExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDIDMODE }

• enum ClConfigurationEnums {
    ClConfiguration_Base,
    ClConfiguration_Medium,
    ClConfiguration_Full,
    ClConfiguration_DualBase,
    ClConfiguration_EightyBit,
    NUM_CLCONFIGURATION }

• enum ClTimeSlotsCountEnums {
    ClTimeSlotsCount_One,
    ClTimeSlotsCount_Two,
    ClTimeSlotsCount_Three,
    NUM_CLTIMESLOTSCOUNT }

• enum CxpLinkConfigurationStatusEnums {
    CxpLinkConfigurationStatus_None,
    CxpLinkConfigurationStatus_Pending,
    CxpLinkConfigurationStatus_CXP1_X1,
    CxpLinkConfigurationStatus_CXP2_X1,

```

```

CxpLinkConfigurationStatus_CXP3_X1,
CxpLinkConfigurationStatus_CXP5_X1,
CxpLinkConfigurationStatus_CXP6_X1,
CxpLinkConfigurationStatus_CXP1_X2,
CxpLinkConfigurationStatus_CXP2_X2,
CxpLinkConfigurationStatus_CXP3_X2,
CxpLinkConfigurationStatus_CXP5_X2,
CxpLinkConfigurationStatus_CXP6_X2,
CxpLinkConfigurationStatus_CXP1_X3,
CxpLinkConfigurationStatus_CXP2_X3,
CxpLinkConfigurationStatus_CXP3_X3,
CxpLinkConfigurationStatus_CXP5_X3,
CxpLinkConfigurationStatus_CXP6_X3,
CxpLinkConfigurationStatus_CXP1_X4,
CxpLinkConfigurationStatus_CXP2_X4,
CxpLinkConfigurationStatus_CXP3_X4,
CxpLinkConfigurationStatus_CXP5_X4,
CxpLinkConfigurationStatus_CXP6_X4,
CxpLinkConfigurationStatus_CXP1_X5,
CxpLinkConfigurationStatus_CXP2_X5,
CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,
CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

```

- **enum** CxpLinkConfigurationPreferredEnums {


```

CxpLinkConfigurationPreferred_CXP1_X1,
CxpLinkConfigurationPreferred_CXP2_X1,
CxpLinkConfigurationPreferred_CXP3_X1,
CxpLinkConfigurationPreferred_CXP5_X1,
CxpLinkConfigurationPreferred_CXP6_X1,
CxpLinkConfigurationPreferred_CXP1_X2,
CxpLinkConfigurationPreferred_CXP2_X2,
CxpLinkConfigurationPreferred_CXP3_X2,
CxpLinkConfigurationPreferred_CXP5_X2,
CxpLinkConfigurationPreferred_CXP6_X2,
CxpLinkConfigurationPreferred_CXP1_X3,
CxpLinkConfigurationPreferred_CXP2_X3,
CxpLinkConfigurationPreferred_CXP3_X3,
CxpLinkConfigurationPreferred_CXP5_X3,
CxpLinkConfigurationPreferred_CXP6_X3,
CxpLinkConfigurationPreferred_CXP1_X4,
CxpLinkConfigurationPreferred_CXP2_X4,
CxpLinkConfigurationPreferred_CXP3_X4,
CxpLinkConfigurationPreferred_CXP5_X4,
CxpLinkConfigurationPreferred_CXP6_X4,
CxpLinkConfigurationPreferred_CXP1_X5,
CxpLinkConfigurationPreferred_CXP2_X5,
CxpLinkConfigurationPreferred_CXP3_X5,
CxpLinkConfigurationPreferred_CXP5_X5,
CxpLinkConfigurationPreferred_CXP6_X5,
CxpLinkConfigurationPreferred_CXP1_X6,
CxpLinkConfigurationPreferred_CXP2_X6,
CxpLinkConfigurationPreferred_CXP3_X6,

```

- CxpLinkConfigurationPreferred_CXP5_X6,
 - CxpLinkConfigurationPreferred_CXP6_X6,
 - NUM_CXPLINKCONFIGURATIONPREFERRED }
- enum CxpLinkConfigurationEnums {
 - CxpLinkConfiguration_Auto,
 - CxpLinkConfiguration_CXP1_X1,
 - CxpLinkConfiguration_CXP2_X1,
 - CxpLinkConfiguration_CXP3_X1,
 - CxpLinkConfiguration_CXP5_X1,
 - CxpLinkConfiguration_CXP6_X1,
 - CxpLinkConfiguration_CXP1_X2,
 - CxpLinkConfiguration_CXP2_X2,
 - CxpLinkConfiguration_CXP3_X2,
 - CxpLinkConfiguration_CXP5_X2,
 - CxpLinkConfiguration_CXP6_X2,
 - CxpLinkConfiguration_CXP1_X3,
 - CxpLinkConfiguration_CXP2_X3,
 - CxpLinkConfiguration_CXP3_X3,
 - CxpLinkConfiguration_CXP5_X3,
 - CxpLinkConfiguration_CXP6_X3,
 - CxpLinkConfiguration_CXP1_X4,
 - CxpLinkConfiguration_CXP2_X4,
 - CxpLinkConfiguration_CXP3_X4,
 - CxpLinkConfiguration_CXP5_X4,
 - CxpLinkConfiguration_CXP6_X4,
 - CxpLinkConfiguration_CXP1_X5,
 - CxpLinkConfiguration_CXP2_X5,
 - CxpLinkConfiguration_CXP3_X5,
 - CxpLinkConfiguration_CXP5_X5,
 - CxpLinkConfiguration_CXP6_X5,
 - CxpLinkConfiguration_CXP1_X6,
 - CxpLinkConfiguration_CXP2_X6,
 - CxpLinkConfiguration_CXP3_X6,
 - CxpLinkConfiguration_CXP5_X6,
 - CxpLinkConfiguration_CXP6_X6,
 - NUM_CXPLINKCONFIGURATION }
- enum CxpConnectionTestModeEnums {
 - CxpConnectionTestMode_Off,
 - CxpConnectionTestMode_Mode1,
 - NUM_CXPCONNECTIONTESTMODE }
- enum CxpPoCxpStatusEnums {
 - CxpPoCxpStatus_Auto,
 - CxpPoCxpStatus_Off,
 - CxpPoCxpStatus_Tripped,
 - NUM_CXPPOCXPSTATUS }
- enum InferenceBoxType {
 - INFERENCE_BOX_TYPE_RECTANGLE = 0,
 - INFERENCE_BOX_TYPE_CIRCLE = 1,
 - INFERENCE_BOX_TYPE_ROTATED_RECTANGLE = 2 }

Inference Bounding Box Type.
- enum Error {
 - SPINNAKER_ERR_SUCCESS = 0,
 - SPINNAKER_ERR_ERROR = -1001,
 - SPINNAKER_ERR_NOT_INITIALIZED = -1002,
 - SPINNAKER_ERR_NOT_IMPLEMENTED = -1003,
 - SPINNAKER_ERR_RESOURCE_IN_USE = -1004,
 - SPINNAKER_ERR_ACCESS_DENIED = -1005,
 - SPINNAKER_ERR_INVALID_HANDLE = -1006,

```

SPINNAKER_ERR_INVALID_ID = -1007,
SPINNAKER_ERR_NO_DATA = -1008,
SPINNAKER_ERR_INVALID_PARAMETER = -1009,
SPINNAKER_ERR_IO = -1010,
SPINNAKER_ERR_TIMEOUT = -1011,
SPINNAKER_ERR_ABORT = -1012,
SPINNAKER_ERR_INVALID_BUFFER = -1013,
SPINNAKER_ERR_NOT_AVAILABLE = -1014,
SPINNAKER_ERR_INVALID_ADDRESS = -1015,
SPINNAKER_ERR_BUFFER_TOO_SMALL = -1016,
SPINNAKER_ERR_INVALID_INDEX = -1017,
SPINNAKER_ERR_PARSING_CHUNK_DATA = -1018,
SPINNAKER_ERR_INVALID_VALUE = -1019,
SPINNAKER_ERR_RESOURCE_EXHAUSTED = -1020,
SPINNAKER_ERR_OUT_OF_MEMORY = -1021,
SPINNAKER_ERR_BUSY = -1022,
GENICAM_ERR_INVALID_ARGUMENT = -2001,
GENICAM_ERR_OUT_OF_RANGE = -2002,
GENICAM_ERR_PROPERTY = -2003,
GENICAM_ERR_RUN_TIME = -2004,
GENICAM_ERR_LOGICAL = -2005,
GENICAM_ERR_ACCESS = -2006,
GENICAM_ERR_TIMEOUT = -2007,
GENICAM_ERR_DYNAMIC_CAST = -2008,
GENICAM_ERR_GENERIC = -2009,
GENICAM_ERR_BAD_ALLOCATION = -2010,
SPINNAKER_ERR_IM_CONVERT = -3001,
SPINNAKER_ERR_IM_COPY = -3002,
SPINNAKER_ERR_IM_MALLOC = -3003,
SPINNAKER_ERR_IM_NOT_SUPPORTED = -3004,
SPINNAKER_ERR_IM_HISTOGRAM_RANGE = -3005,
SPINNAKER_ERR_IM_HISTOGRAM_MEAN = -3006,
SPINNAKER_ERR_IM_MIN_MAX = -3007,
SPINNAKER_ERR_IM_COLOR_CONVERSION = -3008,
SPINNAKER_ERR_IM_DECOMPRESSION = -3009,
SPINNAKER_ERR_CUSTOM_ID = -10000 }

```

Spinnaker enum definitions.

- enum EventType {
 SPINNAKER_EVENT_ARRIVAL_REMOVAL,
 SPINNAKER_EVENT_DEVICE,
 SPINNAKER_EVENT_DEVICE_SPECIFIC,
 SPINNAKER_EVENT_NEW_BUFFER,
 SPINNAKER_EVENT_LOGGING_EVENT,
 SPINNAKER_EVENT_UNKNOWN,
 SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL }

Event types in Spinnaker.

- enum PixelFormatNamespaceID {
 SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
 SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
 SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
 SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
 SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,
 SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }

This enum represents the namespace in which the TL specific pixel format resides.

- enum ColorProcessingAlgorithm {
 DEFAULT,
 NO_COLOR_PROCESSING,
 NEAREST_NEIGHBOR,


```

NEAREST_NEIGHBOR_AVG,
BILINEAR,
EDGE_SENSING,
HQ_LINEAR,
IPP,
DIRECTIONAL_FILTER,
RIGOROUS,
WEIGHTED_DIRECTIONAL_FILTER }

```

Color processing algorithms.

- enum `ImageFileFormat` {
`FROM_FILE_EXT` = -1,
`PGM`,
`PPM`,
`BMP`,
`JPEG`,
`JPEG2000`,
`TIFF`,
`PNG`,
`RAW`,
`JPEG12_C`,
`IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

File formats to be used for saving images to disk.

- enum `ImageStatus` {
`IMAGE_UNKNOWN_ERROR` = -1,
`IMAGE_NO_ERROR` = 0,
`IMAGE_CRC_CHECK_FAILED` = 1,
`IMAGE_DATA_OVERFLOW` = 2,
`IMAGE_MISSING_PACKETS`,
`IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT`,
`IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT`,
`IMAGE_PACKETID_INCONSISTENT`,
`IMAGE_MISSING_LEADER` = 7,
`IMAGE_MISSING_TRAILER`,
`IMAGE_DATA_INCOMPLETE`,
`IMAGE_INFO_INCONSISTENT`,
`IMAGE_CHUNK_DATA_INVALID` = 11,
`IMAGE_NO_SYSTEM_RESOURCES` = 12 }

Status of images returned from `GetNextImage()` call.

- enum `StatisticsChannel` {
`GREY`,
`RED`,
`GREEN`,
`BLUE`,
`HUE`,
`SATURATION`,
`LIGHTNESS`,
`NUM_STATISTICS_CHANNELS` }

Channels that allow statistics to be calculated.

- enum `SpinnakerLogLevel` {
`LOG_LEVEL_OFF` = -1,
`LOG_LEVEL_FATAL` = 0,
`LOG_LEVEL_ALERT` = 100,
`LOG_LEVEL_CRIT` = 200,
`LOG_LEVEL_ERROR` = 300,
`LOG_LEVEL_WARN` = 400,
`LOG_LEVEL_NOTICE` = 500,
`LOG_LEVEL_INFO` = 600,

```
LOG_LEVEL_DEBUG = 700,
LOG_LEVEL_NOTSET = 800 }
```

log levels

- enum `PayloadTypeInfoIDs` {
`PAYLOAD_TYPE_UNKNOWN` = 0,
`PAYLOAD_TYPE_IMAGE` = 1,
`PAYLOAD_TYPE_RAW_DATA` = 2,
`PAYLOAD_TYPE_FILE` = 3,
`PAYLOAD_TYPE_CHUNK_DATA` = 4,
`PAYLOAD_TYPE_JPEG` = 5,
`PAYLOAD_TYPE_JPEG2000` = 6,
`PAYLOAD_TYPE_H264` = 7,
`PAYLOAD_TYPE_CHUNK_ONLY` = 8,
`PAYLOAD_TYPE_DEVICE_SPECIFIC` = 9,
`PAYLOAD_TYPE_MULTI_PART` = 10,
`PAYLOAD_TYPE_CUSTOM_ID` = 1000,
`PAYLOAD_TYPE_EXTENDED_CHUNK` = 1001,
`PAYLOAD_TYPE_LOSSLESS_COMPRESSED` = 1002,
`PAYLOAD_TYPE_LOSSY_COMPRESSED` = 1003,
`PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED` = 1004,
`PAYLOAD_TYPE_CHUNK_DATA_LOSSLESS_COMPRESSED` = 1005,
`PAYLOAD_TYPE_CHUNK_DATA_LOSSY_COMPRESSED` = 1006 }
- enum `ActionCommandStatus` {
`ACTION_COMMAND_STATUS_OK` = 0,
`ACTION_COMMAND_STATUS_NO_REF_TIME`,
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,
`ACTION_COMMAND_STATUS_ACTION_LATE`,
`ACTION_COMMAND_STATUS_ERROR` }

Possible Status Codes Returned from Action Command.

- enum `PixelFormatIntType` {
`IntType_UINT8`,
`IntType_INT8`,
`IntType_UINT10`,
`IntType_UINT10p`,
`IntType_UINT10P`,
`IntType_UINT12`,
`IntType_UINT12p`,
`IntType_UINT12P`,
`IntType_UINT14`,
`IntType_UINT16`,
`IntType_INT16`,
`IntType_FLOAT32`,
`IntType_UNKNOWN` }

Possible integer types and packing used in a pixel format.

- enum `BufferOwnership` {
`BUFFER_OWNERSHIP_SYSTEM`,
`BUFFER_OWNERSHIP_USER` }
- enum `CCMColorTemperature` {
`TUNGSTEN_2800K`,
`WARM_FLUORESCENT_3000K`,
`COOL_FLUORESCENT_4000K`,
`SUNNY_5000K`,
`CLOUDY_6500K`,
`SHADE_8000K` }
- enum `CCMType` {
`LINEAR_3X3`,
`POLYNOMIAL_9X3` }

- enum `CCMSensor` { `IMX250` }
- enum `StreamTypeEnum` {
`StreamType_GigEVision`,
`StreamType_CameraLink`,
`StreamType_CameraLinkHS`,
`StreamType_CoaXPress`,
`StreamType_USB3Vision`,
`StreamType_Custom`,
`NUMSTREAMTYPE` }

The enum definitions for TL Device nodes from the transport layer .xml files.

- enum `StreamBufferCountModeEnum` {
`StreamBufferCountMode_Manual`,
`StreamBufferCountMode_Auto`,
`NUMSTREAMBUFFERCOUNTMODE` }
- enum `StreamBufferHandlingModeEnum` {
`StreamBufferHandlingMode_OldestFirst`,
`StreamBufferHandlingMode_OldestFirstOverwrite`,
`StreamBufferHandlingMode_NewestOnly`,
`StreamBufferHandlingMode_NewestFirst`,
`NUMSTREAMBUFFERHANDLINGMODE` }
- enum `DeviceTypeEnum` {
`DeviceType_GigEVision`,
`DeviceType_CameraLink`,
`DeviceType_CameraLinkHS`,
`DeviceType_CoaXPress`,
`DeviceType_USB3Vision`,
`DeviceType_Custom`,
`NUMDEVICETYPE` }
- enum `DeviceAccessStatusEnum` {
`DeviceAccessStatus_Unknown`,
`DeviceAccessStatus_ReadWrite`,
`DeviceAccessStatus_ReadOnly`,
`DeviceAccessStatus_NoAccess`,
`DeviceAccessStatus_Busy`,
`DeviceAccessStatus_OpenReadWrite`,
`DeviceAccessStatus_OpenReadOnly`,
`NUMDEVICEACCESSSTATUS` }
- enum `GevCCPEnum` {
`GevCCP_EnumEntry_GevCCP_OpenAccess`,
`GevCCP_EnumEntry_GevCCP_ExclusiveAccess`,
`GevCCP_EnumEntry_GevCCP_ControlAccess`,
`NUMGEVCCP` }
- enum `GUIXMLLocationEnum` {
`GUIXMLLocation_Device`,
`GUIXMLLocation_Host`,
`NUMGUIXMLLOCATION` }
- enum `GenICamXMLLocationEnum` {
`GenICamXMLLocation_Device`,
`GenICamXMLLocation_Host`,
`NUMGENICAMXMLLOCATION` }
- enum `DeviceEndiannessMechanismEnum` {
`DeviceEndiannessMechanism_Legacy`,
`DeviceEndiannessMechanism_Standard`,
`NUMDEVICEENDIANESSMECHANISM` }
- enum `DeviceCurrentSpeedEnum` {
`DeviceCurrentSpeed_UnknownSpeed`,
`DeviceCurrentSpeed_LowSpeed`,

- DeviceCurrentSpeed_FullSpeed,
- DeviceCurrentSpeed_HighSpeed,
- DeviceCurrentSpeed_SuperSpeed,
- NUMDEVICECURRENTSPEED }
- enum InterfaceTypeEnum {
 - InterfaceType_GigEVision,
 - InterfaceType_CameraLink,
 - InterfaceType_CameraLinkHS,
 - InterfaceType_CoaXPress,
 - InterfaceType_USB3Vision,
 - InterfaceType_Custom,
 - NUMINTERFACETYPE }
- enum POEStatusEnum {
 - POEStatus_NotSupported,
 - POEStatus_PowerOff,
 - POEStatus_PowerOn,
 - NUMPOESTATUS }
- enum FilterDriverStatusEnum {
 - FilterDriverStatus_NotSupported,
 - FilterDriverStatus_Disabled,
 - FilterDriverStatus_Enabled,
 - NUMFILTERDRIVERSTATUS }
- enum TLTypeEnum {
 - TLType_GigEVision,
 - TLType_CameraLink,
 - TLType_CameraLinkHS,
 - TLType_CoaXPress,
 - TLType_USB3Vision,
 - TLType_Mixed,
 - TLType_Custom,
 - NUMTLTYPE }

Functions

- class [DEPRECATED_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API A↔
 VIRecorder
Provides the functionality for the user to record images to an AVI file.
- template<class T, class B >
 bool [operator==](#) (const std::nullptr_t, const [BasePtr](#)< T, B > &rhs)
Pointer equal.

Variables

- const uint64_t [EVENT_TIMEOUT_NONE](#) = 0
Timeout values for getting next image, device, or interface event.
- const uint64_t [EVENT_TIMEOUT_INFINITE](#) = 0xFFFFFFFFFFFFFFFF

14.7 Spinnaker::GenApi Namespace Reference

Classes

- class [AutoLock](#)

- class [BooleanNode](#)
Interface for string properties.
- class [CategoryNode](#)
Interface for string properties.
- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.
- class [CChunkAdapterDcam](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterGeneric](#)
- class [CChunkAdapterGEV](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.
- class [CChunkPort](#)
Port attachable to a chunk in a buffer.
- class [CEnumerationTRef](#)
Interface for string properties.
- class [CEventAdapter](#)
Delivers Events to ports.
- class [CEventAdapter1394](#)
Distribute the events to the node map.
- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.
- class [CEventAdapterGEV](#)
Connects a GigE Event to a node map.
- class [CEventAdapterU3V](#)
Connects a U3V Event to a node map.
- class [CEventPort](#)
Port attachable to an event.
- class [CFeatureBag](#)
Bag holding streamable features of a nodetree.
- class [CFloatPtr](#)
SmartPointer for IFloat interface pointer.
- class [CGeneric_XMLLoaderParams](#)
Empty base class used by class [CNodeMapRef](#) as generic template argument.
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [CNodeCallback](#)
callback body instance for INode pointers
- class [CNodeMapFactory](#)
The node map factory is used for creating node maps from camera description files.
- class [CNodeMapRef](#)
SmartPointer for NodeMaps with create function.
- class [CNodeMapRefT](#)
SmartPointer template for NodeMaps with create function.
- class [CommandNode](#)
Interface for string properties.
- class [Counter](#)
Definition of a simple [Counter](#) class.

- class [CPointer](#)
Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.
- class [CPortImpl](#)
Standard implementation for a port.
- class [CPortWriteList](#)
Container holding a list of port write commands.
- class [CRegisterPortImpl](#)
Standard implementation for a port using a register based transport layer.
- class [CSelectorSet](#)
The set of selectors selecting a given node.
- class [CTestPortStruct](#)
Implements a register spaces based on a C++ struct.
- class [double_autovector_t](#)
Vector of doubles with reference counting.
- class [EAccessModeClass](#)
Holds conversion methods for the access mode enumeration.
- class [ECachingModeClass](#)
Holds conversion methods for the caching mode enumeration.
- class [EDisplayNotationClass](#)
Holds conversion methods for the notation type of floats.
- class [EEndianessClass](#)
Holds conversion methods for the endianess enumeration.
- class [EGenApiSchemaVersionClass](#)
helper class converting `EGenApiSchemaVersion` from and to string
- class [EInputDirectionClass](#)
Holds conversion methods for the notation type of floats.
- class [ENamespaceClass](#)
Holds conversion methods for the namespace enumeration.
- class [EnumEntryNode](#)
[Interface](#) for string properties.
- class [EnumNode](#)
[Interface](#) for string properties.
- class [ERepresentationClass](#)
Holds conversion methods for the representation enumeration.
- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [ESlopeClass](#)
Holds conversion methods for the converter formulas.
- class [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- class [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [FileProtocolAdapter](#)
Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.
- class [FloatNode](#)
[Interface](#) for string properties.
- class [FloatRegNode](#)
[Interface](#) for string properties.
- class [Function_NodeCallback](#)

- Container for a function pointer.*
- class [IDevFileStreamBase](#)
- class [IDevFileStreamBuf](#)
- class [int64_autovector_t](#)
- Vector of integers with reference counting.*
- class [IntegerNode](#)
- Interface for string properties.*
- class [IntRegNode](#)
- Interface for string properties.*
- class [Member_NodeCallback](#)
- Container for a member function pointer.*
- class [Node](#)
- class common to all nodes*
- class [NodeMap](#)
- Smart pointer template for NodeMaps with create function.*
- class [ODevFileStreamBase](#)
- class [ODevFileStreamBuf](#)
- class [PortNode](#)
- Interface for value properties.*
- class [PortRecorder](#)
- Interface for recording write commands on a port.*
- class [PortReplay](#)
- Interface for replaying write commands on a port.*
- class [RegisterNode](#)
- Interface for string properties.*
- class [SpinTestCamera](#)
- class [StringNode](#)
- Interface for string properties.*
- class [StringRegNode](#)
- Interface for string properties.*
- class [ValueNode](#)
- Interface for value properties.*

Typedefs

- typedef [BooleanNode](#) [CBooleanRef](#)
- typedef [CategoryNode](#) [CCategoryRef](#)
- typedef [CommandNode](#) [CCommandRef](#)
- typedef [EnumEntryNode](#) [CEnumEntryRef](#)
- typedef [EnumNode](#) [CEnumerationRef](#)
- typedef [ODevFileStreamBase](#)< char, std::char_traits< char > > [ODevFileStream](#)
- typedef [IDevFileStreamBase](#)< char, std::char_traits< char > > [IDevFileStream](#)
- typedef [FloatNode](#) [CFloatRef](#)
- typedef node_vector [NodeList_t](#)
- a list of node references*
- typedef intptr_t [CallbackHandleType](#)
- the callback handle for nodes*
- typedef [IntegerNode](#) [CIntegerRef](#)
- typedef [Node](#) [CNodeRef](#)
- typedef [Node](#) [CSelectorRef](#)
- typedef [NodeMap](#) [CNodeMapRef](#)

- typedef [CPointer](#)< [IBase](#) > [CBasePtr](#)
SmartPointer for IBase interface pointer.
- typedef [CPointer](#)< [INode](#), [IBase](#) > [CNodePtr](#)
SmartPointer for INode interface pointer.
- typedef [CPointer](#)< [IValue](#) > [CValuePtr](#)
SmartPointer for IValue interface pointer.
- typedef [CPointer](#)< [ICategory](#) > [CCategoryPtr](#)
SmartPointer for ICategory interface pointer.
- typedef [CPointer](#)< [IBoolean](#) > [CBooleanPtr](#)
SmartPointer for IBoolean interface pointer.
- typedef [CPointer](#)< [IInteger](#) > [CIntegerPtr](#)
SmartPointer for IInteger interface pointer.
- typedef [CPointer](#)< [IString](#) > [CStringPtr](#)
SmartPointer for IString interface pointer.
- typedef [CPointer](#)< [IRegister](#) > [CRegisterPtr](#)
SmartPointer for IRegister interface pointer.
- typedef [CPointer](#)< [IEnumeration](#) > [CEnumerationPtr](#)
SmartPointer for IEnumeration interface pointer.
- typedef [CPointer](#)< [IEnumEntry](#) > [CEnumEntryPtr](#)
SmartPointer for IEnumEntry interface pointer.
- typedef [CPointer](#)< [IPort](#) > [CPortPtr](#)
SmartPointer for IPort interface pointer.
- typedef [CPointer](#)< [IPortReplay](#) > [CPortReplayPtr](#)
SmartPointer for IPortReplay interface pointer.
- typedef [CPointer](#)< [IPortRecorder](#) > [CPortRecorderPtr](#)
SmartPointer for IPortRecorder interface pointer.
- typedef [CPointer](#)< [IPortWriteList](#), [IPortWriteList](#) > [CPortWriteListPtr](#)
SmartPointer for IPortWriteList interface pointer.
- typedef [CPointer](#)< [IChunkPort](#) > [CChunkPortPtr](#)
SmartPointer for IChunkPort interface pointer.
- typedef [CPointer](#)< [INodeMap](#), [INodeMap](#) > [CNodeMapPtr](#)
SmartPointer for INodeMap interface pointer.
- typedef [CPointer](#)< [INodeMapDyn](#), [INodeMap](#) > [CNodeMapDynPtr](#)
SmartPointer for INodeMapDyn interface pointer.
- typedef [CPointer](#)< [IDeviceInfo](#), [INodeMap](#) > [CDeviceInfoPtr](#)
SmartPointer for IDeviceInfo interface pointer.
- typedef [CPointer](#)< [ISelector](#) > [CSelectorPtr](#)
SmartPointer for ISelector interface pointer.
- typedef [CPointer](#)< [ICommand](#) > [CCommandPtr](#)
SmartPointer for ICommand interface pointer.
- typedef [CPointer](#)< [IPortConstruct](#) > [CPortConstructPtr](#)
SmartPointer for IPortConstruct interface pointer.
- typedef [PortNode](#) [CPortRef](#)
- typedef [PortRecorder](#) [CPortRecorderRef](#)
Reference to an IPortRecorder pointer.
- typedef [RegisterNode](#) [CRegisterRef](#)
- typedef [StringNode](#) [CStringRef](#)
- typedef [GenlCam::gcstring_vector](#) [StringList_t](#)
A list of strings.
- typedef [ValueNode](#) [CValueRef](#)

Enumerations

- enum [GVCP_MESSAGE_TAGS](#) {
[TAG_EVENT_CMD](#) = 0xc0,
[TAG_EVENTDATA_CMD](#) = 0xc2 }
- enum [ECallbackType](#) {
[cbPostInsideLock](#) = 1,
[cbPostOutsideLock](#) = 2 }
the type of callback
- enum [ECacheUsage_t](#) {
[CacheUsage_Automatic](#),
[CacheUsage_ForceWrite](#),
[CacheUsage_ForceRead](#),
[CacheUsage_Ignore](#) }
Lists the cache usage strategies.
- enum [EContentType_t](#) {
[ContentType_Xml](#),
[ContentType_ZippedXml](#) }
Lists the processable file types.
- enum [ESign](#) {
[Signed](#),
[Unsigned](#),
[_UndefinedSign](#) }
signed or unsigned integers
- enum [EAccessMode](#) {
[NI](#),
[NA](#),
[WO](#),
[RO](#),
[RW](#),
[_UndefinedAccesMode](#),
[_CycleDetectAccesMode](#) }
access mode of a node
- enum [EVisibility](#) {
[Beginner](#) = 0,
[Expert](#) = 1,
[Guru](#) = 2,
[Invisible](#) = 3,
[_UndefinedVisibility](#) = 99 }
recommended visibility of a node
- enum [ECachingMode](#) {
[NoCache](#),
[WriteThrough](#),
[WriteAround](#),
[_UndefinedCachingMode](#) }
caching mode of a register
- enum [ERepresentation](#) {
[Linear](#),
[Logarithmic](#),
[Boolean](#),
[PureNumber](#),
[HexNumber](#),
[IPV4Address](#),
[MACAddress](#),
[_UndefinedRepresentation](#) }
recommended representation of a node value

- enum [EEndianness](#) {
[BigEndian](#),
[LittleEndian](#),
[_UndefinedEndian](#) }
Endianness of a value in a register.
- enum [ENameSpace](#) {
[Custom](#),
[Standard](#),
[_UndefinedNameSpace](#) }
Defines if a node name is standard or custom.
- enum [EStandardNameSpace](#) {
[None](#),
[GEV](#),
[IIDC](#),
[CL](#),
[USB](#),
[_UndefinedStandardNameSpace](#) }
Defines from which standard namespace a node name comes from.
- enum [EYesNo](#) {
[Yes](#) = 1,
[No](#) = 0,
[_UndefinedYesNo](#) = 2 }
Defines the choices of a Yes/No alternative.
- enum [ESlope](#) {
[Increasing](#),
[Decreasing](#),
[Varying](#),
[Automatic](#),
[_UndefinedESlope](#) }
typedef for formula type
- enum [EXMLValidation](#) {
[xvLoad](#) = 0x00000001L,
[xvCycles](#) = 0x00000002L,
[xvSFNC](#) = 0x00000004L,
[xvDefault](#) = 0x00000000L,
[xvAll](#) = 0xffffffffL,
[_UndefinedEXMLValidation](#) = 0x80000000L }
typedef describing the different validity checks which can be performed on an XML file
- enum [EDisplayNotation](#) {
[fnAutomatic](#),
[fnFixed](#),
[fnScientific](#),
[_UndefinedEDisplayNotation](#) }
typedef for float notation
- enum [EInterfaceType](#) {
[intfIValue](#),
[intfIBase](#),
[intfInteger](#),
[intfBoolean](#),
[intfCommand](#),
[intfFloat](#),
[intfString](#),
[intfRegister](#),
[intfCategory](#),
[intfEnumeration](#),
[intfEnumEntry](#),
[intfIPort](#) }

- typedef for interface type*
- enum [ELinkType](#) {
[ctParentNodes](#),
[ctReadingChildren](#),
[ctWritingChildren](#),
[ctInvalidatingChildren](#),
[ctDependingNodes](#),
[ctTerminalNodes](#) }
- typedef for link type*
- enum [EIncMode](#) {
[noIncrement](#),
[fixedIncrement](#),
[listIncrement](#) }
- typedef for increment mode*
- enum [EInputDirection](#) {
[idFrom](#),
[idTo](#),
[idNone](#) }
- typedef for link type*
- enum [EGenApiSchemaVersion](#) {
[v1_0](#) = 1,
[v1_1](#) = 2,
[_Undefined](#) = -1 }
- GenApi schema version.*

Functions

- void [SPINNAKER_API SET_GUID](#) (SPIN_GUID &name, uint32_t l, uint16_t w1, uint16_t w2, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4, uint8_t b5, uint8_t b6, uint8_t b7, uint8_t b8)
- virtual void [operator=](#) (bool Value)
Set node value.
- virtual bool [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const
Get node value.
- virtual [EYesNo](#) [CacheChunkData](#) () const =0
Indicates if the chunk a adapter must hold a cached version of the chunk data.
- virtual bool [IsDone](#) (bool [Verify](#)=true)=0
Query whether the command is executed.
- virtual [GenICam::gcstring](#) [GetVendorName](#) ()=0
Get the vendor name.
- virtual [GenICam::gcstring](#) [GetToolTip](#) ()=0
Get tool tip.
- virtual [GenICam::gcstring](#) [GetStandardNameSpace](#) ()=0
Get the standard name space.
- virtual void [GetGenApiVersion](#) ([GenICam::Version_t](#) &Version, uint16_t &Build)=0
Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) ([GenICam::Version_t](#) &Version)=0
Get the schema version number.
- virtual void [GetDeviceVersion](#) ([GenICam::Version_t](#) &Version)=0
Get the version of the device description file.
- virtual [GenICam::gcstring](#) [GetProductGuid](#) ()=0

- Get the Guid describing the product.*

 - virtual [GenlCam::gcstring GetVersionGuid](#) ()=0
- Get the Guid describing the product version.*

 - virtual [GenlCam::gcstring GetSymbolic](#) () const =0
- Get symbolic enum value.*

 - virtual double [GetNumericValue](#) ()=0
- Get double number associated with the entry.*

 - virtual bool [IsSelfClearing](#) ()=0
- Indicates if the corresponding EnumEntry is self clearing.*

 - virtual void [GetEntries](#) ([NodeList_t](#) &Entries)=0
- Get list of entry nodes.*

 - virtual [IEnumeration](#) & [operator=](#) (const [GenlCam::gcstring](#) &ValueStr)=0
- Set string node value.*

 - virtual void [SetIntValue](#) (int64_t Value, bool [Verify](#)=true)=0
- Set integer node value.*

 - virtual [GenlCam::gcstring operator*](#) ()=0
- Get string node value.*

 - virtual int64_t [GetIntValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
- Get integer node value.*

 - virtual [IEnumEntry](#) * [GetEntryByName](#) (const [GenlCam::gcstring](#) &Symbolic)=0
- Get an entry node by name.*

 - virtual [IEnumEntry](#) * [GetEntry](#) (const int64_t IntValue)=0
- Get an entry node by its IntValue.*

 - virtual [IEnumEntry](#) * [GetCurrentEntry](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
- Get the current entry.*

 - virtual [IEnumeration](#) & [operator=](#) (EnumT Value)=0
- Set node value.*

 - virtual [IEnumEntry](#) * [GetEntry](#) (const EnumT Value)=0
- returns the EnumEntry object belonging to the Value*

 - virtual [IFloat](#) & [operator=](#) (double Value)=0
- Set node value.*

 - virtual double [GetMin](#) ()=0
- Get minimum value allowed.*

 - virtual double [GetMax](#) ()=0
- Get maximum value allowed.*

 - virtual bool [HasInc](#) ()=0
- True if the float has a constant increment.*

 - virtual [EIncMode](#) [GetIncMode](#) ()=0
- Get increment mode.*

 - virtual double [GetInc](#) ()=0
- Get the constant increment if there is any.*

 - virtual [double_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)=0
- Get list of valid value.*

 - virtual [ERepresentation](#) [GetRepresentation](#) ()=0
- Get recommended representation.*

 - virtual [GenlCam::gcstring](#) [GetUnit](#) () const =0
- Get the physical unit name.*

 - virtual [EDisplayNotation](#) [GetDisplayNotation](#) () const =0
- Get the way the float should be converted to a string.*

 - virtual int64_t [GetDisplayPrecision](#) () const =0
- Get the precision to be used when converting the float to a string.*

- virtual void [ImposeMin](#) (double Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)=0
Restrict maximum value.
- virtual [Integer](#) & [operator=](#) (int64_t Value)=0
Set node value.
- virtual void [ImposeMin](#) (int64_t Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (int64_t Value)=0
Restrict maximum value.
- virtual [GenApi::ENamespace](#) [GetNameSpace](#) () const =0
Get name space.
- virtual [EVisibility](#) [GetVisibility](#) () const =0
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()=0
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const =0
Is the node value cacheable.
- virtual [EYesNo](#) [IsAccessModeCacheable](#) () const =0
True if the AccessMode can be cached.
- virtual [ECachingMode](#) [GetCachingMode](#) () const =0
Get Caching Mode.
- virtual int64_t [GetPollingTime](#) () const =0
recommended polling time (for non-cacheable nodes)
- virtual [GenICam::gcstring](#) [GetDescription](#) () const =0
Get a long description of the node.
- virtual [GenICam::gcstring](#) [GetDisplayName](#) () const =0
Get a name string for display.
- virtual [GenICam::gcstring](#) [GetDeviceName](#) () const =0
Get a name of the device.
- virtual void [GetChildren](#) ([GenApi::NodeList_t](#) &Children, [ELinkType](#) LinkType=[ctReadingChildren](#)) const =0
Get all nodes this node directly depends on.
- virtual void [GetParents](#) ([GenApi::NodeList_t](#) &Parents) const =0
Gets all nodes this node is directly depending on.
- virtual [CallbackHandleType](#) [RegisterCallback](#) ([CNodeCallback](#) *pCallback)=0
Register change callback Takes ownership of the CNodeCallback object.
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)=0
De register change callback Destroys CNodeCallback object.
- virtual [INodeMap](#) * [GetNodeMap](#) () const =0
Retrieves the central node map.
- virtual [GenICam::gcstring](#) [GetEventID](#) () const =0
Get the EventId of the node.
- virtual bool [IsStreamable](#) () const =0
True if the node is streamable.
- virtual void [GetPropertyNames](#) ([GenICam::gcstring_vector](#) &PropertyNames) const =0
Returns a list of the names all properties set during initialization.
- virtual bool [GetProperty](#) (const [GenICam::gcstring](#) &PropertyName, [GenICam::gcstring](#) &ValueStr, [GenICam::gcstring](#) &AttributeStr)=0
Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.
- virtual void [ImposeAccessMode](#) ([EAccessMode](#) ImposedAccessMode)=0

- Imposes an access mode to the natural access mode of the node.*

 - virtual void **ImposeVisibility** (**EVisibility** ImposedVisibility)=0

Imposes a visibility to the natural visibility of the node.
- virtual **INode** * **GetAlias** () const =0

Retrieves the a node which describes the same feature in a different way.
- virtual **INode** * **GetCastAlias** () const =0

Retrieves the a node which describes the same feature so that it can be casted.
- virtual **GenICam::gcstring** **GetDocuURL** () const =0

Gets a URL pointing to the documentation of that feature.
- virtual bool **IsDeprecated** () const =0

True if the node should not be used any more.
- virtual **EInterfaceType** **GetPrincipalInterfaceType** () const =0

Get the type of the main interface of a node.
- virtual bool **IsFeature** () const =0

True if the node can be reached via category nodes from a category node named "Root".
- virtual bool **operator==** (int nullPtr) const =0
- virtual bool **operator!=** (int nullPtr) const =0
- bool **IsReadable** (**EAccessMode** AccessMode)

Tests if readable.
- bool **IsReadable** (const **IBase** *p)

Checks if a node is readable.
- bool **IsReadable** (const **IBase** &r)

Checks if a node is readable.
- bool **IsWritable** (**EAccessMode** AccessMode)

Tests if writable.
- bool **IsWritable** (const **IBase** *p)

Checks if a node is writable.
- bool **IsWritable** (const **IBase** &r)

Checks if a node is writable.
- bool **IsImplemented** (**EAccessMode** AccessMode)

Tests if implemented.
- bool **IsImplemented** (const **IBase** *p)

Checks if a node is implemented.
- bool **IsImplemented** (const **IBase** &r)

Checks if a node is implemented.
- bool **IsAvailable** (**EAccessMode** AccessMode)

Tests if available.
- bool **IsAvailable** (const **IBase** *p)

Checks if a node is available.
- bool **IsAvailable** (const **IBase** &r)

Checks if a node is available.
- **EAccessMode** **Combine** (**EAccessMode** Peter, **EAccessMode** Paul)

Computes which access mode the two guards allow together.
- bool **IsVisible** (**EVisibility** Visibility, **EVisibility** MaxVisibility)

Tests Visibility CAVE : this relies on the EVisibility enum's coding.
- **EVisibility** **Combine** (**EVisibility** Peter, **EVisibility** Paul)

Computes which visibility the two guards allow together.
- bool **IsCacheable** (**ECachingMode** CachingMode)

Tests Cacheability.
- **ECachingMode** **Combine** (**ECachingMode** Peter, **ECachingMode** Paul)

Computes which CachingMode results from a combination.

- virtual [INode](#) * [GetNode](#) (const [GenICam::gcstring](#) &Name) const =0
Retrieves the node from the central map by Name.
- virtual void [InvalidateNodes](#) () const =0
Invalidates all nodes.
- virtual bool [Connect](#) ([IPort](#) *pPort, const [GenICam::gcstring](#) &PortName) const =0
Connects a port to a port node with given name.
- virtual bool [Connect](#) ([IPort](#) *pPort) const =0
Connects a port to the standard port "Device".
- virtual void [Poll](#) (int64_t ElapsedTime)=0
Fires nodes which have a polling time.
- virtual [CLock](#) & [GetLock](#) () const =0
Returns the lock which guards the node map.
- virtual uint64_t [GetNumNodes](#) () const =0
Get the number of nodes in the map.
- virtual void [LoadXMLFromFile](#) (const [GenICam::gcstring](#) &FileName)=0
Loads an XML from a file.
- virtual void [LoadXMLFromFileInject](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectFileName)=0
Loads an XML from a file with injection.
- virtual void [LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)=0
Loads an XML from a string.
- virtual void [LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLData, const [GenICam::gcstring](#) &InjectXMLData)=0
Loads an XML from a string with injection.
- virtual void [PreprocessXMLFromFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32_t XMLValidation=[xv↵Default](#))=0
Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void [MergeXMLFiles](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectedFileName, const [GenICam::gcstring](#) &OutputFileName)=0
Injects an XML file into a target file.
- virtual void [ExtractIndependentSubtree](#) (const [GenICam::gcstring](#) &XMLData, const [GenICam::gcstring](#) &InjectXMLData, const [GenICam::gcstring](#) &SubTreeRootNodeName, [GenICam::gcstring](#) &Extracted↵Subtree)=0
Extract independent subtree.
- virtual void [GetSupportedSchemaVersions](#) ([GenICam::gcstring_vector](#) &SchemaVersions)=0
Gets a list of supported schema versions.
- virtual void [LoadXMLFromZIPFile](#) (const [GenICam::gcstring](#) &ZipFileName)=0
Loads an XML from a ZIP file.
- virtual void [LoadXMLFromZIPData](#) (const void *zipData, size_t zipSize)=0
Loads an XML from a ZIP data buffer.
- virtual void [PreprocessXMLFromZIPFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32_t XMLValidation=[xv↵Default](#))=0
Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))=0
Writes a chunk of bytes to the port.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()=0
Determines if the port adapter must perform an endianness swap.
- virtual void [Replay](#) ([IPort](#) *pPort)=0

- Replays the write command to the given port interface.*
- virtual void [SetCookie](#) (const int64_t Value)=0
Sets a cookie in case the port implementation want to cache a command list.
- virtual int64_t [GetCookie](#) ()=0
Gets the cookie a port implementation may have set for caching a command list.
- virtual void [StopRecording](#) ()=0
Stops recording.
- virtual void [Get](#) (uint8_t *pBuffer, int64_t Length, bool Verify=false, bool IgnoreCache=false)=0
Fills a buffer with the register's contents.
- virtual int64_t [GetLength](#) ()=0
Retrieves the Length of the register [Bytes].
- virtual int64_t [GetAddress](#) ()=0
Retrieves the Address of the register.
- virtual void [GetSelectedFeatures](#) (FeatureList_t &) const =0
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) (FeatureList_t &) const =0
retrieve the group of features selecting this node
- virtual bool [SetNext](#) (bool Tick=true)=0
Sets digit to next value.
- virtual void [Restore](#) ()=0
Restores the selectors' values found at creation.
- virtual [GenICam::gcstring ToString](#) ()=0
Returns a string representation of the digit.
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)=0
Retrieves an ordered list of selectors.
- virtual int64_t [GetMaxLength](#) ()=0
Retrieves the maximum length of the string in bytes.
- virtual [GenICam::gcstring ToString](#) (bool Verify=false, bool IgnoreCache=false)=0
Get content of the node as string.
- virtual void [FromString](#) (const [GenICam::gcstring](#) &ValueStr, bool Verify=true)=0
Set content of the node as string.
- virtual bool [IsValueCacheValid](#) () const =0
Checks if the value comes from cache or is requested from another node.
- template<class Function >
[CNodeCallback](#) * [make_NodeCallback](#) (INode *pNode, Function function, [ECallbackType](#) CallbackType)
make a new callback object for C functions
- template<class Function >
intptr_t [Register](#) (INode *pNode, Function f, [ECallbackType](#) CallbackType=[cbPostInsideLock](#))
Register a C-function as a callback.
- template<class Client , class Member >
[CNodeCallback](#) * [make_NodeCallback](#) (INode *pNode, Client &client, Member member, [ECallbackType](#) CallbackType)
make a new callback object for member functions
- template<class Client , class Member >
intptr_t [Register](#) (INode *pNode, Client &c, Member m, [ECallbackType](#) CallbackType=[cbPostInsideLock](#))
Register a C++-member function a callback.
- [SPINNAKER_API](#) void [Deregister](#) ([GenApi::CallbackHandleType](#) pCallbackInfo)
Unregistering callback by handle.
- [SPINNAKER_API](#) IDestroy * [CastToIDestroy](#) (INodeMap *pNodeMap)
makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)

- `template<class TCameraParams >`
`void _LoadXMLFromFile (const GenICam::gcstring &FileName)`
- `template<class TCameraParams >`
`void _LoadXMLFromZIPFile (const GenICam::gcstring &ZipFileName)`
- `template<class TCameraParams >`
`void _LoadXMLFromFileInject (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)`
- `template<class TCameraParams >`
`void _LoadXMLFromString (const GenICam::gcstring &XMLData)`
- `template<class TCameraParams >`
`void _LoadXMLFromZIPData (const void *zipData, size_t zipSize)`
- `template<class TCameraParams >`
`void _LoadXMLFromStringInject (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)`
- `template<class TCameraParams >`
`void _GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions)`
- `template<class TCameraParams >`
`GenICam::gcstring _GetDeviceName ()`
- `template<class TCameraParams >`
`void _Poll (int64_t ElapsedTime)`
- `template<class TCameraParams >`
`void _GetNodes (NodeList_t &Nodes)`
- `template<class TCameraParams >`
`INode * _GetNode (const GenICam::gcstring &key)`
- `template<class TCameraParams >`
`void _InvalidateNodes ()`
- `template<class TCameraParams >`
`bool _Connect (IPort *pPort, const GenICam::gcstring &PortName)`
- `template<class TCameraParams >`
`bool _Connect (IPort *pPort)`
- `template<class TCameraParams >`
`bool _ClearXMLCache ()`
- `virtual void PersistFeature (IValue &item)=0`
Stores a feature.
- `SPINNAKER_API std::istream & EatComments (std::istream &is)`
*Helper function ignoring lines starting with comment character '#'.
Reads in persistent data from a stream.*
- `SPINNAKER_API std::istream & operator>> (std::istream &is, CFeatureBag &FeatureBag)`
Reads in persistent data from a stream.
- `SPINNAKER_API std::ostream & operator<< (std::ostream &os, const CFeatureBag &FeatureBag)`
writes out persistent data to a stream
- `template<class T, class B >`
`bool IsReadable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is readable.
- `template<class T, class B >`
`bool IsWritable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Writable.
- `template<class T, class B >`
`bool IsImplemented (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Implemented.
- `template<class T, class B >`
`bool IsAvailable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Available.
- `GenICam::gcstring GetInterfaceName (IBase *pBase)`
Returns the name of the main interface as string DEPRICATED, use IBase::GetPrincipalInterfaceType() instead.
- `virtual void SetNumEnums (int NumEnums)=0`
sets the number of enum values

Variables

- [interface SPINNAKER_API_ABSTRACT IBase](#)
Base interface common to all nodes.
- `const uint8_t COMMAND_MAGIC = 0x42`
- `const uint32_t U3V_EVENT_PREFIX = 0x45563355`
- `const uint16_t GENCP_EVENT_CMD_ID = 0x0C00`
- `const size_t GENCP_COMMAND_HEADER_SIZE = sizeof(U3V_COMMAND_HEADER)`
- `const size_t GENCP_EVENT_BASIC_SIZE = sizeof(U3V_EVENT_MESSAGE)`
- [interface SPINNAKER_API_ABSTRACT IBoolean](#)
Interface for Boolean properties.
- [interface SPINNAKER_API_ABSTRACT bool Verify = true\) = 0](#)
- [interface SPINNAKER_API_ABSTRACT ICategory](#)
Gives access to a category node.
- [interface SPINNAKER_API_ABSTRACT IChunkPort](#)
Interface for ports attached to a chunk.
- [interface SPINNAKER_API_ABSTRACT ICommand](#)
Interface for command like properties.
- [interface SPINNAKER_API_ABSTRACT IDestroy](#)
Interface to destroy an object.
- [interface SPINNAKER_API_ABSTRACT IDeviceInfo](#)
Interface to get information about the device (= nodemap)
- [interface SPINNAKER_API_ABSTRACT IEnumEntry](#)
Interface of single enum value.
- [interface SPINNAKER_API_ABSTRACT IEnumeration](#)
Interface for enumeration properties.
- `template<typename EnumT >`
[interface SPINNAKER_API_ABSTRACT IEnumerationT](#)
Interface for enumeration properties.
- `template<typename EnumT >`
[interface SPINNAKER_API_ABSTRACT virtual public IEnumReference](#)
Interface to construct an enum reference.
- [interface SPINNAKER_API_ABSTRACT IFloat](#)
Interface for float properties.
- [interface SPINNAKER_API_ABSTRACT IInteger](#)
Interface for integer properties.
- [interface SPINNAKER_API_ABSTRACT INode](#)
Interface common to all nodes.
- [interface SPINNAKER_API_ABSTRACT virtual public IReference](#)
Interface to construct a reference.
- [interface SPINNAKER_API_ABSTRACT INodeMap](#)
Interface to access the node map.
- [interface SPINNAKER_API_ABSTRACT INodeMapDyn](#)
Interface to access the node map.
- [interface SPINNAKER_API_ABSTRACT IPort](#)
Interface for ports.
- [interface SPINNAKER_API_ABSTRACT int64_t Address](#)
- [interface SPINNAKER_API_ABSTRACT int64_t int64_t Length = 0](#)
- [interface SPINNAKER_API IPortConstruct](#)
Interface for ports.
- [interface SPINNAKER_API_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER_API_ABSTRACT IPortReplay](#)

- Interface for replaying write commands on a port.*
- [interface SPINNAKER_API_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER_API_ABSTRACT IPortRecorder](#)
- Interface for recording write commands on a port.*
- [interface SPINNAKER_API_ABSTRACT IRegister](#)
- Interface for registers.*
- [interface SPINNAKER_API_ABSTRACT ISelector](#)
- Interface for groups of features selected by a single one.*
- [interface SPINNAKER_API_ABSTRACT ISelectorDigit](#)
- Interface of a "digit" of the "counter" formed by the selector set.*
- [interface SPINNAKER_API_ABSTRACT IString](#)
- Interface for string properties.*
- [interface SPINNAKER_API_ABSTRACT IValue](#)
- Interface for value properties.*
- [interface SPINNAKER_API_ABSTRACT IPersistScript](#)
- Basic interface to persist values to.*

14.7.1 Typedef Documentation

14.7.1.1 IDevFileStream

```
typedef IDevFileStreamBase<char, std::char_traits<char> > IDevFileStream
```

14.7.1.2 ODevFileStream

```
typedef ODevFileStreamBase<char, std::char_traits<char> > ODevFileStream
```

14.7.2 Enumeration Type Documentation

14.7.2.1 GVCP_MESSAGE_TAGS

```
enum GVCP\_MESSAGE\_TAGS
```

Enumerator

TAG_EVENT_CMD	
TAG_EVENTDATA_CMD	

14.7.3 Function Documentation

14.7.3.1 PersistFeature()

```
virtual void Spinnaker::GenApi::PersistFeature (
    IValue & item ) [pure virtual]
```

Stores a feature.

14.7.3.2 SET_GUID()

```
void SPINNAKER_API Spinnaker::GenApi::SET_GUID (
    SPIN_GUID & name,
    uint32_t l,
    uint16_t w1,
    uint16_t w2,
    uint8_t b1,
    uint8_t b2,
    uint8_t b3,
    uint8_t b4,
    uint8_t b5,
    uint8_t b6,
    uint8_t b7,
    uint8_t b8 )
```

14.7.4 Variable Documentation

14.7.4.1 COMMAND_MAGIC

```
const uint8_t COMMAND_MAGIC = 0x42
```

14.7.4.2 GENCP_COMMAND_HEADER_SIZE

```
const size_t GENCP_COMMAND_HEADER_SIZE = sizeof(U3V_COMMAND_HEADER)
```

14.7.4.3 GENCP_EVENT_BASIC_SIZE

```
const size_t GENCP_EVENT_BASIC_SIZE = sizeof(U3V_EVENT_MESSAGE)
```

14.7.4.4 GENCP_EVENT_CMD_ID

```
const uint16_t GENCP_EVENT_CMD_ID = 0x0C00
```

14.7.4.5 IPersistScript

```
interface SPINNAKER_API_ABSTRACT IPersistScript
```

Initial value:

```
{
    virtual void SetInfo(GenICam::gcstring & Info) = 0
}
```

Basic interface to persist values to.

14.7.4.6 U3V_EVENT_PREFIX

```
const uint32_t U3V_EVENT_PREFIX = 0x45563355
```

14.8 Spinnaker::GenICam Namespace Reference

Classes

- class [AutoLock](#)
- class [CGlobalLock](#)
 - Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)
 - Unlocks the global lock object on destruction.*
- class [CLock](#)
 - A lock class.*
- class [CLockEx](#)
 - This class is for testing purposes only.*
- class [gcstring](#)
- class [LockableObject](#)
 - Instance-Lock for an object.*
- struct [Version_t](#)
 - Version.*

Functions

- [SPINNAKER_API](#) void [ThrowBadAlloc](#) ()
- [std::istream](#) & [getline](#) ([std::istream](#) &is, [Spinnaker::GenICam::gcstring](#) &str)
STL getline.
- [std::istream](#) & [getline](#) ([std::istream](#) &is, [Spinnaker::GenICam::gcstring](#) &str, char delim)
STL getline.
- [template](#)<typename Td , typename Ts >
Td [INTEGRAL_CAST2](#) (Ts s)
This verifies at runtime if there was no loss of data if an type Ts (e.g.
- [template](#)<typename T >
T [INTEGRAL_CAST](#) (int64_t ll)
This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.
- [SPINNAKER_API](#) bool [DoesEnvironmentVariableExist](#) (const [Spinnaker::GenICam::gcstring](#) &VariableName)
Returns true if an environment variable exists.
- [SPINNAKER_API](#) [gcstring](#) [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName)
Retrieve the value of an environment variable.
- [SPINNAKER_API](#) bool [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName, [gcstring](#) &VariableContent)
Retrieve the value of an environment variable.
- [SPINNAKER_API](#) [gcstring](#) [UrlEncode](#) (const [gcstring](#) &Input)
Converts \ to / and replaces all unsafe characters by their xx equivalent.
- [SPINNAKER_API](#) [gcstring](#) [UrlDecode](#) (const [gcstring](#) &Input)
Replaces xx escapes by their char equivalent.
- [SPINNAKER_API](#) void [ReplaceEnvironmentVariables](#) ([gcstring](#) &Buffer, bool ReplaceBlankBy20=false)
Replaces in a string and replace ' ' with %20.
- [SPINNAKER_API](#) [gcstring](#) [GetGenICamCacheFolder](#) (void)
Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling [SetGenICamCacheFolder\(\)](#).
- [SPINNAKER_API](#) [gcstring](#) [GetGenICamLogConfig](#) (void)
Retrieve the path of the GenICam logging properties file.
- [SPINNAKER_API](#) [gcstring](#) [GetGenICamCLProtocolFolder](#) (void)
Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling [SetGenICamCLProtocolFolder\(\)](#).
- [SPINNAKER_API](#) void [SetGenICamCacheFolder](#) (const [gcstring](#) &path)
Stores the path of the GenICam cache folder.
- [SPINNAKER_API](#) void [SetGenICamLogConfig](#) (const [gcstring](#) &path)
Stores the path of the GenICam logging properties file.
- [SPINNAKER_API](#) void [SetGenICamCLProtocolFolder](#) (const [gcstring](#) &path)
Stores the path of the CLProtocol folder.
- [SPINNAKER_API](#) void [Tokenize](#) (const [gcstring](#) &str, [gcstring_vector](#) &tokens, const [gcstring](#) &delimiters=" ")
splits str input string into a list of tokens using the delimiter
- [SPINNAKER_API](#) void [GetFiles](#) (const [gcstring](#) &FileTemplate, [gcstring_vector](#) &FileNames, const bool DirectoriesOnly=false)
Gets a list of files or directories matching a given FileTemplate.
- [SPINNAKER_API](#) [gcstring](#) [GetModulePathFromFunction](#) (void *pFunction)
Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

14.8.1 Function Documentation

14.8.1.1 `getline()` [1/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

STL `getline`.

14.8.1.2 `getline()` [2/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str,
    char delim ) [inline]
```

STL `getline`.

14.8.1.3 `ThrowBadAlloc()`

```
SPINNAKER_API void Spinnaker::GenICam::ThrowBadAlloc ( )
```

14.9 Spinnaker::Video Namespace Reference

Classes

- struct [AVIOption](#)
Options for saving AVI files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [MJPGOption](#)
Options for saving MJPG files.
- class [SpinVideo](#)
Provides the functionality for the user to record images to an AVI/MP4 file.

Chapter 15

Class Documentation

15.1 ActionCommandResult Struct Reference

Action Command Result.

Public Attributes

- unsigned int [DeviceAddress](#)
- [ActionCommandStatus](#) Status

15.1.1 Detailed Description

Action Command Result.

15.1.2 Member Data Documentation

15.1.2.1 DeviceAddress

`unsigned int DeviceAddress`

15.1.2.2 Status

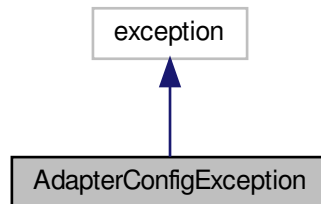
`ActionCommandStatus Status`

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

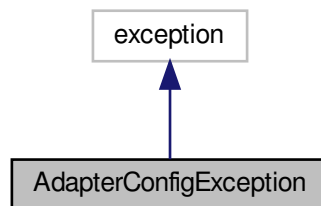
- `include/SpinnakerDefs.h`

15.2 AdapterConfigException Class Reference

Inheritance diagram for AdapterConfigException:



Collaboration diagram for AdapterConfigException:



Public Member Functions

- [AdapterConfigException](#) (const [AdapterConfig::AdapterConfigErr](#) errCode)
- [AdapterConfigException](#) (const [AdapterConfig::AdapterConfigErr](#) errCode, std::string param)
- [AdapterConfig::AdapterConfigErr ErrCode](#) () const
- std::string [GetParamStr](#) () const

15.2.1 Constructor & Destructor Documentation

15.2.1.1 AdapterConfigException() [1/2]

```
AdapterConfigException (
    const AdapterConfig::AdapterConfigErr errCode ) [inline]
```

15.2.1.2 AdapterConfigException() [2/2]

```
AdapterConfigException (
    const AdapterConfig::AdapterConfigErr errCode,
    std::string param ) [inline]
```

15.2.2 Member Function Documentation

15.2.2.1 ErrCode()

```
AdapterConfig::AdapterConfigErr ErrCode ( ) const [inline]
```

15.2.2.2 GetParamStr()

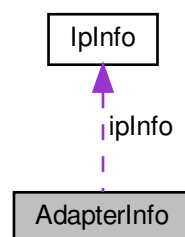
```
std::string GetParamStr ( ) const [inline]
```

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

- include/[AdapterConfig.h](#)

15.3 AdapterInfo Struct Reference

Collaboration diagram for AdapterInfo:



Public Member Functions

- [AdapterInfo](#) ()

Public Attributes

- `std::string` [adapterName](#)
- `std::string` [adapterGUID](#)
- `std::string` [adapterMACAddress](#)
- `std::string` [adapterDescription](#)
- `bool` [dhcpEnabled](#)
- `IpInfo` [ipInfo](#)
- `std::string` [receiveBuffersRegKey](#)
- `std::string` [transmitBuffersRegKey](#)
- `std::string` [jumboPacketsRegKey](#)
- `unsigned int` [transmitBuffers](#)
- `unsigned int` [receiveBuffers](#)
- `unsigned int` [jumboPackets](#)
- `unsigned int` [receiveBuffersMin](#)
- `unsigned int` [receiveBuffersMax](#)
- `unsigned int` [receiveBuffersStep](#)
- `unsigned int` [transmitBuffersMin](#)
- `unsigned int` [transmitBuffersMax](#)
- `unsigned int` [transmitBuffersStep](#)
- `std::vector< unsigned int >` [jumboPacketValidValues](#)

15.3.1 Constructor & Destructor Documentation

15.3.1.1 AdapterInfo()

```
AdapterInfo ( ) [inline]
```

15.3.2 Member Data Documentation

15.3.2.1 adapterDescription

```
std::string adapterDescription
```

15.3.2.2 adapterGUID

```
std::string adapterGUID
```

15.3.2.3 adapterMACAddress

`std::string adapterMACAddress`

15.3.2.4 adapterName

`std::string adapterName`

15.3.2.5 dhcpEnabled

`bool dhcpEnabled`

15.3.2.6 ipInfo

`IpInfo ipInfo`

15.3.2.7 jumboPackets

`unsigned int jumboPackets`

15.3.2.8 jumboPacketsRegKey

`std::string jumboPacketsRegKey`

15.3.2.9 jumboPacketValidValues

`std::vector<unsigned int> jumboPacketValidValues`

15.3.2.10 receiveBuffers

`unsigned int receiveBuffers`

15.3.2.11 receiveBuffersMax

```
unsigned int receiveBuffersMax
```

15.3.2.12 receiveBuffersMin

```
unsigned int receiveBuffersMin
```

15.3.2.13 receiveBuffersRegKey

```
std::string receiveBuffersRegKey
```

15.3.2.14 receiveBuffersStep

```
unsigned int receiveBuffersStep
```

15.3.2.15 transmitBuffers

```
unsigned int transmitBuffers
```

15.3.2.16 transmitBuffersMax

```
unsigned int transmitBuffersMax
```

15.3.2.17 transmitBuffersMin

```
unsigned int transmitBuffersMin
```

15.3.2.18 transmitBuffersRegKey

```
std::string transmitBuffersRegKey
```

15.3.2.19 transmitBuffersStep

```
unsigned int transmitBuffersStep
```

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

- include/[AdapterConfig.h](#)

15.4 AttachStatistics_t Struct Reference

Delivers information about the attached chunks and nodes.

Public Attributes

- int [NumChunkPorts](#)
Number of chunk ports found in the node map.
- int [NumChunks](#)
Number of chunks found in the buffer.
- int [NumAttachedChunks](#)
Number of chunks from the buffer attached to a chunk port.

15.4.1 Detailed Description

Delivers information about the attached chunks and nodes.

15.4.2 Member Data Documentation

15.4.2.1 NumAttachedChunks

```
int NumAttachedChunks
```

Number of chunks from the buffer attached to a chunk port.

15.4.2.2 NumChunkPorts

```
int NumChunkPorts
```

Number of chunk ports found in the node map.

15.4.2.3 NumChunks

`int NumChunks`

Number of chunks found in the buffer.

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

- `include/SpinGenApi/ChunkAdapter.h`

15.5 AutoLock Class Reference

Public Member Functions

- [AutoLock \(CLOCK &lock\)](#)
- [~AutoLock \(\)](#)

15.5.1 Constructor & Destructor Documentation

15.5.1.1 AutoLock()

```
AutoLock (
    CLOCK & lock ) [inline]
```

15.5.1.2 ~AutoLock()

```
~AutoLock ( ) [inline]
```

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

- `include/SpinGenApi/GCSynch.h`

15.6 AutoLock Class Reference

Public Member Functions

- [AutoLock \(CLOCK &lock\)](#)
- [~AutoLock \(\)](#)

15.6.1 Constructor & Destructor Documentation

15.6.1.1 AutoLock()

```
AutoLock (
    CLock & lock ) [inline]
```

15.6.1.2 ~AutoLock()

```
~AutoLock ( ) [inline]
```

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

- include/SpinGenApi/[Synch.h](#)

15.7 AVIOption Struct Reference

Options for saving AVI files.

Public Member Functions

- [AVIOption](#) ()

Public Attributes

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [reserved](#) [256]
Reserved for future use.

15.7.1 Detailed Description

Options for saving AVI files.

15.7.2 Constructor & Destructor Documentation

15.7.2.1 AVIOption()

```
AVIOption ( ) [inline]
```

15.7.3 Member Data Documentation

15.7.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

15.7.3.2 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

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

- [include/SpinVideoDefs.h](#)

15.8 BasePtr< T, B > Class Template Reference

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

Public Member Functions

- [BasePtr](#) (void) throw ()
Default constructor.
- virtual [~BasePtr](#) (void)
- [BasePtr](#) (const [BasePtr](#) &other) throw ()
Copy constructor.
- virtual [BasePtr](#) & [operator=](#) (const [BasePtr](#) &rhs)
Assign INode Pointer.
- virtual [BasePtr](#) & [operator=](#) (const int nMustBeNull)
- virtual [BasePtr](#) & [operator=](#) (const long nMustBeNull)
- virtual [BasePtr](#) & [operator=](#) (const std::nullptr_t nullptr)
- virtual [operator T*](#) (void) const
Dereferencing.
- virtual T * [operator->](#) (void) const
Dereferencing.

- virtual bool [IsValid](#) () const throw ()
True if the pointer is valid.
- virtual [operator bool](#) (void) const throw ()
True if the pointer is valid.
- virtual bool [operator==](#) (const [BasePtr](#) &rT) const
Pointer equal.
- virtual bool [operator==](#) (std::nullptr_t) const
Pointer equal.
- virtual bool [operator==](#) (int nMustBeNull) const
Pointer equal.
- virtual bool [operator==](#) (long nMustBeNull) const
Pointer equal.
- virtual T * [get](#) () const
[get\(\)](#)

Protected Attributes

- PointerData * [m_pT](#)
Underlying raw pointer.

15.8.1 Detailed Description

```
template<class T, class B = T>
class Spinnaker::BasePtr< T, B >
```

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

15.8.2 Constructor & Destructor Documentation

15.8.2.1 BasePtr() [1/2]

```
BasePtr (
    void ) throw )
```

Default constructor.

15.8.2.2 ~BasePtr()

```
virtual ~BasePtr (
    void ) [virtual]
```

15.8.2.3 BasePtr() [2/2]

```
BasePtr (
    const BasePtr< T, B > & other ) throw )
```

Copy constructor.

15.8.3 Member Function Documentation

15.8.3.1 get()

```
virtual T* get ( ) const [virtual]
```

[get\(\)](#)

15.8.3.2 IsValid()

```
virtual bool IsValid ( ) const throw ) [virtual]
```

True if the pointer is valid.

15.8.3.3 operator bool()

```
virtual operator bool (
    void ) const throw ) [virtual]
```

True if the pointer is valid.

15.8.3.4 operator T*()

```
virtual operator T* (
    void ) const [virtual]
```

Dereferencing.

15.8.3.5 operator->()

```
virtual T* operator-> (
    void ) const [virtual]
```

Dereferencing.

15.8.3.6 operator=() [1/4]

```
virtual BasePtr& operator= (
    const BasePtr< T, B > & rhs ) [virtual]
```

Assign INode Pointer.

15.8.3.7 operator=() [2/4]

```
virtual BasePtr& operator= (
    const int nMustBeNull ) [virtual]
```

15.8.3.8 operator=() [3/4]

```
virtual BasePtr& operator= (
    const long nMustBeNull ) [virtual]
```

15.8.3.9 operator=() [4/4]

```
virtual BasePtr& operator= (
    const std::nullptr_t nullptr ) [virtual]
```

15.8.3.10 operator==() [1/4]

```
virtual bool operator== (
    const BasePtr< T, B > & rT ) const [virtual]
```

Pointer equal.

15.8.3.11 operator==() [2/4]

```
virtual bool operator==(
    std::nullptr_t ) const [virtual]
```

Pointer equal.

15.8.3.12 operator==() [3/4]

```
virtual bool operator==(
    int nMustBeNull ) const [virtual]
```

Pointer equal.

15.8.3.13 operator==() [4/4]

```
virtual bool operator==(
    long nMustBeNull ) const [virtual]
```

Pointer equal.

15.8.4 Member Data Documentation

15.8.4.1 m_pT

```
PointerData* m_pT [protected]
```

Underlying raw pointer.

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

- include/[BasePtr.h](#)

15.9 BMPOption Struct Reference

Options for saving Bitmap image.

Public Member Functions

- [BMPOption](#) ()

Public Attributes

- bool [indexedColor_8bit](#)
- unsigned int [reserved](#) [16]

Reserved for future use.

15.9.1 Detailed Description

Options for saving Bitmap image.

15.9.2 Constructor & Destructor Documentation

15.9.2.1 BMPOption()

```
BMPOption ( ) [inline]
```

15.9.3 Member Data Documentation

15.9.3.1 indexedColor_8bit

```
bool indexedColor_8bit
```

15.9.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

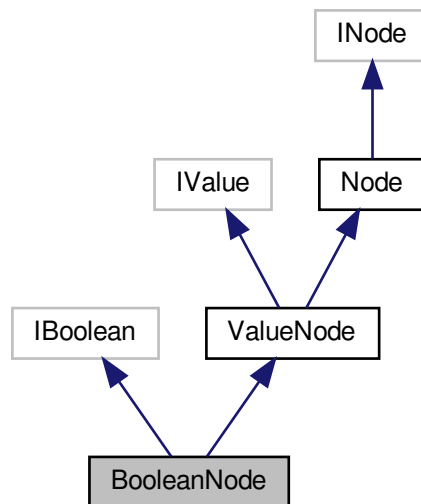
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

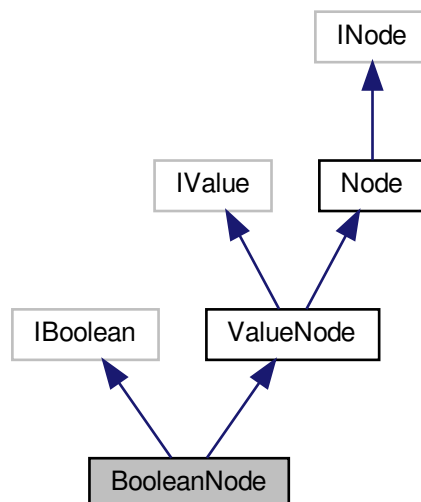
15.10 BooleanNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for BooleanNode:



Collaboration diagram for BooleanNode:



Public Member Functions

- [BooleanNode](#) ()
- [BooleanNode](#) (std::shared_ptr< Node::NodeImpl > pBoolean)
- virtual [~BooleanNode](#) ()
- void [SetValue](#) (bool Value, bool [Verify](#)=true)
Set node value.
- virtual void [operator=](#) (bool Value)
Set node value.
- bool [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false) const
Get node value.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.10.1 Detailed Description

[Interface](#) for string properties.

15.10.2 Constructor & Destructor Documentation

15.10.2.1 BooleanNode() [1/2]

```
BooleanNode ( )
```

15.10.2.2 BooleanNode() [2/2]

```
BooleanNode (
    std::shared_ptr< Node::NodeImpl > pBoolean )
```

15.10.2.3 ~BooleanNode()

```
virtual ~BooleanNode ( ) [virtual]
```

15.10.3 Member Function Documentation

15.10.3.1 GetValue()

```
bool GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) const
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked.
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false).

Returns

The value read.

15.10.3.2 operator=()

```
virtual void operator= (
    bool Value ) [virtual]
```

Set node value.

15.10.3.3 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

15.10.3.4 SetValue()

```
void SetValue (
    bool Value,
    bool Verify = true )
```

Set node value.

Parameters

<i>Value</i>	The value to set.
<i>Verify</i>	Enables AccessMode and Range verification (default = true).

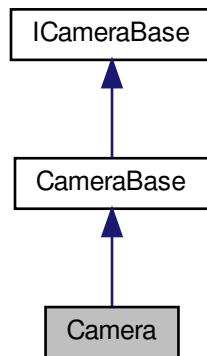
The documentation for this class was generated from the following file:

- include/SpinGenApi/[BooleanNode.h](#)

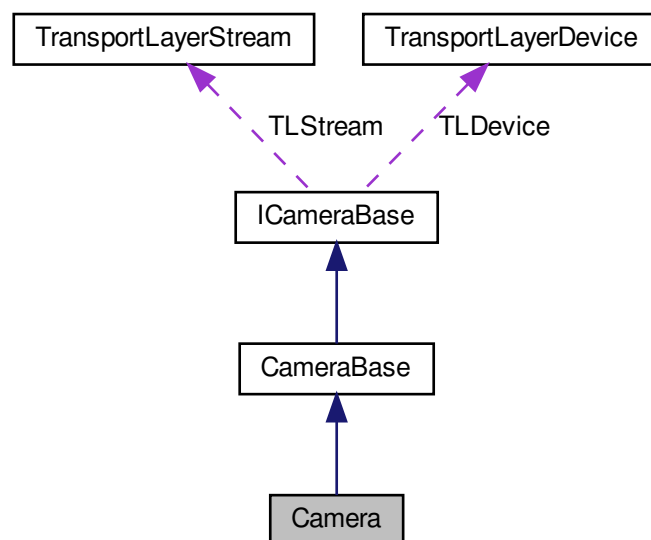
15.11 Camera Class Reference

The camera object class.

Inheritance diagram for Camera:



Collaboration diagram for Camera:



Public Member Functions

- `~Camera()`
- `void Init()`

Public Attributes

- [GenApi::Integer](#) & [LUTIndex](#)
Description:
Control the index (offset) of the coefficient to access in the selected LUT.
- [GenApi::Boolean](#) & [LUTEnable](#)
Description:
Activates the selected LUT.
- [GenApi::Integer](#) & [LUTValue](#)
Description:
Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.
- [GenApi::EnumerationT< LUTSelectorEnums >](#) & [LUTSelector](#)
Description:
Selects which LUT to control.
- [GenApi::Float](#) & [ExposureTime](#)
Description:
Exposure time in microseconds when Exposure Mode is Timed.
- [GenApi::Command](#) & [AcquisitionStop](#)
Description: *This command stops the acquisition of images.*
- [GenApi::Float](#) & [AcquisitionResultingFrameRate](#)
Description: *Resulting frame rate in Hertz.*
- [GenApi::Float](#) & [AcquisitionLineRate](#)
Description: *Controls the rate (in Hertz) at which the Lines in a Frame are captured.*
- [GenApi::Command](#) & [AcquisitionStart](#)
Description: *This command starts the acquisition of images.*
- [GenApi::Command](#) & [TriggerSoftware](#)
Description:
Generates an internal trigger if Trigger Source is set to Software.
- [GenApi::EnumerationT< ExposureModeEnums >](#) & [ExposureMode](#)
Description:
Sets the operation mode of the Exposure.
- [GenApi::EnumerationT< AcquisitionModeEnums >](#) & [AcquisitionMode](#)
Description: *Sets the acquisition mode of the device.*
- [GenApi::Integer](#) & [AcquisitionFrameCount](#)
Description:
Number of images to acquire during a multi frame acquisition.
- [GenApi::EnumerationT< TriggerSourceEnums >](#) & [TriggerSource](#)
Description:
Specifies the internal signal or physical input line to use as the trigger source.
- [GenApi::EnumerationT< TriggerActivationEnums >](#) & [TriggerActivation](#)
Description: *Specifies the activation mode of the trigger.*
- [GenApi::EnumerationT< SensorShutterModeEnums >](#) & [SensorShutterMode](#)
Description: *Sets the shutter mode of the device.*
- [GenApi::Float](#) & [TriggerDelay](#)
Description:
Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.
- [GenApi::EnumerationT< TriggerModeEnums >](#) & [TriggerMode](#)
Description:
Controls whether or not trigger is active.
- [GenApi::Float](#) & [AcquisitionFrameRate](#)
Description: *User controlled acquisition frame rate in Hertz Visibility:*
- [GenApi::EnumerationT< TriggerOverlapEnums >](#) & [TriggerOverlap](#)
Description: *Specifies the overlap mode of the trigger.*
- [GenApi::EnumerationT< TriggerSelectorEnums >](#) & [TriggerSelector](#)

- Description: Selects the type of trigger to configure.*

 - [GenApi::IBoolean](#) & [AcquisitionFrameRateEnable](#)

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.
- [GenApi::IEnumerationT< ExposureAutoEnums >](#) & [ExposureAuto](#)

Description: Sets the automatic exposure mode Visibility:
- [GenApi::Integer](#) & [AcquisitionBurstFrameCount](#)

Description:
This feature is used only if the FrameBurstStart trigger is enabled and the FrameBurstEnd trigger is disabled.
- [GenApi::Integer](#) & [EventTest](#)

Description: Returns the unique identifier of the Test type of Event.
- [GenApi::Integer](#) & [EventTestTimestamp](#)

Description: Returns the Timestamp of the Test Event.
- [GenApi::Integer](#) & [EventExposureEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End Event.
- [GenApi::Integer](#) & [EventExposureEnd](#)

Description: Returns the unique identifier of the Exposure End type of Event.
- [GenApi::Integer](#) & [EventExposureEndTimestamp](#)

Description: Returns the Timestamp of the Exposure End Event.
- [GenApi::Integer](#) & [EventError](#)

Description: Returns the unique identifier of the Error type of Event.
- [GenApi::Integer](#) & [EventErrorTimestamp](#)

Description: Returns the Timestamp of the Error Event.
- [GenApi::Integer](#) & [EventErrorCode](#)

Description: Returns the error code for the error that happened Visibility:
- [GenApi::Integer](#) & [EventErrorFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Error Event.
- [GenApi::IEnumerationT< EventSelectorEnums >](#) & [EventSelector](#)

Description: Selects which Event to enable or disable.
- [GenApi::IBoolean](#) & [EventSerialReceiveOverflow](#)

Description: Returns the status of the event serial receive overflow.
- [GenApi::Integer](#) & [EventSerialPortReceive](#)

Description: Returns the unique identifier of the Serial Port Receive type of Event.
- [GenApi::Integer](#) & [EventSerialPortReceiveTimestamp](#)

Description: Returns the Timestamp of the Serial Port Receive Event.
- [GenApi::IString](#) & [EventSerialData](#)

Description: Returns the serial data that was received.
- [GenApi::Integer](#) & [EventSerialDataLength](#)

Description: Returns the length of the received serial data that was included in the event payload.
- [GenApi::IEnumerationT< EventNotificationEnums >](#) & [EventNotification](#)

Description: Enables/Disables the selected event.
- [GenApi::Integer](#) & [LogicBlockLUTRowIndex](#)

Description: Controls the row of the truth table to access in the selected LUT.
- [GenApi::IEnumerationT< LogicBlockSelectorEnums >](#) & [LogicBlockSelector](#)

Description: Selects which LogicBlock to configure Visibility:
- [GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums >](#) & [LogicBlockLUTInputActivation](#)

Description: Selects the activation mode of the Logic Input Source signal.
- [GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums >](#) & [LogicBlockLUTInputSelector](#)

Description: Controls which LogicBlockLUT Input Source & Activation to access.
- [GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums >](#) & [LogicBlockLUTInputSource](#)

Description: Selects the source for the input into the Logic LUT.
- [GenApi::IBoolean](#) & [LogicBlockLUTOutputValue](#)

- Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.*

 - [GenApi::Integer](#) & [LogicBlockLUTOutputValueAll](#)

Description: Sets the value of all the output bits in the selected LUT.
- [GenApi::EnumerationT](#) < [LogicBlockLUTSelectorEnums](#) > & [LogicBlockLUTSelector](#)

Description: Selects which LogicBlock LUT to configure Visibility:
- [GenApi::IFloat](#) & [ColorTransformationValue](#)

Description:
Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.
- [GenApi::IBoolean](#) & [ColorTransformationEnable](#)

Description:
Enables/disables the color transform selected with ColorTransformationSelector.
- [GenApi::EnumerationT](#) < [ColorTransformationSelectorEnums](#) > & [ColorTransformationSelector](#)

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.
- [GenApi::EnumerationT](#) < [RgbTransformLightSourceEnums](#) > & [RgbTransformLightSource](#)

Description:
Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources.
- [GenApi::IFloat](#) & [Saturation](#)

Description: Controls the color saturation.
- [GenApi::IBoolean](#) & [SaturationEnable](#)

Description: Enables/disables Saturation adjustment.
- [GenApi::EnumerationT](#) < [ColorTransformationValueSelectorEnums](#) > & [ColorTransformationValueSelector](#)

Description:
Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module
- [GenApi::Integer](#) & [TimestampLatchValue](#)

Description: Returns the latched value of the timestamp counter.
- [GenApi::Command](#) & [TimestampReset](#)

Description: Resets the current value of the device timestamp counter.
- [GenApi::IString](#) & [DeviceUserID](#)

Description: User-programmable device identifier.
- [GenApi::IFloat](#) & [DeviceTemperature](#)

Description: Device temperature in degrees Celsius (C).
- [GenApi::Integer](#) & [MaxDeviceResetTime](#)

Description: Time to wait until device reset complete (ms).
- [GenApi::Integer](#) & [DeviceTLVersionMinor](#)

Description:
Minor version of the Transport Layer of the device.
- [GenApi::IString](#) & [DeviceSerialNumber](#)

Description:
Device's serial number.
- [GenApi::IString](#) & [DeviceVendorName](#)

Description: Name of the manufacturer of the device.
- [GenApi::EnumerationT](#) < [DeviceRegistersEndiannessEnums](#) > & [DeviceRegistersEndianness](#)

Description: Endianness of the registers of the device.
- [GenApi::IString](#) & [DeviceManufacturerInfo](#)

Description: Manufacturer information about the device.
- [GenApi::Integer](#) & [DeviceLinkSpeed](#)

Description:
Indicates the speed of transmission negotiated on the specified Link.
- [GenApi::Integer](#) & [LinkUptime](#)

Description: Time since the last phy negotiation (enumeration).
- [GenApi::Integer](#) & [DeviceEventChannelCount](#)

Description:
Indicates the number of event channels supported by the device.

- [GenApi::ICommand](#) & [TimestampLatch](#)
Description: Latches the current timestamp counter into TimestampLatchValue.
- [GenApi::IEnumerationT< DeviceScanTypeEnums >](#) & [DeviceScanType](#)
Description: Scan type of the sensor of the device.
- [GenApi::ICommand](#) & [DeviceReset](#)
Description: This is a command that immediately resets and reboots the device.
- [GenApi::IEnumerationT< DeviceCharacterSetEnums >](#) & [DeviceCharacterSet](#)
Description:
Character set used by the strings of the device's bootstrap registers.
- [GenApi::Integer](#) & [DeviceLinkThroughputLimit](#)
Description:
Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.
- [GenApi::IString](#) & [DeviceFirmwareVersion](#)
Description: Version of the firmware on the device.
- [GenApi::Integer](#) & [DeviceStreamChannelCount](#)
Description:
Indicates the number of streaming channels supported by the device.
- [GenApi::IEnumerationT< DeviceTLTypeEnums >](#) & [DeviceTLType](#)
Description: Transport Layer type of the device.
- [GenApi::IString](#) & [DeviceVersion](#)
Description: Version of the device.
- [GenApi::IEnumerationT< DevicePowerSupplySelectorEnums >](#) & [DevicePowerSupplySelector](#)
Description:
Selects the power supply source to control or read.
- [GenApi::IString](#) & [SensorDescription](#)
Description: Returns Sensor Description Visibility:
- [GenApi::IString](#) & [DeviceModelName](#)
Description: Model of the device.
- [GenApi::Integer](#) & [DeviceTLVersionMajor](#)
Description:
Major version of the Transport Layer of the device.
- [GenApi::IEnumerationT< DeviceTemperatureSelectorEnums >](#) & [DeviceTemperatureSelector](#)
Description:
Selects the location within the device, where the temperature will be measured.
- [GenApi::Integer](#) & [EnumerationCount](#)
Description: Number of enumerations since uptime.
- [GenApi::IFloat](#) & [PowerSupplyCurrent](#)
Description:
Indicates the output current of the selected power supply (A).
- [GenApi::IString](#) & [DeviceID](#)
Description: Device identifier (serial number).
- [GenApi::Integer](#) & [DeviceUptime](#)
Description: Total time since the device was powered up in seconds.
- [GenApi::Integer](#) & [DeviceLinkCurrentThroughput](#)
Description: Current bandwidth of streamed data.
- [GenApi::Integer](#) & [DeviceMaxThroughput](#)
Description:
Maximum bandwidth of the data that can be streamed out of the device.
- [GenApi::ICommand](#) & [FactoryReset](#)
Description: Returns all user tables to factory default Visibility:
- [GenApi::IFloat](#) & [PowerSupplyVoltage](#)
Description:
Indicates the current voltage of the selected power supply (V).

- [GenApi::IEnumerationT < DeviceIndicatorModeEnums > & DeviceIndicatorMode](#)
Description: Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).
- [GenApi::IFloat & DeviceLinkBandwidthReserve](#)
*Description:
Percentage of streamed data bandwidth reserved for packet resend.*
- [GenApi::Integer & AasRoiOffsetY](#)
*Description:
Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::Integer & AasRoiOffsetX](#)
*Description:
Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::IEnumerationT < AutoExposureControlPriorityEnums > & AutoExposureControlPriority](#)
*Description:
Selects whether to adjust gain or exposure first.*
- [GenApi::IFloat & BalanceWhiteAutoLowerLimit](#)
*Description:
Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.*
- [GenApi::IFloat & BalanceWhiteAutoDamping](#)
*Description:
Controls how quickly 'BalanceWhiteAuto' adjusts the values for Red and Blue BalanceRatio in response to changing conditions.*
- [GenApi::Integer & AasRoiHeight](#)
*Description:
Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::IFloat & AutoExposureGreyValueUpperLimit](#)
*Description:
The highest value in percentage that the target mean may reach.*
- [GenApi::IFloat & AutoExposureTargetGreyValue](#)
*Description:
This is the user-specified target grey level (image mean) to apply to the current image.*
- [GenApi::IFloat & AutoExposureGainLowerLimit](#)
*Description:
The smallest gain that auto exposure can set.*
- [GenApi::IFloat & AutoExposureGreyValueLowerLimit](#)
*Description:
The lowest value in percentage that the target mean may reach.*
- [GenApi::IEnumerationT < AutoExposureMeteringModeEnums > & AutoExposureMeteringMode](#)
*Description:
Selects a metering mode: average, spot, or partial metering.*
- [GenApi::IFloat & AutoExposureExposureTimeUpperLimit](#)
*Description:
The largest exposure time that auto exposure can set.*
- [GenApi::IFloat & AutoExposureGainUpperLimit](#)
*Description:
The largest gain that auto exposure can set.*
- [GenApi::IFloat & AutoExposureControlLoopDamping](#)
*Description:
It controls how fast the exposure and gain get settled.*
- [GenApi::IFloat & AutoExposureEVCompensation](#)
*Description:
The EV compensation value used in the exposure compensation.*
- [GenApi::IFloat & AutoExposureExposureTimeLowerLimit](#)

Description:

The smallest exposure time that auto exposure can set.

- [GenApi::IEnumerationT](#) < [BalanceWhiteAutoProfileEnums](#) > & [BalanceWhiteAutoProfile](#)

Description: *Selects the profile used by BalanceWhiteAuto.*

- [GenApi::IEnumerationT](#) < [AutoAlgorithmSelectorEnums](#) > & [AutoAlgorithmSelector](#)

Description:

Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

- [GenApi::IEnumerationT](#) < [AutoExposureTargetGreyValueAutoEnums](#) > & [AutoExposureTargetGreyValue↔Auto](#)

Description:

This indicates whether the target image grey level is automatically set by the camera or manually set by the user.

- [GenApi::IBoolean](#) & [AasRoiEnable](#)

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithm↔Selector feature.

- [GenApi::IEnumerationT](#) < [AutoExposureLightingModeEnums](#) > & [AutoExposureLightingMode](#)

Description:

Selects a lighting mode: Backlight, Frontlight or Normal (default).

- [GenApi::IInteger](#) & [AasRoiWidth](#)

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

- [GenApi::IFloat](#) & [BalanceWhiteAutoUpperLimit](#)

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

- [GenApi::IInteger](#) & [LinkErrorCount](#)

Description: *Counts the number of error on the link.*

- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationDHCP](#)

Description: *Controls whether the DHCP IP configuration scheme is activated on the given logical link.*

- [GenApi::IInteger](#) & [GevInterfaceSelector](#)

Description: *Selects which logical link to control.*

- [GenApi::IInteger](#) & [GevSCPD](#)

Description: *Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.*

- [GenApi::IInteger](#) & [GevTimestampTickFrequency](#)

Description: *Indicates the number of timestamp ticks in 1 second (frequency in Hz).*

- [GenApi::IInteger](#) & [GevSCPSPacketSize](#)

Description: *Specifies the stream packet size (in bytes) to send on this channel.*

- [GenApi::IInteger](#) & [GevCurrentDefaultGateway](#)

Description: *Reports the default gateway IP address to be used on the given logical link.*

- [GenApi::IBoolean](#) & [GevSCCFGUnconditionalStreaming](#)

Description: *Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).*

- [GenApi::IInteger](#) & [GevMCTT](#)

Description: *Indicates the transmission timeout of the message channel.*

- [GenApi::IBoolean](#) & [GevSCPSPDoNotFragment](#)

Description: *The state of this feature is copied into the "do not fragment" bit of the IP header of each stream packet.*

- [GenApi::IInteger](#) & [GevCurrentSubnetMask](#)

Description: *Reports the subnet mask of the given logical link.*

- [GenApi::IInteger](#) & [GevStreamChannelSelector](#)

Description: *Selects the stream channel to control.*

- [GenApi::IInteger](#) & [GevCurrentIPAddress](#)

Description: *Reports the IP address for the given logical link.*

- [GenApi::IInteger](#) & [GevMCSP](#)

- Description: Indicates the source port of the message channel.*

 - [GenApi::Integer](#) & [GevGVCPPendingTimeout](#)

Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.
 - [GenApi::EnumerationT<GevIEEE1588StatusEnums>](#) & [GevIEEE1588Status](#)

Description: Provides the status of the IEEE 1588 clock.
 - [GenApi::IString](#) & [GevFirstURL](#)

Description: The first choice of URL for the XML device description file.
 - [GenApi::Integer](#) & [GevMACAddress](#)

Description: MAC address of the logical link.
 - [GenApi::Integer](#) & [GevPersistentSubnetMask](#)

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.
 - [GenApi::Integer](#) & [GevMCPHostPort](#)

Description: The port to which the device must send messages Visibility:
 - [GenApi::Integer](#) & [GevSCPHostPort](#)

Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.
 - [GenApi::IBoolean](#) & [GevGVCPPendingAck](#)

Description: Enables the generation of PENDING_ACK.
 - [GenApi::Integer](#) & [GevSCPIInterfaceIndex](#)

Description: Index of the logical link to use.
 - [GenApi::IBoolean](#) & [GevSupportedOption](#)

Description: Returns if the selected GEV option is supported.
 - [GenApi::EnumerationT<GevIEEE1588ModeEnums>](#) & [GevIEEE1588Mode](#)

Description: Provides the mode of the IEEE 1588 clock.
 - [GenApi::IBoolean](#) & [GevCurrentIPConfigurationLLA](#)

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.
 - [GenApi::Integer](#) & [GevSCSP](#)

Description: Indicates the source port of the stream channel.
 - [GenApi::IBoolean](#) & [GevIEEE1588](#)

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.
 - [GenApi::IBoolean](#) & [GevSCCFGExtendedChunkData](#)

Description: Enables cameras to use the extended chunk data payload type for this stream channel.
 - [GenApi::Integer](#) & [GevPersistentIPAddress](#)

Description: Controls the Persistent IP address for this logical link.
 - [GenApi::IBoolean](#) & [GevCurrentIPConfigurationPersistentIP](#)

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.
 - [GenApi::EnumerationT<GevIEEE1588ClockAccuracyEnums>](#) & [GevIEEE1588ClockAccuracy](#)

Description: Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.
 - [GenApi::Integer](#) & [GevHeartbeatTimeout](#)

Description: Indicates the current heartbeat timeout in milliseconds.
 - [GenApi::Integer](#) & [GevPersistentDefaultGateway](#)

Description: Controls the persistent default gateway for this logical link.
 - [GenApi::EnumerationT<GevCCPEnums>](#) & [GevCCP](#)

Description: Controls the device access privilege of an application.
 - [GenApi::Integer](#) & [GevMCDA](#)

Description: Controls the destination IP address of the message channel Visibility:
 - [GenApi::Integer](#) & [GevSCDA](#)

Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

- [GenApi::Integer](#) & [GevSCPDirection](#)
Description: Transmit or Receive of the channel Visibility:
- [GenApi::Boolean](#) & [GevSCPSFireTestPacket](#)
Description: Sends a test packet.
- [GenApi::String](#) & [GevSecondURL](#)
Description: The second choice of URL to the XML device description file.
- [GenApi::EnumerationT<GevSupportedOptionSelectorEnums>](#) & [GevSupportedOptionSelector](#)
Description: Selects the GEV option to interrogate for existing support.
- [GenApi::Boolean](#) & [GevGVCPHeartbeatDisable](#)
Description: Disables the GVCP heartbeat.
- [GenApi::Integer](#) & [GevMCRC](#)
Description: Indicates the number of retries of the message channel.
- [GenApi::Boolean](#) & [GevSCPSBigEndian](#)
Description: Endianess of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [GevNumberOfInterfaces](#)
Description: Indicates the number of physical network interfaces supported by this device.
- [GenApi::Integer](#) & [TLParamsLocked](#)
Description: Visibility:
- [GenApi::Integer](#) & [PayloadSize](#)
Description: Provides the number of bytes transferred for each image or chunk on the stream channel.
- [GenApi::Integer](#) & [PacketResendRequestCount](#)
Description: Counts the number of resend requests received from the host.
- [GenApi::Boolean](#) & [SharpeningEnable](#)
Description:
Enables/disables the sharpening feature.
- [GenApi::EnumerationT<BlackLevelSelectorEnums>](#) & [BlackLevelSelector](#)
Description:
Selects which black level to control.
- [GenApi::Boolean](#) & [GammaEnable](#)
Description: Enables/disables gamma correction.
- [GenApi::Boolean](#) & [SharpeningAuto](#)
Description:
Enables/disables the auto sharpening feature.
- [GenApi::Boolean](#) & [BlackLevelClampingEnable](#)
Description:
Enable the black level auto clamping feature which performing dark current compensation.
- [GenApi::Float](#) & [BalanceRatio](#)
Description:
Controls the balance ratio of the selected color relative to green.
- [GenApi::EnumerationT<BalanceWhiteAutoEnums>](#) & [BalanceWhiteAuto](#)
Description:
White Balance compensates for color shifts caused by different lighting conditions.
- [GenApi::Float](#) & [SharpeningThreshold](#)
Description:
Controls the minimum intensity gradient change to invoke sharpening.
- [GenApi::EnumerationT<GainAutoEnums>](#) & [GainAuto](#)
Description:
Sets the automatic gain mode.
- [GenApi::Float](#) & [Sharpening](#)
Description:
Controls the amount to sharpen a signal.
- [GenApi::Float](#) & [Gain](#)
Description:
Controls the amplification of the video signal in dB.

- [GenApi::IEnumerationT< BalanceRatioSelectorEnums > & BalanceRatioSelector](#)
 Description:
Selects a balance ratio to configure once a balance ratio control has been selected.
- [GenApi::IEnumerationT< GainSelectorEnums > & GainSelector](#)
 Description: *Selects which gain to control.*
- [GenApi::IFloat & BlackLevel](#)
 Description:
Controls the offset of the video signal in percent.
- [GenApi::Integer & BlackLevelRaw](#)
 Description:
Controls the offset of the video signal in camera specific units.
- [GenApi::IFloat & Gamma](#)
 Description: *Controls the gamma correction of pixel intensity.*
- [GenApi::Integer & DefectTableIndex](#)
 Description:
Controls the offset of the element to access in the defective pixel location table.
- [GenApi::ICommand & DefectTableFactoryRestore](#)
 Description: *Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.*
- [GenApi::Integer & DefectTableCoordinateY](#)
 Description:
Returns the Y coordinate of the defective pixel at DefectTableIndex within the defective pixel table.
- [GenApi::ICommand & DefectTableSave](#)
 Description: *Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.*
- [GenApi::IEnumerationT< DefectCorrectionModeEnums > & DefectCorrectionMode](#)
 Description: *Controls the method used for replacing defective pixels.*
- [GenApi::Integer & DefectTableCoordinateX](#)
 Description:
Returns the X coordinate of the defective pixel at DefectTableIndex within the defective pixel table.
- [GenApi::Integer & DefectTablePixelCount](#)
 Description:
The number of defective pixel locations in the current table.
- [GenApi::IBoolean & DefectCorrectStaticEnable](#)
 Description: *Enables/Disables table-based defective pixel correction.*
- [GenApi::ICommand & DefectTableApply](#)
 Description: *Applies the current defect table, so that any changes made affect images captured by the camera.*
- [GenApi::IBoolean & UserSetFeatureEnable](#)
 Description: *Whether or not the selected feature is saved to user sets.*
- [GenApi::ICommand & UserSetSave](#)
 Description:
Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.
- [GenApi::IEnumerationT< UserSetSelectorEnums > & UserSetSelector](#)
 Description:
Selects the feature User Set to load, save or configure.
- [GenApi::ICommand & UserSetLoad](#)
 Description:
Loads the User Set specified by UserSetSelector to the device and makes it active.
- [GenApi::IEnumerationT< UserSetDefaultEnums > & UserSetDefault](#)
 Description:
Selects the feature User Set to load and make active by default when the device is restarted.
- [GenApi::IEnumerationT< SerialPortBaudRateEnums > & SerialPortBaudRate](#)
 Description: *This feature controls the baud rate used by the selected serial port.*
- [GenApi::Integer & SerialPortDataBits](#)
 Description: *This feature controls the number of data bits used by the selected serial port.*

- [GenApi::IEnumerationT< SerialPortParityEnums > & SerialPortParity](#)
Description: This feature controls the parity used by the selected serial port.
- [GenApi::Integer & SerialTransmitQueueMaxCharacterCount](#)
Description: >Returns the maximum number of characters in the serial port transmit queue.
- [GenApi::Integer & SerialReceiveQueueCurrentCharacterCount](#)
Description: Returns the number of characters currently in the serial port receive queue.
- [GenApi::IEnumerationT< SerialPortSelectorEnums > & SerialPortSelector](#)
Description: Selects which serial port of the device to control.
- [GenApi::IEnumerationT< SerialPortStopBitsEnums > & SerialPortStopBits](#)
Description: This feature controls the number of stop bits used by the selected serial port.
- [GenApi::ICommand & SerialReceiveQueueClear](#)
Description: This is a command that clears the device serial port receive queue.
- [GenApi::Integer & SerialReceiveFramingErrorCount](#)
Description: Returns the number of framing errors that have occurred on the serial port.
- [GenApi::Integer & SerialTransmitQueueCurrentCharacterCount](#)
Description: Returns the number of characters currently in the serial port transmit queue.
- [GenApi::Integer & SerialReceiveParityErrorCount](#)
Description: Returns the number of parity errors that have occurred on the serial port.
- [GenApi::IEnumerationT< SerialPortSourceEnums > & SerialPortSource](#)
Description: Specifies the physical input Line on which to receive serial data.
- [GenApi::Integer & SerialReceiveQueueMaxCharacterCount](#)
Description: >Returns the maximum number of characters in the serial port receive queue.
- [GenApi::Integer & SequencerSetStart](#)
Description: Sets the first sequencer set to be used.
- [GenApi::IEnumerationT< SequencerModeEnums > & SequencerMode](#)
Description: Controls whether or not a sequencer is active.
- [GenApi::IEnumerationT< SequencerConfigurationValidEnums > & SequencerConfigurationValid](#)
Description:
Display whether the current sequencer configuration is valid to run.
- [GenApi::IEnumerationT< SequencerSetValidEnums > & SequencerSetValid](#)
Description:
Displays whether the currently selected sequencer set's register contents are valid to use.
- [GenApi::Integer & SequencerSetSelector](#)
Description:
Selects the sequencer set to which subsequent settings apply.
- [GenApi::IEnumerationT< SequencerTriggerActivationEnums > & SequencerTriggerActivation](#)
Description:
Specifies the activation mode of the sequencer trigger.
- [GenApi::IEnumerationT< SequencerConfigurationModeEnums > & SequencerConfigurationMode](#)
Description:
Controls whether or not a sequencer is in configuration mode.
- [GenApi::ICommand & SequencerSetSave](#)
Description:
Saves the current device configuration to the currently selected sequencer set.
- [GenApi::IEnumerationT< SequencerTriggerSourceEnums > & SequencerTriggerSource](#)
Description:
Specifies the internal signal or physical input line to use as the sequencer trigger source.
- [GenApi::Integer & SequencerSetActive](#)
Description: Displays the currently active sequencer set.
- [GenApi::Integer & SequencerSetNext](#)
Description: Specifies the next sequencer set.
- [GenApi::ICommand & SequencerSetLoad](#)

Description:

Loads currently selected sequencer to the current device configuration.

- [GenApi::Integer](#) & [SequencerPathSelector](#)

Description:

Selects branching path to be used for subsequent settings.

- [GenApi::Boolean](#) & [SequencerFeatureEnable](#)

Description:

Enables the selected feature and makes it active in all sequencer sets.

- [GenApi::Integer](#) & [TransferBlockCount](#)

Description: Specifies the number of data blocks (images) that the device should stream before stopping.

- [GenApi::Command](#) & [TransferStart](#)

Description: Starts the streaming of data blocks (images) out of the device.

- [GenApi::Integer](#) & [TransferQueueMaxBlockCount](#)

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

- [GenApi::Integer](#) & [TransferQueueCurrentBlockCount](#)

Description: Returns number of data blocks (images) currently in the transfer queue.

- [GenApi::EnumerationT< TransferQueueModeEnums >](#) & [TransferQueueMode](#)

Description: Specifies the operation mode of the transfer queue.

- [GenApi::EnumerationT< TransferOperationModeEnums >](#) & [TransferOperationMode](#)

Description: Selects the operation mode of the transfer.

- [GenApi::Command](#) & [TransferStop](#)

Description: Stops the streaming of data block (images).

- [GenApi::Integer](#) & [TransferQueueOverflowCount](#)

Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

- [GenApi::EnumerationT< TransferControlModeEnums >](#) & [TransferControlMode](#)

Description: Selects the control method for the transfers.

- [GenApi::Float](#) & [ChunkBlackLevel](#)

Description: Returns the black level used to capture the image.

- [GenApi::Integer](#) & [ChunkFrameID](#)

Description: Returns the image frame ID.

- [GenApi::String](#) & [ChunkSerialData](#)

Description: Returns the serial data that was received.

- [GenApi::Float](#) & [ChunkExposureTime](#)

Description: Returns the exposure time used to capture the image.

- [GenApi::Integer](#) & [ChunkCompressionMode](#)

Description: Returns the compression mode of the last image payload.

- [GenApi::Float](#) & [ChunkCompressionRatio](#)

Description: Returns the compression ratio of the last image payload.

- [GenApi::Boolean](#) & [ChunkSerialReceiveOverflow](#)

Description: Returns the status of the chunk serial receive overflow.

- [GenApi::Integer](#) & [ChunkTimestamp](#)

Description: Returns the Timestamp of the image.

- [GenApi::Boolean](#) & [ChunkModeActive](#)

Description: Activates the inclusion of Chunk data in the payload of the image.

- [GenApi::Integer](#) & [ChunkExposureEndLineStatusAll](#)

Description: Returns the status of all the I/O lines at the end of exposure event.

- [GenApi::EnumerationT< ChunkGainSelectorEnums >](#) & [ChunkGainSelector](#)

Description: Selects which gain to retrieve Visibility:

- [GenApi::EnumerationT< ChunkSelectorEnums >](#) & [ChunkSelector](#)

Description: Selects which chunk data to enable or disable.

- [GenApi::EnumerationT< ChunkBlackLevelSelectorEnums >](#) & [ChunkBlackLevelSelector](#)

- Description: Selects which black level to retrieve Visibility:*

 - [GenApi::Integer](#) & [ChunkWidth](#)

Description: Returns the width of the image included in the payload.
- [GenApi::Integer](#) & [ChunkImage](#)

Description: Returns the image payload.
- [GenApi::Integer](#) & [ChunkHeight](#)

Description: Returns the height of the image included in the payload.
- [GenApi::EnumerationT< ChunkPixelFormatEnums >](#) & [ChunkPixelFormat](#)

Description: Format of the pixel provided by the camera Visibility:
- [GenApi::Float](#) & [ChunkGain](#)

Description: Returns the gain used to capture the image.
- [GenApi::Integer](#) & [ChunkSequencerSetActive](#)

Description: Returns the index of the active set of the running sequencer included in the payload.
- [GenApi::Integer](#) & [ChunkCRC](#)

Description: Returns the CRC of the image payload.
- [GenApi::Integer](#) & [ChunkOffsetX](#)

Description: Returns the Offset X of the image included in the payload.
- [GenApi::Integer](#) & [ChunkOffsetY](#)

Description: Returns the Offset Y of the image included in the payload.
- [GenApi::Boolean](#) & [ChunkEnable](#)

Description: Enables the inclusion of the selected Chunk data in the payload of the image.
- [GenApi::Integer](#) & [ChunkSerialDataLength](#)

Description: Returns the length of the received serial data that was included in the payload.
- [GenApi::Integer](#) & [FileAccessOffset](#)

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.
- [GenApi::Integer](#) & [FileAccessLength](#)

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.
- [GenApi::EnumerationT< FileOperationStatusEnums >](#) & [FileOperationStatus](#)

Description: Represents the file operation execution status.
- [GenApi::Command](#) & [FileOperationExecute](#)

Description:
This is a command that executes the selected file operation on the selected file.
- [GenApi::EnumerationT< FileOpenModeEnums >](#) & [FileOpenMode](#)

Description:
The mode of the file when it is opened.
- [GenApi::Integer](#) & [FileOperationResult](#)

Description: Represents the file operation result.
- [GenApi::EnumerationT< FileOperationSelectorEnums >](#) & [FileOperationSelector](#)

Description:
Sets operation to execute on the selected file when the execute command is given.
- [GenApi::EnumerationT< FileSelectorEnums >](#) & [FileSelector](#)

Description:
Selects which file is being operated on.
- [GenApi::Integer](#) & [FileSize](#)

Description: Represents the size of the selected file in bytes.
- [GenApi::EnumerationT< BinningSelectorEnums >](#) & [BinningSelector](#)

Description:
Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.
- [GenApi::Integer](#) & [PixelDynamicRangeMin](#)

Description: Minimum value that can be returned during the digitization process.
- [GenApi::Integer](#) & [PixelDynamicRangeMax](#)

Description: Maximum value that can be returned during the digitization process.

- [GenApi::Integer & OffsetY](#)
Description:
Vertical offset from the origin to the ROI (in pixels).
- [GenApi::Integer & BinningHorizontal](#)
Description:
Number of horizontal photo-sensitive cells to combine together.
- [GenApi::Integer & Width](#)
Description:
Width of the image provided by the device (in pixels).
- [GenApi::EnumerationT< TestPatternGeneratorSelectorEnums > & TestPatternGeneratorSelector](#)
Description:
Selects which test pattern generator is controlled by the TestPattern feature.
- [GenApi::IFloat & CompressionRatio](#)
Description: *Reports the ratio between the uncompressed image size and compressed image size.*
- [GenApi::EnumerationT< CompressionSaturationPriorityEnums > & CompressionSaturationPriority](#)
Description: *When FrameRate is enabled, camera drops frames if datarate is saturated.*
- [GenApi::IBoolean & ReverseX](#)
Description: *Horizontally flips the image sent by the device.*
- [GenApi::IBoolean & ReverseY](#)
Description: *Vertically flips the image sent by the device.*
- [GenApi::EnumerationT< TestPatternEnums > & TestPattern](#)
Description:
Selects the type of test pattern that is generated by the device as image source.
- [GenApi::EnumerationT< PixelColorFilterEnums > & PixelColorFilter](#)
Description: *Type of color filter that is applied to the image.*
- [GenApi::Integer & WidthMax](#)
Description:
Maximum width of the image (in pixels).
- [GenApi::EnumerationT< AdcBitDepthEnums > & AdcBitDepth](#)
Description:
Selects which ADC bit depth to use.
- [GenApi::Integer & BinningVertical](#)
Description:
Number of vertical photo-sensitive cells to combine together.
- [GenApi::EnumerationT< DecimationHorizontalModeEnums > & DecimationHorizontalMode](#)
Description:
The mode used to reduce the horizontal resolution when DecimationHorizontal is used.
- [GenApi::EnumerationT< BinningVerticalModeEnums > & BinningVerticalMode](#)
Description: *Visibility:*
- [GenApi::Integer & OffsetX](#)
Description:
Horizontal offset from the origin to the ROI (in pixels).
- [GenApi::Integer & HeightMax](#)
Description: *Maximum height of the image (in pixels).*
- [GenApi::Integer & DecimationHorizontal](#)
Description:
Horizontal decimation of the image.
- [GenApi::EnumerationT< PixelSizeEnums > & PixelSize](#)
Description: *Total size in bits of a pixel of the image.*
- [GenApi::Integer & SensorHeight](#)
Description: *Effective height of the sensor in pixels.*
- [GenApi::EnumerationT< DecimationSelectorEnums > & DecimationSelector](#)
Description: *Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.*

- [GenApi::IBoolean & IspEnable](#)
Description:
Controls whether the image processing core is used for optional pixel format mode (i.e.
- [GenApi::IBoolean & AdaptiveCompressionEnable](#)
Description: Controls whether lossless compression adapts to the image content.
- [GenApi::IEnumerationT< ImageCompressionModeEnums > & ImageCompressionMode](#)
Description: Visibility:
- [GenApi::Integer & DecimationVertical](#)
Description:
Vertical decimation of the image.
- [GenApi::Integer & Height](#)
Description:
Height of the image provided by the device (in pixels).
- [GenApi::IEnumerationT< BinningHorizontalModeEnums > & BinningHorizontalMode](#)
Description: Visibility:
- [GenApi::IEnumerationT< PixelFormatEnums > & PixelFormat](#)
Description: Format of the pixel provided by the camera.
- [GenApi::Integer & SensorWidth](#)
Description: Effective width of the sensor in pixels.
- [GenApi::IEnumerationT< DecimationVerticalModeEnums > & DecimationVerticalMode](#)
Description:
The mode used to reduce the vertical resolution when DecimationVertical is used.
- [GenApi::ICommand & TestEventGenerate](#)
Description: This command generates a test event and sends it to the host.
- [GenApi::ICommand & TriggerEventTest](#)
Description: This command generates a test event and sends it to the host.
- [GenApi::Integer & GuiXmlManifestAddress](#)
Description: Location of the GUI XML manifest table.
- [GenApi::Integer & Test0001](#)
Description: For testing only.
- [GenApi::IBoolean & V3_3Enable](#)
Description: Internally generated 3.3V rail.
- [GenApi::IEnumerationT< LineModeEnums > & LineMode](#)
Description: Controls if the physical Line is used to Input or Output a signal.
- [GenApi::IEnumerationT< LineSourceEnums > & LineSource](#)
Description: Selects which internal acquisition or I/O source signal to output on the selected line.
- [GenApi::IEnumerationT< LineInputFilterSelectorEnums > & LineInputFilterSelector](#)
Description: Selects the kind of input filter to configure: Deglitch or Debounce.
- [GenApi::IBoolean & UserOutputValue](#)
Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).
- [GenApi::Integer & UserOutputValueAll](#)
Description: Returns the current status of all the user output status bits in a hexadecimal representation (UserOutput 0 status corresponds to bit 0, UserOutput 1 status with bit 1, etc).
- [GenApi::IEnumerationT< UserOutputSelectorEnums > & UserOutputSelector](#)
Description: Selects which bit of the User Output register is set by UserOutputValue.
- [GenApi::IBoolean & LineStatus](#)
Description: Returns the current status of the selected input or output Line Visibility:
- [GenApi::IEnumerationT< LineFormatEnums > & LineFormat](#)
Description: Displays the current electrical format of the selected physical input or output Line.
- [GenApi::Integer & LineStatusAll](#)
Description: Returns the current status of all the line status bits in a hexadecimal representation (Line 0 status corresponds to bit 0, Line 1 status with bit 1, etc).

- [GenApi::IEnumerationT< LineSelectorEnums > & LineSelector](#)
Description: Selects the physical line (or pin) of the external device connector to configure Visibility:
- [GenApi::IEnumerationT< ExposureActiveModeEnums > & ExposureActiveMode](#)
Description: Control sensor active exposure mode.
- [GenApi::IBoolean & LineInverter](#)
Description: Controls the inversion of the signal of the selected input or output line.
- [GenApi::IFloat & LineFilterWidth](#)
Description: Filter width in microseconds for the selected line and filter combination Visibility:
- [GenApi::IEnumerationT< CounterTriggerActivationEnums > & CounterTriggerActivation](#)
Description: Selects the activation mode of the trigger to start the counter.
- [GenApi::Integer & CounterValue](#)
Description: Current counter value Visibility:
- [GenApi::IEnumerationT< CounterSelectorEnums > & CounterSelector](#)
Description: Selects which counter to configure Visibility:
- [GenApi::Integer & CounterValueAtReset](#)
Description: Value of the selected Counter when it was reset by a trigger.
- [GenApi::IEnumerationT< CounterStatusEnums > & CounterStatus](#)
Description: Returns the current status of the counter.
- [GenApi::IEnumerationT< CounterTriggerSourceEnums > & CounterTriggerSource](#)
Description: Selects the source of the trigger to start the counter Visibility:
- [GenApi::Integer & CounterDelay](#)
Description: Sets the delay (or number of events) before the CounterStart event is generated.
- [GenApi::IEnumerationT< CounterResetSourceEnums > & CounterResetSource](#)
Description: Selects the signal that will be the source to reset the counter.
- [GenApi::IEnumerationT< CounterEventSourceEnums > & CounterEventSource](#)
Description: Selects the event that will increment the counter Visibility:
- [GenApi::IEnumerationT< CounterEventActivationEnums > & CounterEventActivation](#)
Description: Selects the activation mode of the event to increment the Counter.
- [GenApi::Integer & CounterDuration](#)
Description: Sets the duration (or number of events) before the CounterEnd event is generated.
- [GenApi::IEnumerationT< CounterResetActivationEnums > & CounterResetActivation](#)
Description: Selects the Activation mode of the Counter Reset Source signal.
- [GenApi::IEnumerationT< DeviceTypeEnums > & DeviceType](#)
Description: Returns the device type.
- [GenApi::IString & DeviceFamilyName](#)
Description: Identifier of the product family of the device.
- [GenApi::Integer & DeviceSFNCVersionMajor](#)
Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer & DeviceSFNCVersionMinor](#)
Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer & DeviceSFNCVersionSubMinor](#)
Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer & DeviceManifestEntrySelector](#)
Description: Selects the manifest entry to reference.
- [GenApi::Integer & DeviceManifestXMLMajorVersion](#)
Description: Indicates the major version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer & DeviceManifestXMLMinorVersion](#)
Description: Indicates the minor version number of the [GenICam XML](#) file of the selected manifest entry.

- [GenApi::Integer](#) & [DeviceManifestXMLSubMinorVersion](#)
Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMajorVersion](#)
Description: Indicates the major version number of the schema file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMinorVersion](#)
Description: Indicates the minor version number of the schema file of the selected manifest entry.
- [GenApi::IString](#) & [DeviceManifestPrimaryURL](#)
Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.
- [GenApi::IString](#) & [DeviceManifestSecondaryURL](#)
Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceTLVersionSubMinor](#)
Description: Sub minor version of the Transport Layer of the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMajor](#)
Description: Major version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMinor](#)
Description: Minor version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceConnectionSelector](#)
Description: Selects which Connection of the device to control.
- [GenApi::Integer](#) & [DeviceConnectionSpeed](#)
Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.
- [GenApi::IEnumerationT](#) < [DeviceConnectionStatusEnums](#) > & [DeviceConnectionStatus](#)
Description: Indicates the status of the specified Connection.
- [GenApi::Integer](#) & [DeviceLinkSelector](#)
Description: Selects which Link of the device to control.
- [GenApi::IEnumerationT](#) < [DeviceLinkThroughputLimitModeEnums](#) > & [DeviceLinkThroughputLimitMode](#)
Description: Controls if the DeviceLinkThroughputLimit is active.
- [GenApi::Integer](#) & [DeviceLinkConnectionCount](#)
Description: Returns the number of physical connection of the device used by a particular Link.
- [GenApi::IEnumerationT](#) < [DeviceLinkHeartbeatModeEnums](#) > & [DeviceLinkHeartbeatMode](#)
Description: Activate or deactivate the Link's heartbeat.
- [GenApi::IFloat](#) & [DeviceLinkHeartbeatTimeout](#)
Description: Controls the current heartbeat timeout of the specific Link.
- [GenApi::IFloat](#) & [DeviceLinkCommandTimeout](#)
Description: Indicates the command timeout of the specified Link.
- [GenApi::Integer](#) & [DeviceStreamChannelSelector](#)
Description: Selects the stream channel to control.
- [GenApi::IEnumerationT](#) < [DeviceStreamChannelTypeEnums](#) > & [DeviceStreamChannelType](#)
Description: Reports the type of the stream channel.
- [GenApi::Integer](#) & [DeviceStreamChannelLink](#)
Description: Index of device's Link to use for streaming the specified stream channel.
- [GenApi::IEnumerationT](#) < [DeviceStreamChannelEndiannessEnums](#) > & [DeviceStreamChannelEndianness](#)
Description: Endianness of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [DeviceStreamChannelPacketSize](#)
Description: Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceStart](#)
Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceEnd](#)
Description: Indicate to the device the end of feature persistence.
- [GenApi::ICommand](#) & [DeviceRegistersStreamingStart](#)
Description: Prepare the device for registers streaming without checking for consistency.

- [GenApi::ICommand](#) & [DeviceRegistersStreamingEnd](#)
Description: Announce the end of registers streaming.
- [GenApi::ICommand](#) & [DeviceRegistersCheck](#)
Description: Perform the validation of the current register set for consistency.
- [GenApi::IBoolean](#) & [DeviceRegistersValid](#)
Description: Returns if the current register set is valid and consistent.
- [GenApi::IEnumerationT< DeviceClockSelectorEnums >](#) & [DeviceClockSelector](#)
Description: Selects the clock frequency to access from the device.
- [GenApi::IFloat](#) & [DeviceClockFrequency](#)
Description: Returns the frequency of the selected Clock.
- [GenApi::IEnumerationT< DeviceSerialPortSelectorEnums >](#) & [DeviceSerialPortSelector](#)
Description: Selects which serial port of the device to control.
- [GenApi::IEnumerationT< DeviceSerialPortBaudRateEnums >](#) & [DeviceSerialPortBaudRate](#)
Description: This feature controls the baud rate used by the selected serial port.
- [GenApi::Integer](#) & [Timestamp](#)
Description: Reports the current value of the device timestamp counter.
- [GenApi::IEnumerationT< SensorTapsEnums >](#) & [SensorTaps](#)
Description: Number of taps of the camera sensor.
- [GenApi::IEnumerationT< SensorDigitizationTapsEnums >](#) & [SensorDigitizationTaps](#)
Description: Number of digitized samples outputted simultaneously by the camera A/D conversion stage.
- [GenApi::IEnumerationT< RegionSelectorEnums >](#) & [RegionSelector](#)
Description: Selects the Region of interest to control.
- [GenApi::IEnumerationT< RegionModeEnums >](#) & [RegionMode](#)
Description: Controls if the selected Region of interest is active and streaming.
- [GenApi::IEnumerationT< RegionDestinationEnums >](#) & [RegionDestination](#)
Description: Control the destination of the selected region.
- [GenApi::IEnumerationT< ImageComponentSelectorEnums >](#) & [ImageComponentSelector](#)
Description: Selects a component to activate data streaming from.
- [GenApi::IBoolean](#) & [ImageComponentEnable](#)
Description: Controls if the selected component streaming is active.
- [GenApi::Integer](#) & [LinePitch](#)
Description: Total number of bytes between 2 successive lines.
- [GenApi::IEnumerationT< PixelFormatInfoSelectorEnums >](#) & [PixelFormatInfoSelector](#)
Description: Select the pixel format for which the information will be returned.
- [GenApi::Integer](#) & [PixelFormatInfoID](#)
Description: Returns the value used by the streaming channels to identify the selected pixel format.
- [GenApi::IEnumerationT< DeinterlacingEnums >](#) & [Deinterlacing](#)
Description: Controls how the device performs de-interlacing.
- [GenApi::IEnumerationT< ImageCompressionRateOptionEnums >](#) & [ImageCompressionRateOption](#)
Description: Two rate controlling options are offered: fixed bit rate or fixed quality.
- [GenApi::Integer](#) & [ImageCompressionQuality](#)
Description: Control the quality of the produced compressed stream.
- [GenApi::IFloat](#) & [ImageCompressionBitrate](#)
Description: Control the rate of the produced compressed stream.
- [GenApi::IEnumerationT< ImageCompressionJPEGFormatOptionEnums >](#) & [ImageCompressionJPEGFormatOption](#)
Description: When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.
- [GenApi::ICommand](#) & [AcquisitionAbort](#)
Description: Aborts the Acquisition immediately.
- [GenApi::ICommand](#) & [AcquisitionArm](#)

- Description: Arms the device before an AcquisitionStart command.*

 - [GenApi::IEnumerationT< AcquisitionStatusSelectorEnums > & AcquisitionStatusSelector](#)

Description: Selects the internal acquisition signal to read using AcquisitionStatus.
- [GenApi::IBoolean & AcquisitionStatus](#)

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.
- [GenApi::IInteger & TriggerDivider](#)

Description: Specifies a division factor for the incoming trigger pulses.
- [GenApi::IInteger & TriggerMultiplier](#)

Description: Specifies a multiplication factor for the incoming trigger pulses.
- [GenApi::IEnumerationT< ExposureTimeModeEnums > & ExposureTimeMode](#)

Description: Sets the configuration mode of the ExposureTime feature.
- [GenApi::IEnumerationT< ExposureTimeSelectorEnums > & ExposureTimeSelector](#)

Description: Selects which exposure time is controlled by the ExposureTime feature.
- [GenApi::IEnumerationT< GainAutoBalanceEnums > & GainAutoBalance](#)

Description: Sets the mode for automatic gain balancing between the sensor color channels or taps.
- [GenApi::IEnumerationT< BlackLevelAutoEnums > & BlackLevelAuto](#)

Description: Controls the mode for automatic black level adjustment.
- [GenApi::IEnumerationT< BlackLevelAutoBalanceEnums > & BlackLevelAutoBalance](#)

Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.
- [GenApi::IEnumerationT< WhiteClipSelectorEnums > & WhiteClipSelector](#)

Description: Selects which White Clip to control.
- [GenApi::IFloat & WhiteClip](#)

Description: Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.
- [GenApi::IRegister & LUTValueAll](#)

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.
- [GenApi::IInteger & UserOutputValueAllMask](#)

Description: Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.
- [GenApi::ICommand & CounterReset](#)

Description: Does a software reset of the selected Counter and starts it.
- [GenApi::IEnumerationT< TimerSelectorEnums > & TimerSelector](#)

Description: Selects which Timer to configure.
- [GenApi::IFloat & TimerDuration](#)

Description: Sets the duration (in microseconds) of the Timer pulse.
- [GenApi::IFloat & TimerDelay](#)

Description: Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.
- [GenApi::ICommand & TimerReset](#)

Description: Does a software reset of the selected timer and starts it.
- [GenApi::IFloat & TimerValue](#)

Description: Reads or writes the current value (in microseconds) of the selected Timer.
- [GenApi::IEnumerationT< TimerStatusEnums > & TimerStatus](#)

Description: Returns the current status of the Timer.
- [GenApi::IEnumerationT< TimerTriggerSourceEnums > & TimerTriggerSource](#)

Description: Selects the source of the trigger to start the Timer.
- [GenApi::IEnumerationT< TimerTriggerActivationEnums > & TimerTriggerActivation](#)

Description: Selects the activation mode of the trigger to start the Timer.
- [GenApi::IEnumerationT< EncoderSelectorEnums > & EncoderSelector](#)

Description: Selects which Encoder to configure.
- [GenApi::IEnumerationT< EncoderSourceAEnums > & EncoderSourceA](#)

- Description: Selects the signal which will be the source of the A input of the Encoder.*

 - [GenApi::IEnumerationT< EncoderSourceBEnums > & EncoderSourceB](#)
- Description: Selects the signal which will be the source of the B input of the Encoder.*

 - [GenApi::IEnumerationT< EncoderModeEnums > & EncoderMode](#)
- Description: Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.*

 - [GenApi::Integer & EncoderDivider](#)
- Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.*

 - [GenApi::IEnumerationT< EncoderOutputModeEnums > & EncoderOutputMode](#)
- Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.*

 - [GenApi::IEnumerationT< EncoderStatusEnums > & EncoderStatus](#)
- Description: Returns the motion status of the encoder.*

 - [GenApi::IFloat & EncoderTimeout](#)
- Description: Sets the maximum time interval between encoder counter increments before the status turns to static.*

 - [GenApi::IEnumerationT< EncoderResetSourceEnums > & EncoderResetSource](#)
- Description: Selects the signals that will be the source to reset the Encoder.*

 - [GenApi::IEnumerationT< EncoderResetActivationEnums > & EncoderResetActivation](#)
- Description: Selects the Activation mode of the Encoder Reset Source signal.*

 - [GenApi::ICommand & EncoderReset](#)
- Description: Does a software reset of the selected Encoder and starts it.*

 - [GenApi::Integer & EncoderValue](#)
- Description: Reads or writes the current value of the position counter of the selected Encoder.*

 - [GenApi::Integer & EncoderValueAtReset](#)
- Description: Reads the value of the of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.*

 - [GenApi::IEnumerationT< SoftwareSignalSelectorEnums > & SoftwareSignalSelector](#)
- Description: Selects which Software Signal features to control.*

 - [GenApi::ICommand & SoftwareSignalPulse](#)
- Description: Generates a pulse signal that can be used as a software trigger.*

 - [GenApi::IEnumerationT< ActionUnconditionalModeEnums > & ActionUnconditionalMode](#)
- Description: Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.*

 - [GenApi::Integer & ActionDeviceKey](#)
- Description: Provides the device key that allows the device to check the validity of action commands.*

 - [GenApi::Integer & ActionQueueSize](#)
- Description: Indicates the size of the scheduled action commands queue.*

 - [GenApi::Integer & ActionSelector](#)
- Description: Selects to which Action Signal further Action settings apply.*

 - [GenApi::Integer & ActionGroupMask](#)
- Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.*

 - [GenApi::Integer & ActionGroupKey](#)
- Description: Provides the key that the device will use to validate the action on reception of the action protocol message.*

 - [GenApi::Integer & EventAcquisitionTrigger](#)
- Description: Returns the unique Identifier of the Acquisition Trigger type of Event.*

 - [GenApi::Integer & EventAcquisitionTriggerTimestamp](#)
- Description: Returns the Timestamp of the Acquisition Trigger Event.*

 - [GenApi::Integer & EventAcquisitionTriggerFrameID](#)
- Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger Event.*

 - [GenApi::Integer & EventAcquisitionStart](#)

- Description: Returns the unique Identifier of the Acquisition Start type of Event.*

 - [GenApi::Integer](#) & [EventAcquisitionStartTimestamp](#)

Description: Returns the Timestamp of the Acquisition Start Event.
- [GenApi::Integer](#) & [EventAcquisitionStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start Event.
- [GenApi::Integer](#) & [EventAcquisitionEnd](#)

Description: Returns the unique Identifier of the Acquisition End type of Event.
- [GenApi::Integer](#) & [EventAcquisitionEndTimestamp](#)

Description: Returns the Timestamp of the Acquisition End Event.
- [GenApi::Integer](#) & [EventAcquisitionEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferStart](#)

Description: Returns the unique Identifier of the Acquisition Transfer Start type of Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferStartTimestamp](#)

Description: Returns the Timestamp of the Acquisition Transfer Start Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferEnd](#)

Description: Returns the unique Identifier of the Acquisition Transfer End type of Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferEndTimestamp](#)

Description: Returns the Timestamp of the Acquisition Transfer End Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End Event.
- [GenApi::Integer](#) & [EventAcquisitionError](#)

Description: Returns the unique Identifier of the Acquisition Error type of Event.
- [GenApi::Integer](#) & [EventAcquisitionErrorTimestamp](#)

Description: Returns the Timestamp of the Acquisition Error Event.
- [GenApi::Integer](#) & [EventAcquisitionErrorFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.
- [GenApi::Integer](#) & [EventFrameTrigger](#)

Description: Returns the unique Identifier of the FrameTrigger type of Event.
- [GenApi::Integer](#) & [EventFrameTriggerTimestamp](#)

Description: Returns the Timestamp of the FrameTrigger Event.
- [GenApi::Integer](#) & [EventFrameTriggerFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.
- [GenApi::Integer](#) & [EventFrameStart](#)

Description: Returns the unique Identifier of the Frame Start type of Event.
- [GenApi::Integer](#) & [EventFrameStartTimestamp](#)

Description: Returns the Timestamp of the Frame Start Event.
- [GenApi::Integer](#) & [EventFrameStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start Event.
- [GenApi::Integer](#) & [EventFrameEnd](#)

Description: Returns the unique Identifier of the Frame End type of Event.
- [GenApi::Integer](#) & [EventFrameEndTimestamp](#)

Description: Returns the Timestamp of the Frame End Event.
- [GenApi::Integer](#) & [EventFrameEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End Event.
- [GenApi::Integer](#) & [EventFrameBurstStart](#)

Description: Returns the unique Identifier of the Frame Burst Start type of Event.
- [GenApi::Integer](#) & [EventFrameBurstStartTimestamp](#)

Description: Returns the Timestamp of the Frame Burst Start Event.

- [GenApi::Integer](#) & [EventFrameBurstStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start Event.
- [GenApi::Integer](#) & [EventFrameBurstEnd](#)
Description: Returns the unique Identifier of the Frame Burst End type of Event.
- [GenApi::Integer](#) & [EventFrameBurstEndTimestamp](#)
Description: Returns the Timestamp of the Frame Burst End Event.
- [GenApi::Integer](#) & [EventFrameBurstEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End Event.
- [GenApi::Integer](#) & [EventFrameTransferStart](#)
Description: Returns the unique Identifier of the Frame Transfer Start type of Event.
- [GenApi::Integer](#) & [EventFrameTransferStartTimestamp](#)
Description: Returns the Timestamp of the Frame Transfer Start Event.
- [GenApi::Integer](#) & [EventFrameTransferStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start Event.
- [GenApi::Integer](#) & [EventFrameTransferEnd](#)
Description: Returns the unique Identifier of the Frame Transfer End type of Event.
- [GenApi::Integer](#) & [EventFrameTransferEndTimestamp](#)
Description: Returns the Timestamp of the Frame Transfer End Event.
- [GenApi::Integer](#) & [EventFrameTransferEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End Event.
- [GenApi::Integer](#) & [EventExposureStart](#)
Description: Returns the unique Identifier of the Exposure Start type of Event.
- [GenApi::Integer](#) & [EventExposureStartTimestamp](#)
Description: Returns the Timestamp of the Exposure Start Event.
- [GenApi::Integer](#) & [EventExposureStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start Event.
- [GenApi::Integer](#) & [EventStream0TransferStart](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Start type of Event.
- [GenApi::Integer](#) & [EventStream0TransferStartTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Start Event.
- [GenApi::Integer](#) & [EventStream0TransferStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start Event.
- [GenApi::Integer](#) & [EventStream0TransferEnd](#)
Description: Returns the unique Identifier of the Stream 0 Transfer End type of Event.
- [GenApi::Integer](#) & [EventStream0TransferEndTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer End Event.
- [GenApi::Integer](#) & [EventStream0TransferEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End Event.
- [GenApi::Integer](#) & [EventStream0TransferPause](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of Event.
- [GenApi::Integer](#) & [EventStream0TransferPauseTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Pause Event.
- [GenApi::Integer](#) & [EventStream0TransferPauseFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause Event.
- [GenApi::Integer](#) & [EventStream0TransferResume](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of Event.
- [GenApi::Integer](#) & [EventStream0TransferResumeTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Resume Event.
- [GenApi::Integer](#) & [EventStream0TransferResumeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockStart](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockStartTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Block Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockEnd](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockEndTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockTrigger](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstStart](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstStartTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstEnd](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstEndTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.
- [GenApi::Integer](#) & [EventStream0TransferOverflow](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.
- [GenApi::Integer](#) & [EventStream0TransferOverflowTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.
- [GenApi::Integer](#) & [EventStream0TransferOverflowFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.
- [GenApi::Integer](#) & [EventSequencerSetChange](#)
Description: Returns the unique Identifier of the Sequencer Set Change type of Event.
- [GenApi::Integer](#) & [EventSequencerSetChangeTimestamp](#)
Description: Returns the Timestamp of the Sequencer Set Change Event.
- [GenApi::Integer](#) & [EventSequencerSetChangeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.
- [GenApi::Integer](#) & [EventCounter0Start](#)
Description: Returns the unique Identifier of the Counter 0 Start type of Event.
- [GenApi::Integer](#) & [EventCounter0StartTimestamp](#)
Description: Returns the Timestamp of the Counter 0 Start Event.

- [GenApi::Integer](#) & [EventCounter0StartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.
- [GenApi::Integer](#) & [EventCounter1Start](#)
Description: Returns the unique Identifier of the Counter 1 Start type of Event.
- [GenApi::Integer](#) & [EventCounter1StartTimestamp](#)
Description: Returns the Timestamp of the Counter 1 Start Event.
- [GenApi::Integer](#) & [EventCounter1StartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start Event.
- [GenApi::Integer](#) & [EventCounter0End](#)
Description: Returns the unique Identifier of the Counter 0 End type of Event.
- [GenApi::Integer](#) & [EventCounter0EndTimestamp](#)
Description: Returns the Timestamp of the Counter 0 End Event.
- [GenApi::Integer](#) & [EventCounter0EndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End Event.
- [GenApi::Integer](#) & [EventCounter1End](#)
Description: Returns the unique Identifier of the Counter 1 End type of Event.
- [GenApi::Integer](#) & [EventCounter1EndTimestamp](#)
Description: Returns the Timestamp of the Counter 1 End Event.
- [GenApi::Integer](#) & [EventCounter1EndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End Event.
- [GenApi::Integer](#) & [EventTimer0Start](#)
Description: Returns the unique Identifier of the Timer 0 Start type of Event.
- [GenApi::Integer](#) & [EventTimer0StartTimestamp](#)
Description: Returns the Timestamp of the Timer 0 Start Event.
- [GenApi::Integer](#) & [EventTimer0StartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start Event.
- [GenApi::Integer](#) & [EventTimer1Start](#)
Description: Returns the unique Identifier of the Timer 1 Start type of Event.
- [GenApi::Integer](#) & [EventTimer1StartTimestamp](#)
Description: Returns the Timestamp of the Timer 1 Start Event.
- [GenApi::Integer](#) & [EventTimer1StartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start Event.
- [GenApi::Integer](#) & [EventTimer0End](#)
Description: Returns the unique Identifier of the Timer 0 End type of Event.
- [GenApi::Integer](#) & [EventTimer0EndTimestamp](#)
Description: Returns the Timestamp of the Timer 0 End Event.
- [GenApi::Integer](#) & [EventTimer0EndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End Event.
- [GenApi::Integer](#) & [EventTimer1End](#)
Description: Returns the unique Identifier of the Timer 1 End type of Event.
- [GenApi::Integer](#) & [EventTimer1EndTimestamp](#)
Description: Returns the Timestamp of the Timer 1 End Event.
- [GenApi::Integer](#) & [EventTimer1EndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End Event.
- [GenApi::Integer](#) & [EventEncoder0Stopped](#)
Description: Returns the unique Identifier of the Encoder 0 Stopped type of Event.
- [GenApi::Integer](#) & [EventEncoder0StoppedTimestamp](#)
Description: Returns the Timestamp of the Encoder 0 Stopped Event.
- [GenApi::Integer](#) & [EventEncoder0StoppedFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped Event.
- [GenApi::Integer](#) & [EventEncoder1Stopped](#)

- Description: Returns the unique Identifier of the Encoder 1 Stopped type of Event.*

 - [GenApi::Integer](#) & [EventEncoder1StoppedTimestamp](#)

Description: Returns the Timestamp of the Encoder 1 Stopped Event.
- [GenApi::Integer](#) & [EventEncoder1StoppedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped Event.
- [GenApi::Integer](#) & [EventEncoder0Restarted](#)

Description: Returns the unique Identifier of the Encoder 0 Restarted type of Event.
- [GenApi::Integer](#) & [EventEncoder0RestartedTimestamp](#)

Description: Returns the Timestamp of the Encoder 0 Restarted Event.
- [GenApi::Integer](#) & [EventEncoder0RestartedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted Event.
- [GenApi::Integer](#) & [EventEncoder1Restarted](#)

Description: Returns the unique Identifier of the Encoder 1 Restarted type of Event.
- [GenApi::Integer](#) & [EventEncoder1RestartedTimestamp](#)

Description: Returns the Timestamp of the Encoder 1 Restarted Event.
- [GenApi::Integer](#) & [EventEncoder1RestartedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted Event.
- [GenApi::Integer](#) & [EventLine0RisingEdge](#)

Description: Returns the unique Identifier of the Line 0 Rising Edge type of Event.
- [GenApi::Integer](#) & [EventLine0RisingEdgeTimestamp](#)

Description: Returns the Timestamp of the Line 0 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine0RisingEdgeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine1RisingEdge](#)

Description: Returns the unique Identifier of the Line 1 Rising Edge type of Event.
- [GenApi::Integer](#) & [EventLine1RisingEdgeTimestamp](#)

Description: Returns the Timestamp of the Line 1 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine1RisingEdgeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine0FallingEdge](#)

Description: Returns the unique Identifier of the Line 0 Falling Edge type of Event.
- [GenApi::Integer](#) & [EventLine0FallingEdgeTimestamp](#)

Description: Returns the Timestamp of the Line 0 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine0FallingEdgeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine1FallingEdge](#)

Description: Returns the unique Identifier of the Line 1 Falling Edge type of Event.
- [GenApi::Integer](#) & [EventLine1FallingEdgeTimestamp](#)

Description: Returns the Timestamp of the Line 1 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine1FallingEdgeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine0AnyEdge](#)

Description: Returns the unique Identifier of the Line 0 Any Edge type of Event.
- [GenApi::Integer](#) & [EventLine0AnyEdgeTimestamp](#)

Description: Returns the Timestamp of the Line 0 Any Edge Event.
- [GenApi::Integer](#) & [EventLine0AnyEdgeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge Event.
- [GenApi::Integer](#) & [EventLine1AnyEdge](#)

Description: Returns the unique Identifier of the Line 1 Any Edge type of Event.
- [GenApi::Integer](#) & [EventLine1AnyEdgeTimestamp](#)

Description: Returns the Timestamp of the Line 1 Any Edge Event.

- [GenApi::Integer & EventLine1AnyEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge Event.
- [GenApi::Integer & EventLinkTrigger0](#)
Description: Returns the unique Identifier of the Link Trigger 0 type of Event.
- [GenApi::Integer & EventLinkTrigger0Timestamp](#)
Description: Returns the Timestamp of the Link Trigger 0 Event.
- [GenApi::Integer & EventLinkTrigger0FrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 Event.
- [GenApi::Integer & EventLinkTrigger1](#)
Description: Returns the unique Identifier of the Link Trigger 1 type of Event.
- [GenApi::Integer & EventLinkTrigger1Timestamp](#)
Description: Returns the Timestamp of the Link Trigger 1 Event.
- [GenApi::Integer & EventLinkTrigger1FrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 Event.
- [GenApi::Integer & EventActionLate](#)
Description: Returns the unique Identifier of the Action Late type of Event.
- [GenApi::Integer & EventActionLateTimestamp](#)
Description: Returns the Timestamp of the Action Late Event.
- [GenApi::Integer & EventActionLateFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late Event.
- [GenApi::Integer & EventLinkSpeedChange](#)
Description: Returns the unique Identifier of the Link Speed Change type of Event.
- [GenApi::Integer & EventLinkSpeedChangeTimestamp](#)
Description: Returns the Timestamp of the Link Speed Change Event.
- [GenApi::Integer & EventLinkSpeedChangeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change Event.
- [GenApi::IRegister & FileAccessBuffer](#)
Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
- [GenApi::Integer & SourceCount](#)
Description: Controls or returns the number of sources supported by the device.
- [GenApi::IEnumerationT< SourceSelectorEnums > & SourceSelector](#)
Description: Selects the source to control.
- [GenApi::IEnumerationT< TransferSelectorEnums > & TransferSelector](#)
Description: Selects which stream transfers are currently controlled by the selected Transfer features.
- [GenApi::Integer & TransferBurstCount](#)
Description: Number of Block(s) to transfer for each TransferBurstStart trigger.
- [GenApi::ICommand & TransferAbort](#)
Description: Aborts immediately the streaming of data block(s).
- [GenApi::ICommand & TransferPause](#)
Description: Pauses the streaming of data Block(s).
- [GenApi::ICommand & TransferResume](#)
Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.
- [GenApi::IEnumerationT< TransferTriggerSelectorEnums > & TransferTriggerSelector](#)
Description: Selects the type of transfer trigger to configure.
- [GenApi::IEnumerationT< TransferTriggerModeEnums > & TransferTriggerMode](#)
Description: Controls if the selected trigger is active.
- [GenApi::IEnumerationT< TransferTriggerSourceEnums > & TransferTriggerSource](#)
Description: Specifies the signal to use as the trigger source for transfers.
- [GenApi::IEnumerationT< TransferTriggerActivationEnums > & TransferTriggerActivation](#)
Description: Specifies the activation mode of the transfer control trigger.

- [GenApi::IEnumerationT< TransferStatusSelectorEnums > & TransferStatusSelector](#)
Description: Selects which status of the transfer module to read.
- [GenApi::IBoolean & TransferStatus](#)
Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.
- [GenApi::IEnumerationT< TransferComponentSelectorEnums > & TransferComponentSelector](#)
Description: Selects the color component for the control of the TransferStreamChannel feature.
- [GenApi::Integer & TransferStreamChannel](#)
Description: Selects the streaming channel that will be used to transfer the selected stream of data.
- [GenApi::IEnumerationT< Scan3dDistanceUnitEnums > & Scan3dDistanceUnit](#)
Description: Specifies the unit used when delivering calibrated distance data.
- [GenApi::IEnumerationT< Scan3dCoordinateSystemEnums > & Scan3dCoordinateSystem](#)
Description: Specifies the Coordinate system to use for the device.
- [GenApi::IEnumerationT< Scan3dOutputModeEnums > & Scan3dOutputMode](#)
Description: Controls the Calibration and data organization of the device, naming the coordinates transmitted.
- [GenApi::IEnumerationT< Scan3dCoordinateSystemReferenceEnums > & Scan3dCoordinateSystemReference](#)
Description: Defines coordinate system reference location.
- [GenApi::IEnumerationT< Scan3dCoordinateSelectorEnums > & Scan3dCoordinateSelector](#)
Description: Selects the individual coordinates in the vectors for 3D information/transformation.
- [GenApi::IFloat & Scan3dCoordinateScale](#)
Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.
- [GenApi::IFloat & Scan3dCoordinateOffset](#)
Description: Offset when transforming a pixel from relative coordinates to world coordinates.
- [GenApi::IBoolean & Scan3dInvalidDataFlag](#)
Description: Enables the definition of a non-valid flag value in the data stream.
- [GenApi::IFloat & Scan3dInvalidDataValue](#)
Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.
- [GenApi::IFloat & Scan3dAxisMin](#)
Description: Minimum valid transmitted coordinate value of the selected Axis.
- [GenApi::IFloat & Scan3dAxisMax](#)
Description: Maximum valid transmitted coordinate value of the selected Axis.
- [GenApi::IEnumerationT< Scan3dCoordinateTransformSelectorEnums > & Scan3dCoordinateTransformSelector](#)
Description: Sets the index to read/write a coordinate transform value.
- [GenApi::IFloat & Scan3dTransformValue](#)
Description: Specifies the transform value selected.
- [GenApi::IEnumerationT< Scan3dCoordinateReferenceSelectorEnums > & Scan3dCoordinateReferenceSelector](#)
Description: Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.
- [GenApi::IFloat & Scan3dCoordinateReferenceValue](#)
Description: Returns the reference value selected.
- [GenApi::Integer & ChunkPartSelector](#)
Description: Selects the part to access in chunk data in a multipart transmission.
- [GenApi::IEnumerationT< ChunkImageComponentEnums > & ChunkImageComponent](#)
Description: Returns the component of the payload image.
- [GenApi::Integer & ChunkPixelDynamicRangeMin](#)
Description: Returns the minimum value of dynamic range of the image included in the payload.
- [GenApi::Integer & ChunkPixelDynamicRangeMax](#)
Description: Returns the maximum value of dynamic range of the image included in the payload.
- [GenApi::Integer & ChunkTimestampLatchValue](#)

- Description: Returns the last Timestamp latched with the TimestampLatch command.*

 - [GenApi::Integer](#) & [ChunkLineStatusAll](#)
- Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.*

 - [GenApi::EnumerationT< ChunkCounterSelectorEnums >](#) & [ChunkCounterSelector](#)
- Description: Selects which counter to retrieve data from.*

 - [GenApi::Integer](#) & [ChunkCounterValue](#)
- Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.*

 - [GenApi::EnumerationT< ChunkTimerSelectorEnums >](#) & [ChunkTimerSelector](#)
- Description: Selects which Timer to retrieve data from.*

 - [GenApi::IFloat](#) & [ChunkTimerValue](#)
- Description: Returns the value of the selected Timer at the time of the FrameStart internal event.*

 - [GenApi::EnumerationT< ChunkEncoderSelectorEnums >](#) & [ChunkEncoderSelector](#)
- Description: Selects which Encoder to retrieve data from.*

 - [GenApi::Integer](#) & [ChunkScanLineSelector](#)
- Description: Index for vector representation of one chunk value per line in an image.*

 - [GenApi::Integer](#) & [ChunkEncoderValue](#)
- Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.*

 - [GenApi::EnumerationT< ChunkEncoderStatusEnums >](#) & [ChunkEncoderStatus](#)
- Description: Returns the motion status of the selected encoder.*

 - [GenApi::EnumerationT< ChunkExposureTimeSelectorEnums >](#) & [ChunkExposureTimeSelector](#)
- Description: Selects which exposure time is read by the ChunkExposureTime feature.*

 - [GenApi::Integer](#) & [ChunkLinePitch](#)
- Description: Returns the LinePitch of the image included in the payload.*

 - [GenApi::EnumerationT< ChunkSourceIDEnums >](#) & [ChunkSourceID](#)
- Description: Returns the identifier of Source that the image comes from.*

 - [GenApi::EnumerationT< ChunkRegionIDEnums >](#) & [ChunkRegionID](#)
- Description: Returns the identifier of Region that the image comes from.*

 - [GenApi::Integer](#) & [ChunkTransferBlockID](#)
- Description: Returns the unique identifier of the transfer block used to transport the payload.*

 - [GenApi::EnumerationT< ChunkTransferStreamIDEnums >](#) & [ChunkTransferStreamID](#)
- Description: Returns identifier of the stream that generated this block.*

 - [GenApi::Integer](#) & [ChunkTransferQueueCurrentBlockCount](#)
- Description: Returns the current number of blocks in the transfer queue.*

 - [GenApi::Integer](#) & [ChunkStreamChannelID](#)
- Description: Returns identifier of the stream channel used to carry the block.*

 - [GenApi::EnumerationT< ChunkScan3dDistanceUnitEnums >](#) & [ChunkScan3dDistanceUnit](#)
- Description: Returns the Distance Unit of the payload image.*

 - [GenApi::EnumerationT< ChunkScan3dOutputModeEnums >](#) & [ChunkScan3dOutputMode](#)
- Description: Returns the Calibrated Mode of the payload image.*

 - [GenApi::EnumerationT< ChunkScan3dCoordinateSystemEnums >](#) & [ChunkScan3dCoordinateSystem](#)
- Description: Returns the Coordinate [System](#) of the image included in the payload.*

 - [GenApi::EnumerationT< ChunkScan3dCoordinateSystemReferenceEnums >](#) & [ChunkScan3dCoordinateSystemReference](#)
- Description: Returns the Coordinate [System](#) Position of the image included in the payload.*

 - [GenApi::EnumerationT< ChunkScan3dCoordinateSelectorEnums >](#) & [ChunkScan3dCoordinateSelector](#)
- Description: Selects which Coordinate to retrieve data from.*

 - [GenApi::IFloat](#) & [ChunkScan3dCoordinateScale](#)
- Description: Returns the Scale for the selected coordinate axis of the image included in the payload.*

 - [GenApi::IFloat](#) & [ChunkScan3dCoordinateOffset](#)
- Description: Returns the Offset for the selected coordinate axis of the image included in the payload.*

- [GenApi::IBoolean](#) & [ChunkScan3dInvalidDataFlag](#)
Description: Returns if a specific non-valid data flag is used in the data stream.
- [GenApi::IFloat](#) & [ChunkScan3dInvalidDataValue](#)
Description: Returns the Invalid Data Value used for the image included in the payload.
- [GenApi::IFloat](#) & [ChunkScan3dAxisMin](#)
Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.
- [GenApi::IFloat](#) & [ChunkScan3dAxisMax](#)
Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.
- [GenApi::IEnumerationT< ChunkScan3dCoordinateTransformSelectorEnums >](#) & [ChunkScan3dCoordinateTransformSelector](#)
Description: Selector for transform values.
- [GenApi::IFloat](#) & [ChunkScan3dTransformValue](#)
Description: Returns the transform value.
- [GenApi::IEnumerationT< ChunkScan3dCoordinateReferenceSelectorEnums >](#) & [ChunkScan3dCoordinateReferenceSelector](#)
Description: Selector to read a coordinate system reference value defining the transform of a point from one system to the other.
- [GenApi::IFloat](#) & [ChunkScan3dCoordinateReferenceValue](#)
Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.
- [GenApi::IInteger](#) & [TestPendingAck](#)
Description: Tests the device's pending acknowledge feature.
- [GenApi::IEnumerationT< DeviceTapGeometryEnums >](#) & [DeviceTapGeometry](#)
Description: This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.
- [GenApi::IEnumerationT< GevPhysicalLinkConfigurationEnums >](#) & [GevPhysicalLinkConfiguration](#)
Description: Controls the principal physical link configuration to use on next restart/power-up of the device.
- [GenApi::IEnumerationT< GevCurrentPhysicalLinkConfigurationEnums >](#) & [GevCurrentPhysicalLinkConfiguration](#)
Description: Indicates the current physical link configuration of the device.
- [GenApi::IInteger](#) & [GevActiveLinkCount](#)
Description: Indicates the current number of active logical links.
- [GenApi::IBoolean](#) & [GevPAUSEFrameReception](#)
Description: Controls whether incoming PAUSE Frames are handled on the given logical link.
- [GenApi::IBoolean](#) & [GevPAUSEFrameTransmission](#)
Description: Controls whether PAUSE Frames can be generated on the given logical link.
- [GenApi::IEnumerationT< GevIPConfigurationStatusEnums >](#) & [GevIPConfigurationStatus](#)
Description: Reports the current IP configuration status.
- [GenApi::IInteger](#) & [GevDiscoveryAckDelay](#)
Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.
- [GenApi::IEnumerationT< GevGVCPExtendedStatusCodesSelectorEnums >](#) & [GevGVCPExtendedStatusCodesSelector](#)
Description: Selects the GigE Vision version to control extended status codes for.
- [GenApi::IBoolean](#) & [GevGVCPExtendedStatusCodes](#)
Description: Enables the generation of extended status codes.
- [GenApi::IInteger](#) & [GevPrimaryApplicationSwitchoverKey](#)
Description: Controls the key to use to authenticate primary application switchover requests.
- [GenApi::IEnumerationT< GevGVSPExtendedIDModeEnums >](#) & [GevGVSPExtendedIDMode](#)
Description: Enables the extended IDs mode.
- [GenApi::IInteger](#) & [GevPrimaryApplicationSocket](#)
Description: Returns the UDP source port of the primary application.
- [GenApi::IInteger](#) & [GevPrimaryApplicationIPAddress](#)

- Description: Returns the address of the primary application.*

 - [GenApi::IBoolean](#) & [GevSCCFGPacketResendDestination](#)

Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.
- [GenApi::IBoolean](#) & [GevSCCFGAllInTransmission](#)

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.
- [GenApi::Integer](#) & [GevSCZoneCount](#)

Description: Reports the number of zones per block transmitted on the selected stream channel.
- [GenApi::Integer](#) & [GevSCZoneDirectionAll](#)

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.
- [GenApi::IBoolean](#) & [GevSCZoneConfigurationLock](#)

Description: Controls whether the selected stream channel multi-zone configuration is locked.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesTransmitted](#)

Description: Reports the number of transmitted PAUSE frames.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesReceived](#)

Description: Reports the number of received PAUSE frames.
- [GenApi::IEnumerationT](#) < [CIConfigurationEnums](#) > & [CIConfiguration](#)

Description: This [Camera](#) Link specific feature describes the configuration used by the camera.
- [GenApi::IEnumerationT](#) < [CITimeSlotsCountEnums](#) > & [CITimeSlotsCount](#)

Description: This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.
- [GenApi::IEnumerationT](#) < [CxpLinkConfigurationStatusEnums](#) > & [CxpLinkConfigurationStatus](#)

Description: This feature indicates the current and active Link configuration used by the Device.
- [GenApi::IEnumerationT](#) < [CxpLinkConfigurationPreferredEnums](#) > & [CxpLinkConfigurationPreferred](#)

Description: Provides the Link configuration that allows the Transmitter Device to operate in its default mode.
- [GenApi::IEnumerationT](#) < [CxpLinkConfigurationEnums](#) > & [CxpLinkConfiguration](#)

Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.
- [GenApi::Integer](#) & [CxpConnectionSelector](#)

Description: Selects the CoaXPress physical connection to control.
- [GenApi::IEnumerationT](#) < [CxpConnectionTestModeEnums](#) > & [CxpConnectionTestMode](#)

Description: Enables the test mode for an individual physical connection of the Device.
- [GenApi::Integer](#) & [CxpConnectionTestErrorCount](#)

Description: Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.
- [GenApi::Integer](#) & [CxpConnectionTestPacketCount](#)

Description: Reports the current count for test packets recieved by the device on the connection selected by CxpConnectionSelector.
- [GenApi::ICommand](#) & [CxpPoCxpAuto](#)

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.
- [GenApi::ICommand](#) & [CxpPoCxpTurnOff](#)

Description: Disable Power over CoaXPress (PoCXP) for the Link.
- [GenApi::ICommand](#) & [CxpPoCxpTripReset](#)

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).
- [GenApi::IEnumerationT](#) < [CxpPoCxpStatusEnums](#) > & [CxpPoCxpStatus](#)

Description: Returns the Power over CoaXPress (PoCXP) status of the Device.
- [GenApi::Integer](#) & [ChunkInferenceFrameId](#)

Description: Returns the frame ID associated with the most recent inference result.
- [GenApi::Integer](#) & [ChunkInferenceResult](#)

Description: Returns the chunk data inference result.
- [GenApi::IFloat](#) & [ChunkInferenceConfidence](#)

Description: Returns the chunk data inference confidence percentage.
- [GenApi::IRegister](#) & [ChunkInferenceBoundingBoxResult](#)

Description: Returns the chunk inference bounding box result data.

Protected Member Functions

- [Camera](#) ()

Additional Inherited Members

15.11.1 Detailed Description

The camera object class.

15.11.2 Constructor & Destructor Documentation

15.11.2.1 `~Camera()`

`~Camera` ()

15.11.2.2 `Camera()`

`Camera` () [protected]

15.11.3 Member Function Documentation

15.11.3.1 `Init()`

`void Init` () [virtual]

Implements [ICameraBase](#).

15.11.4 Member Data Documentation

15.11.4.1 AasRoiEnable

`GenApi::IBoolean& AasRoiEnable`

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the `AutoAlgorithmSelector` feature.

Visibility:

15.11.4.2 AasRoiHeight

`GenApi::IInteger& AasRoiHeight`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the `AutoAlgorithmSelector` feature.

Visibility:

15.11.4.3 AasRoiOffsetX

`GenApi::IInteger& AasRoiOffsetX`

Description:

Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the `AutoAlgorithmSelector` feature.

Visibility:

15.11.4.4 AasRoiOffsetY

`GenApi::IInteger& AasRoiOffsetY`

Description:

Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the `AutoAlgorithmSelector` feature.

Visibility:

15.11.4.5 AasRoiWidth

`GenApi::Integer& AasRoiWidth`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the `AutoAlgorithmSelector` feature.

Visibility:

15.11.4.6 AcquisitionAbort

`GenApi::ICommand& AcquisitionAbort`

Description: Aborts the Acquisition immediately.

This will end the capture without completing the current Frame or waiting on a trigger. If no Acquisition is in progress, the command is ignored. Visibility: Expert

15.11.4.7 AcquisitionArm

`GenApi::ICommand& AcquisitionArm`

Description: Arms the device before an `AcquisitionStart` command.

This optional command validates all the current features for consistency and prepares the device for a fast start of the Acquisition. Visibility: Expert

15.11.4.8 AcquisitionBurstFrameCount

`GenApi::Integer& AcquisitionBurstFrameCount`

Description:

This feature is used only if the `FrameBurstStart` trigger is enabled and the `FrameBurstEnd` trigger is disabled.

Note that the total number of frames captured is also conditioned by `AcquisitionFrameCount` if `AcquisitionMode` is `MultiFrame` and ignored if `AcquisitionMode` is `Single`.

Visibility:

15.11.4.9 AcquisitionFrameCount

`GenApi::Integer& AcquisitionFrameCount`

Description:

Number of images to acquire during a multi frame acquisition.

Visibility:

15.11.4.10 AcquisitionFrameRate

`GenApi::Float& AcquisitionFrameRate`

Description: User controlled acquisition frame rate in Hertz Visibility:

15.11.4.11 AcquisitionFrameRateEnable

`GenApi::Boolean& AcquisitionFrameRateEnable`

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

Visibility:

15.11.4.12 AcquisitionLineRate

`GenApi::Float& AcquisitionLineRate`

Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.

Visibility:

15.11.4.13 AcquisitionMode

`GenApi::EnumerationT<AcquisitionModeEnums>& AcquisitionMode`

Description: Sets the acquisition mode of the device.

Continuous: acquires images continuously. Multi Frame: acquires a specified number of images before stopping acquisition. Single Frame: acquires 1 image before stopping acquisition. Visibility:

15.11.4.14 AcquisitionResultingFrameRate

`GenApi::Float& AcquisitionResultingFrameRate`

Description: Resulting frame rate in Hertz.

If this does not equal the Acquisition Frame Rate it is because the Exposure Time is greater than the frame time.

Visibility:

15.11.4.15 AcquisitionStart

`GenApi::ICommand& AcquisitionStart`

Description: This command starts the acquisition of images.

Visibility:

15.11.4.16 AcquisitionStatus

`GenApi::IBoolean& AcquisitionStatus`

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.

Visibility: Expert

15.11.4.17 AcquisitionStatusSelector

`GenApi::IEnumerationT<AcquisitionStatusSelectorEnums>& AcquisitionStatusSelector`

Description: Selects the internal acquisition signal to read using AcquisitionStatus.

Visibility: Expert

15.11.4.18 AcquisitionStop

`GenApi::ICommand& AcquisitionStop`

Description: This command stops the acquisition of images.

Visibility:

15.11.4.19 ActionDeviceKey

`GenApi::IInteger& ActionDeviceKey`

Description: Provides the device key that allows the device to check the validity of action commands.

The device internal assertion of an action signal is only authorized if the ActionDeviceKey and the action device key value in the protocol message are equal. Visibility: Guru

15.11.4.20 ActionGroupKey

`GenApi::IInteger& ActionGroupKey`

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

15.11.4.21 ActionGroupMask

`GenApi::Integer& ActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

15.11.4.22 ActionQueueSize

`GenApi::Integer& ActionQueueSize`

Description: Indicates the size of the scheduled action commands queue.

This number represents the maximum number of scheduled action commands that can be pending at a given point in time. Visibility: Guru

15.11.4.23 ActionSelector

`GenApi::Integer& ActionSelector`

Description: Selects to which Action Signal further Action settings apply.

Visibility: Guru

15.11.4.24 ActionUnconditionalMode

`GenApi::EnumerationT<ActionUnconditionalModeEnums>& ActionUnconditionalMode`

Description: Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

Visibility: Guru

15.11.4.25 AdaptiveCompressionEnable

`GenApi::Boolean& AdaptiveCompressionEnable`

Description: Controls whether lossless compression adapts to the image content.

If disabled, a fixed encoding table is used. Visibility:

15.11.4.26 AdcBitDepth

`GenApi::IEnumerationT<AdcBitDepthEnums>& AdcBitDepth`

Description:

Selects which ADC bit depth to use.

A higher ADC bit depth results in better image quality but slower maximum frame rate.

Visibility:

15.11.4.27 aPAUSEMACCtrlFramesReceived

`GenApi::IInteger& aPAUSEMACCtrlFramesReceived`

Description: Reports the number of received PAUSE frames.

Visibility: Guru

15.11.4.28 aPAUSEMACCtrlFramesTransmitted

`GenApi::IInteger& aPAUSEMACCtrlFramesTransmitted`

Description: Reports the number of transmitted PAUSE frames.

Visibility: Guru

15.11.4.29 AutoAlgorithmSelector

`GenApi::IEnumerationT<AutoAlgorithmSelectorEnums>& AutoAlgorithmSelector`

Description:

Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

Visibility:

15.11.4.30 AutoExposureControlLoopDamping

`GenApi::IFloat& AutoExposureControlLoopDamping`

Description:

It controls how fast the exposure and gain get settled.

If the value is too small, it may cause the system to be unstable. Range is from 0.0 to 1.0. Default = 0.2.

Visibility:

15.11.4.31 AutoExposureControlPriority

`GenApi::IEnumerationT<AutoExposureControlPriorityEnums>& AutoExposureControlPriority`

Description:

Selects whether to adjust gain or exposure first.

When gain priority is selected, the camera fixes the gain to 0 dB, and the exposure is adjusted according to the target grey level. If the maximum exposure is reached before the target grey level is hit, the gain starts to change to meet the target. This mode is used to have the minimum noise. When exposure priority is selected, the camera sets the exposure to a small value (default is 5 ms). The gain is adjusted according to the target grey level. If maximum gain is reached before the target grey level is hit, the exposure starts to change to meet the target. This mode is used to capture fast motion.

Visibility:

15.11.4.32 AutoExposureEVCompensation

`GenApi::IFloat& AutoExposureEVCompensation`

Description:

The EV compensation value used in the exposure compensation.

This allows you to adjust the resultant image intensity with one control. A positive value makes the image brighter. A negative value makes the image darker. Range from -3 to 3 with a step of 1/3. Default = 0.

Visibility:

15.11.4.33 AutoExposureExposureTimeLowerLimit

`GenApi::IFloat& AutoExposureExposureTimeLowerLimit`

Description:

The smallest exposure time that auto exposure can set.

Visibility:

15.11.4.34 AutoExposureExposureTimeUpperLimit

`GenApi::IFloat& AutoExposureExposureTimeUpperLimit`

Description:

The largest exposure time that auto exposure can set.

Visibility:

15.11.4.35 AutoExposureGainLowerLimit

`GenApi::IFloat& AutoExposureGainLowerLimit`

Description:

The smallest gain that auto exposure can set.

Visibility:

15.11.4.36 AutoExposureGainUpperLimit

`GenApi::IFloat& AutoExposureGainUpperLimit`

Description:

The largest gain that auto exposure can set.

Visibility:

15.11.4.37 AutoExposureGreyValueLowerLimit

`GenApi::IFloat& AutoExposureGreyValueLowerLimit`

Description:

The lowest value in percentage that the target mean may reach.

Visibility:

15.11.4.38 AutoExposureGreyValueUpperLimit

`GenApi::IFloat& AutoExposureGreyValueUpperLimit`

Description:

The highest value in percentage that the target mean may reach.

Visibility:

15.11.4.39 AutoExposureLightingMode

`GenApi::IEnumerationT<AutoExposureLightingModeEnums>& AutoExposureLightingMode`

Description:

Selects a lighting mode: Backlight, Frontlight or Normal (default).

a. Backlight compensation: used when a strong light is coming from the back of the object. b. Frontlight compensation: used when a strong light is shining in the front of the object while the background is dark. c. Normal lighting: used when the object is not under backlight or frontlight conditions. When normal lighting is selected, metering modes are available.

Visibility:

15.11.4.40 AutoExposureMeteringMode

`GenApi::IEnumerationT<AutoExposureMeteringModeEnums>& AutoExposureMeteringMode`

Description:

Selects a metering mode: average, spot, or partial metering.

a. Average: Measures the light from the entire scene uniformly to determine the final exposure value. Every portion of the exposed area has the same contribution. b. Spot: Measures a small area (about 3%) in the center of the scene while the rest of the scene is ignored. This mode is used when the scene has a high contrast and the object of interest is relatively small. c. Partial: Measures the light from a larger area (about 11%) in the center of the scene. This mode is used when very dark or bright regions appear at the edge of the frame. Note: Metering mode is available only when Lighting Mode Selector is Normal.

Visibility:

15.11.4.41 AutoExposureTargetGreyValue

`GenApi::IFloat& AutoExposureTargetGreyValue`

Description:

This is the user-specified target grey level (image mean) to apply to the current image.

Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

15.11.4.42 AutoExposureTargetGreyValueAuto

`GenApi::IEnumerationT<AutoExposureTargetGreyValueAutoEnums>& AutoExposureTargetGreyValueAuto`

Description:

This indicates whether the target image grey level is automatically set by the camera or manually set by the user.

Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

15.11.4.43 BalanceRatio

`GenApi::IFloat& BalanceRatio`

Description:

Controls the balance ratio of the selected color relative to green.

Used for white balancing.

Visibility:

15.11.4.44 BalanceRatioSelector

`GenApi::IEnumerationT<BalanceRatioSelectorEnums>& BalanceRatioSelector`

Description:

Selects a balance ratio to configure once a balance ratio control has been selected.

Visibility:

15.11.4.45 BalanceWhiteAuto

`GenApi::IEnumerationT<BalanceWhiteAutoEnums>& BalanceWhiteAuto`

Description:

White Balance compensates for color shifts caused by different lighting conditions.

It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

Visibility:

15.11.4.46 BalanceWhiteAutoDamping

`GenApi::IFloat& BalanceWhiteAutoDamping`

Description:

Controls how quickly 'BalanceWhiteAuto' adjusts the values for Red and Blue BalanceRatio in response to changing conditions.

Higher damping means the changes are more gradual.

Visibility:

15.11.4.47 BalanceWhiteAutoLowerLimit

`GenApi::IFloat& BalanceWhiteAutoLowerLimit`

Description:

Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.

Visibility:

15.11.4.48 BalanceWhiteAutoProfile

`GenApi::IEnumerationT<BalanceWhiteAutoProfileEnums>& BalanceWhiteAutoProfile`

Description: Selects the profile used by BalanceWhiteAuto.

Visibility:

15.11.4.49 BalanceWhiteAutoUpperLimit

`GenApi::IFloat& BalanceWhiteAutoUpperLimit`

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

Visibility:

15.11.4.50 BinningHorizontal

`GenApi::Integer& BinningHorizontal`

Description:

Number of horizontal photo-sensitive cells to combine together.

This reduces the horizontal resolution (width) of the image. A value of 1 indicates that no horizontal binning is performed by the camera. This value must be 1 for decimation to be active.

Visibility:

15.11.4.51 BinningHorizontalMode

`GenApi::EnumerationT<BinningHorizontalModeEnums>& BinningHorizontalMode`

Description: Visibility:

15.11.4.52 BinningSelector

`GenApi::EnumerationT<BinningSelectorEnums>& BinningSelector`

Description:

Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

Visibility:

15.11.4.53 BinningVertical

`GenApi::Integer& BinningVertical`

Description:

Number of vertical photo-sensitive cells to combine together.

This reduces the vertical resolution (height) of the image. A value of 1 indicates that no vertical binning is performed by the camera. This value must be 1 for decimation to be active.

Visibility:

15.11.4.54 BinningVerticalMode

`GenApi::IEnumerationT<BinningVerticalModeEnums>& BinningVerticalMode`

Description: Visibility:

15.11.4.55 BlackLevel

`GenApi::IFloat& BlackLevel`

Description:

Controls the offset of the video signal in percent.

Visibility:

15.11.4.56 BlackLevelAuto

`GenApi::IEnumerationT<BlackLevelAutoEnums>& BlackLevelAuto`

Description: Controls the mode for automatic black level adjustment.

The exact algorithm used to implement this adjustment is device-specific. Visibility: Expert

15.11.4.57 BlackLevelAutoBalance

`GenApi::IEnumerationT<BlackLevelAutoBalanceEnums>& BlackLevelAutoBalance`

Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.

The black level coefficients of each channel are adjusted so they are matched. Visibility: Expert

15.11.4.58 BlackLevelClampingEnable

`GenApi::IBoolean& BlackLevelClampingEnable`

Description:

Enable the black level auto clamping feature which performing dark current compensation.

Visibility:

15.11.4.59 BlackLevelRaw

`GenApi::Integer& BlackLevelRaw`

Description:

Controls the offset of the video signal in camera specific units.

Visibility:

15.11.4.60 BlackLevelSelector

`GenApi::EnumerationT<BlackLevelSelectorEnums>& BlackLevelSelector`

Description:

Selects which black level to control.

Only All can be set by the user. Analog and Digital are read-only.

Visibility:

15.11.4.61 ChunkBlackLevel

`GenApi::Float& ChunkBlackLevel`

Description: Returns the black level used to capture the image.

Visibility:

15.11.4.62 ChunkBlackLevelSelector

`GenApi::EnumerationT<ChunkBlackLevelSelectorEnums>& ChunkBlackLevelSelector`

Description: Selects which black level to retrieve Visibility:

15.11.4.63 ChunkCompressionMode

`GenApi::Integer& ChunkCompressionMode`

Description: Returns the compression mode of the last image payload.

Visibility:

15.11.4.64 ChunkCompressionRatio

`GenApi::IFloat& ChunkCompressionRatio`

Description: Returns the compression ratio of the last image payload.

Visibility:

15.11.4.65 ChunkCounterSelector

`GenApi::IEnumerationT<ChunkCounterSelectorEnums>& ChunkCounterSelector`

Description: Selects which counter to retrieve data from.

Visibility: Expert

15.11.4.66 ChunkCounterValue

`GenApi::IInteger& ChunkCounterValue`

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

15.11.4.67 ChunkCRC

`GenApi::IInteger& ChunkCRC`

Description: Returns the CRC of the image payload.

Visibility:

15.11.4.68 ChunkEnable

`GenApi::IBoolean& ChunkEnable`

Description: Enables the inclusion of the selected Chunk data in the payload of the image.

Visibility:

15.11.4.69 ChunkEncoderSelector

`GenApi::IEnumerationT<ChunkEncoderSelectorEnums>& ChunkEncoderSelector`

Description: Selects which Encoder to retrieve data from.

Visibility: Expert

15.11.4.70 ChunkEncoderStatus

`GenApi::IEnumerationT<ChunkEncoderStatusEnums>& ChunkEncoderStatus`

Description: Returns the motion status of the selected encoder.

Visibility: Expert

15.11.4.71 ChunkEncoderValue

`GenApi::IInteger& ChunkEncoderValue`

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.

Visibility: Expert

15.11.4.72 ChunkExposureEndLineStatusAll

`GenApi::IInteger& ChunkExposureEndLineStatusAll`

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

15.11.4.73 ChunkExposureTime

`GenApi::IFloat& ChunkExposureTime`

Description: Returns the exposure time used to capture the image.

Visibility:

15.11.4.74 ChunkExposureTimeSelector

`GenApi::IEnumerationT<ChunkExposureTimeSelectorEnums>& ChunkExposureTimeSelector`

Description: Selects which exposure time is read by the ChunkExposureTime feature.

Visibility: Expert

15.11.4.75 ChunkFrameID

`GenApi::IInteger& ChunkFrameID`

Description: Returns the image frame ID.

Visibility:

15.11.4.76 ChunkGain

`GenApi::IFloat& ChunkGain`

Description: Returns the gain used to capture the image.

Visibility:

15.11.4.77 ChunkGainSelector

`GenApi::IEnumerationT<ChunkGainSelectorEnums>& ChunkGainSelector`

Description: Selects which gain to retrieve Visibility:

15.11.4.78 ChunkHeight

`GenApi::IInteger& ChunkHeight`

Description: Returns the height of the image included in the payload.

Visibility:

15.11.4.79 ChunkImage

`GenApi::IInteger& ChunkImage`

Description: Returns the image payload.

Visibility:

15.11.4.80 ChunkImageComponent

`GenApi::IEnumerationT<ChunkImageComponentEnums>& ChunkImageComponent`

Description: Returns the component of the payload image.

This can be used to identify the image component of a generic part in a multipart transfer. Visibility: Expert

15.11.4.81 ChunkInferenceBoundingBoxResult

`GenApi::IRegister& ChunkInferenceBoundingBoxResult`

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

15.11.4.82 ChunkInferenceConfidence

`GenApi::IFloat& ChunkInferenceConfidence`

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

15.11.4.83 ChunkInferenceFrameId

`GenApi::IInteger& ChunkInferenceFrameId`

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

15.11.4.84 ChunkInferenceResult

`GenApi::IInteger& ChunkInferenceResult`

Description: Returns the chunk data inference result.

Visibility: Expert

15.11.4.85 ChunkLinePitch

`GenApi::IInteger& ChunkLinePitch`

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

15.11.4.86 ChunkLineStatusAll

`GenApi::IInteger& ChunkLineStatusAll`

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

15.11.4.87 ChunkModeActive

`GenApi::IBoolean& ChunkModeActive`

Description: Activates the inclusion of Chunk data in the payload of the image.

Visibility:

15.11.4.88 ChunkOffsetX

`GenApi::Integer& ChunkOffsetX`

Description: Returns the Offset X of the image included in the payload.

Visibility:

15.11.4.89 ChunkOffsetY

`GenApi::Integer& ChunkOffsetY`

Description: Returns the Offset Y of the image included in the payload.

Visibility:

15.11.4.90 ChunkPartSelector

`GenApi::Integer& ChunkPartSelector`

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

15.11.4.91 ChunkPixelDynamicRangeMax

`GenApi::Integer& ChunkPixelDynamicRangeMax`

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

15.11.4.92 ChunkPixelDynamicRangeMin

`GenApi::Integer& ChunkPixelDynamicRangeMin`

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

15.11.4.93 ChunkPixelFormat

`GenApi::EnumerationT<ChunkPixelFormatEnums>& ChunkPixelFormat`

Description: Format of the pixel provided by the camera Visibility:

15.11.4.94 ChunkRegionID

`GenApi::IEnumerationT<ChunkRegionIDEnums>& ChunkRegionID`

Description: Returns the identifier of Region that the image comes from.

Visibility: Expert

15.11.4.95 ChunkScan3dAxisMax

`GenApi::IFloat& ChunkScan3dAxisMax`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.11.4.96 ChunkScan3dAxisMin

`GenApi::IFloat& ChunkScan3dAxisMin`

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.11.4.97 ChunkScan3dCoordinateOffset

`GenApi::IFloat& ChunkScan3dCoordinateOffset`

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.11.4.98 ChunkScan3dCoordinateReferenceSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateReferenceSelectorEnums>& ChunkScan3dCoordinateReferenceSelector`

Description: Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

Visibility: Expert

15.11.4.99 ChunkScan3dCoordinateReferenceValue

`GenApi::IFloat& ChunkScan3dCoordinateReferenceValue`

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.

Visibility: Expert

15.11.4.100 ChunkScan3dCoordinateScale

`GenApi::IFloat& ChunkScan3dCoordinateScale`

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.11.4.101 ChunkScan3dCoordinateSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateSelectorEnums>& ChunkScan3dCoordinateSelector`

Description: Selects which Coordinate to retrieve data from.

Visibility: Expert

15.11.4.102 ChunkScan3dCoordinateSystem

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemEnums>& ChunkScan3dCoordinateSystem`

Description: Returns the Coordinate [System](#) of the image included in the payload.

Visibility: Expert

15.11.4.103 ChunkScan3dCoordinateSystemReference

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemReferenceEnums>& ChunkScan3dCoordinate↵
SystemReference`

Description: Returns the Coordinate [System](#) Position of the image included in the payload.

Visibility: Expert

15.11.4.104 ChunkScan3dCoordinateTransformSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateTransformSelectorEnums>& ChunkScan3dCoordinate↵
TransformSelector`

Description: Selector for transform values.

Visibility: Expert

15.11.4.105 ChunkScan3dDistanceUnit

`GenApi::IEnumerationT<ChunkScan3dDistanceUnitEnums>& ChunkScan3dDistanceUnit`

Description: Returns the Distance Unit of the payload image.

Visibility: Expert

15.11.4.106 ChunkScan3dInvalidDataFlag

`GenApi::IBoolean& ChunkScan3dInvalidDataFlag`

Description: Returns if a specific non-valid data flag is used in the data stream.

Visibility: Expert

15.11.4.107 ChunkScan3dInvalidDataValue

`GenApi::IFloat& ChunkScan3dInvalidDataValue`

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

15.11.4.108 ChunkScan3dOutputMode

`GenApi::IEnumerationT<ChunkScan3dOutputModeEnums>& ChunkScan3dOutputMode`

Description: Returns the Calibrated Mode of the payload image.

Visibility: Expert

15.11.4.109 ChunkScan3dTransformValue

`GenApi::IFloat& ChunkScan3dTransformValue`

Description: Returns the transform value.

Visibility: Expert

15.11.4.110 ChunkScanLineSelector

`GenApi::IInteger& ChunkScanLineSelector`

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

15.11.4.111 ChunkSelector

`GenApi::IEnumerationT<ChunkSelectorEnums>& ChunkSelector`

Description: Selects which chunk data to enable or disable.

Visibility:

15.11.4.112 ChunkSequencerSetActive

`GenApi::Integer& ChunkSequencerSetActive`

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

15.11.4.113 ChunkSerialData

`GenApi::String& ChunkSerialData`

Description: Returns the serial data that was received.

Visibility:

15.11.4.114 ChunkSerialDataLength

`GenApi::Integer& ChunkSerialDataLength`

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

15.11.4.115 ChunkSerialReceiveOverflow

`GenApi::Boolean& ChunkSerialReceiveOverflow`

Description: Returns the status of the chunk serial receive overflow.

Visibility:

15.11.4.116 ChunkSourceID

`GenApi::EnumerationT<ChunkSourceIDEnums>& ChunkSourceID`

Description: Returns the identifier of Source that the image comes from.

Visibility: Expert

15.11.4.117 ChunkStreamChannelID

`GenApi::Integer& ChunkStreamChannelID`

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

15.11.4.118 ChunkTimerSelector

`GenApi::IEnumerationT<ChunkTimerSelectorEnums>& ChunkTimerSelector`

Description: Selects which Timer to retrieve data from.

Visibility: Expert

15.11.4.119 ChunkTimerValue

`GenApi::IFloat& ChunkTimerValue`

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

15.11.4.120 ChunkTimestamp

`GenApi::IInteger& ChunkTimestamp`

Description: Returns the Timestamp of the image.

Visibility:

15.11.4.121 ChunkTimestampLatchValue

`GenApi::IInteger& ChunkTimestampLatchValue`

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

15.11.4.122 ChunkTransferBlockID

`GenApi::IInteger& ChunkTransferBlockID`

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

15.11.4.123 ChunkTransferQueueCurrentBlockCount

`GenApi::IInteger& ChunkTransferQueueCurrentBlockCount`

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

15.11.4.124 ChunkTransferStreamID

`GenApi::IEnumerationT<ChunkTransferStreamIDEnums>& ChunkTransferStreamID`

Description: Returns identifier of the stream that generated this block.

Visibility: Expert

15.11.4.125 ChunkWidth

`GenApi::IInteger& ChunkWidth`

Description: Returns the width of the image included in the payload.

Visibility:

15.11.4.126 ClConfiguration

`GenApi::IEnumerationT<ClConfigurationEnums>& ClConfiguration`

Description: This [Camera](#) Link specific feature describes the configuration used by the camera.

It helps especially when a camera is capable of operation in a non-standard configuration, and when the features [PixelSize](#), [SensorDigitizationTaps](#), and [DeviceTapGeometry](#) do not provide enough information for interpretation of the image data provided by the camera. Visibility: Beginner

15.11.4.127 ClTimeSlotsCount

`GenApi::IEnumerationT<ClTimeSlotsCountEnums>& ClTimeSlotsCount`

Description: This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Visibility: Expert

15.11.4.128 ColorTransformationEnable

`GenApi::IBoolean& ColorTransformationEnable`

Description:

Enables/disables the color transform selected with [ColorTransformationSelector](#).

For RGB to YUV this is read-only. Enabling/disabling RGB to YUV can only be done by changing pixel format.

Visibility:

15.11.4.129 ColorTransformationSelector

`GenApi::IEnumerationT<ColorTransformationSelectorEnums>& ColorTransformationSelector`

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

Visibility:

15.11.4.130 ColorTransformationValue

`GenApi::IFloat& ColorTransformationValue`

Description:

Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

Visibility:

15.11.4.131 ColorTransformationValueSelector

`GenApi::IEnumerationT<ColorTransformationValueSelectorEnums>& ColorTransformationValueSelector`

Description:

Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module

Visibility:

15.11.4.132 CompressionRatio

`GenApi::IFloat& CompressionRatio`

Description: Reports the ratio between the uncompressed image size and compressed image size.

Visibility:

15.11.4.133 CompressionSaturationPriority

`GenApi::IEnumerationT<CompressionSaturationPriorityEnums>& CompressionSaturationPriority`

Description: When FrameRate is enabled, camera drops frames if datarate is saturated.

If FrameRate is disabled, camera adjusts the framerate to match the maximum achievable datarate. Visibility:

15.11.4.134 CounterDelay

`GenApi::Integer& CounterDelay`

Description: Sets the delay (or number of events) before the CounterStart event is generated.

Visibility:

15.11.4.135 CounterDuration

`GenApi::Integer& CounterDuration`

Description: Sets the duration (or number of events) before the CounterEnd event is generated.

Visibility:

15.11.4.136 CounterEventActivation

`GenApi::EnumerationT<CounterEventActivationEnums>& CounterEventActivation`

Description: Selects the activation mode of the event to increment the Counter.

Visibility:

15.11.4.137 CounterEventSource

`GenApi::EnumerationT<CounterEventSourceEnums>& CounterEventSource`

Description: Selects the event that will increment the counter Visibility:

15.11.4.138 CounterReset

`GenApi::Command& CounterReset`

Description: Does a software reset of the selected Counter and starts it.

The counter starts counting events immediately after the reset unless a Counter trigger is active. CounterReset can be used to reset the Counter independently from the CounterResetSource. To disable the counter temporarily, set CounterEventSource to Off. Visibility: Expert

15.11.4.139 CounterResetActivation

`GenApi::EnumerationT<CounterResetActivationEnums>& CounterResetActivation`

Description: Selects the Activation mode of the Counter Reset Source signal.

Visibility:

15.11.4.140 CounterResetSource

`GenApi::IEnumerationT<CounterResetSourceEnums>& CounterResetSource`

Description: Selects the signal that will be the source to reset the counter.

Visibility:

15.11.4.141 CounterSelector

`GenApi::IEnumerationT<CounterSelectorEnums>& CounterSelector`

Description: Selects which counter to configure Visibility:

15.11.4.142 CounterStatus

`GenApi::IEnumerationT<CounterStatusEnums>& CounterStatus`

Description: Returns the current status of the counter.

Visibility:

15.11.4.143 CounterTriggerActivation

`GenApi::IEnumerationT<CounterTriggerActivationEnums>& CounterTriggerActivation`

Description: Selects the activation mode of the trigger to start the counter.

Visibility:

15.11.4.144 CounterTriggerSource

`GenApi::IEnumerationT<CounterTriggerSourceEnums>& CounterTriggerSource`

Description: Selects the source of the trigger to start the counter Visibility:

15.11.4.145 CounterValue

`GenApi::IInteger& CounterValue`

Description: Current counter value Visibility:

15.11.4.146 CounterValueAtReset

`GenApi::Integer& CounterValueAtReset`

Description: Value of the selected Counter when it was reset by a trigger.

Visibility:

15.11.4.147 CxpConnectionSelector

`GenApi::Integer& CxpConnectionSelector`

Description: Selects the CoaXPress physical connection to control.

Visibility: Expert

15.11.4.148 CxpConnectionTestErrorCount

`GenApi::Integer& CxpConnectionTestErrorCount`

Description: Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

15.11.4.149 CxpConnectionTestMode

`GenApi::EnumerationT<CxpConnectionTestModeEnums>& CxpConnectionTestMode`

Description: Enables the test mode for an individual physical connection of the Device.

Visibility: Expert

15.11.4.150 CxpConnectionTestPacketCount

`GenApi::Integer& CxpConnectionTestPacketCount`

Description: Reports the current count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

15.11.4.151 CxpLinkConfiguration

`GenApi::EnumerationT<CxpLinkConfigurationEnums>& CxpLinkConfiguration`

Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.

In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus. Visibility: Beginner

15.11.4.152 CxpLinkConfigurationPreferred

`GenApi::IEnumerationT<CxpLinkConfigurationPreferredEnums>& CxpLinkConfigurationPreferred`

Description: Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

Visibility: Expert

15.11.4.153 CxpLinkConfigurationStatus

`GenApi::IEnumerationT<CxpLinkConfigurationStatusEnums>& CxpLinkConfigurationStatus`

Description: This feature indicates the current and active Link configuration used by the Device.

Visibility: Beginner

15.11.4.154 CxpPoCxpAuto

`GenApi::ICommand& CxpPoCxpAuto`

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

15.11.4.155 CxpPoCxpStatus

`GenApi::IEnumerationT<CxpPoCxpStatusEnums>& CxpPoCxpStatus`

Description: Returns the Power over CoaXPress (PoCXP) status of the Device.

Visibility: Expert

15.11.4.156 CxpPoCxpTripReset

`GenApi::ICommand& CxpPoCxpTripReset`

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

Visibility: Expert

15.11.4.157 CxpPoCxpTurnOff

`GenApi::ICommand& CxpPoCxpTurnOff`

Description: Disable Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

15.11.4.158 DecimationHorizontal

`GenApi::Integer& DecimationHorizontal`

Description:

Horizontal decimation of the image.

This reduces the horizontal resolution (width) of the image by only retaining a single pixel within a window whose size is the decimation factor specified here. A value of 1 indicates that no horizontal decimation is performed by the camera. This value must be 1 for binning to be active.

Visibility:

15.11.4.159 DecimationHorizontalMode

`GenApi::EnumerationT<DecimationHorizontalModeEnums>& DecimationHorizontalMode`

Description:

The mode used to reduce the horizontal resolution when DecimationHorizontal is used.

The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Visibility:

15.11.4.160 DecimationSelector

`GenApi::EnumerationT<DecimationSelectorEnums>& DecimationSelector`

Description: Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

Visibility:

15.11.4.161 DecimationVertical

`GenApi::Integer& DecimationVertical`

Description:

Vertical decimation of the image.

This reduces the vertical resolution (height) of the image by only retaining a single pixel within a window whose size is the decimation factor specified here. A value of 1 indicates that no vertical decimation is performed by the camera. This value must be 1 for binning to be active.

Visibility:

15.11.4.162 DecimationVerticalMode

`GenApi::IEnumerationT<DecimationVerticalModeEnums>& DecimationVerticalMode`

Description:

The mode used to reduce the vertical resolution when DecimationVertical is used.

The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Visibility:

15.11.4.163 DefectCorrectionMode

`GenApi::IEnumerationT<DefectCorrectionModeEnums>& DefectCorrectionMode`

Description: Controls the method used for replacing defective pixels.

Visibility:

15.11.4.164 DefectCorrectStaticEnable

`GenApi::IBoolean& DefectCorrectStaticEnable`

Description: Enables/Disables table-based defective pixel correction.

Visibility:

15.11.4.165 DefectTableApply

`GenApi::ICommand& DefectTableApply`

Description: Applies the current defect table, so that any changes made affect images captured by the camera.

This writes the table to volatile memory, so changes to the table are lost if the camera loses power. To save the table to non-volatile memory, use DefectTableSave.

Visibility:

15.11.4.166 DefectTableCoordinateX

`GenApi::IInteger& DefectTableCoordinateX`

Description:

Returns the X coordinate of the defective pixel at DefectTableIndex within the defective pixel table.

Changes made do not take effect in captured images until the command DefectTableApply is written.

Visibility:

15.11.4.167 DefectTableCoordinateY

`GenApi::Integer& DefectTableCoordinateY`

Description:

Returns the Y coordinate of the defective pixel at DefectTableIndex within the defective pixel table.

Changes made do not take effect in captured images until the command DefectTableApply is written.

Visibility:

15.11.4.168 DefectTableFactoryRestore

`GenApi::ICommand& DefectTableFactoryRestore`

Description: Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.

This permanently overwrites any changes made to the defect table.

Visibility:

15.11.4.169 DefectTableIndex

`GenApi::Integer& DefectTableIndex`

Description:

Controls the offset of the element to access in the defective pixel location table.

Visibility:

15.11.4.170 DefectTablePixelCount

`GenApi::Integer& DefectTablePixelCount`

Description:

The number of defective pixel locations in the current table.

Visibility:

15.11.4.171 DefectTableSave

`GenApi::ICommand& DefectTableSave`

Description: Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.

This overwrites the existing defective pixel table. The new table is loaded whenever the camera powers up.

Visibility:

15.11.4.172 Deinterlacing

`GenApi::IEnumerationT<DeinterlacingEnums>& Deinterlacing`

Description: Controls how the device performs de-interlacing.

Visibility: Beginner

15.11.4.173 DeviceCharacterSet

`GenApi::IEnumerationT<DeviceCharacterSetEnums>& DeviceCharacterSet`

Description:

Character set used by the strings of the device's bootstrap registers.

Visibility:

15.11.4.174 DeviceClockFrequency

`GenApi::IFloat& DeviceClockFrequency`

Description: Returns the frequency of the selected Clock.

Visibility: Expert

15.11.4.175 DeviceClockSelector

`GenApi::IEnumerationT<DeviceClockSelectorEnums>& DeviceClockSelector`

Description: Selects the clock frequency to access from the device.

Visibility: Expert

15.11.4.176 DeviceConnectionSelector

`GenApi::Integer& DeviceConnectionSelector`

Description: Selects which Connection of the device to control.

Visibility: Beginner

15.11.4.177 DeviceConnectionSpeed

`GenApi::Integer& DeviceConnectionSpeed`

Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.

15.11.4.178 DeviceConnectionStatus

`GenApi::EnumerationT<DeviceConnectionStatusEnums>& DeviceConnectionStatus`

Description: Indicates the status of the specified Connection.

Visibility: Expert

15.11.4.179 DeviceEventChannelCount

`GenApi::Integer& DeviceEventChannelCount`

Description:

Indicates the number of event channels supported by the device.

Visibility:

15.11.4.180 DeviceFamilyName

`GenApi::IString& DeviceFamilyName`

Description: Identifier of the product family of the device.

Visibility: Beginner

15.11.4.181 DeviceFeaturePersistenceEnd

`GenApi::ICommand& DeviceFeaturePersistenceEnd`

Description: Indicate to the device the end of feature persistence.

Visibility: Guru

15.11.4.182 DeviceFeaturePersistenceStart

`GenApi::ICommand& DeviceFeaturePersistenceStart`

Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.

Visibility: Guru

15.11.4.183 DeviceFirmwareVersion

`GenApi::IString& DeviceFirmwareVersion`

Description: Version of the firmware on the device.

Visibility:

15.11.4.184 DeviceGenCPVersionMajor

`GenApi::IInteger& DeviceGenCPVersionMajor`

Description: Major version of the GenCP protocol supported by the device.

Visibility: Beginner

15.11.4.185 DeviceGenCPVersionMinor

`GenApi::IInteger& DeviceGenCPVersionMinor`

Description: Minor version of the GenCP protocol supported by the device.

Visibility: Beginner

15.11.4.186 DeviceID

`GenApi::IString& DeviceID`

Description: Device identifier (serial number).

Visibility:

15.11.4.187 DeviceIndicatorMode

`GenApi::IEnumerationT<DeviceIndicatorModeEnums>& DeviceIndicatorMode`

Description: Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

Visibility:

15.11.4.188 DeviceLinkBandwidthReserve

`GenApi::IFloat& DeviceLinkBandwidthReserve`

Description:

Percentage of streamed data bandwidth reserved for packet resend.

Visibility:

15.11.4.189 DeviceLinkCommandTimeout

`GenApi::IFloat& DeviceLinkCommandTimeout`

Description: Indicates the command timeout of the specified Link.

This corresponds to the maximum response time of the device for a command sent on that link. Visibility: Guru

15.11.4.190 DeviceLinkConnectionCount

`GenApi::IInteger& DeviceLinkConnectionCount`

Description: Returns the number of physical connection of the device used by a particular Link.

Visibility: Beginner

15.11.4.191 DeviceLinkCurrentThroughput

`GenApi::IInteger& DeviceLinkCurrentThroughput`

Description: Current bandwidth of streamed data.

Visibility:

15.11.4.192 DeviceLinkHeartbeatMode

`GenApi::IEnumerationT<DeviceLinkHeartbeatModeEnums>& DeviceLinkHeartbeatMode`

Description: Activate or deactivate the Link's heartbeat.

Visibility: Expert

15.11.4.193 DeviceLinkHeartbeatTimeout

`GenApi::IFloat& DeviceLinkHeartbeatTimeout`

Description: Controls the current heartbeat timeout of the specific Link.

Visibility: Guru

15.11.4.194 DeviceLinkSelector

`GenApi::Integer& DeviceLinkSelector`

Description: Selects which Link of the device to control.

Visibility: Beginner

15.11.4.195 DeviceLinkSpeed

`GenApi::Integer& DeviceLinkSpeed`

Description:

Indicates the speed of transmission negotiated on the specified Link.

(Bps)

Visibility:

15.11.4.196 DeviceLinkThroughputLimit

`GenApi::Integer& DeviceLinkThroughputLimit`

Description:

Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.

If necessary, delays will be uniformly inserted between transport layer packets in order to control the peak bandwidth.

Visibility:

15.11.4.197 DeviceLinkThroughputLimitMode

`GenApi::EnumerationT<DeviceLinkThroughputLimitModeEnums>& DeviceLinkThroughputLimitMode`

Description: Controls if the DeviceLinkThroughputLimit is active.

When disabled, lower level TL specific features are expected to control the throughput. When enabled, Device↔LinkThroughputLimit controls the overall throughput. Visibility: Expert

15.11.4.198 DeviceManifestEntrySelector

`GenApi::Integer& DeviceManifestEntrySelector`

Description: Selects the manifest entry to reference.

Visibility: Guru

15.11.4.199 DeviceManifestPrimaryURL

`GenApi::IString& DeviceManifestPrimaryURL`

Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

15.11.4.200 DeviceManifestSchemaMajorVersion

`GenApi::Integer& DeviceManifestSchemaMajorVersion`

Description: Indicates the major version number of the schema file of the selected manifest entry.

Visibility: Guru

15.11.4.201 DeviceManifestSchemaMinorVersion

`GenApi::Integer& DeviceManifestSchemaMinorVersion`

Description: Indicates the minor version number of the schema file of the selected manifest entry.

Visibility: Guru

15.11.4.202 DeviceManifestSecondaryURL

`GenApi::IString& DeviceManifestSecondaryURL`

Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

15.11.4.203 DeviceManifestXMLMajorVersion

`GenApi::Integer& DeviceManifestXMLMajorVersion`

Description: Indicates the major version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

15.11.4.204 DeviceManifestXMLMinorVersion

`GenApi::Integer& DeviceManifestXMLMinorVersion`

Description: Indicates the minor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

15.11.4.205 DeviceManifestXMLSubMinorVersion

`GenApi::Integer& DeviceManifestXMLSubMinorVersion`

Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

15.11.4.206 DeviceManufacturerInfo

`GenApi::IString& DeviceManufacturerInfo`

Description: Manufacturer information about the device.

Visibility:

15.11.4.207 DeviceMaxThroughput

`GenApi::Integer& DeviceMaxThroughput`

Description:

Maximum bandwidth of the data that can be streamed out of the device.

This can be used to estimate if the physical connection(s) can sustain transfer of free-running images from the camera at its maximum speed.

Visibility:

15.11.4.208 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Model of the device.

Visibility:

15.11.4.209 DevicePowerSupplySelector

`GenApi::IEnumerationT<DevicePowerSupplySelectorEnums>& DevicePowerSupplySelector`

Description:

Selects the power supply source to control or read.

Visibility:

15.11.4.210 DeviceRegistersCheck

`GenApi::ICommand& DeviceRegistersCheck`

Description: Perform the validation of the current register set for consistency.

This will update the DeviceRegistersValid flag. Visibility: Expert

15.11.4.211 DeviceRegistersEndianness

`GenApi::IEnumerationT<DeviceRegistersEndiannessEnums>& DeviceRegistersEndianness`

Description: Endianness of the registers of the device.

Visibility:

15.11.4.212 DeviceRegistersStreamingEnd

`GenApi::ICommand& DeviceRegistersStreamingEnd`

Description: Announce the end of registers streaming.

This will do a register set validation for consistency and activate it. This will also update the DeviceRegistersValid flag. Visibility: Guru

15.11.4.213 DeviceRegistersStreamingStart

`GenApi::ICommand& DeviceRegistersStreamingStart`

Description: Prepare the device for registers streaming without checking for consistency.

Visibility: Guru

15.11.4.214 DeviceRegistersValid

`GenApi::IBoolean& DeviceRegistersValid`

Description: Returns if the current register set is valid and consistent.

Visibility: Expert

15.11.4.215 DeviceReset

```
GenApi::ICommand& DeviceReset
```

Description: This is a command that immediately resets and reboots the device.

Visibility:

15.11.4.216 DeviceScanType

```
GenApi::IEnumerationT<DeviceScanTypeEnums>& DeviceScanType
```

Description: Scan type of the sensor of the device.

Visibility:

15.11.4.217 DeviceSerialNumber

```
GenApi::IString& DeviceSerialNumber
```

Description:

Device's serial number.

This string is a unique identifier of the device.

Visibility:

15.11.4.218 DeviceSerialPortBaudRate

```
GenApi::IEnumerationT<DeviceSerialPortBaudRateEnums>& DeviceSerialPortBaudRate
```

Description: This feature controls the baud rate used by the selected serial port.

Visibility: Expert

15.11.4.219 DeviceSerialPortSelector

```
GenApi::IEnumerationT<DeviceSerialPortSelectorEnums>& DeviceSerialPortSelector
```

Description: Selects which serial port of the device to control.

Visibility: Expert

15.11.4.220 DeviceSFNCVersionMajor

`GenApi::Integer& DeviceSFNCVersionMajor`

Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

15.11.4.221 DeviceSFNCVersionMinor

`GenApi::Integer& DeviceSFNCVersionMinor`

Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

15.11.4.222 DeviceSFNCVersionSubMinor

`GenApi::Integer& DeviceSFNCVersionSubMinor`

Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

15.11.4.223 DeviceStreamChannelCount

`GenApi::Integer& DeviceStreamChannelCount`

Description:

Indicates the number of streaming channels supported by the device.

Visibility:

15.11.4.224 DeviceStreamChannelEndianness

`GenApi::EnumerationT<DeviceStreamChannelEndiannessEnums>& DeviceStreamChannelEndianness`

Description: Endianness of multi-byte pixel data for this stream.

Visibility: Guru

15.11.4.225 DeviceStreamChannelLink

`GenApi::Integer& DeviceStreamChannelLink`

Description: Index of device's Link to use for streaming the specified stream channel.

Visibility: Guru

15.11.4.226 DeviceStreamChannelPacketSize

`GenApi::Integer& DeviceStreamChannelPacketSize`

Description: Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.

Visibility: Expert

15.11.4.227 DeviceStreamChannelSelector

`GenApi::Integer& DeviceStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility: Expert

15.11.4.228 DeviceStreamChannelType

`GenApi::EnumerationT<DeviceStreamChannelTypeEnums>& DeviceStreamChannelType`

Description: Reports the type of the stream channel.

Visibility: Guru

15.11.4.229 DeviceTapGeometry

`GenApi::EnumerationT<DeviceTapGeometryEnums>& DeviceTapGeometry`

Description: This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

Visibility: Expert

15.11.4.230 DeviceTemperature

`GenApi::Float& DeviceTemperature`

Description: Device temperature in degrees Celsius (C).

Visibility:

15.11.4.231 DeviceTemperatureSelector

`GenApi::IEnumerationT<DeviceTemperatureSelectorEnums>& DeviceTemperatureSelector`

Description:

Selects the location within the device, where the temperature will be measured.

Visibility:

15.11.4.232 DeviceTLType

`GenApi::IEnumerationT<DeviceTLTypeEnums>& DeviceTLType`

Description: Transport Layer type of the device.

Visibility:

15.11.4.233 DeviceTLVersionMajor

`GenApi::IInteger& DeviceTLVersionMajor`

Description:

Major version of the Transport Layer of the device.

Visibility:

15.11.4.234 DeviceTLVersionMinor

`GenApi::IInteger& DeviceTLVersionMinor`

Description:

Minor version of the Transport Layer of the device.

Visibility:

15.11.4.235 DeviceTLVersionSubMinor

`GenApi::IInteger& DeviceTLVersionSubMinor`

Description: Sub minor version of the Transport Layer of the device.

Visibility: Beginner

15.11.4.236 DeviceType

```
GenApi::IEnumerationT<DeviceTypeEnums>& DeviceType
```

Description: Returns the device type.

Visibility: Guru

15.11.4.237 DeviceUptime

```
GenApi::IInteger& DeviceUptime
```

Description: Total time since the device was powered up in seconds.

Visibility:

15.11.4.238 DeviceUserID

```
GenApi::IString& DeviceUserID
```

Description: User-programmable device identifier.

Visibility:

15.11.4.239 DeviceVendorName

```
GenApi::IString& DeviceVendorName
```

Description: Name of the manufacturer of the device.

Visibility:

15.11.4.240 DeviceVersion

```
GenApi::IString& DeviceVersion
```

Description: Version of the device.

Visibility:

15.11.4.241 EncoderDivider

```
GenApi::IInteger& EncoderDivider
```

Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.

Visibility: Expert

15.11.4.242 EncoderMode

`GenApi::IEnumerationT<EncoderModeEnums>& EncoderMode`

Description: Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.

Visibility: Expert

15.11.4.243 EncoderOutputMode

`GenApi::IEnumerationT<EncoderOutputModeEnums>& EncoderOutputMode`

Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.

Visibility: Expert

15.11.4.244 EncoderReset

`GenApi::ICommand& EncoderReset`

Description: Does a software reset of the selected Encoder and starts it.

The Encoder starts counting events immediately after the reset. EncoderReset can be used to reset the Encoder independently from the EncoderResetSource. Visibility: Expert

15.11.4.245 EncoderResetActivation

`GenApi::IEnumerationT<EncoderResetActivationEnums>& EncoderResetActivation`

Description: Selects the Activation mode of the Encoder Reset Source signal.

Visibility: Expert

15.11.4.246 EncoderResetSource

`GenApi::IEnumerationT<EncoderResetSourceEnums>& EncoderResetSource`

Description: Selects the signals that will be the source to reset the Encoder.

Visibility: Expert

15.11.4.247 EncoderSelector

`GenApi::IEnumerationT<EncoderSelectorEnums>& EncoderSelector`

Description: Selects which Encoder to configure.

Visibility: Expert

15.11.4.248 EncoderSourceA

`GenApi::IEnumerationT<EncoderSourceAEnums>& EncoderSourceA`

Description: Selects the signal which will be the source of the A input of the Encoder.

Visibility: Expert

15.11.4.249 EncoderSourceB

`GenApi::IEnumerationT<EncoderSourceBEnums>& EncoderSourceB`

Description: Selects the signal which will be the source of the B input of the Encoder.

Visibility: Expert

15.11.4.250 EncoderStatus

`GenApi::IEnumerationT<EncoderStatusEnums>& EncoderStatus`

Description: Returns the motion status of the encoder.

Visibility: Expert

15.11.4.251 EncoderTimeout

`GenApi::IFloat& EncoderTimeout`

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

Visibility: Expert

15.11.4.252 EncoderValue

`GenApi::IInteger& EncoderValue`

Description: Reads or writes the current value of the position counter of the selected Encoder.

Visibility: Expert

15.11.4.253 EncoderValueAtReset

`GenApi::IInteger& EncoderValueAtReset`

Description: Reads the value of the of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

Visibility: Expert

15.11.4.254 EnumerationCount

`GenApi::Integer& EnumerationCount`

Description: Number of enumerations since uptime.

Visibility:

15.11.4.255 EventAcquisitionEnd

`GenApi::Integer& EventAcquisitionEnd`

Description: Returns the unique Identifier of the Acquisition End type of Event.

Visibility: Expert

15.11.4.256 EventAcquisitionEndFrameID

`GenApi::Integer& EventAcquisitionEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End Event.

Visibility: Expert

15.11.4.257 EventAcquisitionEndTimestamp

`GenApi::Integer& EventAcquisitionEndTimestamp`

Description: Returns the Timestamp of the Acquisition End Event.

Visibility: Expert

15.11.4.258 EventAcquisitionError

`GenApi::Integer& EventAcquisitionError`

Description: Returns the unique Identifier of the Acquisition Error type of Event.

Visibility: Expert

15.11.4.259 EventAcquisitionErrorFrameID

`GenApi::Integer& EventAcquisitionErrorFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.

Visibility: Expert

15.11.4.260 EventAcquisitionErrorTimestamp

`GenApi::Integer& EventAcquisitionErrorTimestamp`

Description: Returns the Timestamp of the Acquisition Error Event.

Visibility: Expert

15.11.4.261 EventAcquisitionStart

`GenApi::Integer& EventAcquisitionStart`

Description: Returns the unique Identifier of the Acquisition Start type of Event.

Visibility: Expert

15.11.4.262 EventAcquisitionStartFrameID

`GenApi::Integer& EventAcquisitionStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start Event.

Visibility: Expert

15.11.4.263 EventAcquisitionStartTimestamp

`GenApi::Integer& EventAcquisitionStartTimestamp`

Description: Returns the Timestamp of the Acquisition Start Event.

Visibility: Expert

15.11.4.264 EventAcquisitionTransferEnd

`GenApi::Integer& EventAcquisitionTransferEnd`

Description: Returns the unique Identifier of the Acquisition Transfer End type of Event.

Visibility: Expert

15.11.4.265 EventAcquisitionTransferEndFrameID

`GenApi::Integer& EventAcquisitionTransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End Event.

Visibility: Expert

15.11.4.266 EventAcquisitionTransferEndTimestamp

`GenApi::Integer& EventAcquisitionTransferEndTimestamp`

Description: Returns the Timestamp of the Acquisition Transfer End Event.

Visibility: Expert

15.11.4.267 EventAcquisitionTransferStart

`GenApi::Integer& EventAcquisitionTransferStart`

Description: Returns the unique Identifier of the Acquisition Transfer Start type of Event.

Visibility: Expert

15.11.4.268 EventAcquisitionTransferStartFrameID

`GenApi::Integer& EventAcquisitionTransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start Event.

Visibility: Expert

15.11.4.269 EventAcquisitionTransferStartTimestamp

`GenApi::Integer& EventAcquisitionTransferStartTimestamp`

Description: Returns the Timestamp of the Acquisition Transfer Start Event.

Visibility: Expert

15.11.4.270 EventAcquisitionTrigger

`GenApi::Integer& EventAcquisitionTrigger`

Description: Returns the unique Identifier of the Acquisition Trigger type of Event.

Visibility: Expert

15.11.4.271 EventAcquisitionTriggerFrameID

`GenApi::Integer& EventAcquisitionTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger Event.

Visibility: Expert

15.11.4.272 EventAcquisitionTriggerTimestamp

`GenApi::Integer& EventAcquisitionTriggerTimestamp`

Description: Returns the Timestamp of the Acquisition Trigger Event.

Visibility: Expert

15.11.4.273 EventActionLate

`GenApi::Integer& EventActionLate`

Description: Returns the unique Identifier of the Action Late type of Event.

Visibility: Expert

15.11.4.274 EventActionLateFrameID

`GenApi::Integer& EventActionLateFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late Event.

Visibility: Expert

15.11.4.275 EventActionLateTimestamp

`GenApi::Integer& EventActionLateTimestamp`

Description: Returns the Timestamp of the Action Late Event.

Visibility: Expert

15.11.4.276 EventCounter0End

`GenApi::Integer& EventCounter0End`

Description: Returns the unique Identifier of the Counter 0 End type of Event.

Visibility: Expert

15.11.4.277 EventCounter0EndFrameID

`GenApi::Integer& EventCounter0EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End Event.

Visibility: Expert

15.11.4.278 EventCounter0EndTimestamp

`GenApi::Integer& EventCounter0EndTimestamp`

Description: Returns the Timestamp of the Counter 0 End Event.

Visibility: Expert

15.11.4.279 EventCounter0Start

`GenApi::Integer& EventCounter0Start`

Description: Returns the unique Identifier of the Counter 0 Start type of Event.

Visibility: Expert

15.11.4.280 EventCounter0StartFrameID

`GenApi::Integer& EventCounter0StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.

Visibility: Expert

15.11.4.281 EventCounter0StartTimestamp

`GenApi::Integer& EventCounter0StartTimestamp`

Description: Returns the Timestamp of the Counter 0 Start Event.

Visibility: Expert

15.11.4.282 EventCounter1End

`GenApi::Integer& EventCounter1End`

Description: Returns the unique Identifier of the Counter 1 End type of Event.

Visibility: Expert

15.11.4.283 EventCounter1EndFrameID

`GenApi::Integer& EventCounter1EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End Event.

Visibility: Expert

15.11.4.284 EventCounter1EndTimestamp

`GenApi::Integer& EventCounter1EndTimestamp`

Description: Returns the Timestamp of the Counter 1 End Event.

Visibility: Expert

15.11.4.285 EventCounter1Start

`GenApi::Integer& EventCounter1Start`

Description: Returns the unique Identifier of the Counter 1 Start type of Event.

Visibility: Expert

15.11.4.286 EventCounter1StartFrameID

`GenApi::Integer& EventCounter1StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start Event.

Visibility: Expert

15.11.4.287 EventCounter1StartTimestamp

`GenApi::Integer& EventCounter1StartTimestamp`

Description: Returns the Timestamp of the Counter 1 Start Event.

Visibility: Expert

15.11.4.288 EventEncoder0Restarted

`GenApi::Integer& EventEncoder0Restarted`

Description: Returns the unique Identifier of the Encoder 0 Restarted type of Event.

Visibility: Expert

15.11.4.289 EventEncoder0RestartedFrameID

`GenApi::Integer& EventEncoder0RestartedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted Event.

Visibility: Expert

15.11.4.290 EventEncoder0RestartedTimestamp

`GenApi::Integer& EventEncoder0RestartedTimestamp`

Description: Returns the Timestamp of the Encoder 0 Restarted Event.

Visibility: Expert

15.11.4.291 EventEncoder0Stopped

`GenApi::Integer& EventEncoder0Stopped`

Description: Returns the unique Identifier of the Encoder 0 Stopped type of Event.

Visibility: Expert

15.11.4.292 EventEncoder0StoppedFrameID

`GenApi::Integer& EventEncoder0StoppedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped Event.

Visibility: Expert

15.11.4.293 EventEncoder0StoppedTimestamp

`GenApi::Integer& EventEncoder0StoppedTimestamp`

Description: Returns the Timestamp of the Encoder 0 Stopped Event.

Visibility: Expert

15.11.4.294 EventEncoder1Restarted

`GenApi::Integer& EventEncoder1Restarted`

Description: Returns the unique Identifier of the Encoder 1 Restarted type of Event.

Visibility: Expert

15.11.4.295 EventEncoder1RestartedFrameID

`GenApi::Integer& EventEncoder1RestartedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted Event.

Visibility: Expert

15.11.4.296 EventEncoder1RestartedTimestamp

`GenApi::Integer& EventEncoder1RestartedTimestamp`

Description: Returns the Timestamp of the Encoder 1 Restarted Event.

Visibility: Expert

15.11.4.297 EventEncoder1Stopped

`GenApi::Integer& EventEncoder1Stopped`

Description: Returns the unique Identifier of the Encoder 1 Stopped type of Event.

Visibility: Expert

15.11.4.298 EventEncoder1StoppedFrameID

`GenApi::Integer& EventEncoder1StoppedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped Event.

Visibility: Expert

15.11.4.299 EventEncoder1StoppedTimestamp

`GenApi::Integer& EventEncoder1StoppedTimestamp`

Description: Returns the Timestamp of the Encoder 1 Stopped Event.

Visibility: Expert

15.11.4.300 EventError

`GenApi::Integer& EventError`

Description: Returns the unique identifier of the Error type of Event.

Visibility:

15.11.4.301 EventErrorCode

`GenApi::Integer& EventErrorCode`

Description: Returns the error code for the error that happened Visibility:

15.11.4.302 EventErrorFrameID

`GenApi::Integer& EventErrorFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Error Event.

Visibility:

15.11.4.303 EventErrorTimestamp

`GenApi::Integer& EventErrorTimestamp`

Description: Returns the Timestamp of the Error Event.

Visibility:

15.11.4.304 EventExposureEnd

`GenApi::Integer& EventExposureEnd`

Description: Returns the unique identifier of the Exposure End type of Event.

Visibility:

15.11.4.305 EventExposureEndFrameID

`GenApi::Integer& EventExposureEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End Event.

Visibility:

15.11.4.306 EventExposureEndTimestamp

`GenApi::Integer& EventExposureEndTimestamp`

Description: Returns the Timestamp of the Exposure End Event.

Visibility:

15.11.4.307 EventExposureStart

`GenApi::Integer& EventExposureStart`

Description: Returns the unique Identifier of the Exposure Start type of Event.

Visibility: Expert

15.11.4.308 EventExposureStartFrameID

`GenApi::Integer& EventExposureStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start Event.

Visibility: Expert

15.11.4.309 EventExposureStartTimestamp

`GenApi::Integer& EventExposureStartTimestamp`

Description: Returns the Timestamp of the Exposure Start Event.

Visibility: Expert

15.11.4.310 EventFrameBurstEnd

`GenApi::Integer& EventFrameBurstEnd`

Description: Returns the unique Identifier of the Frame Burst End type of Event.

Visibility: Expert

15.11.4.311 EventFrameBurstEndFrameID

`GenApi::Integer& EventFrameBurstEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End Event.

Visibility: Expert

15.11.4.312 EventFrameBurstEndTimestamp

`GenApi::Integer& EventFrameBurstEndTimestamp`

Description: Returns the Timestamp of the Frame Burst End Event.

Visibility: Expert

15.11.4.313 EventFrameBurstStart

`GenApi::Integer& EventFrameBurstStart`

Description: Returns the unique Identifier of the Frame Burst Start type of Event.

Visibility: Expert

15.11.4.314 EventFrameBurstStartFrameID

`GenApi::Integer& EventFrameBurstStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start Event.

Visibility: Expert

15.11.4.315 EventFrameBurstStartTimestamp

`GenApi::Integer& EventFrameBurstStartTimestamp`

Description: Returns the Timestamp of the Frame Burst Start Event.

Visibility: Expert

15.11.4.316 EventFrameEnd

`GenApi::Integer& EventFrameEnd`

Description: Returns the unique Identifier of the Frame End type of Event.

Visibility: Expert

15.11.4.317 EventFrameEndFrameID

`GenApi::Integer& EventFrameEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End Event.

Visibility: Expert

15.11.4.318 EventFrameEndTimestamp

`GenApi::Integer& EventFrameEndTimestamp`

Description: Returns the Timestamp of the Frame End Event.

Visibility: Expert

15.11.4.319 EventFrameStart

`GenApi::Integer& EventFrameStart`

Description: Returns the unique Identifier of the Frame Start type of Event.

Visibility: Expert

15.11.4.320 EventFrameStartFrameID

`GenApi::Integer& EventFrameStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start Event.

Visibility: Expert

15.11.4.321 EventFrameStartTimestamp

`GenApi::Integer& EventFrameStartTimestamp`

Description: Returns the Timestamp of the Frame Start Event.

Visibility: Expert

15.11.4.322 EventFrameTransferEnd

`GenApi::Integer& EventFrameTransferEnd`

Description: Returns the unique Identifier of the Frame Transfer End type of Event.

Visibility: Expert

15.11.4.323 EventFrameTransferEndFrameID

`GenApi::Integer& EventFrameTransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End Event.

Visibility: Expert

15.11.4.324 EventFrameTransferEndTimestamp

`GenApi::Integer& EventFrameTransferEndTimestamp`

Description: Returns the Timestamp of the Frame Transfer End Event.

Visibility: Expert

15.11.4.325 EventFrameTransferStart

`GenApi::Integer& EventFrameTransferStart`

Description: Returns the unique Identifier of the Frame Transfer Start type of Event.

Visibility: Expert

15.11.4.326 EventFrameTransferStartFrameID

`GenApi::Integer& EventFrameTransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start Event.

Visibility: Expert

15.11.4.327 EventFrameTransferStartTimestamp

`GenApi::Integer& EventFrameTransferStartTimestamp`

Description: Returns the Timestamp of the Frame Transfer Start Event.

Visibility: Expert

15.11.4.328 EventFrameTrigger

`GenApi::Integer& EventFrameTrigger`

Description: Returns the unique Identifier of the FrameTrigger type of Event.

It can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type event received. Visibility: Expert

15.11.4.329 EventFrameTriggerFrameID

`GenApi::Integer& EventFrameTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

Visibility: Expert

15.11.4.330 EventFrameTriggerTimestamp

`GenApi::Integer& EventFrameTriggerTimestamp`

Description: Returns the Timestamp of the FrameTrigger Event.

It can be used to determine precisely when the event occurred. Visibility: Expert

15.11.4.331 EventLine0AnyEdge

`GenApi::Integer& EventLine0AnyEdge`

Description: Returns the unique Identifier of the Line 0 Any Edge type of Event.

Visibility: Expert

15.11.4.332 EventLine0AnyEdgeFrameID

`GenApi::Integer& EventLine0AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge Event.

Visibility: Expert

15.11.4.333 EventLine0AnyEdgeTimestamp

`GenApi::Integer& EventLine0AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Any Edge Event.

Visibility: Expert

15.11.4.334 EventLine0FallingEdge

`GenApi::Integer& EventLine0FallingEdge`

Description: Returns the unique Identifier of the Line 0 Falling Edge type of Event.

Visibility: Expert

15.11.4.335 EventLine0FallingEdgeFrameID

`GenApi::Integer& EventLine0FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.

Visibility: Expert

15.11.4.336 EventLine0FallingEdgeTimestamp

`GenApi::Integer& EventLine0FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Falling Edge Event.

Visibility: Expert

15.11.4.337 EventLine0RisingEdge

`GenApi::Integer& EventLine0RisingEdge`

Description: Returns the unique Identifier of the Line 0 Rising Edge type of Event.

Visibility: Expert

15.11.4.338 EventLine0RisingEdgeFrameID

`GenApi::Integer& EventLine0RisingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event.

Visibility: Expert

15.11.4.339 EventLine0RisingEdgeTimestamp

`GenApi::Integer& EventLine0RisingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Rising Edge Event.

Visibility: Expert

15.11.4.340 EventLine1AnyEdge

`GenApi::Integer& EventLine1AnyEdge`

Description: Returns the unique Identifier of the Line 1 Any Edge type of Event.

Visibility: Expert

15.11.4.341 EventLine1AnyEdgeFrameID

`GenApi::Integer& EventLine1AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge Event.

Visibility: Expert

15.11.4.342 EventLine1AnyEdgeTimestamp

`GenApi::Integer& EventLine1AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Any Edge Event.

Visibility: Expert

15.11.4.343 EventLine1FallingEdge

`GenApi::Integer& EventLine1FallingEdge`

Description: Returns the unique Identifier of the Line 1 Falling Edge type of Event.

Visibility: Expert

15.11.4.344 EventLine1FallingEdgeFrameID

`GenApi::Integer& EventLine1FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge Event.

Visibility: Expert

15.11.4.345 EventLine1FallingEdgeTimestamp

`GenApi::Integer& EventLine1FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Falling Edge Event.

Visibility: Expert

15.11.4.346 EventLine1RisingEdge

`GenApi::Integer& EventLine1RisingEdge`

Description: Returns the unique Identifier of the Line 1 Rising Edge type of Event.

Visibility: Expert

15.11.4.347 EventLine1RisingEdgeFrameID

`GenApi::Integer& EventLine1RisingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event.

Visibility: Expert

15.11.4.348 EventLine1RisingEdgeTimestamp

`GenApi::Integer& EventLine1RisingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Rising Edge Event.

Visibility: Expert

15.11.4.349 EventLinkSpeedChange

`GenApi::Integer& EventLinkSpeedChange`

Description: Returns the unique Identifier of the Link Speed Change type of Event.

Visibility: Expert

15.11.4.350 EventLinkSpeedChangeFrameID

`GenApi::Integer& EventLinkSpeedChangeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change Event.

Visibility: Expert

15.11.4.351 EventLinkSpeedChangeTimestamp

`GenApi::Integer& EventLinkSpeedChangeTimestamp`

Description: Returns the Timestamp of the Link Speed Change Event.

Visibility: Expert

15.11.4.352 EventLinkTrigger0

`GenApi::Integer& EventLinkTrigger0`

Description: Returns the unique Identifier of the Link Trigger 0 type of Event.

Visibility: Expert

15.11.4.353 EventLinkTrigger0FrameID

`GenApi::Integer& EventLinkTrigger0FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 Event.

Visibility: Expert

15.11.4.354 EventLinkTrigger0Timestamp

`GenApi::Integer& EventLinkTrigger0Timestamp`

Description: Returns the Timestamp of the Link Trigger 0 Event.

Visibility: Expert

15.11.4.355 EventLinkTrigger1

`GenApi::Integer& EventLinkTrigger1`

Description: Returns the unique Identifier of the Link Trigger 1 type of Event.

Visibility: Expert

15.11.4.356 EventLinkTrigger1FrameID

`GenApi::Integer& EventLinkTrigger1FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 Event.

Visibility: Expert

15.11.4.357 EventLinkTrigger1Timestamp

`GenApi::Integer& EventLinkTrigger1Timestamp`

Description: Returns the Timestamp of the Link Trigger 1 Event.

Visibility: Expert

15.11.4.358 EventNotification

`GenApi::EnumerationT<EventNotificationEnums>& EventNotification`

Description: Enables/Disables the selected event.

Visibility:

15.11.4.359 EventSelector

`GenApi::EnumerationT<EventSelectorEnums>& EventSelector`

Description: Selects which Event to enable or disable.

Visibility:

15.11.4.360 EventSequencerSetChange

`GenApi::Integer& EventSequencerSetChange`

Description: Returns the unique Identifier of the Sequencer Set Change type of Event.

Visibility: Expert

15.11.4.361 EventSequencerSetChangeFrameID

`GenApi::Integer& EventSequencerSetChangeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.

Visibility: Expert

15.11.4.362 EventSequencerSetChangeTimestamp

`GenApi::Integer& EventSequencerSetChangeTimestamp`

Description: Returns the Timestamp of the Sequencer Set Change Event.

Visibility: Expert

15.11.4.363 EventSerialData

`GenApi::IString& EventSerialData`

Description: Returns the serial data that was received.

Visibility:

15.11.4.364 EventSerialDataLength

`GenApi::Integer& EventSerialDataLength`

Description: Returns the length of the received serial data that was included in the event payload.

Visibility:

15.11.4.365 EventSerialPortReceive

`GenApi::Integer& EventSerialPortReceive`

Description: Returns the unique identifier of the Serial Port Receive type of Event.

Visibility:

15.11.4.366 EventSerialPortReceiveTimestamp

`GenApi::Integer& EventSerialPortReceiveTimestamp`

Description: Returns the Timestamp of the Serial Port Receive Event.

Visibility:

15.11.4.367 EventSerialReceiveOverflow

`GenApi::Boolean& EventSerialReceiveOverflow`

Description: Returns the status of the event serial receive overflow.

Visibility:

15.11.4.368 EventStream0TransferBlockEnd

`GenApi::Integer& EventStream0TransferBlockEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.

Visibility: Expert

15.11.4.369 EventStream0TransferBlockEndFrameID

`GenApi::Integer& EventStream0TransferBlockEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.

Visibility: Expert

15.11.4.370 EventStream0TransferBlockEndTimestamp

`GenApi::Integer& EventStream0TransferBlockEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.

Visibility: Expert

15.11.4.371 EventStream0TransferBlockStart

`GenApi::Integer& EventStream0TransferBlockStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of Event.

Visibility: Expert

15.11.4.372 EventStream0TransferBlockStartFrameID

`GenApi::Integer& EventStream0TransferBlockStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.

Visibility: Expert

15.11.4.373 EventStream0TransferBlockStartTimestamp

`GenApi::Integer& EventStream0TransferBlockStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block Start Event.

Visibility: Expert

15.11.4.374 EventStream0TransferBlockTrigger

`GenApi::Integer& EventStream0TransferBlockTrigger`

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.

Visibility: Expert

15.11.4.375 EventStream0TransferBlockTriggerFrameID

`GenApi::Integer& EventStream0TransferBlockTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.

Visibility: Expert

15.11.4.376 EventStream0TransferBlockTriggerTimestamp

`GenApi::Integer& EventStream0TransferBlockTriggerTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.

Visibility: Expert

15.11.4.377 EventStream0TransferBurstEnd

`GenApi::Integer& EventStream0TransferBurstEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.

Visibility: Expert

15.11.4.378 EventStream0TransferBurstEndFrameID

`GenApi::Integer& EventStream0TransferBurstEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.

Visibility: Expert

15.11.4.379 EventStream0TransferBurstEndTimestamp

`GenApi::Integer& EventStream0TransferBurstEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.

Visibility: Expert

15.11.4.380 EventStream0TransferBurstStart

`GenApi::Integer& EventStream0TransferBurstStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.

Visibility: Expert

15.11.4.381 EventStream0TransferBurstStartFrameID

`GenApi::Integer& EventStream0TransferBurstStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.

Visibility: Expert

15.11.4.382 EventStream0TransferBurstStartTimestamp

`GenApi::Integer& EventStream0TransferBurstStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.

Visibility: Expert

15.11.4.383 EventStream0TransferEnd

`GenApi::Integer& EventStream0TransferEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer End type of Event.

Visibility: Expert

15.11.4.384 EventStream0TransferEndFrameID

`GenApi::Integer& EventStream0TransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End Event.

Visibility: Expert

15.11.4.385 EventStream0TransferEndTimestamp

`GenApi::Integer& EventStream0TransferEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer End Event.

Visibility: Expert

15.11.4.386 EventStream0TransferOverflow

`GenApi::Integer& EventStream0TransferOverflow`

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.

Visibility: Expert

15.11.4.387 EventStream0TransferOverflowFrameID

`GenApi::Integer& EventStream0TransferOverflowFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.

Visibility: Expert

15.11.4.388 EventStream0TransferOverflowTimestamp

`GenApi::Integer& EventStream0TransferOverflowTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.

Visibility: Expert

15.11.4.389 EventStream0TransferPause

`GenApi::Integer& EventStream0TransferPause`

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of Event.

Visibility: Expert

15.11.4.390 EventStream0TransferPauseFrameID

`GenApi::Integer& EventStream0TransferPauseFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause Event.

Visibility: Expert

15.11.4.391 EventStream0TransferPauseTimestamp

`GenApi::Integer& EventStream0TransferPauseTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Pause Event.

Visibility: Expert

15.11.4.392 EventStream0TransferResume

`GenApi::Integer& EventStream0TransferResume`

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of Event.

Visibility: Expert

15.11.4.393 EventStream0TransferResumeFrameID

`GenApi::Integer& EventStream0TransferResumeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume Event.

Visibility: Expert

15.11.4.394 EventStream0TransferResumeTimestamp

`GenApi::Integer& EventStream0TransferResumeTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Resume Event.

Visibility: Expert

15.11.4.395 EventStream0TransferStart

`GenApi::Integer& EventStream0TransferStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of Event.

Visibility: Expert

15.11.4.396 EventStream0TransferStartFrameID

`GenApi::Integer& EventStream0TransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start Event.

Visibility: Expert

15.11.4.397 EventStream0TransferStartTimestamp

`GenApi::Integer& EventStream0TransferStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Start Event.

Visibility: Expert

15.11.4.398 EventTest

`GenApi::Integer& EventTest`

Description: Returns the unique identifier of the Test type of Event.

Visibility:

15.11.4.399 EventTestTimestamp

`GenApi::Integer& EventTestTimestamp`

Description: Returns the Timestamp of the Test Event.

Visibility:

15.11.4.400 EventTimer0End

`GenApi::Integer& EventTimer0End`

Description: Returns the unique Identifier of the Timer 0 End type of Event.

Visibility: Expert

15.11.4.401 EventTimer0EndFrameID

`GenApi::Integer& EventTimer0EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End Event.

Visibility: Expert

15.11.4.402 EventTimer0EndTimestamp

`GenApi::Integer& EventTimer0EndTimestamp`

Description: Returns the Timestamp of the Timer 0 End Event.

Visibility: Expert

15.11.4.403 EventTimer0Start

`GenApi::Integer& EventTimer0Start`

Description: Returns the unique Identifier of the Timer 0 Start type of Event.

Visibility: Expert

15.11.4.404 EventTimer0StartFrameID

`GenApi::Integer& EventTimer0StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start Event.

Visibility: Expert

15.11.4.405 EventTimer0StartTimestamp

`GenApi::Integer& EventTimer0StartTimestamp`

Description: Returns the Timestamp of the Timer 0 Start Event.

Visibility: Expert

15.11.4.406 EventTimer1End

`GenApi::Integer& EventTimer1End`

Description: Returns the unique Identifier of the Timer 1 End type of Event.

Visibility: Expert

15.11.4.407 EventTimer1EndFrameID

`GenApi::Integer& EventTimer1EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End Event.

Visibility: Expert

15.11.4.408 EventTimer1EndTimestamp

`GenApi::Integer& EventTimer1EndTimestamp`

Description: Returns the Timestamp of the Timer 1 End Event.

Visibility: Expert

15.11.4.409 EventTimer1Start

`GenApi::Integer& EventTimer1Start`

Description: Returns the unique Identifier of the Timer 1 Start type of Event.

Visibility: Expert

15.11.4.410 EventTimer1StartFrameID

`GenApi::Integer& EventTimer1StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start Event.

Visibility: Expert

15.11.4.411 EventTimer1StartTimestamp

`GenApi::Integer& EventTimer1StartTimestamp`

Description: Returns the Timestamp of the Timer 1 Start Event.

Visibility: Expert

15.11.4.412 ExposureActiveMode

`GenApi::EnumerationT<ExposureActiveModeEnums>& ExposureActiveMode`

Description: Control sensor active exposure mode.

Visibility:

15.11.4.413 ExposureAuto

`GenApi::EnumerationT<ExposureAutoEnums>& ExposureAuto`

Description: Sets the automatic exposure mode Visibility:

15.11.4.414 ExposureMode

`GenApi::EnumerationT<ExposureModeEnums>& ExposureMode`

Description:

Sets the operation mode of the Exposure.

Visibility:

15.11.4.415 ExposureTime

`GenApi::Float& ExposureTime`

Description:

Exposure time in microseconds when Exposure Mode is Timed.

Visibility:

15.11.4.416 ExposureTimeMode

`GenApi::IEnumerationT<ExposureTimeModeEnums>& ExposureTimeMode`

Description: Sets the configuration mode of the ExposureTime feature.

Visibility: Beginner

15.11.4.417 ExposureTimeSelector

`GenApi::IEnumerationT<ExposureTimeSelectorEnums>& ExposureTimeSelector`

Description: Selects which exposure time is controlled by the ExposureTime feature.

This allows for independent control over the exposure components. Visibility: Beginner

15.11.4.418 FactoryReset

`GenApi::ICommand& FactoryReset`

Description: Returns all user tables to factory default Visibility:

15.11.4.419 FileAccessBuffer

`GenApi::IRegister& FileAccessBuffer`

Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.

Visibility: Guru

15.11.4.420 FileAccessLength

`GenApi::IInteger& FileAccessLength`

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

15.11.4.421 FileAccessOffset

`GenApi::IInteger& FileAccessOffset`

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

15.11.4.422 FileOpenMode

`GenApi::IEnumerationT<FileOpenModeEnums>& FileOpenMode`

Description:

The mode of the file when it is opened.

The file can be opened for reading, writing or both. This must be set before opening the file.

Visibility:

15.11.4.423 FileOperationExecute

`GenApi::ICommand& FileOperationExecute`

Description:

This is a command that executes the selected file operation on the selected file.

Visibility:

15.11.4.424 FileOperationResult

`GenApi::Integer& FileOperationResult`

Description: Represents the file operation result.

For Read or Write operations, the number of successfully read/written bytes is returned. Visibility:

15.11.4.425 FileOperationSelector

`GenApi::IEnumerationT<FileOperationSelectorEnums>& FileOperationSelector`

Description:

Sets operation to execute on the selected file when the execute command is given.

Visibility:

15.11.4.426 FileOperationStatus

`GenApi::IEnumerationT<FileOperationStatusEnums>& FileOperationStatus`

Description: Represents the file operation execution status.

Visibility:

15.11.4.427 FileSelector

`GenApi::IEnumerationT<FileSelectorEnums>& FileSelector`

Description:

Selects which file is being operated on.

This must be set before performing any file operations.

Visibility:

15.11.4.428 FileSize

`GenApi::IInteger& FileSize`

Description: Represents the size of the selected file in bytes.

Visibility:

15.11.4.429 Gain

`GenApi::IFloat& Gain`

Description:

Controls the amplification of the video signal in dB.

Visibility:

15.11.4.430 GainAuto

`GenApi::IEnumerationT<GainAutoEnums>& GainAuto`

Description:

Sets the automatic gain mode.

Set to Off for manual control. Set to Once for a single automatic adjustment then return to Off. Set to Continuous for constant adjustment. In automatic modes, the camera adjusts the gain to maximize the dynamic range. Visibility:

15.11.4.431 GainAutoBalance

`GenApi::IEnumerationT<GainAutoBalanceEnums>& GainAutoBalance`

Description: Sets the mode for automatic gain balancing between the sensor color channels or taps.

The gain coefficients of each channel or tap are adjusted so they are matched. Visibility: Beginner

15.11.4.432 GainSelector

`GenApi::IEnumerationT<GainSelectorEnums>& GainSelector`

Description: Selects which gain to control.

The All selection is a total amplification across all channels (or taps).

Visibility:

15.11.4.433 Gamma

`GenApi::IFloat& Gamma`

Description: Controls the gamma correction of pixel intensity.

Visibility:

15.11.4.434 GammaEnable

`GenApi::IBoolean& GammaEnable`

Description: Enables/disables gamma correction.

Visibility:

15.11.4.435 GevActiveLinkCount

`GenApi::IInteger& GevActiveLinkCount`

Description: Indicates the current number of active logical links.

Visibility: Expert

15.11.4.436 GevCCP

`GenApi::IEnumerationT<GevCCPEnums>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility:

15.11.4.437 GevCurrentDefaultGateway

`GenApi::Integer& GevCurrentDefaultGateway`

Description: Reports the default gateway IP address to be used on the given logical link.

Visibility:

15.11.4.438 GevCurrentIPAddress

`GenApi::Integer& GevCurrentIPAddress`

Description: Reports the IP address for the given logical link.

Visibility:

15.11.4.439 GevCurrentIPConfigurationDHCP

`GenApi::Boolean& GevCurrentIPConfigurationDHCP`

Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.

Visibility:

15.11.4.440 GevCurrentIPConfigurationLLA

`GenApi::Boolean& GevCurrentIPConfigurationLLA`

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Visibility:

15.11.4.441 GevCurrentIPConfigurationPersistentIP

`GenApi::Boolean& GevCurrentIPConfigurationPersistentIP`

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.

Visibility:

15.11.4.442 GevCurrentPhysicalLinkConfiguration

`GenApi::EnumerationT<GevCurrentPhysicalLinkConfigurationEnums>& GevCurrentPhysicalLinkConfiguration`

Description: Indicates the current physical link configuration of the device.

Visibility: Expert

15.11.4.443 GevCurrentSubnetMask

`GenApi::Integer& GevCurrentSubnetMask`

Description: Reports the subnet mask of the given logical link.

Visibility:

15.11.4.444 GevDiscoveryAckDelay

`GenApi::Integer& GevDiscoveryAckDelay`

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

Visibility: Expert

15.11.4.445 GevFirstURL

`GenApi::IString& GevFirstURL`

Description: The first choice of URL for the XML device description file.

Visibility:

15.11.4.446 GevGVCPExtendedStatusCodes

`GenApi::IBoolean& GevGVCPExtendedStatusCodes`

Description: Enables the generation of extended status codes.

Visibility: Guru

15.11.4.447 GevGVCPExtendedStatusCodesSelector

`GenApi::IEnumerationT<GevGVCPExtendedStatusCodesSelectorEnums>& GevGVCPExtendedStatusCodesSelector`

Description: Selects the GigE Vision version to control extended status codes for.

Visibility: Guru

15.11.4.448 GevGVCPHeartbeatDisable

`GenApi::IBoolean& GevGVCPHeartbeatDisable`

Description: Disables the GVCP heartbeat.

Visibility:

15.11.4.449 GevGVCPPendingAck

`GenApi::IBoolean & GevGVCPPendingAck`

Description: Enables the generation of PENDING_ACK.

Visibility:

15.11.4.450 GevGVCPPendingTimeout

`GenApi::IInteger & GevGVCPPendingTimeout`

Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.

Visibility:

15.11.4.451 GevGVSPExtendedIDMode

`GenApi::IEnumerationT<GevGVSPExtendedIDModeEnums> & GevGVSPExtendedIDMode`

Description: Enables the extended IDs mode.

Visibility: Expert

15.11.4.452 GevHeartbeatTimeout

`GenApi::IInteger & GevHeartbeatTimeout`

Description: Indicates the current heartbeat timeout in milliseconds.

Visibility:

15.11.4.453 GevIEEE1588

`GenApi::IBoolean & GevIEEE1588`

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

Visibility:

15.11.4.454 GevIEEE1588ClockAccuracy

`GenApi::IEnumerationT<GevIEEE1588ClockAccuracyEnums> & GevIEEE1588ClockAccuracy`

Description: Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

Visibility:

15.11.4.455 GevIEEE1588Mode

`GenApi::IEnumerationT<GevIEEE1588ModeEnums>&   GevIEEE1588Mode`

Description: Provides the mode of the IEEE 1588 clock.

Visibility:

15.11.4.456 GevIEEE1588Status

`GenApi::IEnumerationT<GevIEEE1588StatusEnums>&   GevIEEE1588Status`

Description: Provides the status of the IEEE 1588 clock.

Visibility:

15.11.4.457 GevInterfaceSelector

`GenApi::IInteger&   GevInterfaceSelector`

Description: Selects which logical link to control.

Visibility:

15.11.4.458 GevIPConfigurationStatus

`GenApi::IEnumerationT<GevIPConfigurationStatusEnums>&   GevIPConfigurationStatus`

Description: Reports the current IP configuration status.

Visibility: Beginner

15.11.4.459 GevMACAddress

`GenApi::IInteger&   GevMACAddress`

Description: MAC address of the logical link.

Visibility:

15.11.4.460 GevMCDA

`GenApi::IInteger&   GevMCDA`

Description: Controls the destination IP address of the message channel Visibility:

15.11.4.461 GevMCPHostPort

`GenApi::Integer&   GevMCPHostPort`

Description: The port to which the device must send messages Visibility:

15.11.4.462 GevMCRC

`GenApi::Integer&   GevMCRC`

Description: Indicates the number of retries of the message channel.

Visibility:

15.11.4.463 GevMCSP

`GenApi::Integer&   GevMCSP`

Description: Indicates the source port of the message channel.

Visibility:

15.11.4.464 GevMCTT

`GenApi::Integer&   GevMCTT`

Description: Indicates the transmission timeout of the message channel.

Visibility:

15.11.4.465 GevNumberOfInterfaces

`GenApi::Integer&   GevNumberOfInterfaces`

Description: Indicates the number of physical network interfaces supported by this device.

Visibility:

15.11.4.466 GevPAUSEFrameReception

`GenApi::Boolean&   GevPAUSEFrameReception`

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

Visibility: Expert

15.11.4.467 GevPAUSEFrameTransmission

`GenApi::IBoolean& GevPAUSEFrameTransmission`

Description: Controls whether PAUSE Frames can be generated on the given logical link.

Visibility: Expert

15.11.4.468 GevPersistentDefaultGateway

`GenApi::IInteger& GevPersistentDefaultGateway`

Description: Controls the persistent default gateway for this logical link.

Visibility:

15.11.4.469 GevPersistentIPAddress

`GenApi::IInteger& GevPersistentIPAddress`

Description: Controls the Persistent IP address for this logical link.

Visibility:

15.11.4.470 GevPersistentSubnetMask

`GenApi::IInteger& GevPersistentSubnetMask`

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.

Visibility:

15.11.4.471 GevPhysicalLinkConfiguration

`GenApi::IEnumerationT<GevPhysicalLinkConfigurationEnums>& GevPhysicalLinkConfiguration`

Description: Controls the principal physical link configuration to use on next restart/power-up of the device.

Visibility: Expert

15.11.4.472 GevPrimaryApplicationIPAddress

`GenApi::IInteger& GevPrimaryApplicationIPAddress`

Description: Returns the address of the primary application.

Visibility: Guru

15.11.4.473 GevPrimaryApplicationSocket

`GenApi::Integer&   GevPrimaryApplicationSocket`

Description: Returns the UDP source port of the primary application.

Visibility: Guru

15.11.4.474 GevPrimaryApplicationSwitchoverKey

`GenApi::Integer&   GevPrimaryApplicationSwitchoverKey`

Description: Controls the key to use to authenticate primary application switchover requests.

Visibility: Guru

15.11.4.475 GevSCCFGAllInTransmission

`GenApi::Boolean&   GevSCCFGAllInTransmission`

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

Visibility: Guru

15.11.4.476 GevSCCFGExtendedChunkData

`GenApi::Boolean&   GevSCCFGExtendedChunkData`

Description: Enables cameras to use the extended chunk data payload type for this stream channel.

Visibility:

15.11.4.477 GevSCCFGPacketResendDestination

`GenApi::Boolean&   GevSCCFGPacketResendDestination`

Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.

When True, the source IP address provided in the packet resend command packet is used. When False, the value set in the `GevSCDA[GevStreamChannelSelector]` feature is used. Visibility: Guru

15.11.4.478 GevSCCFGUnconditionalStreaming

`GenApi::Boolean&   GevSCCFGUnconditionalStreaming`

Description: Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).

Visibility:

15.11.4.479 GevSCDA

`GenApi::Integer&   GevSCDA`

Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

Visibility:

15.11.4.480 GevSCPD

`GenApi::Integer&   GevSCPD`

Description: Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.

This can be used as a crude flow-control mechanism if the application or the network infrastructure cannot keep up with the packets coming from the device. Visibility:

15.11.4.481 GevSCPDirection

`GenApi::Integer&   GevSCPDirection`

Description: Transmit or Receive of the channel Visibility:

15.11.4.482 GevSCPHostPort

`GenApi::Integer&   GevSCPHostPort`

Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.

Visibility:

15.11.4.483 GevSCPInterfaceIndex

`GenApi::Integer&   GevSCPInterfaceIndex`

Description: Index of the logical link to use.

Visibility:

15.11.4.484 GevSCPSBigEndian

`GenApi::Boolean&   GevSCPSBigEndian`

Description: Endianess of multi-byte pixel data for this stream.

Visibility:

15.11.4.485 GevSCPSDoNotFragment

`GenApi::IBoolean& GevSCPSDoNotFragment`

Description: The state of this feature is copied into the "do not fragment" bit of the IP header of each stream packet.

Visibility:

15.11.4.486 GevSCPSFireTestPacket

`GenApi::IBoolean& GevSCPSFireTestPacket`

Description: Sends a test packet.

Visibility:

15.11.4.487 GevSCSPacketSize

`GenApi::IInteger& GevSCSPacketSize`

Description: Specifies the stream packet size (in bytes) to send on this channel.

Visibility:

15.11.4.488 GevSCSP

`GenApi::IInteger& GevSCSP`

Description: Indicates the source port of the stream channel.

Visibility:

15.11.4.489 GevSCZoneConfigurationLock

`GenApi::IBoolean& GevSCZoneConfigurationLock`

Description: Controls whether the selected stream channel multi-zone configuration is locked.

When locked, the GVSP transmitter is not allowed to change the number of zones and their direction during block acquisition and transmission. Visibility: Guru

15.11.4.490 GevSCZoneCount

`GenApi::IInteger& GevSCZoneCount`

Description: Reports the number of zones per block transmitted on the selected stream channel.

Visibility: Guru

15.11.4.491 GevSCZoneDirectionAll

`GenApi::Integer&   GevSCZoneDirectionAll`

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.

Visibility: Guru

15.11.4.492 GevSecondURL

`GenApi::String&   GevSecondURL`

Description: The second choice of URL to the XML device description file.

Visibility:

15.11.4.493 GevStreamChannelSelector

`GenApi::Integer&   GevStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility:

15.11.4.494 GevSupportedOption

`GenApi::Boolean&   GevSupportedOption`

Description: Returns if the selected GEV option is supported.

Visibility:

15.11.4.495 GevSupportedOptionSelector

`GenApi::EnumerationT<GevSupportedOptionSelectorEnums>&   GevSupportedOptionSelector`

Description: Selects the GEV option to interrogate for existing support.

Visibility:

15.11.4.496 GevTimestampTickFrequency

`GenApi::Integer&   GevTimestampTickFrequency`

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

Visibility:

15.11.4.497 GuiXmlManifestAddress

`GenApi::Integer& GuiXmlManifestAddress`

Description: Location of the GUI XML manifest table.

Visibility:

15.11.4.498 Height

`GenApi::Integer& Height`

Description:

Height of the image provided by the device (in pixels).

Visibility:

15.11.4.499 HeightMax

`GenApi::Integer& HeightMax`

Description: Maximum height of the image (in pixels).

This dimension is calculated after vertical binning. HeightMax does not take into account the current Region of interest (Height or OffsetY). Visibility:

15.11.4.500 ImageComponentEnable

`GenApi::Boolean& ImageComponentEnable`

Description: Controls if the selected component streaming is active.

Visibility: Beginner

15.11.4.501 ImageComponentSelector

`GenApi::EnumerationT<ImageComponentSelectorEnums>& ImageComponentSelector`

Description: Selects a component to activate data streaming from.

Visibility: Beginner

15.11.4.502 ImageCompressionBitrate

`GenApi::Float& ImageCompressionBitrate`

Description: Control the rate of the produced compressed stream.

Visibility: Expert

15.11.4.503 ImageCompressionJPEGFormatOption

`GenApi::IEnumerationT<ImageCompressionJPEGFormatOptionEnums>& ImageCompressionJPEGFormatOption`

Description: When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

Visibility: Expert

15.11.4.504 ImageCompressionMode

`GenApi::IEnumerationT<ImageCompressionModeEnums>& ImageCompressionMode`

Description: Visibility:

15.11.4.505 ImageCompressionQuality

`GenApi::IInteger& ImageCompressionQuality`

Description: Control the quality of the produced compressed stream.

Visibility: Expert

15.11.4.506 ImageCompressionRateOption

`GenApi::IEnumerationT<ImageCompressionRateOptionEnums>& ImageCompressionRateOption`

Description: Two rate controlling options are offered: fixed bit rate or fixed quality.

The exact implementation to achieve one or the other is vendor-specific. Visibility: Expert

15.11.4.507 IspEnable

`GenApi::IBoolean& IspEnable`

Description:

Controls whether the image processing core is used for optional pixel format mode (i.e. mono).

Visibility:

15.11.4.508 LineFilterWidth

`GenApi::IFloat& LineFilterWidth`

Description: Filter width in microseconds for the selected line and filter combination Visibility:

15.11.4.509 LineFormat

`GenApi::IEnumerationT<LineFormatEnums>& LineFormat`

Description: Displays the current electrical format of the selected physical input or output Line.

Visibility:

15.11.4.510 LineInputFilterSelector

`GenApi::IEnumerationT<LineInputFilterSelectorEnums>& LineInputFilterSelector`

Description: Selects the kind of input filter to configure: Deglitch or Debounce.

Visibility:

15.11.4.511 LineInverter

`GenApi::IBoolean& LineInverter`

Description: Controls the inversion of the signal of the selected input or output line.

Visibility:

15.11.4.512 LineMode

`GenApi::IEnumerationT<LineModeEnums>& LineMode`

Description: Controls if the physical Line is used to Input or Output a signal.

Visibility:

15.11.4.513 LinePitch

`GenApi::IInteger& LinePitch`

Description: Total number of bytes between 2 successive lines.

This feature is used to facilitate alignment of image data. Visibility: Expert

15.11.4.514 LineSelector

`GenApi::IEnumerationT<LineSelectorEnums>& LineSelector`

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:

15.11.4.515 LineSource

`GenApi::IEnumerationT<LineSourceEnums>& LineSource`

Description: Selects which internal acquisition or I/O source signal to output on the selected line.

LineMode must be Output. Visibility:

15.11.4.516 LineStatus

`GenApi::IBoolean& LineStatus`

Description: Returns the current status of the selected input or output Line Visibility:

15.11.4.517 LineStatusAll

`GenApi::IInteger& LineStatusAll`

Description: Returns the current status of all the line status bits in a hexadecimal representation (Line 0 status corresponds to bit 0, Line 1 status with bit 1, etc).

This allows simultaneous reading of all line statuses at once. Visibility:

15.11.4.518 LinkErrorCount

`GenApi::IInteger& LinkErrorCount`

Description: Counts the number of error on the link.

Visibility:

15.11.4.519 LinkUptime

`GenApi::IInteger& LinkUptime`

Description: Time since the last phy negotiation (enumeration).

Visibility:

15.11.4.520 LogicBlockLUTInputActivation

```
GenApi::IEnumerationT<LogicBlockLUTInputActivationEnums>& LogicBlockLUTInputActivation
```

Description: Selects the activation mode of the Logic Input Source signal.

Visibility:

15.11.4.521 LogicBlockLUTInputSelector

```
GenApi::IEnumerationT<LogicBlockLUTInputSelectorEnums>& LogicBlockLUTInputSelector
```

Description: Controls which LogicBlockLUT Input Source & Activation to access.

Visibility:

15.11.4.522 LogicBlockLUTInputSource

```
GenApi::IEnumerationT<LogicBlockLUTInputSourceEnums>& LogicBlockLUTInputSource
```

Description: Selects the source for the input into the Logic LUT.

Visibility:

15.11.4.523 LogicBlockLUTOutputValue

```
GenApi::IBoolean& LogicBlockLUTOutputValue
```

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

Visibility:

15.11.4.524 LogicBlockLUTOutputValueAll

```
GenApi::IInteger& LogicBlockLUTOutputValueAll
```

Description: Sets the value of all the output bits in the selected LUT.

Visibility:

15.11.4.525 LogicBlockLUTRowIndex

```
GenApi::IInteger& LogicBlockLUTRowIndex
```

Description: Controls the row of the truth table to access in the selected LUT.

Visibility:

15.11.4.526 LogicBlockLUTSelector

`GenApi::IEnumerationT<LogicBlockLUTSelectorEnums>& LogicBlockLUTSelector`

Description: Selects which LogicBlock LUT to configure Visibility:

15.11.4.527 LogicBlockSelector

`GenApi::IEnumerationT<LogicBlockSelectorEnums>& LogicBlockSelector`

Description: Selects which LogicBlock to configure Visibility:

15.11.4.528 LUTEnable

`GenApi::IBoolean& LUTEnable`

Description:

Activates the selected LUT.

Visibility:

15.11.4.529 LUTIndex

`GenApi::IInteger& LUTIndex`

Description:

Control the index (offset) of the coefficient to access in the selected LUT.

Visibility:

15.11.4.530 LUTSelector

`GenApi::IEnumerationT<LUTSelectorEnums>& LUTSelector`

Description:

Selects which LUT to control.

Visibility:

15.11.4.531 LUTValue

`GenApi::Integer& LUTValue`

Description:

Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.

Visibility:

15.11.4.532 LUTValueAll

`GenApi::IRegister& LUTValueAll`

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.

Visibility: Guru

15.11.4.533 MaxDeviceResetTime

`GenApi::Integer& MaxDeviceResetTime`

Description: Time to wait until device reset complete (ms).

Visibility:

15.11.4.534 OffsetX

`GenApi::Integer& OffsetX`

Description:

Horizontal offset from the origin to the ROI (in pixels).

Visibility:

15.11.4.535 OffsetY

`GenApi::Integer& OffsetY`

Description:

Vertical offset from the origin to the ROI (in pixels).

Visibility:

15.11.4.536 PacketResendRequestCount

`GenApi::Integer& PacketResendRequestCount`

Description: Counts the number of resend requests received from the host.

Visibility:

15.11.4.537 PayloadSize

`GenApi::Integer& PayloadSize`

Description: Provides the number of bytes transferred for each image or chunk on the stream channel.

Visibility:

15.11.4.538 PixelColorFilter

`GenApi::EnumerationT<PixelColorFilterEnums>& PixelColorFilter`

Description: Type of color filter that is applied to the image.

Only applies to Bayer pixel formats. All others have no color filter.

Visibility:

15.11.4.539 PixelDynamicRangeMax

`GenApi::Integer& PixelDynamicRangeMax`

Description: Maximum value that can be returned during the digitization process.

This corresponds to the brightest value of the camera. For color cameras, this returns the biggest value that each color component can take.

Visibility:

15.11.4.540 PixelDynamicRangeMin

`GenApi::Integer& PixelDynamicRangeMin`

Description: Minimum value that can be returned during the digitization process.

This corresponds to the darkest value of the camera. For color cameras, this returns the smallest value that each color component can take.

Visibility:

15.11.4.541 PixelFormat

`GenApi::IEnumerationT<PixelFormatEnums>& PixelFormat`

Description: Format of the pixel provided by the camera.

Visibility:

15.11.4.542 PixelFormatInfoID

`GenApi::IInteger& PixelFormatInfoID`

Description: Returns the value used by the streaming channels to identify the selected pixel format.

Visibility: Guru

15.11.4.543 PixelFormatInfoSelector

`GenApi::IEnumerationT<PixelFormatInfoSelectorEnums>& PixelFormatInfoSelector`

Description: Select the pixel format for which the information will be returned.

Visibility: Guru

15.11.4.544 PixelSize

`GenApi::IEnumerationT<PixelSizeEnums>& PixelSize`

Description: Total size in bits of a pixel of the image.

Visibility:

15.11.4.545 PowerSupplyCurrent

`GenApi::IFloat& PowerSupplyCurrent`

Description:

Indicates the output current of the selected power supply (A).

Visibility:

15.11.4.546 PowerSupplyVoltage

`GenApi::IFloat& PowerSupplyVoltage`

Description:

Indicates the current voltage of the selected power supply (V).

Visibility:

15.11.4.547 RegionDestination

`GenApi::IEnumerationT<RegionDestinationEnums>& RegionDestination`

Description: Control the destination of the selected region.

Visibility: Expert

15.11.4.548 RegionMode

`GenApi::IEnumerationT<RegionModeEnums>& RegionMode`

Description: Controls if the selected Region of interest is active and streaming.

Visibility: Beginner

15.11.4.549 RegionSelector

`GenApi::IEnumerationT<RegionSelectorEnums>& RegionSelector`

Description: Selects the Region of interest to control.

The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently. Visibility: Beginner

15.11.4.550 ReverseX

`GenApi::IBoolean& ReverseX`

Description: Horizontally flips the image sent by the device.

The region of interest is applied after flipping. For color cameras the bayer pixel format is affected. For example, BayerRG16 changes to BayerGR16.

Visibility:

15.11.4.551 ReverseY

`GenApi::IBoolean& ReverseY`

Description: Vertically flips the image sent by the device.

The region of interest is applied after flipping. For color cameras the bayer pixel format is affected. For example, BayerRG16 changes to BayerGB16.

Visibility:

15.11.4.552 RgbTransformLightSource

`GenApi::IEnumerationT<RgbTransformLightSourceEnums>& RgbTransformLightSource`

Description:

Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources.

Selecting a value also sets the white balance ratios (BalanceRatioRed and BalanceRatioBlue), but those can be overwritten through manual or auto white balance.

Visibility:

15.11.4.553 Saturation

`GenApi::IFloat& Saturation`

Description: Controls the color saturation.

Visibility:

15.11.4.554 SaturationEnable

`GenApi::IBoolean& SaturationEnable`

Description: Enables/disables Saturation adjustment.

Visibility:

15.11.4.555 Scan3dAxisMax

`GenApi::IFloat& Scan3dAxisMax`

Description: Maximum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

15.11.4.556 Scan3dAxisMin

`GenApi::IFloat& Scan3dAxisMin`

Description: Minimum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

15.11.4.557 Scan3dCoordinateOffset

```
GenApi::IFloat& Scan3dCoordinateOffset
```

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

15.11.4.558 Scan3dCoordinateReferenceSelector

```
GenApi::IEnumerationT<Scan3dCoordinateReferenceSelectorEnums>& Scan3dCoordinateReferenceSelector
```

Description: Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

Visibility: Expert

15.11.4.559 Scan3dCoordinateReferenceValue

```
GenApi::IFloat& Scan3dCoordinateReferenceValue
```

Description: Returns the reference value selected.

Reads the value of a rotation or translation value for the current (Anchor or Transformed) coordinate system transformation to the Reference system. Visibility: Expert

15.11.4.560 Scan3dCoordinateScale

```
GenApi::IFloat& Scan3dCoordinateScale
```

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

15.11.4.561 Scan3dCoordinateSelector

```
GenApi::IEnumerationT<Scan3dCoordinateSelectorEnums>& Scan3dCoordinateSelector
```

Description: Selects the individual coordinates in the vectors for 3D information/transformation.

Visibility: Expert

15.11.4.562 Scan3dCoordinateSystem

```
GenApi::IEnumerationT<Scan3dCoordinateSystemEnums>& Scan3dCoordinateSystem
```

Description: Specifies the Coordinate system to use for the device.

Visibility: Beginner

15.11.4.563 Scan3dCoordinateSystemReference

```
GenApi::IEnumerationT<Scan3dCoordinateSystemReferenceEnums>& Scan3dCoordinateSystemReference
```

Description: Defines coordinate system reference location.

Visibility: Expert

15.11.4.564 Scan3dCoordinateTransformSelector

```
GenApi::IEnumerationT<Scan3dCoordinateTransformSelectorEnums>& Scan3dCoordinateTransformSelector
```

Description: Sets the index to read/write a coordinate transform value.

Visibility: Expert

15.11.4.565 Scan3dDistanceUnit

```
GenApi::IEnumerationT<Scan3dDistanceUnitEnums>& Scan3dDistanceUnit
```

Description: Specifies the unit used when delivering calibrated distance data.

Visibility: Beginner

15.11.4.566 Scan3dInvalidDataFlag

```
GenApi::IBoolean& Scan3dInvalidDataFlag
```

Description: Enables the definition of a non-valid flag value in the data stream.

Note that the confidence output is an alternate recommended way to identify non-valid pixels. Using an Scan3dInvalidDataValue may give processing penalties due to special handling. Visibility: Expert

15.11.4.567 Scan3dInvalidDataValue

```
GenApi::IFloat& Scan3dInvalidDataValue
```

Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.

Visibility: Expert

15.11.4.568 Scan3dOutputMode

```
GenApi::IEnumerationT<Scan3dOutputModeEnums>& Scan3dOutputMode
```

Description: Controls the Calibration and data organization of the device, naming the coordinates transmitted.

Visibility: Expert

15.11.4.569 Scan3dTransformValue

`GenApi::IFloat& Scan3dTransformValue`

Description: Specifies the transform value selected.

For translations (Scan3dCoordinateTransformSelector = TranslationX/Y/Z) it is expressed in the distance unit of the system, for rotations (Scan3dCoordinateTransformSelector =RotationX/Y/Z) in degrees. Visibility: Expert

15.11.4.570 SensorDescription

`GenApi::IString& SensorDescription`

Description: Returns Sensor Description Visibility:

15.11.4.571 SensorDigitizationTaps

`GenApi::IEnumerationT<SensorDigitizationTapsEnums>& SensorDigitizationTaps`

Description: Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

Visibility: Expert

15.11.4.572 SensorHeight

`GenApi::IInteger& SensorHeight`

Description: Effective height of the sensor in pixels.

Visibility:

15.11.4.573 SensorShutterMode

`GenApi::IEnumerationT<SensorShutterModeEnums>& SensorShutterMode`

Description: Sets the shutter mode of the device.

Visibility:

15.11.4.574 SensorTaps

`GenApi::IEnumerationT<SensorTapsEnums>& SensorTaps`

Description: Number of taps of the camera sensor.

Visibility: Expert

15.11.4.575 SensorWidth

`GenApi::Integer& SensorWidth`

Description: Effective width of the sensor in pixels.

Visibility:

15.11.4.576 SequencerConfigurationMode

`GenApi::EnumerationT<SequencerConfigurationModeEnums>& SequencerConfigurationMode`

Description:

Controls whether or not a sequencer is in configuration mode.

Visibility:

15.11.4.577 SequencerConfigurationValid

`GenApi::EnumerationT<SequencerConfigurationValidEnums>& SequencerConfigurationValid`

Description:

Display whether the current sequencer configuration is valid to run.

Visibility:

15.11.4.578 SequencerFeatureEnable

`GenApi::Boolean& SequencerFeatureEnable`

Description:

Enables the selected feature and makes it active in all sequencer sets.

Visibility:

15.11.4.579 SequencerMode

`GenApi::EnumerationT<SequencerModeEnums>& SequencerMode`

Description: Controls whether or not a sequencer is active.

Visibility:

15.11.4.580 SequencerPathSelector

`GenApi::Integer& SequencerPathSelector`

Description:

Selects branching path to be used for subsequent settings.

Visibility:

15.11.4.581 SequencerSetActive

`GenApi::Integer& SequencerSetActive`

Description: Displays the currently active sequencer set.

Visibility:

15.11.4.582 SequencerSetLoad

`GenApi::Command& SequencerSetLoad`

Description:

Loads currently selected sequencer to the current device configuration.

Visibility:

15.11.4.583 SequencerSetNext

`GenApi::Integer& SequencerSetNext`

Description: Specifies the next sequencer set.

Visibility:

15.11.4.584 SequencerSetSave

`GenApi::Command& SequencerSetSave`

Description:

Saves the current device configuration to the currently selected sequencer set.

Visibility:

15.11.4.585 SequencerSetSelector

`GenApi::Integer& SequencerSetSelector`

Description:

Selects the sequencer set to which subsequent settings apply.

Visibility:

15.11.4.586 SequencerSetStart

`GenApi::Integer& SequencerSetStart`

Description: Sets the first sequencer set to be used.

Visibility:

15.11.4.587 SequencerSetValid

`GenApi::EnumerationT<SequencerSetValidEnums>& SequencerSetValid`

Description:

Displays whether the currently selected sequencer set's register contents are valid to use.

Visibility:

15.11.4.588 SequencerTriggerActivation

`GenApi::EnumerationT<SequencerTriggerActivationEnums>& SequencerTriggerActivation`

Description:

Specifies the activation mode of the sequencer trigger.

Visibility:

15.11.4.589 SequencerTriggerSource

`GenApi::EnumerationT<SequencerTriggerSourceEnums>& SequencerTriggerSource`

Description:

Specifies the internal signal or physical input line to use as the sequencer trigger source.

Visibility:

15.11.4.590 SerialPortBaudRate

`GenApi::IEnumerationT<SerialPortBaudRateEnums>& SerialPortBaudRate`

Description: This feature controls the baud rate used by the selected serial port.

Visibility:

15.11.4.591 SerialPortDataBits

`GenApi::IInteger& SerialPortDataBits`

Description: This feature controls the number of data bits used by the selected serial port.

Possible values that can be used are between 5 and 9. Visibility:

15.11.4.592 SerialPortParity

`GenApi::IEnumerationT<SerialPortParityEnums>& SerialPortParity`

Description: This feature controls the parity used by the selected serial port.

Visibility:

15.11.4.593 SerialPortSelector

`GenApi::IEnumerationT<SerialPortSelectorEnums>& SerialPortSelector`

Description: Selects which serial port of the device to control.

Visibility:

15.11.4.594 SerialPortSource

`GenApi::IEnumerationT<SerialPortSourceEnums>& SerialPortSource`

Description: Specifies the physical input Line on which to receive serial data.

Visibility:

15.11.4.595 SerialPortStopBits

`GenApi::IEnumerationT<SerialPortStopBitsEnums>& SerialPortStopBits`

Description: This feature controls the number of stop bits used by the selected serial port.

Visibility:

15.11.4.596 SerialReceiveFramingErrorCount

`GenApi::Integer& SerialReceiveFramingErrorCount`

Description: Returns the number of framing errors that have occurred on the serial port.

Visibility:

15.11.4.597 SerialReceiveParityErrorCount

`GenApi::Integer& SerialReceiveParityErrorCount`

Description: Returns the number of parity errors that have occurred on the serial port.

Visibility:

15.11.4.598 SerialReceiveQueueClear

`GenApi::Command& SerialReceiveQueueClear`

Description: This is a command that clears the device serial port receive queue.

Visibility:

15.11.4.599 SerialReceiveQueueCurrentCharacterCount

`GenApi::Integer& SerialReceiveQueueCurrentCharacterCount`

Description: Returns the number of characters currently in the serial port receive queue.

Visibility:

15.11.4.600 SerialReceiveQueueMaxCharacterCount

`GenApi::Integer& SerialReceiveQueueMaxCharacterCount`

Description: >Returns the maximum number of characters in the serial port receive queue.

Visibility:

15.11.4.601 SerialTransmitQueueCurrentCharacterCount

`GenApi::Integer& SerialTransmitQueueCurrentCharacterCount`

Description: Returns the number of characters currently in the serial port transmit queue.

Visibility:

15.11.4.602 SerialTransmitQueueMaxCharacterCount

`GenApi::Integer& SerialTransmitQueueMaxCharacterCount`

Description: >Returns the maximum number of characters in the serial port transmit queue.

Visibility:

15.11.4.603 Sharpening

`GenApi::Float& Sharpening`

Description:

Controls the amount to sharpen a signal.

The sharpened amount is proportional to the difference between a pixel and its neighbors. A negative value smooths out the difference, while a positive value amplifies the difference. You can boost by a maximum of 8x, but smoothing is limited to 1x (in float). Default value: 2.0

Visibility:

15.11.4.604 SharpeningAuto

`GenApi::Boolean& SharpeningAuto`

Description:

Enables/disables the auto sharpening feature.

When enabled, the camera automatically determines the sharpening threshold based on the noise level of the camera.

Visibility:

15.11.4.605 SharpeningEnable

`GenApi::Boolean& SharpeningEnable`

Description:

Enables/disables the sharpening feature.

Sharpening is disabled by default.

Visibility:

15.11.4.606 SharpeningThreshold

`GenApi::IFloat& SharpeningThreshold`

Description:

Controls the minimum intensity gradient change to invoke sharpening.

When "Sharpening Auto" is enabled, this is determined automatically by the device. The threshold is specified as a fraction of the total intensity range, and ranges from 0 to 0.25. A threshold higher than 25% produces little to no difference than 25%. High thresholds sharpen only areas with significant intensity changes. Low thresholds sharpen more areas.

Visibility:

15.11.4.607 SoftwareSignalPulse

`GenApi::ICommand& SoftwareSignalPulse`

Description: Generates a pulse signal that can be used as a software trigger.

This command can be used to trigger other modules that accept a SoftwareSignal as trigger source. Visibility: Beginner

15.11.4.608 SoftwareSignalSelector

`GenApi::IEnumerationT<SoftwareSignalSelectorEnums>& SoftwareSignalSelector`

Description: Selects which Software Signal features to control.

Visibility: Beginner

15.11.4.609 SourceCount

`GenApi::IInteger& SourceCount`

Description: Controls or returns the number of sources supported by the device.

Visibility: Beginner

15.11.4.610 SourceSelector

`GenApi::IEnumerationT<SourceSelectorEnums>& SourceSelector`

Description: Selects the source to control.

Visibility: Beginner

15.11.4.611 Test0001

`GenApi::Integer& Test0001`

Description: For testing only.

Visibility:

15.11.4.612 TestEventGenerate

`GenApi::ICommand& TestEventGenerate`

Description: This command generates a test event and sends it to the host.

Visibility:

15.11.4.613 TestPattern

`GenApi::IEnumerationT<TestPatternEnums>& TestPattern`

Description:

Selects the type of test pattern that is generated by the device as image source.

Visibility:

15.11.4.614 TestPatternGeneratorSelector

`GenApi::IEnumerationT<TestPatternGeneratorSelectorEnums>& TestPatternGeneratorSelector`

Description:

Selects which test pattern generator is controlled by the TestPattern feature.

Visibility:

15.11.4.615 TestPendingAck

`GenApi::Integer& TestPendingAck`

Description: Tests the device's pending acknowledge feature.

When this feature is written, the device waits a time period corresponding to the value of TestPendingAck before acknowledging the write. Visibility: Guru

15.11.4.616 TimerDelay

`GenApi::IFloat& TimerDelay`

Description: Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.

Visibility: Expert

15.11.4.617 TimerDuration

`GenApi::IFloat& TimerDuration`

Description: Sets the duration (in microseconds) of the Timer pulse.

Visibility: Expert

15.11.4.618 TimerReset

`GenApi::ICommand& TimerReset`

Description: Does a software reset of the selected timer and starts it.

The timer starts immediately after the reset unless a timer trigger is active. Visibility: Expert

15.11.4.619 TimerSelector

`GenApi::IEnumerationT<TimerSelectorEnums>& TimerSelector`

Description: Selects which Timer to configure.

Visibility: Expert

15.11.4.620 TimerStatus

`GenApi::IEnumerationT<TimerStatusEnums>& TimerStatus`

Description: Returns the current status of the Timer.

Visibility: Expert

15.11.4.621 TimerTriggerActivation

`GenApi::IEnumerationT<TimerTriggerActivationEnums>& TimerTriggerActivation`

Description: Selects the activation mode of the trigger to start the Timer.

Visibility: Expert

15.11.4.622 TimerTriggerSource

`GenApi::IEnumerationT<TimerTriggerSourceEnums>& TimerTriggerSource`

Description: Selects the source of the trigger to start the Timer.

Visibility: Expert

15.11.4.623 TimerValue

`GenApi::IFloat& TimerValue`

Description: Reads or writes the current value (in microseconds) of the selected Timer.

Visibility: Expert

15.11.4.624 Timestamp

`GenApi::IInteger& Timestamp`

Description: Reports the current value of the device timestamp counter.

Visibility: Expert

15.11.4.625 TimestampLatch

`GenApi::ICommand& TimestampLatch`

Description: Latches the current timestamp counter into TimestampLatchValue.

Visibility:

15.11.4.626 TimestampLatchValue

`GenApi::IInteger& TimestampLatchValue`

Description: Returns the latched value of the timestamp counter.

Visibility:

15.11.4.627 TimestampReset

`GenApi::ICommand& TimestampReset`

Description: Resets the current value of the device timestamp counter.

Visibility:

15.11.4.628 TLParamsLocked

`GenApi::Integer& TLParamsLocked`

Description: Visibility:

15.11.4.629 TransferAbort

`GenApi::ICommand& TransferAbort`

Description: Aborts immediately the streaming of data block(s).

Aborting the transfer will result in the lost of the data that is present or currently entering in the block queue. However, the next new block received will be stored in the queue and transferred to the host when the streaming is restarted. If implemented, this feature should be available when the TransferControlMode is set to "UserControlled". Visibility: Expert

15.11.4.630 TransferBlockCount

`GenApi::Integer& TransferBlockCount`

Description: Specifies the number of data blocks (images) that the device should stream before stopping.

This feature is only active if the Transfer Operation Mode is set to Multi Block. Visibility:

15.11.4.631 TransferBurstCount

`GenApi::Integer& TransferBurstCount`

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.

Visibility: Expert

15.11.4.632 TransferComponentSelector

`GenApi::IEnumerationT<TransferComponentSelectorEnums>& TransferComponentSelector`

Description: Selects the color component for the control of the TransferStreamChannel feature.

Visibility: Guru

15.11.4.633 TransferControlMode

`GenApi::IEnumerationT<TransferControlModeEnums>& TransferControlMode`

Description: Selects the control method for the transfers.

Basic and Automatic start transmitting data as soon as there is enough data to fill a link layer packet. User Controlled allows you to directly control the transfer of blocks. Visibility:

15.11.4.634 TransferOperationMode

```
GenApi::IEnumerationT<TransferOperationModeEnums>& TransferOperationMode
```

Description: Selects the operation mode of the transfer.

Continuous is similar to Basic/Automatic but you can start/stop the transfer while acquisition runs independently. Multi Block transmits a specified number of blocks and then stops. Visibility:

15.11.4.635 TransferPause

```
GenApi::ICommand& TransferPause
```

Description: Pauses the streaming of data Block(s).

Pausing the streaming will immediately suspend the ongoing data transfer even if a block is partially transferred. The device will resume its transmission at the reception of a TransferResume command. Visibility: Guru

15.11.4.636 TransferQueueCurrentBlockCount

```
GenApi::IInteger& TransferQueueCurrentBlockCount
```

Description: Returns number of data blocks (images) currently in the transfer queue.

Visibility:

15.11.4.637 TransferQueueMaxBlockCount

```
GenApi::IInteger& TransferQueueMaxBlockCount
```

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

15.11.4.638 TransferQueueMode

```
GenApi::IEnumerationT<TransferQueueModeEnums>& TransferQueueMode
```

Description: Specifies the operation mode of the transfer queue.

Visibility:

15.11.4.639 TransferQueueOverflowCount

```
GenApi::IInteger& TransferQueueOverflowCount
```

Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

Visibility:

15.11.4.640 TransferResume

`GenApi::ICommand& TransferResume`

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.

Visibility: Guru

15.11.4.641 TransferSelector

`GenApi::IEnumerationT<TransferSelectorEnums>& TransferSelector`

Description: Selects which stream transfers are currently controlled by the selected Transfer features.

Visibility: Expert

15.11.4.642 TransferStart

`GenApi::ICommand& TransferStart`

Description: Starts the streaming of data blocks (images) out of the device.

This feature is available when the Transfer Control Mode is set to User Controlled. Visibility:

15.11.4.643 TransferStatus

`GenApi::IBoolean& TransferStatus`

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.

Visibility: Guru

15.11.4.644 TransferStatusSelector

`GenApi::IEnumerationT<TransferStatusSelectorEnums>& TransferStatusSelector`

Description: Selects which status of the transfer module to read.

Visibility: Guru

15.11.4.645 TransferStop

`GenApi::ICommand& TransferStop`

Description: Stops the streaming of data block (images).

The current block transmission is completed. This feature is available when the Transfer Control Mode is set to User Controlled. Visibility:

15.11.4.646 TransferStreamChannel

`GenApi::Integer& TransferStreamChannel`

Description: Selects the streaming channel that will be used to transfer the selected stream of data.

In general, this feature can be omitted and the default streaming channel will be used. Visibility: Guru

15.11.4.647 TransferTriggerActivation

`GenApi::EnumerationT<TransferTriggerActivationEnums>& TransferTriggerActivation`

Description: Specifies the activation mode of the transfer control trigger.

Visibility: Guru

15.11.4.648 TransferTriggerMode

`GenApi::EnumerationT<TransferTriggerModeEnums>& TransferTriggerMode`

Description: Controls if the selected trigger is active.

Visibility: Guru

15.11.4.649 TransferTriggerSelector

`GenApi::EnumerationT<TransferTriggerSelectorEnums>& TransferTriggerSelector`

Description: Selects the type of transfer trigger to configure.

Visibility: Guru

15.11.4.650 TransferTriggerSource

`GenApi::EnumerationT<TransferTriggerSourceEnums>& TransferTriggerSource`

Description: Specifies the signal to use as the trigger source for transfers.

Visibility: Guru

15.11.4.651 TriggerActivation

`GenApi::EnumerationT<TriggerActivationEnums>& TriggerActivation`

Description: Specifies the activation mode of the trigger.

Visibility:

15.11.4.652 TriggerDelay

`GenApi::IFloat& TriggerDelay`

Description:

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

Visibility:

15.11.4.653 TriggerDivider

`GenApi::IInteger& TriggerDivider`

Description: Specifies a division factor for the incoming trigger pulses.

Visibility: Expert

15.11.4.654 TriggerEventTest

`GenApi::ICommand& TriggerEventTest`

Description: This command generates a test event and sends it to the host.

Visibility:

15.11.4.655 TriggerMode

`GenApi::IEnumerationT<TriggerModeEnums>& TriggerMode`

Description:

Controls whether or not trigger is active.

Visibility:

15.11.4.656 TriggerMultiplier

`GenApi::IInteger& TriggerMultiplier`

Description: Specifies a multiplication factor for the incoming trigger pulses.

It is used generally used in conjunction with TriggerDivider to control the ratio of triggers that are accepted.

Visibility: Expert

15.11.4.657 TriggerOverlap

`GenApi::IEnumerationT<TriggerOverlapEnums>& TriggerOverlap`

Description: Specifies the overlap mode of the trigger.

Visibility:

15.11.4.658 TriggerSelector

`GenApi::IEnumerationT<TriggerSelectorEnums>& TriggerSelector`

Description: Selects the type of trigger to configure.

Visibility:

15.11.4.659 TriggerSoftware

`GenApi::ICommand& TriggerSoftware`

Description:

Generates an internal trigger if Trigger Source is set to Software.

Visibility:

15.11.4.660 TriggerSource

`GenApi::IEnumerationT<TriggerSourceEnums>& TriggerSource`

Description:

Specifies the internal signal or physical input line to use as the trigger source.

Visibility:

15.11.4.661 UserOutputSelector

`GenApi::IEnumerationT<UserOutputSelectorEnums>& UserOutputSelector`

Description: Selects which bit of the User Output register is set by UserOutputValue.

Visibility:

15.11.4.662 UserOutputValue

`GenApi::IBoolean& UserOutputValue`

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).

Visibility:

15.11.4.663 UserOutputValueAll

`GenApi::IInteger& UserOutputValueAll`

Description: Returns the current status of all the user output status bits in a hexadecimal representation (UserOutput 0 status corresponds to bit 0, UserOutput 1 status with bit 1, etc).

This allows simultaneous reading of all user output statuses at once. Visibility:

15.11.4.664 UserOutputValueAllMask

`GenApi::IInteger& UserOutputValueAllMask`

Description: Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.

If the UserOutputValueAllMask feature is present, setting the user Output register using UserOutputValueAll will only change the bits that have a corresponding bit in the mask set to one. Visibility: Expert

15.11.4.665 UserSetDefault

`GenApi::IEnumerationT<UserSetDefaultEnums>& UserSetDefault`

Description:

Selects the feature User Set to load and make active by default when the device is restarted.

Visibility:

15.11.4.666 UserSetFeatureEnable

`GenApi::IBoolean& UserSetFeatureEnable`

Description: Whether or not the selected feature is saved to user sets.

Visibility:

15.11.4.667 UserSetLoad

`GenApi::ICommand& UserSetLoad`

Description:

Loads the User Set specified by UserSetSelector to the device and makes it active.

Visibility:

15.11.4.668 UserSetSave

`GenApi::ICommand& UserSetSave`

Description:

Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.

Visibility:

15.11.4.669 UserSetSelector

`GenApi::IEnumerationT<UserSetSelectorEnums>& UserSetSelector`

Description:

Selects the feature User Set to load, save or configure.

Visibility:

15.11.4.670 V3_3Enable

`GenApi::IBoolean& V3_3Enable`

Description: Internally generated 3.3V rail.

Enable to supply external circuits with power. This is different than standard logic outputs in that it is comparatively slow to switch but can supply a more significant amount of power. This is only available on some pins. Visibility:

15.11.4.671 WhiteClip

`GenApi::IFloat& WhiteClip`

Description: Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.

The video signal will never exceed the white clipping point: it will saturate at that level. Visibility: Expert

15.11.4.672 WhiteClipSelector

`GenApi::IEnumerationT<WhiteClipSelectorEnums>& WhiteClipSelector`

Description: Selects which White Clip to control.

Visibility: Expert

15.11.4.673 Width

`GenApi::IInteger& Width`

Description:

Width of the image provided by the device (in pixels).

Visibility:

15.11.4.674 WidthMax

`GenApi::IInteger& WidthMax`

Description:

Maximum width of the image (in pixels).

The dimension is calculated after horizontal binning. WidthMax does not take into account the current Region of interest (Width or OffsetX).

Visibility:

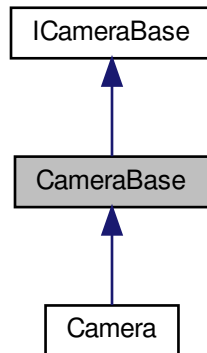
The documentation for this class was generated from the following file:

- `include/Camera.h`

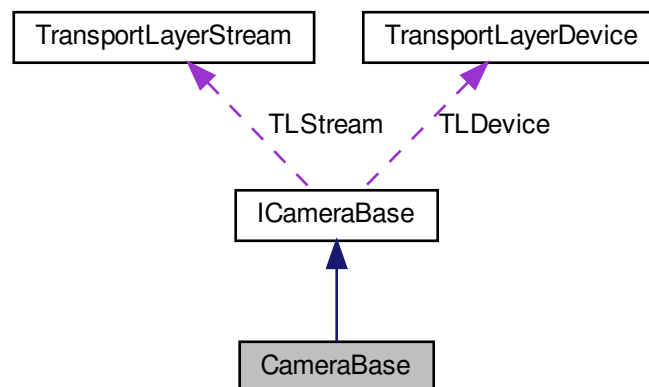
15.12 CameraBase Class Reference

The base class for the camera object.

Inheritance diagram for CameraBase:



Collaboration diagram for CameraBase:



Public Member Functions

- virtual [~CameraBase](#) (void)

Virtual Destructor.

- void [Init](#) ()

Init Connects to camera with read and write access, retrieves remote device and GUI XML and generates node map references.

- void **DelInit** ()
DelInit Disconnects camera port, resets camera back to read access and frees [GenICam](#) node map and GUI XML.
- bool **IsInitialized** ()
IsInitialized Checks if camera is initialized.
- bool **IsValid** ()
IsValid Checks a flag to determine if camera is still valid for use.
- [GenApi::INodeMap](#) & **GetNodeMap** () const
GetNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file.
- [GenApi::INodeMap](#) & **GetTLDeviceNodeMap** () const
GetTLDeviceNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Device module.
- [GenApi::INodeMap](#) & **GetTLStreamNodeMap** () const
GetTLStreamNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Stream module.
- [GenApi::EAccessMode](#) **GetAccessMode** () const
GetAccessMode Returns the access mode that the software has on the [Camera](#).
- void **ReadPort** (uint64_t iAddress, void *pBuffer, size_t iSize)
- void **WritePort** (uint64_t iAddress, const void *pBuffer, size_t iSize)
- void **BeginAcquisition** ()
BeginAcquisition Starts the image acquisition engine.
- void **EndAcquisition** ()
EndAcquisition Stops the image acquisition engine.
- [BufferOwnership](#) **GetBufferOwnership** () const
GetBufferOwnership Gets data buffer ownership.
- void **SetBufferOwnership** (const [BufferOwnership](#) mode)
SetBufferOwnership Sets data buffer ownership.
- uint64_t **GetUserBufferCount** () const
GetUserBufferCount Gets the number of user memory buffers.
- uint64_t **GetUserBufferSize** () const
GetUserBufferSize Gets the size of one user memory buffer (in bytes).
- uint64_t **GetUserBufferTotalSize** () const
GetUserBufferTotalSize Gets the total size of all the user memory buffers (in bytes).
- void **SetUserBuffers** (void *const pMemBuffers, uint64_t totalSize)
SetUserBuffers Specify contiguous user allocated memory to use as data buffers.
- void **SetUserBuffers** (void **const ppMemBuffers, const uint64_t bufferCount, const uint64_t bufferSize)
SetUserBuffers Specify non-contiguous user allocated memory to use as data buffers.
- [ImagePtr](#) **GetNextImage** (uint64_t grabTimeout=[EVENT_TIMEOUT_INFINITE](#), uint64_t streamID=0)
GetNextImage Gets the next image that was received by the transport layer.
- [GenICam::gcstring](#) **GetUniqueID** ()
GetUniqueID This returns a unique id string that identifies the camera.
- bool **IsStreaming** () const
IsStreaming Returns true if the camera is currently streaming or false if it is not.
- [GenICam::gcstring](#) **GetGuiXml** () const
Returns the GUI XML that can be passed into the [Spinnaker](#) GUI framework.
- void **RegisterEventHandler** ([EventHandler](#) &evtHandlerToRegister)
RegisterEventHandler([EventHandler](#) &) Registers a specific event handler for the camera.
- void **RegisterEventHandler** ([EventHandler](#) &evtHandlerToRegister, const [GenICam::gcstring](#) &eventName)
RegisterEventHandler([EventHandler](#) &, const [GenICam::gcstring](#) &) Registers a specific event handler for the camera.
- void **UnregisterEventHandler** ([EventHandler](#) &evtHandlerToUnregister)
UnregisterEventHandler Unregisters an event handler for the camera Event handlers should be unregistered first before calling camera [DelInit](#)()

- unsigned int [GetNumImagesInUse](#) ()
GetNumImagesInUse Returns the number of images that are currently in use.
- unsigned int [GetNumDataStreams](#) ()
GetNumDataStreams Returns the number of streams that a device supports.
- unsigned int [DiscoverMaxPacketSize](#) ()
DiscoverMaxPacketSize Returns the largest packet size that can be safely used on the interface that device is connected to.
- void [ForceIP](#) ()
ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Protected Member Functions

- [CameraBase](#) (void)
Default constructor.
- [CameraBase](#) (const [CameraBase](#) &)
Copy constructor.
- [CameraBase](#) & [operator=](#) (const [CameraBase](#) &)
Assignment operator.

Friends

- class [InterfacelImpl](#)

Additional Inherited Members

15.12.1 Detailed Description

The base class for the camera object.

15.12.2 Constructor & Destructor Documentation

15.12.2.1 [~CameraBase\(\)](#)

```
virtual ~CameraBase (  
    void ) [virtual]
```

Virtual Destructor.

15.12.2.2 CameraBase() [1/2]

```
CameraBase (
    void ) [protected]
```

Default constructor.

15.12.2.3 CameraBase() [2/2]

```
CameraBase (
    const CameraBase & ) [protected]
```

Copy constructor.

15.12.3 Member Function Documentation

15.12.3.1 BeginAcquisition()

```
void BeginAcquisition ( ) [virtual]
```

BeginAcquisition Starts the image acquisition engine.

The camera must be initialized via a call to [Init\(\)](#) before starting an acquisition.

See also

[Init\(\)](#)

Implements [ICameraBase](#).

15.12.3.2 DeInit()

```
void DeInit ( ) [virtual]
```

DeInit Disconnects camera port, resets camera back to read access and frees [GenICam](#) node map and GUI XML.

Do not call more functions that access the remote device such as WritePort/ReadPort after calling [DeInit\(\)](#). Events should also be unregistered before calling camera [DeInit\(\)](#). Otherwise an exception will be thrown in the [DeInit\(\)](#) call and require the user to unregister events before the camera can be re-initialized again.

See also

[Init\(\)](#)

[UnregisterEventHandler\(EventHandler & evtHandlerToUnregister\)](#)

Implements [ICameraBase](#).

15.12.3.3 DiscoverMaxPacketSize()

```
unsigned int DiscoverMaxPacketSize ( ) [virtual]
```

DiscoverMaxPacketSize Returns the largest packet size that can be safely used on the interface that device is connected to.

Returns

The maximum packet size returned.

Implements [ICameraBase](#).

15.12.3.4 EndAcquisition()

```
void EndAcquisition ( ) [virtual]
```

EndAcquisition Stops the image acquisition engine.

If [EndAcquisition\(\)](#) is called without a prior call to [BeginAcquisition\(\)](#) an error message "Camera is not started" will be thrown. All Images that were acquired using [GetNextImage\(\)](#) need to be released first using `image->Release()` before calling [EndAcquisition\(\)](#). All buffers in the input pool and output queue will be discarded when [EndAcquisition\(\)](#) is called.

See also

[Init\(\)](#)
[BeginAcquisition\(\)](#)
[GetNextImage\(grabTimeout \)](#)
[Image::Release\(\)](#)

Implements [ICameraBase](#).

15.12.3.5 ForceIP()

```
void ForceIP ( ) [virtual]
```

ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Implements [ICameraBase](#).

15.12.3.6 GetAccessMode()

```
GenApi::EAccessMode GetAccessMode ( ) const [virtual]
```

GetAccessMode Returns the access mode that the software has on the [Camera](#).

The camera does not need to be initialized before calling this function.

See also

[Init\(\)](#)

Returns

An enumeration value indicating the access mode

Implements [ICameraBase](#).

15.12.3.7 GetBufferOwnership()

```
BufferOwnership GetBufferOwnership ( ) const [virtual]
```

GetBufferOwnership Gets data buffer ownership.

The data buffers can be owned by [System](#) or User. If the system owns the buffers, the memory required for the buffers are allocated and freed by the library. If user owns the buffers, the user is responsible for allocating and ultimately freeing the memory. By default, data buffers are owned by the library.

See also

[SetBufferOwnership\(\)](#)
[SetUserBuffers\(\)](#)

Returns

Buffer ownership (system or user)

Implements [ICameraBase](#).

15.12.3.8 GetGuiXml()

```
GenICam::gcstring GetGuiXml ( ) const [virtual]
```

Returns the GUI XML that can be passed into the [Spinnaker](#) GUI framework.

Returns

[GenICam::gcstring](#) that represents the uncompressed GUI XML file

Implements [ICameraBase](#).

15.12.3.9 GetNextImage()

```
ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT_TIMEOUT_INFINITE,
    uint64_t streamID = 0 ) [virtual]
```

GetNextImage Gets the next image that was received by the transport layer.

This function will block indefinitely until an image arrives. Most cameras support one stream so the default streamID is 0 but if a camera supports multiple streams the user can input the streamID to select from which stream to grab images

See also

[Init\(\)](#)
[BeginAcquisition\(\)](#)
[EndAcquisition\(\)](#)

Parameters

<i>grabTimeout</i>	a 64bit value that represents a timeout in milliseconds
<i>streamID</i>	The stream to grab the image.

Returns

pointer to an [Image](#) object

Implements [ICameraBase](#).

15.12.3.10 GetNodeMap()

```
GenApi::INodeMap& GetNodeMap ( ) const [virtual]
```

GetNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file.

The camera must be initialized by a call to [Init\(\)](#) first before a node map reference can be successfully acquired.

See also

[Init\(\)](#)

Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

15.12.3.11 GetNumDataStreams()

```
unsigned int GetNumDataStreams ( ) [virtual]
```

GetNumDataStreams Returns the number of streams that a device supports.

Returns

The number of data streams

Implements [ICameraBase](#).

15.12.3.12 GetNumImagesInUse()

```
unsigned int GetNumImagesInUse ( ) [virtual]
```

GetNumImagesInUse Returns the number of images that are currently in use.

Each of the images that are currently in use must be cleaned up with a call to `image->Release()` before calling `system->ReleaseInstance()`.

Returns

The number of images that needs to be cleaned up.

Implements [ICameraBase](#).

15.12.3.13 GetTLDeviceNodeMap()

```
GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [virtual]
```

GetTLDeviceNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Device module.

The camera does not need to be initialized before acquiring this node map.

Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

15.12.3.14 GetTLStreamNodeMap()

```
GenApi::INodeMap& GetTLStreamNodeMap ( ) const [virtual]
```

GetTLStreamNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Stream module.

The camera does not need to be initialized before acquiring this node map.

Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

15.12.3.15 GetUniqueID()

```
GenICam::gcstring GetUniqueID ( ) [virtual]
```

GetUniqueID This returns a unique id string that identifies the camera.

This is the camera serial number.

Returns

string that uniquely identifies the camera (serial number)

Implements [ICameraBase](#).

15.12.3.16 GetUserBufferCount()

```
uint64_t GetUserBufferCount ( ) const [virtual]
```

GetUserBufferCount Gets the number of user memory buffers.

This will throw an exception if user memory buffer has not been set. If the user memory is contiguous, this will throw an exception unless [BeginAcquisition\(\)](#) has been called.

See also

[BeginAcquisition\(\)](#)
[SetUserBuffers\(\)](#)

Returns

The number of user memory buffers

Implements [ICameraBase](#).

15.12.3.17 GetUserBufferSize()

```
uint64_t GetUserBufferSize ( ) const [virtual]
```

GetUserBufferSize Gets the size of one user memory buffer (in bytes).

This will throw an exception if user memory buffer has not been set. If the user memory is contiguous, this will throw an exception unless [BeginAcquisition\(\)](#) has been called. To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: $((\text{unsigned int}) (\text{bufferSize} + 1024 - 1) / 1024) * 1024$ where 1024 is the USB3 packet size.

See also

[BeginAcquisition\(\)](#)
[SetUserBuffers\(\)](#)

Returns

The size of one user memory buffer (in bytes)

Implements [ICameraBase](#).

15.12.3.18 GetUserBufferTotalSize()

```
uint64_t GetUserBufferTotalSize ( ) const [virtual]
```

GetUserBufferTotalSize Gets the total size of all the user memory buffers (in bytes).

This will throw an exception if user memory buffer has not been set. The total size should be [GetUserBufferSize\(\)](#) multiplied by [GetUserBufferCount\(\)](#) or larger.

See also

[GetUserBufferCount\(\)](#)
[GetUserBufferSize\(\)](#)
[SetUserBuffers\(\)](#)

Returns

The total size of all the user memory buffers (in bytes)

Implements [ICameraBase](#).

15.12.3.19 Init()

```
void Init ( ) [virtual]
```

Init Connects to camera with read and write access, retrieves remote device and GUI XML and generates node map references.

This function needs to be called before any camera related API calls such as [BeginAcquisition\(\)](#), [EndAcquisition\(\)](#), [GetNodeMap\(\)](#), [GetNextImage\(\)](#).

See also

[BeginAcquisition\(\)](#)
[EndAcquisition\(\)](#)
[GetNodeMap\(\)](#)
[GetNextImage\(\)](#)

Implements [ICameraBase](#).

15.12.3.20 IsInitialized()

```
bool IsInitialized ( ) [virtual]
```

IsInitialized Checks if camera is initialized.

This function needs to return true in order to retrieve a valid NodeMap from the [GetNodeMap\(\)](#) call.

See also

[GetNodeMap\(\)](#)

Returns

If camera is initialized or not

Implements [ICameraBase](#).

15.12.3.21 IsStreaming()

```
bool IsStreaming ( ) const [virtual]
```

IsStreaming Returns true if the camera is currently streaming or false if it is not.

See also

[Init\(\)](#)

Returns

returns true if camera is streaming and false otherwise.

Implements [ICameraBase](#).

15.12.3.22 IsValid()

```
bool IsValid ( ) [virtual]
```

IsValid Checks a flag to determine if camera is still valid for use.

Returns

If camera is valid or not

Implements [ICameraBase](#).

15.12.3.23 operator=()

```
CameraBase& operator= (
    const CameraBase & ) [protected]
```

Assignment operator.

15.12.3.24 ReadPort()

```
void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

15.12.3.25 RegisterEventHandler() [1/2]

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [virtual]
```

[RegisterEventHandler\(EventHandler &\)](#) Registers a specific event handler for the camera.

The camera has to be initialized first with a call to [Init\(\)](#) before registering handlers for events.

See also

[Init\(\)](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the camera

Implements [ICameraBase](#).

15.12.3.26 RegisterEventHandler() [2/2]

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister,
    const GenICam::gcstring & eventName ) [virtual]
```

[RegisterEventHandler\(EventHandler &, const GenICam::gcstring&\)](#) Registers a specific event handler for the camera.

See also

[Init\(\)](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the camera
<i>eventName</i>	The event name to register

Implements [ICameraBase](#).

15.12.3.27 SetBufferOwnership()

```
void SetBufferOwnership (
    const BufferOwnership mode ) [virtual]
```

SetBufferOwnership Sets data buffer ownership.

The data buffers can be owned by [System](#) or User. If the system owns the buffers, the memory required for the buffers are allocated and freed by the library. If user owns the buffers, the user is responsible for allocating and ultimately freeing the memory. By default, data buffers are owned by the library.

See also

[GetBufferOwnership\(\)](#)
[SetUserBuffers\(\)](#)

Parameters

<i>mode</i>	System owned or User owned buffers

Implements [ICameraBase](#).

15.12.3.28 SetUserBuffers() [1/2]

```
void SetUserBuffers (
    void *const pMemBuffers,
    uint64_t totalSize ) [virtual]
```

SetUserBuffers Specify contiguous user allocated memory to use as data buffers.

To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: $((\text{unsigned int}) (\text{bufferSize} + 1024 - 1) / 1024) * 1024$ where 1024 is the USB3 packet size.

See also

[GetBufferOwnership\(\)](#)
[SetBufferOwnership\(\)](#)
[GetUserBufferCount\(\)](#)
[GetUserBufferSize\(\)](#)
[GetUserBufferTotalSize\(\)](#)

Parameters

<i>pMemBuffers</i>	Pointer to memory buffers to be written to
<i>totalSize</i>	The total size of the memory allocated for the user buffers (in bytes)

Implements [ICameraBase](#).

15.12.3.29 SetUserBuffers() [2/2]

```
void SetUserBuffers (
    void **const ppMemBuffers,
    const uint64_t bufferCount,
    const uint64_t bufferSize ) [virtual]
```

SetUserBuffers Specify non-contiguous user allocated memory to use as data buffers.

Each pointer to a buffer must have enough memory to hold one image. To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: $((\text{unsigned int}) (\text{bufferSize} + 1024 - 1) / 1024) * 1024$ where 1024 is the USB3 packet size.

See also

[GetBufferOwnership\(\)](#)
[SetBufferOwnership\(\)](#)
[GetUserBufferCount\(\)](#)
[GetUserBufferSize\(\)](#)
[GetUserBufferTotalSize\(\)](#)

Parameters

<i>ppMemBuffers</i>	Pointer to pointers that each point to a single user memory buffer to be written to
<i>bufferCount</i>	The number of user memory buffers
<i>bufferSize</i>	The size of the memory allocated for each user buffer (in bytes)

Implements [ICameraBase](#).

15.12.3.30 UnregisterEventHandler()

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

UnregisterEventHandler Unregisters an event handler for the camera Event handlers should be unregistered first before calling camera [Delnit\(\)](#).

Otherwise an exception will be thrown in the [Delnit\(\)](#) call and require the user to unregister event handlers before the camera can be re-initialized again.

See also

[Delnit\(\)](#)

Parameters

<i>evtHandlerToUnregister</i>	The event handler to unregister from the camera

Implements [ICameraBase](#).

15.12.3.31 WritePort()

```
void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

15.12.4 Friends And Related Function Documentation

15.12.4.1 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

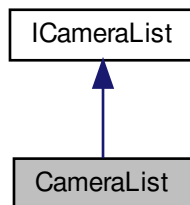
The documentation for this class was generated from the following file:

- include/[CameraBase.h](#)

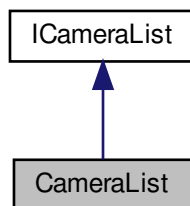
15.13 CameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for CameraList:



Collaboration diagram for CameraList:



Public Member Functions

- [CameraList](#) (void)
Default constructor.
- virtual [~CameraList](#) (void)
Virtual destructor.
- [CameraList](#) (const [CameraList](#) &iface)
Copy constructor.
- [CameraList](#) & [operator=](#) (const [CameraList](#) &iface)
Assignment operator.
- [CameraPtr](#) [operator\[\]](#) (unsigned int index)
Array subscription operators.
- unsigned int [GetSize](#) () const
Returns the size of the camera list.
- [CameraPtr](#) [GetByIndex](#) (unsigned int index) const

- Returns a pointer to a camera object at the "index".*
- [CameraPtr GetBySerial](#) (std::string serialNumber) const
Returns a pointer to a camera object with the specified serial number.
- [CameraPtr GetByDeviceID](#) (std::string deviceID) const
Returns a pointer to a camera object with the specified device identifier.
- void [Clear](#) ()
Clears the list of cameras and destroys their corresponding reference counted objects.
- void [RemoveByIndex](#) (unsigned int index)
Removes a camera at "index" and destroys its corresponding reference counted object.
- void [RemoveBySerial](#) (std::string serialNumber)
Removes a camera using its serial number and destroys its corresponding reference counted object.
- void [RemoveByDeviceID](#) (std::string deviceID)
Removes a camera using its unique device identifier and destroys its corresponding reference counted object.
- void [Append](#) (const [CameraList](#) &list)
Appends a copy of the camera list.

Additional Inherited Members

15.13.1 Detailed Description

Used to hold a list of camera objects.

15.13.2 Constructor & Destructor Documentation

15.13.2.1 [CameraList\(\)](#) [1/2]

```
CameraList (
    void )
```

Default constructor.

15.13.2.2 [~CameraList\(\)](#)

```
virtual ~CameraList (
    void ) [virtual]
```

Virtual destructor.

15.13.2.3 CameraList() [2/2]

```
CameraList (
    const CameraList & iface )
```

Copy constructor.

15.13.3 Member Function Documentation

15.13.3.1 Append()

```
void Append (
    const CameraList & list ) [virtual]
```

Appends a copy of the camera list.

Parameters

<i>list</i>	Another CameraList object, whose elements are added to this list.

Implements [ICameraList](#).

15.13.3.2 Clear()

```
void Clear ( ) [virtual]
```

Clears the list of cameras and destroys their corresponding reference counted objects.

This is necessary in order to clean up the parent interface. It is important that the camera list is destroyed or is cleared before calling `system->ReleaseInstance()` or else the call to `system->ReleaseInstance()` will result in an error message thrown that a reference to the camera is still held.

See also

[System:ReleaseInstance\(\)](#)

Implements [ICameraList](#).

15.13.3.3 GetByDeviceID()

```
CameraPtr GetByDeviceID (
    std::string deviceID ) const [virtual]
```

Returns a pointer to a camera object with the specified device identifier.

This function will return a NULL [CameraPtr](#) if no matching device identifier is found.

Parameters

<i>deviceID</i>	The unique device identifier of the camera object to retrieve

Returns

A pointer to a camera object.

Implements [ICameraList](#).

15.13.3.4 GetByIndex()

```
CameraPtr GetByIndex (
    unsigned int index ) const [virtual]
```

Returns a pointer to a camera object at the "index".

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

<i>index</i>	The index at which to retrieve the camera object

Returns

A pointer to a camera object.

Implements [ICameraList](#).

15.13.3.5 GetBySerial()

```
CameraPtr GetBySerial (
    std::string serialNumber ) const [virtual]
```

Returns a pointer to a camera object with the specified serial number.

This function will return a NULL [CameraPtr](#) if no matching camera serial is found.

Parameters

<i>serialNumber</i>	The serial number of the camera object to retrieve

Returns

A pointer to a camera object.

Implements [ICameraList](#).

15.13.3.6 GetSize()

```
unsigned int GetSize ( ) const [virtual]
```

Returns the size of the camera list.

The size is the number of [Camera](#) objects stored in the list.

Returns

An integer that represents the list size.

Implements [ICameraList](#).

15.13.3.7 operator=()

```
CameraList& operator= (
    const CameraList & iface )
```

Assignment operator.

15.13.3.8 operator[]()

```
CameraPtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [ICameraList](#).

15.13.3.9 RemoveByDeviceID()

```
void RemoveByDeviceID (
    std::string deviceID ) [virtual]
```

Removes a camera using its unique device identifier and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_NOT_AVAILABLE error if no matching device identifier is found.

Parameters

<i>deviceID</i>	The unique device identifier of the camera object to retrieve

Implements [ICameraList](#).

15.13.3.10 RemoveByIndex()

```
void RemoveByIndex (
    unsigned int index ) [virtual]
```

Removes a camera at "index" and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

<i>index</i>	The index at which to remove the Camera object

Implements [ICameraList](#).

15.13.3.11 RemoveBySerial()

```
void RemoveBySerial (
    std::string serialNumber ) [virtual]
```

Removes a camera using its serial number and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_NOT_AVAILABLE error if no matching camera serial is found.

Parameters

<i>serialNumber</i>	The serial number of the Camera object to remove

Implements [ICameraList](#).

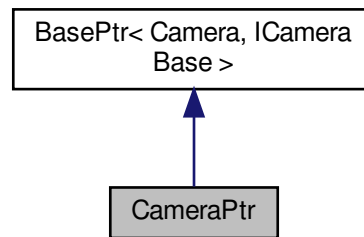
The documentation for this class was generated from the following file:

- include/[CameraList.h](#)

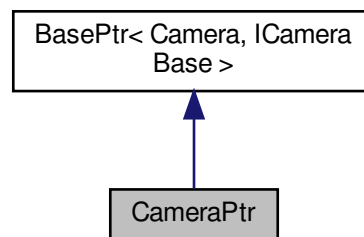
15.14 CameraPtr Class Reference

A reference tracked pointer to a camera object.

Inheritance diagram for CameraPtr:



Collaboration diagram for CameraPtr:



Public Member Functions

- [CameraPtr](#) () throw ()
Default constructor.
- [CameraPtr](#) (const int) throw ()
Default constructor.
- [CameraPtr](#) (const long) throw ()
Default constructor with argument.
- [CameraPtr](#) (const std::nullptr_t) throw ()

Additional Inherited Members

15.14.1 Detailed Description

A reference tracked pointer to a camera object.

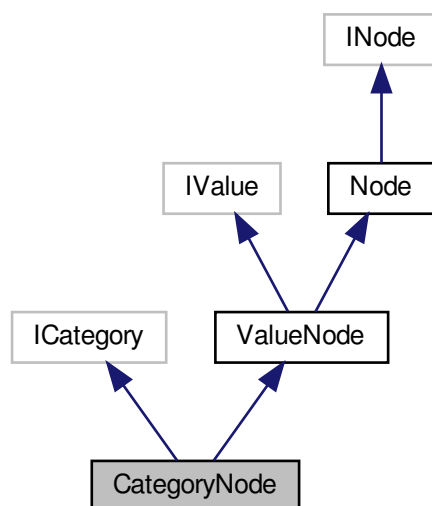
The documentation for this class was generated from the following file:

- include/[CameraPtr.h](#)

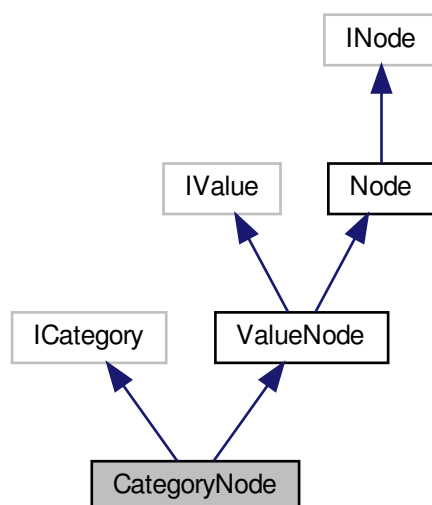
15.15 CategoryNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for CategoryNode:



Collaboration diagram for CategoryNode:



Public Member Functions

- [CategoryNode](#) ()
- [CategoryNode](#) (std::shared_ptr< Node::NodeImpl > pCategory)
- virtual [~CategoryNode](#) ()
- virtual void [GetFeatures](#) (FeatureList_t &Features) const
Get all features of the category (including sub-categories)
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.15.1 Detailed Description

[Interface](#) for string properties.

15.15.2 Constructor & Destructor Documentation

15.15.2.1 [CategoryNode\(\)](#) [1/2]

[CategoryNode](#) ()

15.15.2.2 [CategoryNode\(\)](#) [2/2]

[CategoryNode](#) (
 std::shared_ptr< Node::NodeImpl > pCategory)

15.15.2.3 [~CategoryNode\(\)](#)

virtual [~CategoryNode](#) () [virtual]

15.15.3 Member Function Documentation

15.15.3.1 GetFeatures()

```
virtual void GetFeatures (
    FeatureList_t & Features ) const [virtual]
```

Get all features of the category (including sub-categories)

15.15.3.2 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

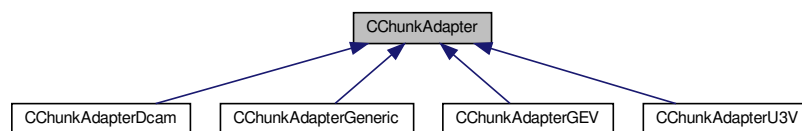
The documentation for this class was generated from the following file:

- include/SpinGenApi/[CategoryNode.h](#)

15.16 CChunkAdapter Class Reference

Connects a chunked buffer to a node map.

Inheritance diagram for CChunkAdapter:



Public Member Functions

- virtual [~CChunkAdapter](#) ()
Destructor.
- void [AttachNodeMap](#) (INodeMap *pNodeMap)
Attaches to a node map and retrieves the chunk ports.
- void [DetachNodeMap](#) ()
Detaches from the node map.
- virtual bool [CheckBufferLayout](#) (uint8_t *pBuffer, int64_t BufferLength)=0
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, int64_t BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)=0
Attaches a buffer to the matching ChunkPort.
- void [DetachBuffer](#) ()
Detaches a buffer.
- void [UpdateBuffer](#) (uint8_t *pBaseAddress)
Updates the base address of the buffer.
- void [ClearCaches](#) ()
Clears the chunk caches.

Protected Member Functions

- [CChunkAdapter](#) ([INodeMap](#) *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
Serves as default constructor.

Protected Attributes

- void * [m_pChunkAdapter](#)

15.16.1 Detailed Description

Connects a chunked buffer to a node map.

15.16.2 Constructor & Destructor Documentation

15.16.2.1 ~CChunkAdapter()

```
virtual ~CChunkAdapter ( ) [virtual]
```

Destructor.

15.16.2.2 CChunkAdapter()

```
CChunkAdapter (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 ) [protected]
```

Serves as default constructor.

15.16.3 Member Function Documentation

15.16.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [pure virtual]
```

Attaches a buffer to the matching ChunkPort.

Implemented in [CChunkAdapterDcam](#), [CChunkAdapterGeneric](#), [CChunkAdapterGEV](#), and [CChunkAdapterU3V](#).

15.16.3.2 AttachNodeMap()

```
void AttachNodeMap (
    INodeMap * pNodeMap )
```

Attaches to a node map and retrieves the chunk ports.

15.16.3.3 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [pure virtual]
```

Checks if a buffer contains chunks in a known format.

Implemented in [CChunkAdapterDcam](#), [CChunkAdapterGeneric](#), [CChunkAdapterGEV](#), and [CChunkAdapterU3V](#).

15.16.3.4 ClearCaches()

```
void ClearCaches ( )
```

Clears the chunk caches.

15.16.3.5 DetachBuffer()

```
void DetachBuffer ( )
```

Detaches a buffer.

15.16.3.6 DetachNodeMap()

```
void DetachNodeMap ( )
```

Detaches from the node map.

15.16.3.7 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the buffer.

15.16.4 Member Data Documentation

15.16.4.1 m_pChunkAdapter

```
void* m_pChunkAdapter [protected]
```

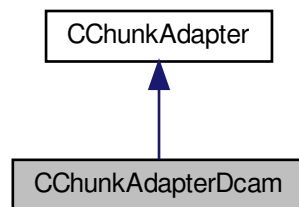
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapter.h](#)

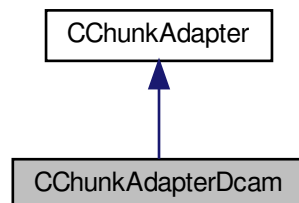
15.17 CChunkAdapterDcam Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterDcam:



Collaboration diagram for CChunkAdapterDcam:



Public Member Functions

- [CChunkAdapterDcam](#) ([INodeMap](#) *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
Constructor.
- virtual [~CChunkAdapterDcam](#) ()
Destructor.
- virtual bool [CheckBufferLayout](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, int64_t BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.
- bool [HasCRC](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if buffer has a CRC attached.
- bool [CheckCRC](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks CRC sum of buffer.

Additional Inherited Members

15.17.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

15.17.2 Constructor & Destructor Documentation

15.17.2.1 CChunkAdapterDcam()

```
CChunkAdapterDcam (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

15.17.2.2 ~CChunkAdapterDcam()

```
virtual ~CChunkAdapterDcam ( ) [virtual]
```

Destructor.

15.17.3 Member Function Documentation

15.17.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.17.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

15.17.3.3 CheckCRC()

```
bool CheckCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

Checks CRC sum of buffer.

15.17.3.4 HasCRC()

```
bool HasCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

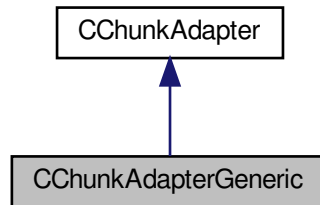
Checks if buffer has a CRC attached.

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

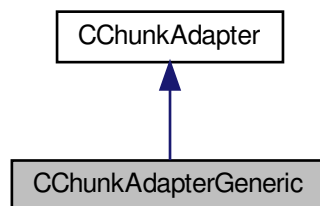
- include/SpinGenApi/[ChunkAdapterDcam.h](#)

15.18 CChunkAdapterGeneric Class Reference

Inheritance diagram for CChunkAdapterGeneric:



Collaboration diagram for CChunkAdapterGeneric:



Public Member Functions

- [CChunkAdapterGeneric](#) (INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
- virtual [~CChunkAdapterGeneric](#) ()
- virtual bool [CheckBufferLayout](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, int64_t BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, [SingleChunkData_t](#) *ChunkData, int64_t NumChunks, [AttachStatistics_t](#) *pAttachStatistics=NULL)
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, [SingleChunkDataStr_t](#) *ChunkData, int64_t NumChunks, [AttachStatistics_t](#) *pAttachStatistics=NULL)

Additional Inherited Members

15.18.1 Constructor & Destructor Documentation

15.18.1.1 CChunkAdapterGeneric()

```
CChunkAdapterGeneric (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

15.18.1.2 ~CChunkAdapterGeneric()

```
virtual ~CChunkAdapterGeneric ( ) [virtual]
```

15.18.2 Member Function Documentation

15.18.2.1 AttachBuffer() [1/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.18.2.2 AttachBuffer() [2/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkData_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

15.18.2.3 AttachBuffer() [3/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkDataStr_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

15.18.2.4 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

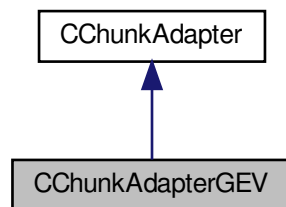
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

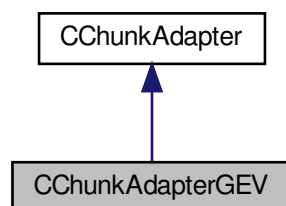
15.19 CChunkAdapterGEV Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterGEV:



Collaboration diagram for CChunkAdapterGEV:



Public Member Functions

- [CChunkAdapterGEV](#) ([INodeMap](#) *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
Constructor.
- virtual [~CChunkAdapterGEV](#) ()
Destructor.
- virtual bool [CheckBufferLayout](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, int64_t BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.

Additional Inherited Members

15.19.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

15.19.2 Constructor & Destructor Documentation

15.19.2.1 CChunkAdapterGEV()

```
CChunkAdapterGEV (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

15.19.2.2 ~CChunkAdapterGEV()

```
virtual ~CChunkAdapterGEV ( ) [virtual]
```

Destructor.

15.19.3 Member Function Documentation

15.19.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.19.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

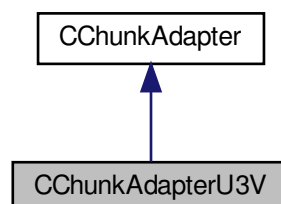
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

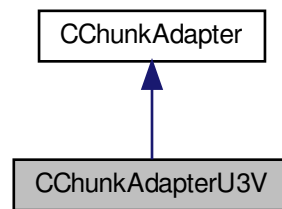
15.20 CChunkAdapterU3V Class Reference

Connects a chunked U3V buffer to a node map.

Inheritance diagram for CChunkAdapterU3V:



Collaboration diagram for CChunkAdapterU3V:



Public Member Functions

- [CChunkAdapterU3V](#) ([INodeMap](#) *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
Constructor.
- virtual [~CChunkAdapterU3V](#) ()
Destructor.
- virtual bool [CheckBufferLayout](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, int64_t BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.

Additional Inherited Members

15.20.1 Detailed Description

Connects a chunked U3V buffer to a node map.

15.20.2 Constructor & Destructor Documentation

15.20.2.1 CChunkAdapterU3V()

```

CChunkAdapterU3V (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
  
```

Constructor.

15.20.2.2 ~CChunkAdapterU3V()

```
virtual ~CChunkAdapterU3V ( ) [virtual]
```

Destructor.

15.20.3 Member Function Documentation

15.20.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.20.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

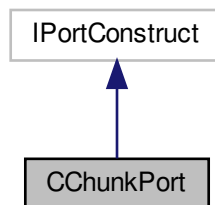
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

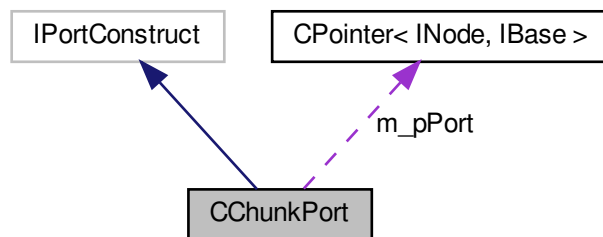
15.21 CChunkPort Class Reference

Port attachable to a chunk in a buffer.

Inheritance diagram for CChunkPort:



Collaboration diagram for CChunkPort:



Public Member Functions

- [CChunkPort](#) ([IPort](#) *pPort=NULL)
Constructor; can attach to a port.
- [~CChunkPort](#) ()
Destructor; detaches from the port.
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Get the access mode of the node.
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.

- virtual void [SetPortImpl](#) (IPort *pPort)
Called from the port node to give the chunk port a pointer to itself.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- void [InvalidateNode](#) ()
- bool [AttachPort](#) (::Spinnaker::GenApi::IPort *pPort)
Attaches the ChunkPort to the Port.
- void [DetachPort](#) ()
Detaches the ChunkPort to the Port.
- void [AttachChunk](#) (uint8_t *pBaseAddress, int64_t ChunkOffset, int64_t [Length](#), bool Cache)
Attaches the Chunk to the ChunkPort.
- void [DetachChunk](#) ()
Detaches the Chunk from the ChunkPort.
- int [GetChunkIDLength](#) ()
Gets the ChunkID length.
- bool [CheckChunkID](#) (uint8_t *pChunkIDBuffer, int ChunkIDLength)
Checks if a ChunkID matches.
- bool [CheckChunkID](#) (uint64_t ChunkID)
Checks if a ChunkID matches, version using uint64_t ID representation.
- void [UpdateBuffer](#) (uint8_t *pBaseAddress)
Updates the base address of the chunk.
- void [ClearCache](#) ()
Clears the chunk cache.

Protected Attributes

- [CNodePtr](#) [m_pPort](#)
- std::shared_ptr< PortAdapter > [m_pPortAdapter](#)
- void * [m_pChunkPort](#)

15.21.1 Detailed Description

Port attachable to a chunk in a buffer.

15.21.2 Constructor & Destructor Documentation

15.21.2.1 CChunkPort()

```
CChunkPort (
    IPort * pPort = NULL )
```

Constructor; can attach to a port.

15.21.2.2 ~CChunkPort()

`~CChunkPort ( )`

Destructor; detaches from the port.

15.21.3 Member Function Documentation

15.21.3.1 AttachChunk()

```
void AttachChunk (
    uint8_t * pBaseAddress,
    int64_t ChunkOffset,
    int64_t Length,
    bool Cache )
```

Attaches the Chunk to the ChunkPort.

15.21.3.2 AttachPort()

```
bool AttachPort (
    ::Spinnaker::GenApi::IPort * pPort )
```

Attaches the ChunkPort to the Port.

15.21.3.3 CheckChunkID() [1/2]

```
bool CheckChunkID (
    uint8_t * pChunkIDBuffer,
    int ChunkIDLength )
```

Checks if a ChunkID matches.

15.21.3.4 CheckChunkID() [2/2]

```
bool CheckChunkID (
    uint64_t ChunkID )
```

Checks if a ChunkID matches, version using uint64_t ID representation.

15.21.3.5 ClearCache()

```
void ClearCache ( )
```

Clears the chunk cache.

15.21.3.6 DetachChunk()

```
void DetachChunk ( )
```

Detaches the Chunk from the ChunkPort.

15.21.3.7 DetachPort()

```
void DetachPort ( )
```

Detaches the ChunkPort to the Port.

15.21.3.8 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

15.21.3.9 GetChunkIDLength()

```
int GetChunkIDLength ( )
```

Gets the ChunkID length.

15.21.3.10 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

15.21.3.11 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.21.3.12 InvalidateNode()

```
void InvalidateNode ( )
```

15.21.3.13 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

15.21.3.14 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

15.21.3.15 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the chunk.

15.21.3.16 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

15.21.4 Member Data Documentation

15.21.4.1 m_pChunkPort

`void* m_pChunkPort` [protected]

15.21.4.2 m_pPort

`CNodePtr m_pPort` [protected]

15.21.4.3 m_pPortAdapter

`std::shared_ptr<PortAdapter> m_pPortAdapter` [protected]

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

- include/SpinGenApi/[ChunkPort.h](#)

15.22 CCMSettings Struct Reference

Public Member Functions

- [CCMSettings](#) ()

Public Attributes

- [CCMColorTemperature](#) ColorTemperature
- [CCMType](#) Type
- [CCMSensor](#) Sensor
- `std::string` [CustomCCMCode](#)

15.22.1 Constructor & Destructor Documentation

15.22.1.1 CCMSettings()

`CCMSettings` () [inline]

15.22.2 Member Data Documentation

15.22.2.1 ColorTemperature

[CCMColorTemperature](#) ColorTemperature

15.22.2.2 CustomCCMCode

`std::string` CustomCCMCode

15.22.2.3 Sensor

[CCMSensor](#) Sensor

15.22.2.4 Type

[CCMType](#) Type

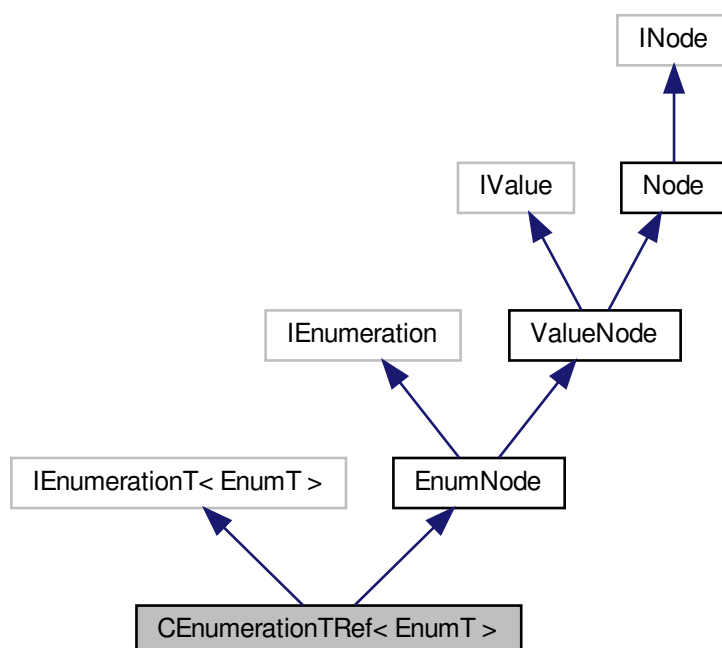
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

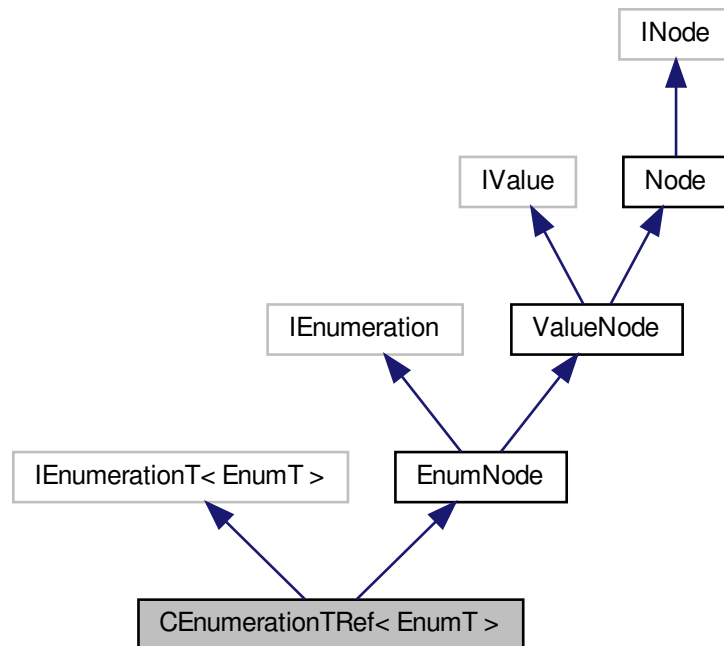
15.23 CEnumerationTRef< EnumT > Class Template Reference

[Interface](#) for string properties.

Inheritance diagram for CEnumerationTRef< EnumT >:



Collaboration diagram for CEnumerationTRef< EnumT >:



Public Member Functions

- `CEnumerationTRef` ()
- `CEnumerationTRef` (std::shared_ptr< Node::NodeImpl > pEnumeration)
- virtual `~CEnumerationTRef` ()
- virtual void `SetValue` (EnumT Value, bool `Verify`=true)
Set node value.
- virtual `IEnumeration` & `operator=` (EnumT Value)
Set node value.
- virtual EnumT `GetValue` (bool `Verify`=false, bool IgnoreCache=false)
Get node value.
- virtual EnumT `operator()` ()
Get node value.
- virtual `IEnumeration` & `operator=` (const `GenlCam::gcstring` &ValueStr)
Set node value.
- virtual `IEnumEntry` * `GetEntry` (const EnumT Value)
returns the `EnumEntry` object belonging to the Value
- virtual `IEnumEntry` * `GetEntry` (const int64_t IntValue)
Get an entry node by its IntValue.
- virtual `IEnumEntry` * `GetCurrentEntry` (bool `Verify`=false, bool IgnoreCache=false)
Get the current entry.
- virtual void `SetReference` (`INode` *pBase)
overload `SetReference` for `EnumerationT`

- virtual void [SetEnumReference](#) (int Index, [GenICam::gcstring](#) Name)
sets the Enum value corresponding to a value
- virtual void [SetNumEnums](#) (int NumEnums)
sets the number of enum values

Additional Inherited Members

15.23.1 Detailed Description

```
template<class EnumT>
class Spinnaker::GenApi::CEnumerationTRef< EnumT >
```

[Interface](#) for string properties.

15.23.2 Constructor & Destructor Documentation

15.23.2.1 CEnumerationTRef() [1/2]

```
CEnumerationTRef ( )
```

15.23.2.2 CEnumerationTRef() [2/2]

```
CEnumerationTRef (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

15.23.2.3 ~CEnumerationTRef()

```
virtual ~CEnumerationTRef ( ) [virtual]
```

15.23.3 Member Function Documentation

15.23.3.1 GetCurrentEntry()

```
virtual IEnumEntry* GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get the current entry.

Reimplemented from [EnumNode](#).

15.23.3.2 GetEntry() [1/2]

```
virtual IEnumEntry* GetEntry (
    const EnumT Value ) [virtual]
```

returns the EnumEntry object belonging to the Value

15.23.3.3 GetEntry() [2/2]

```
virtual IEnumEntry* GetEntry (
    const int64_t IntValue ) [virtual]
```

Get an entry node by its IntValue.

Reimplemented from [EnumNode](#).

15.23.3.4 GetValue()

```
virtual EnumT GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.23.3.5 operator()

```
virtual EnumT operator() ( ) [virtual]
```

Get node value.

15.23.3.6 operator=() [1/2]

```
virtual IEnumeration& operator= (
    EnumT Value ) [virtual]
```

Set node value.

15.23.3.7 operator=() [2/2]

```
virtual IEnumeration& operator= (
    const GenICam::gcstring & ValueStr ) [virtual]
```

Set node value.

Note : the operator= is not inherited thus the operator= versions from IEnumeration must be implemented again

Reimplemented from [EnumNode](#).

15.23.3.8 SetEnumReference()

```
virtual void SetEnumReference (
    int Index,
    GenICam::gcstring Name ) [virtual]
```

sets the Enum value corresponding to a value

15.23.3.9 SetNumEnums()

```
virtual void SetNumEnums (
    int NumEnums ) [virtual]
```

sets the number of enum values

15.23.3.10 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for EnumerationT

Reimplemented from [EnumNode](#).

15.23.3.11 SetValue()

```
virtual void SetValue (
    EnumT Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

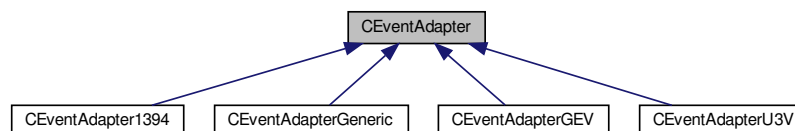
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumNodeT.h](#)

15.24 CEventAdapter Class Reference

Delivers Events to ports.

Inheritance diagram for CEventAdapter:



Public Member Functions

- [CEventAdapter](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapter](#) ()

Destructor.

- virtual void [AttachNodeMap](#) ([INodeMap](#) *pNodeMap)
Attaches to a node map and retrieves the chunk ports.
- virtual void [DetachNodeMap](#) ()
Detaches from the node map.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)=0
Deliver message.

Protected Attributes

- void * [m_pEventAdapter](#)

15.24.1 Detailed Description

Delivers Events to ports.

15.24.2 Constructor & Destructor Documentation

15.24.2.1 CEventAdapter()

```
CEventAdapter (
    INodeMap * pNodeMap = NULL )
```

Constructor.

15.24.2.2 ~CEventAdapter()

```
virtual ~CEventAdapter ( ) [virtual]
```

Destructor.

15.24.3 Member Function Documentation

15.24.3.1 AttachNodeMap()

```
virtual void AttachNodeMap (
    INodeMap * pNodeMap ) [virtual]
```

Attaches to a node map and retrieves the chunk ports.

15.24.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [pure virtual]
```

Deliver message.

Implemented in [CEventAdapterGEV](#), [CEventAdapterU3V](#), [CEventAdapter1394](#), and [CEventAdapterGeneric](#).

15.24.3.3 DetachNodeMap()

```
virtual void DetachNodeMap ( ) [virtual]
```

Detaches from the node emap.

15.24.4 Member Data Documentation

15.24.4.1 m_pEventAdapter

```
void* m_pEventAdapter [protected]
```

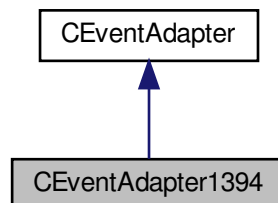
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter.h](#)

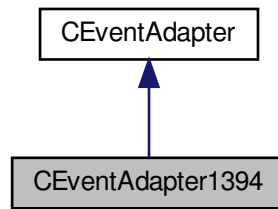
15.25 CEventAdapter1394 Class Reference

Distribute the events to the node map.

Inheritance diagram for CEventAdapter1394:



Collaboration diagram for CEventAdapter1394:



Public Member Functions

- [CEventAdapter1394](#) ([INodeMap](#) *pNodeMap=NULL)
constructor
- virtual [~CEventAdapter1394](#) ()
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (EventData1394 &Event, uint32_t numBytes)
distributes events to node map

Additional Inherited Members

15.25.1 Detailed Description

Distribute the events to the node map.

15.25.2 Constructor & Destructor Documentation

15.25.2.1 CEventAdapter1394()

```

CEventAdapter1394 (
    INodeMap * pNodeMap = NULL ) [explicit]
  
```

constructor

15.25.2.2 ~CEventAdapter1394()

```

virtual ~CEventAdapter1394 ( ) [virtual]
  
```


15.25.3 Member Function Documentation

15.25.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    EventData1394 & Event,
    uint32_t numBytes )
```

distributes events to node map

15.25.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

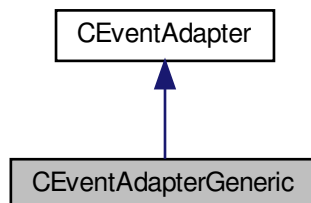
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter1394.h](#)

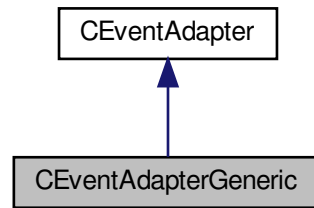
15.26 CEventAdapterGeneric Class Reference

Connects a generic event to a node map.

Inheritance diagram for CEventAdapterGeneric:



Collaboration diagram for CEventAdapterGeneric:



Public Member Functions

- [CEventAdapterGeneric](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterGeneric](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes, const [GenICam::gcstring](#) &EventID)
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes, uint64_t EventID)

Additional Inherited Members

15.26.1 Detailed Description

Connects a generic event to a node map.

15.26.2 Constructor & Destructor Documentation

15.26.2.1 CEventAdapterGeneric()

```

CEventAdapterGeneric (
    INodeMap * pNodeMap = NULL )
  
```

Constructor.

15.26.2.2 ~CEventAdapterGeneric()

```
virtual ~CEventAdapterGeneric ( ) [virtual]
```

Destructor.

15.26.3 Member Function Documentation

15.26.3.1 DeliverMessage() [1/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

15.26.3.2 DeliverMessage() [2/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    const GenICam::gcstring & EventID ) [virtual]
```

15.26.3.3 DeliverMessage() [3/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    uint64_t EventID ) [virtual]
```

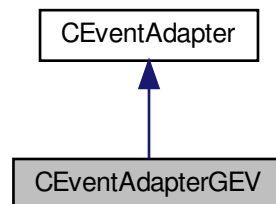
The documentation for this class was generated from the following file:

- `include/SpinGenApi/EventAdapterGeneric.h`

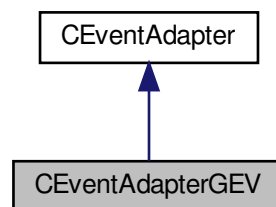
15.27 CEventAdapterGEV Class Reference

Connects a GigE Event to a node map.

Inheritance diagram for CEventAdapterGEV:



Collaboration diagram for CEventAdapterGEV:



Public Member Functions

- [CEventAdapterGEV](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterGEV](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (const [GVCP_EVENT_REQUEST](#) *pEvent)
Delivers the Events listed in the Event packet.
- void [DeliverEventMessage](#) (const [GVCP_EVENTDATA_REQUEST](#) *pEventData)
Delivers the Event + Data listed in the EventData packet.

Additional Inherited Members

15.27.1 Detailed Description

Connects a GigE Event to a node map.

15.27.2 Constructor & Destructor Documentation

15.27.2.1 CEventAdapterGEV()

```
CEventAdapterGEV (
    INodeMap * pNodeMap = NULL )
```

Constructor.

15.27.2.2 ~CEventAdapterGEV()

```
virtual ~CEventAdapterGEV ( ) [virtual]
```

Destructor.

15.27.3 Member Function Documentation

15.27.3.1 DeliverEventMessage() [1/2]

```
void DeliverEventMessage (
    const GVCP_EVENT_REQUEST * pEvent )
```

Delivers the Events listed in the Event packet.

15.27.3.2 DeliverEventMessage() [2/2]

```
void DeliverEventMessage (
    const GVCP_EVENTDATA_REQUEST * pEventData )
```

Delivers the Event + Data listed in the EventData packet.

15.27.3.3 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

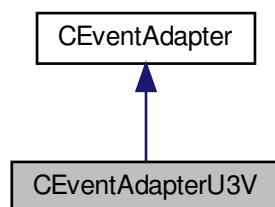
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

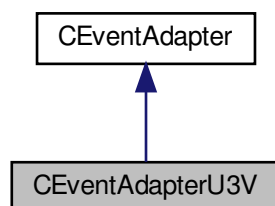
15.28 CEventAdapterU3V Class Reference

Connects a U3V Event to a node map.

Inheritance diagram for CEventAdapterU3V:



Collaboration diagram for CEventAdapterU3V:



Public Member Functions

- [CEventAdapterU3V](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterU3V](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (const [U3V_EVENT_MESSAGE](#) *pEventMessage)
Delivers the Event + Data listed in the packet.

Additional Inherited Members

15.28.1 Detailed Description

Connects a U3V Event to a node map.

15.28.2 Constructor & Destructor Documentation

15.28.2.1 CEventAdapterU3V()

```
CEventAdapterU3V (
    INodeMap * pNodeMap = NULL )
```

Constructor.

15.28.2.2 ~CEventAdapterU3V()

```
virtual ~CEventAdapterU3V ( ) [virtual]
```

Destructor.

15.28.3 Member Function Documentation

15.28.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    const U3V\_EVENT\_MESSAGE * pEventMessage )
```

Delivers the Event + Data listed in the packet.

15.28.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

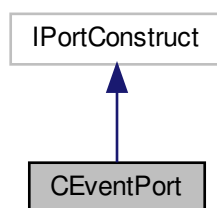
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

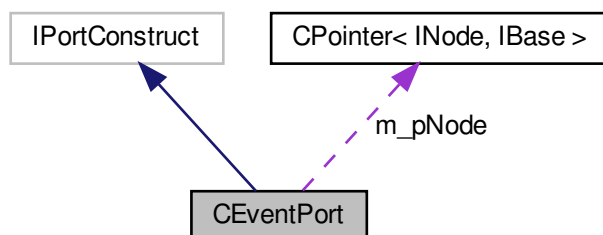
15.29 CEventPort Class Reference

Port attachable to an event.

Inheritance diagram for CEventPort:



Collaboration diagram for CEventPort:



Public Member Functions

- [CEventPort](#) ([INode](#) *pNode=NULL)
Constructor; can attach to a node.
- [~CEventPort](#) ()
Destructor; detaches from the port.
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Get the access mode of the node.
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- virtual void [SetPortImpl](#) (::Spinnaker::GenApi::IPort *pPort)
Called from the port node to give the chunk port a pointer to itself.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- void [InvalidateNode](#) ()
- bool [AttachNode](#) (::Spinnaker::GenApi::INode *pNode)
Attaches to the [Node](#).
- void [DetachNode](#) ()
Detaches from the [Node](#).
- int [GetEventIDLength](#) ()
Gets the EventID length.
- bool [CheckEventID](#) (uint8_t *pEventIDBuffer, int EventIDLength)
Checks if a EventID matches.
- bool [CheckEventID](#) (uint64_t EventID)
Checks if a EventID matches, version using uint64_t ID representation.
- void [AttachEvent](#) (uint8_t *pBaseAddress, int64_t [Length](#))
Attaches the an Event to the EventPort.
- void [DetachEvent](#) ()
Detaches the Event from the EventPort.

Protected Attributes

- [CNodePtr](#) [m_pNode](#)
- std::shared_ptr< [PortAdapter](#) > [m_pPortAdapter](#)
- void * [m_pEventPort](#)

15.29.1 Detailed Description

Port attachable to an event.

15.29.2 Constructor & Destructor Documentation

15.29.2.1 CEventPort()

```
CEventPort (
    INode * pNode = NULL )
```

Constructor; can attach to a node.

15.29.2.2 ~CEventPort()

```
~CEventPort ( )
```

Destructor; detaches from the port.

15.29.3 Member Function Documentation

15.29.3.1 AttachEvent()

```
void AttachEvent (
    uint8_t * pBaseAddress,
    int64_t Length )
```

Attaches the an Event to the EventPort.

15.29.3.2 AttachNode()

```
bool AttachNode (
    ::Spinnaker::GenApi::INode * pNode )
```

Attaches to the [Node](#).

15.29.3.3 CheckEventID() [1/2]

```
bool CheckEventID (
    uint8_t * pEventIDBuffer,
    int EventIDLength )
```

Checks if a EventID matches.

15.29.3.4 CheckEventID() [2/2]

```
bool CheckEventID (
    uint64_t EventID )
```

Checks if a EventID matches, version using uint64_t ID representation.

15.29.3.5 DetachEvent()

```
void DetachEvent ( )
```

Detaches the Event from the EventPort.

15.29.3.6 DetachNode()

```
void DetachNode ( )
```

Detaches from the [Node](#).

15.29.3.7 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

15.29.3.8 GetEventIDLength()

```
int GetEventIDLength ( )
```

Gets the EventID length.

15.29.3.9 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

15.29.3.10 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.29.3.11 InvalidateNode()

```
void InvalidateNode ( )
```

15.29.3.12 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

15.29.3.13 SetPortImpl()

```
virtual void SetPortImpl (
    ::Spinnaker::GenApi::IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

15.29.3.14 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

15.29.4 Member Data Documentation

15.29.4.1 m_pEventPort

```
void* m_pEventPort [protected]
```

15.29.4.2 m_pNode

```
CNodePtr m_pNode [protected]
```

15.29.4.3 m_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

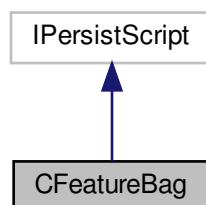
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventPort.h](#)

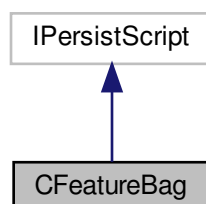
15.30 CFeatureBag Class Reference

Bag holding streamable features of a nodetree.

Inheritance diagram for CFeatureBag:



Collaboration diagram for CFeatureBag:



Public Member Functions

- [CFeatureBag](#) ()
- virtual [~CFeatureBag](#) ()
- virtual void [SetInfo](#) ([GenICam::gcstring](#) &Info)
sets information about the node map
- virtual void [PersistFeature](#) ([IValue](#) &item)
Stores a feature.
- bool [LoadFromBag](#) ([INodeMap](#) *pNodeMap, bool [Verify](#)=true, [GenICam::gcstring_vector](#) *pErrorList=NULL)
Loads the features from the bag to the node tree.
- int64_t [StoreToBag](#) ([INodeMap](#) *pNodeMap, const int MaxNumPersistSkriptEntries=-1)
Stores the streamable nodes to this feature bag.
- bool [operator==](#) (const [CFeatureBag](#) &FeatureBag) const
compares the content of two feature bags
- void * [GetFeatureBagHandle](#) ()

15.30.1 Detailed Description

Bag holding streamable features of a nodetree.

15.30.2 Constructor & Destructor Documentation

15.30.2.1 CFeatureBag()

```
CFeatureBag ( )
```

15.30.2.2 ~CFeatureBag()

```
virtual ~CFeatureBag ( ) [virtual]
```

15.30.3 Member Function Documentation

15.30.3.1 GetFeatureBagHandle()

```
void* GetFeatureBagHandle ( )
```

15.30.3.2 LoadFromBag()

```
bool LoadFromBag (
    INodeMap * pNodeMap,
    bool Verify = true,
    GenICam::gcstring\_vector * pErrorList = NULL )
```

Loads the features from the bag to the node tree.

Parameters

<i>pNodeMap</i>	The node map
<i>Verify</i>	If true, all streamable features are read back
<i>pErrorList</i>	If an error occurs during loading the error message is stored in the list and the loading continues

For Verify=true the list of names in the feature bag is replayed again. If a node is a selector it's value is set to the value from the feature bag. If not the value is read from the camera and compared with the value from the feature bag.

15.30.3.3 operator==()

```
bool operator== (
    const CFeatureBag & FeatureBag ) const
```

compares the content of two feature bags

15.30.3.4 PersistFeature()

```
virtual void PersistFeature (
    IValue & item ) [virtual]
```

Stores a feature.

15.30.3.5 SetInfo()

```
virtual void SetInfo (
    GenICam::gcstring & Info ) [virtual]
```

sets information about the node map

15.30.3.6 StoreToBag()

```
int64_t StoreToBag (
    INodeMap * pNodeMap,
    const int MaxNumPersistSkriptEntries = -1 )
```

Stores the streamable nodes to this feature bag.

Parameters

<i>pNodeMap</i>	The node map to persist
<i>MaxNumPersistSkriptEntries</i>	The max number of entries in the container; -1 means unlimited

Returns

number of entries in the bag

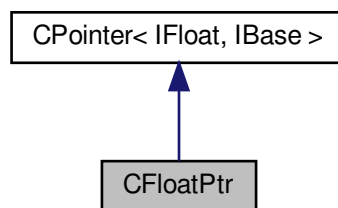
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Persistence.h](#)

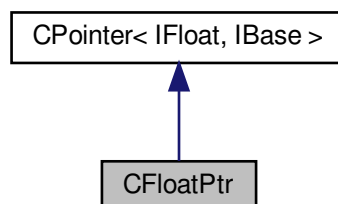
15.31 CFloatPtr Class Reference

SmartPointer for IFloat interface pointer.

Inheritance diagram for CFloatPtr:



Collaboration diagram for CFloatPtr:

**Public Member Functions**

- [CFloatPtr](#) () throw ()
Default constructor.
- [CFloatPtr](#) (IBase *pB)
Constructor from IBase pointer type.
- void [operator=](#) (IBase *pB)
Assign IBase Pointer.
- [IInteger](#) * [GetIntAlias](#) ()
gets the interface of an integer alias node.
- [IEnumeration](#) * [GetEnumAlias](#) ()
gets the interface of an enum alias node.

Additional Inherited Members

15.31.1 Detailed Description

SmartPointer for IFloat interface pointer.

15.31.2 Constructor & Destructor Documentation

15.31.2.1 CFloatPtr() [1/2]

```
CFloatPtr ( ) throw ( ) [inline]
```

Default constructor.

15.31.2.2 CFloatPtr() [2/2]

```
CFloatPtr (
    IBase * pB ) [inline]
```

Constructor from IBase pointer type.

15.31.3 Member Function Documentation

15.31.3.1 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( ) [inline]
```

gets the interface of an enum alias node.

15.31.3.2 GetIntAlias()

```
IInteger* GetIntAlias ( ) [inline]
```

gets the interface of an integer alias node.

15.31.3.3 operator=()

```
void operator= (
    IBase * pB ) [inline]
```

Assign IBase Pointer.

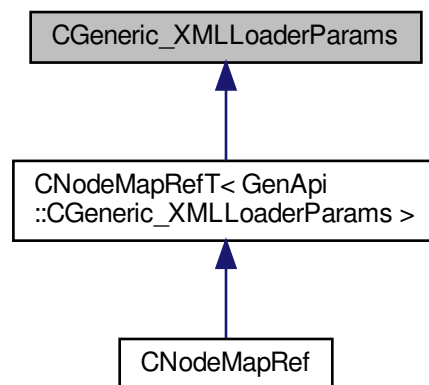
The documentation for this class was generated from the following file:

- include/SpinGenApi/Pointer.h

15.32 CGeneric_XMLLoaderParams Class Reference

Empty base class used by class [CNodeMapRef](#) as generic template argument.

Inheritance diagram for CGeneric_XMLLoaderParams:



Protected Member Functions

- virtual void [_Initialize](#) ([GenApi::INodeMap](#) *)

15.32.1 Detailed Description

Empty base class used by class [CNodeMapRef](#) as generic template argument.

15.32.2 Member Function Documentation

15.32.2.1 _Initialize()

```
virtual void _Initialize (
    GenApi::INodeMap * ) [inline], [protected], [virtual]
```

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

- include/SpinGenApi/NodeMapRef.h

15.33 CGlobalLock Class Reference

Named global lock which can be used over process boundaries.

Public Member Functions

- [CGlobalLock](#) (const char *pszName)
Creates a global lock object name pszName.
- [CGlobalLock](#) (const [gcstring](#) &strName)
Creates a global lock object name strName.
- [~CGlobalLock](#) ()
- bool [IsValid](#) (void) const
tests whether the lock is valid
- bool [Lock](#) (unsigned int timeout_ms)
enters the lock (may block)
- bool [TryLock](#) (void)
tries to enter the lock and returns immediately when not possible
- void [Unlock](#) (void)
leaves the lock

Protected Attributes

- long [m_DebugCount](#)

15.33.1 Detailed Description

Named global lock which can be used over process boundaries.

15.33.2 Constructor & Destructor Documentation

15.33.2.1 CGlobalLock() [1/2]

```
CGlobalLock (
    const char * pszName ) [explicit]
```

Creates a global lock object name pszName.

In case an object with the same name already exists a reference to the existing object will be created. If pszName is NULL an unnamed object will be created.

15.33.2.2 CGlobalLock() [2/2]

```
CGlobalLock (
    const gcstring & strName ) [explicit]
```

Creates a global lock object name strName.

In case an object with the same name already exists a reference to the existing object will be created. If strName is empty an unnamed object will be created.

15.33.2.3 ~CGlobalLock()

```
~CGlobalLock ( )
```

15.33.3 Member Function Documentation**15.33.3.1 IsValid()**

```
bool IsValid (
    void ) const
```

tests whether the lock is valid

15.33.3.2 Lock()

```
bool Lock (
    unsigned int timeout_ms )
```

enters the lock (may block)

15.33.3.3 TryLock()

```
bool TryLock (
    void )
```

tries to enter the lock and returns immediately when not possible

15.33.3.4 Unlock()

```
void Unlock (
    void )
```

leaves the lock

15.33.4 Member Data Documentation

15.33.4.1 m_DebugCount

```
long m_DebugCount [mutable], [protected]
```

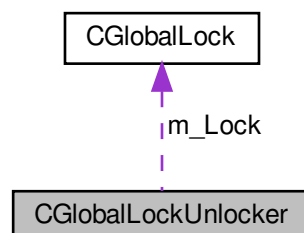
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

15.34 CGlobalLockUnlocker Class Reference

Unlocks the global lock object on destruction.

Collaboration diagram for CGlobalLockUnlocker:



Public Member Functions

- [CGlobalLockUnlocker](#) ([CGlobalLock](#) &lock)
- [~CGlobalLockUnlocker](#) ()
- void [UnlockEarly](#) (void)

This function allows to unlock the object early before the object is destroyed.

Protected Attributes

- [CGlobalLock](#) & [m_Lock](#)
- bool [m_enabled](#)

15.34.1 Detailed Description

Unlocks the global lock object on destruction.

This is for automatic UNLOCKING only. We can't do automatic locking here since there is no returnvalue for constructors

15.34.2 Constructor & Destructor Documentation

15.34.2.1 CGlobalLockUnlocker()

```
CGlobalLockUnlocker (
    CGlobalLock & lock ) [inline]
```

15.34.2.2 ~CGlobalLockUnlocker()

```
~CGlobalLockUnlocker ( ) [inline]
```

15.34.3 Member Function Documentation

15.34.3.1 UnlockEarly()

```
void UnlockEarly (
    void ) [inline]
```

This function allows to unlock the object early before the object is destroyed.

15.34.4 Member Data Documentation

15.34.4.1 m_enabled

`bool m_enabled` [protected]

15.34.4.2 m_Lock

`CGlobalLock& m_Lock` [protected]

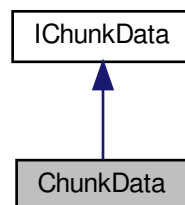
The documentation for this class was generated from the following file:

- `include/SpinGenApi/GCSynch.h`

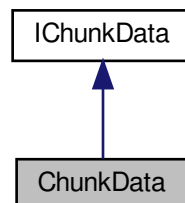
15.35 ChunkData Class Reference

The chunk data which contains additional information about an image.

Inheritance diagram for ChunkData:



Collaboration diagram for ChunkData:



Public Member Functions

- [ChunkData](#) ()
- [ChunkData](#) (const [ChunkData](#) &src)
- virtual [~ChunkData](#) (void)
- void [SetChunks](#) ([GenApi::INodeMap](#) &pNodeMap)
- [float64_t](#) [GetBlackLevel](#) () const
Description: Returns the black level used to capture the image.
- [int64_t](#) [GetFrameID](#) () const
Description: Returns the image frame ID.
- [float64_t](#) [GetExposureTime](#) () const
Description: Returns the exposure time used to capture the image.
- [int64_t](#) [GetCompressionMode](#) () const
Description: Returns the compression mode of the last image payload.
- [float64_t](#) [GetCompressionRatio](#) () const
Description: Returns the compression ratio of the last image payload.
- [int64_t](#) [GetTimestamp](#) () const
Description: Returns the Timestamp of the image.
- [int64_t](#) [GetExposureEndLineStatusAll](#) () const
Description: Returns the status of all the I/O lines at the end of exposure event.
- [int64_t](#) [GetWidth](#) () const
Description: Returns the width of the image included in the payload.
- [int64_t](#) [GetImage](#) () const
Description: Returns the image payload.
- [int64_t](#) [GetHeight](#) () const
Description: Returns the height of the image included in the payload.
- [float64_t](#) [GetGain](#) () const
Description: Returns the gain used to capture the image.
- [int64_t](#) [GetSequencerSetActive](#) () const
Description: Returns the index of the active set of the running sequencer included in the payload.
- [int64_t](#) [GetCRC](#) () const
Description: Returns the CRC of the image payload.
- [int64_t](#) [GetOffsetX](#) () const
Description: Returns the Offset X of the image included in the payload.
- [int64_t](#) [GetOffsetY](#) () const
Description: Returns the Offset Y of the image included in the payload.
- [int64_t](#) [GetSerialDataLength](#) () const
Description: Returns the length of the received serial data that was included in the payload.
- [int64_t](#) [GetPartSelector](#) () const
Description: Selects the part to access in chunk data in a multipart transmission.
- [int64_t](#) [GetPixelDynamicRangeMin](#) () const
Description: Returns the minimum value of dynamic range of the image included in the payload.
- [int64_t](#) [GetPixelDynamicRangeMax](#) () const
Description: Returns the maximum value of dynamic range of the image included in the payload.
- [int64_t](#) [GetTimestampLatchValue](#) () const
Description: Returns the last Timestamp latched with the TimestampLatch command.
- [int64_t](#) [GetLineStatusAll](#) () const
Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.
- [int64_t](#) [GetCounterValue](#) () const
Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.
- [float64_t](#) [GetTimerValue](#) () const

- Description: Returns the value of the selected Timer at the time of the FrameStart internal event.*

 - `int64_t GetScanLineSelector () const`

Description: Index for vector representation of one chunk value per line in an image.
- `int64_t GetEncoderValue () const`

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.
- `int64_t GetLinePitch () const`

Description: Returns the LinePitch of the image included in the payload.
- `int64_t GetTransferBlockID () const`

Description: Returns the unique identifier of the transfer block used to transport the payload.
- `int64_t GetTransferQueueCurrentBlockCount () const`

Description: Returns the current number of blocks in the transfer queue.
- `int64_t GetStreamChannelID () const`

Description: Returns identifier of the stream channel used to carry the block.
- `float64_t GetScan3dCoordinateScale () const`

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dCoordinateOffset () const`

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dInvalidDataValue () const`

Description: Returns the Invalid Data Value used for the image included in the payload.
- `float64_t GetScan3dAxisMin () const`

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dAxisMax () const`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dTransformValue () const`

Description: Returns the transform value.
- `float64_t GetScan3dCoordinateReferenceValue () const`

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.
- `int64_t GetInferenceFrameId () const`

Description: Returns the frame ID associated with the most recent inference result.
- `int64_t GetInferenceResult () const`

Description: Returns the chunk data inference result.
- `float64_t GetInferenceConfidence () const`

Description: Returns the chunk data inference confidence percentage.
- `InferenceBoundingBoxResult GetInferenceBoundingBoxResult () const`

Description: Returns the chunk inference bounding box result data.

Additional Inherited Members

15.35.1 Detailed Description

The chunk data which contains additional information about an image.

15.35.2 Constructor & Destructor Documentation

15.35.2.1 `ChunkData()` [1/2]

```
ChunkData ( )
```

15.35.2.2 `ChunkData()` [2/2]

```
ChunkData (
    const ChunkData & src )
```

15.35.2.3 `~ChunkData()`

```
virtual ~ChunkData (
    void ) [virtual]
```

15.35.3 Member Function Documentation

15.35.3.1 `GetBlackLevel()`

```
float64_t GetBlackLevel ( ) const [virtual]
```

Description: Returns the black level used to capture the image.

Visibility:

Implements [IChunkData](#).

15.35.3.2 `GetCompressionMode()`

```
int64_t GetCompressionMode ( ) const [virtual]
```

Description: Returns the compression mode of the last image payload.

Visibility:

Implements [IChunkData](#).

15.35.3.3 GetCompressionRatio()

```
float64_t GetCompressionRatio ( ) const [virtual]
```

Description: Returns the compression ratio of the last image payload.

Visibility:

Implements [IChunkData](#).

15.35.3.4 GetCounterValue()

```
int64_t GetCounterValue ( ) const [virtual]
```

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.5 GetCRC()

```
int64_t GetCRC ( ) const [virtual]
```

Description: Returns the CRC of the image payload.

Visibility:

Implements [IChunkData](#).

15.35.3.6 GetEncoderValue()

```
int64_t GetEncoderValue ( ) const [virtual]
```

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.7 GetExposureEndLineStatusAll()

```
int64_t GetExposureEndLineStatusAll ( ) const [virtual]
```

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

Implements [IChunkData](#).

15.35.3.8 GetExposureTime()

```
float64_t GetExposureTime ( ) const [virtual]
```

Description: Returns the exposure time used to capture the image.

Visibility:

Implements [IChunkData](#).

15.35.3.9 GetFrameID()

```
int64_t GetFrameID ( ) const [virtual]
```

Description: Returns the image frame ID.

Visibility:

Implements [IChunkData](#).

15.35.3.10 GetGain()

```
float64_t GetGain ( ) const [virtual]
```

Description: Returns the gain used to capture the image.

Visibility:

Implements [IChunkData](#).

15.35.3.11 GetHeight()

```
int64_t GetHeight ( ) const [virtual]
```

Description: Returns the height of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.35.3.12 GetImage()

```
int64_t GetImage ( ) const [virtual]
```

Description: Returns the image payload.

Visibility:

Implements [IChunkData](#).

15.35.3.13 GetInferenceBoundingBoxResult()

```
InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const [virtual]
```

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.14 GetInferenceConfidence()

```
float64\_t GetInferenceConfidence ( ) const [virtual]
```

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.15 GetInferenceFrameId()

```
int64_t GetInferenceFrameId ( ) const [virtual]
```

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.16 GetInferenceResult()

```
int64_t GetInferenceResult ( ) const [virtual]
```

Description: Returns the chunk data inference result.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.17 GetLinePitch()

```
int64_t GetLinePitch ( ) const [virtual]
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.18 GetLineStatusAll()

```
int64_t GetLineStatusAll ( ) const [virtual]
```

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.19 GetOffsetX()

```
int64_t GetOffsetX ( ) const [virtual]
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.35.3.20 GetOffsetY()

```
int64_t GetOffsetY ( ) const [virtual]
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.35.3.21 GetPartSelector()

```
int64_t GetPartSelector ( ) const [virtual]
```

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.22 GetPixelDynamicRangeMax()

```
int64_t GetPixelDynamicRangeMax ( ) const [virtual]
```

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.23 GetPixelDynamicRangeMin()

```
int64_t GetPixelDynamicRangeMin ( ) const [virtual]
```

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.24 GetScan3dAxisMax()

```
float64_t GetScan3dAxisMax ( ) const [virtual]
```

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.25 GetScan3dAxisMin()

```
float64_t GetScan3dAxisMin ( ) const [virtual]
```

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.26 GetScan3dCoordinateOffset()

```
float64_t GetScan3dCoordinateOffset ( ) const [virtual]
```

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.27 GetScan3dCoordinateReferenceValue()

```
float64_t GetScan3dCoordinateReferenceValue ( ) const [virtual]
```

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.28 GetScan3dCoordinateScale()

```
float64_t GetScan3dCoordinateScale ( ) const [virtual]
```

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.29 GetScan3dInvalidDataValue()

```
float64_t GetScan3dInvalidDataValue ( ) const [virtual]
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.30 GetScan3dTransformValue()

```
float64_t GetScan3dTransformValue ( ) const [virtual]
```

Description: Returns the transform value.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.31 GetScanLineSelector()

```
int64_t GetScanLineSelector ( ) const [virtual]
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.32 GetSequencerSetActive()

```
int64_t GetSequencerSetActive ( ) const [virtual]
```

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

Implements [IChunkData](#).

15.35.3.33 GetSerialDataLength()

```
int64_t GetSerialDataLength ( ) const [virtual]
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

Implements [IChunkData](#).

15.35.3.34 GetStreamChannelID()

```
int64_t GetStreamChannelID ( ) const [virtual]
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.35 GetTimerValue()

```
float64_t GetTimerValue ( ) const [virtual]
```

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.36 GetTimestamp()

```
int64_t GetTimestamp ( ) const [virtual]
```

Description: Returns the Timestamp of the image.

Visibility:

Implements [IChunkData](#).

15.35.3.37 GetTimestampLatchValue()

```
int64_t GetTimestampLatchValue ( ) const [virtual]
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.38 GetTransferBlockID()

```
int64_t GetTransferBlockID ( ) const [virtual]
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.39 GetTransferQueueCurrentBlockCount()

```
int64_t GetTransferQueueCurrentBlockCount ( ) const [virtual]
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

Implements [IChunkData](#).

15.35.3.40 GetWidth()

```
int64_t GetWidth ( ) const [virtual]
```

Description: Returns the width of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.35.3.41 SetChunks()

```
void SetChunks (
    GenApi::INodeMap & pNodeMap ) [virtual]
```

Implements [IChunkData](#).

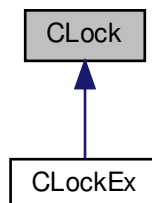
The documentation for this class was generated from the following file:

- [include/ChunkData.h](#)

15.36 CLock Class Reference

A lock class.

Inheritance diagram for CLock:



Public Member Functions

- `CLock ()`
Constructor.
- `~CLock ()`
Destructor.
- `bool TryLock ()`
tries to enter the critical section; returns true if successful
- `void Lock ()`
enters the critical section (may block)
- `void Unlock ()`
leaves the critical section

15.36.1 Detailed Description

A lock class.

15.36.2 Constructor & Destructor Documentation

15.36.2.1 CLock()

`CLock ( )`

Constructor.

15.36.2.2 ~CLock()

`~CLock ( )`

Destructor.

15.36.3 Member Function Documentation

15.36.3.1 Lock()

`void Lock ( )`

enters the critical section (may block)

15.36.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

15.36.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

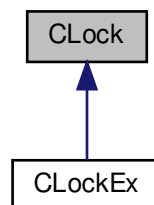
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

15.37 CLock Class Reference

A lock class.

Inheritance diagram for CLock:



Public Member Functions

- [CLock](#) ()
Constructor.
- [CLock](#) (void *pLock)
Constructor.
- [~CLock](#) ()
Destructor.
- bool [TryLock](#) ()
tries to enter the critical section; returns true if successful
- void [Lock](#) ()
enters the critical section (may block)
- void [Unlock](#) ()
leaves the critical section

Protected Attributes

- void * [m_lock](#)
- bool [m_bOwnLock](#)

Friends

- class [NodeMap](#)

15.37.1 Detailed Description

A lock class.

15.37.2 Constructor & Destructor Documentation

15.37.2.1 CLock() [1/2]

[CLock](#) ()

Constructor.

15.37.2.2 CLock() [2/2]

[CLock](#) (
 void * *pLock*)

Constructor.

15.37.2.3 ~CLock()

[~CLock](#) ()

Destructor.

15.37.3 Member Function Documentation

15.37.3.1 Lock()

```
void Lock ( )
```

enters the critical section (may block)

15.37.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

15.37.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

15.37.4 Friends And Related Function Documentation

15.37.4.1 NodeMap

```
friend class NodeMap [friend]
```

15.37.5 Member Data Documentation

15.37.5.1 m_bOwnLock

```
bool m_bOwnLock [protected]
```

15.37.5.2 m_lock

```
void* m_lock [protected]
```

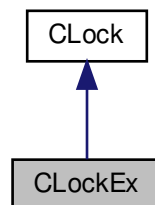
The documentation for this class was generated from the following file:

- [include/SpinGenApi/Synch.h](#)

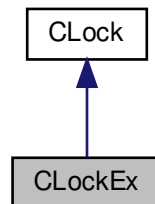
15.38 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Additional Inherited Members

15.38.1 Detailed Description

This class is for testing purposes only.

It should not be used for client code because it exists only for Windows but not for Linux since it uses internal data structures of a Win32 object

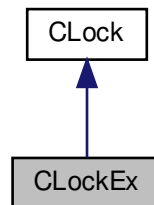
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

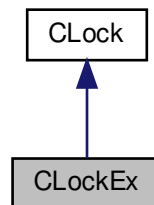
15.39 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Protected Attributes

- void * [m_lockEx](#)

Additional Inherited Members

15.39.1 Detailed Description

This class is for testing purposes only.

It should not be used for client code because it exists only for Windows but not for Linux since it uses internal data structures of a Win32 object

15.39.2 Member Data Documentation

15.39.2.1 m_lockEx

```
void* m_lockEx [protected]
```

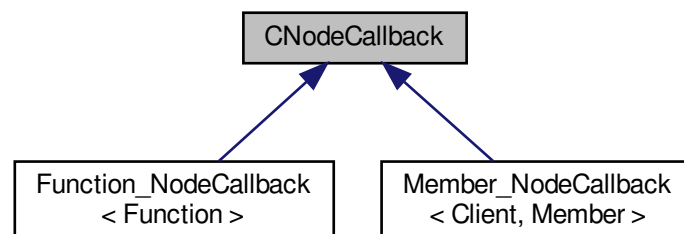
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

15.40 CNodeCallback Class Reference

callback body instance for INode pointers

Inheritance diagram for CNodeCallback:



Public Member Functions

- [CNodeCallback](#) ([INode](#) *pNode, [ECallbackType](#) CallbackType)
- virtual [~CNodeCallback](#) ()
virtual destructor
- virtual void [operator\(\)](#) ([ECallbackType](#) CallbackType) const =0
fires the callback if th type is right
- virtual void [Destroy](#) ()=0
destroys the object
- [INode](#) * [GetNode](#) ()
returns the node the callback is registered to
- [ECallbackType](#) [GetCallbackType](#) ()

Protected Attributes

- [INode * m_pNode](#)
the node were the callback is installed
- [ECallbackType m_CallbackType](#)
the type of the callback

15.40.1 Detailed Description

callback body instance for INode pointers

15.40.2 Constructor & Destructor Documentation

15.40.2.1 CNodeCallback()

```
CNodeCallback (
    INode * pNode,
    ECallbackType CallbackType ) [inline]
```

15.40.2.2 ~CNodeCallback()

```
virtual ~CNodeCallback ( ) [inline], [virtual]
```

virtual destructor

15.40.3 Member Function Documentation

15.40.3.1 Destroy()

```
virtual void Destroy ( ) [pure virtual]
```

destroys the object

Implemented in [Member_NodeCallback< Client, Member >](#), and [Function_NodeCallback< Function >](#).

15.40.3.2 GetCallbackType()

```
ECallbackType GetCallbackType ( ) [inline]
```

15.40.3.3 GetNode()

```
INode* GetNode ( ) [inline]
```

returns the node the callback is registered to

15.40.3.4 operator()

```
virtual void operator() (
    ECallbackType CallbackType ) const [pure virtual]
```

fires the callback if th type is right

Implemented in [Member_NodeCallback< Client, Member >](#), and [Function_NodeCallback< Function >](#).

15.40.4 Member Data Documentation

15.40.4.1 m_CallbackType

```
ECallbackType m_CallbackType [protected]
```

the type of the callback

15.40.4.2 m_pNode

```
INode* m_pNode [protected]
```

the node were the callback is installed

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

- [include/SpinGenApi/NodeCallback.h](#)

15.41 CNodeMapFactory Class Reference

The node map factory is used for creating node maps from camera description files.

Classes

- struct [NodeStatistics_t](#)

Public Member Functions

- [CNodeMapFactory](#) ()
Creates an empty node map factory for assigning a non-empty node map factory later.
- virtual [~CNodeMapFactory](#) ()
Destroys the node map factory data if all references to the data have been released.
- [CNodeMapFactory](#) (const [CNodeMapFactory](#) &)
Creates another reference to the node map factory data.
- [CNodeMapFactory](#) & [operator=](#) (const [CNodeMapFactory](#) &)
Creates another reference to the assigned node map factory data.
- [CNodeMapFactory](#) ([EContentType_t](#) FileType, const [GenlCam::gcstring](#) &FileName, [ECacheUsage_↵](#) t CacheUsage=[CacheUsage_Automatic](#), bool SuppressStringsOnLoad=false)
Creates the node map factory and simply stores the full path to the provided camera description file data.
- [CNodeMapFactory](#) ([EContentType_t](#) ContentType, const void *pData, [size_t](#) DataSize, [ECacheUsage_↵](#) t CacheUsage=[CacheUsage_Automatic](#), bool SuppressStringsOnLoad=false)
Creates the node map factory and simply stores the pointer and the size of the provided camera description file data.
- [CNodeMapFactory](#) (const [GenlCam::gcstring](#) &XmlData, [ECacheUsage_t](#) CacheUsage=[CacheUsage_↵](#) Automatic, bool SuppressStringsOnLoad=false)
Creates the node map factory and copies the provided camera description file string.
- bool [IsEmpty](#) () const
Returns true if nothing is loaded ([IsLoaded\(\)](#)) and no source data is available, e.g.
- void [AddInjectionData](#) ([CNodeMapFactory](#) &injectionData)
Adds a node map factory representing a camera description file to inject.
- void [LoadAndInject](#) ()
Advanced: Loads, Parses, and Injects the camera description files recursively.
- bool [IsLoaded](#) () const
Can be used to check whether the [LoadAndInject\(\)](#) processing step has been performed.
- [CNodeMapFactory](#) [ExtractSubtree](#) (const [GenlCam::gcstring](#) &SubTreeRootNodeName, bool doRename↵ ToRoot=false)
The name of the node that represents the root of the subtree that shall be extracted.
- void [Preprocess](#) ()
Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the [CNode↵](#) DataMap (not part of the public interface).
- bool [IsPreprocessed](#) () const
Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.
- void [ReleaseCameraDescriptionFileData](#) ()
Advanced: Releases any in constructors provided camera description file data buffers or files.
- bool [IsCameraDescriptionFileDataReleased](#) () const
Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.
- [INodeMap](#) * [CreateNodeMap](#) (const [GenlCam::gcstring](#) &DeviceName="Device", bool DoReleaseCamera↵ DescriptionFileData=true)
Creates a node map from the preprocessed memory internal representation of the camera description file(s).

- `INodeMap * CreateNodeMap (CLOCK &UserProvidedLock, const GenICam::gcstring &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true)`
Creates a node map from the preprocessed memory internal representation of the camera description file(s).
- `void GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions) const`
- `GenICam::gcstring ToString () const`
Outputs the pre-processed node map in string form (for debug purpose)
- `GenICam::gcstring ToXml () const`
Outputs the pre-processed node map in XML form (mainly for debug purpose)
- `void GetNodeStatistics (NodeStatistics_t &NodeStatistics)`
- `const GenICam::gcstring ApplyStyleSheet (const GenICam::gcstring &StyleSheetFileName)`
Applies a style sheet to the pre-processed node map.

Static Public Member Functions

- `static INodeMap * CreateEmptyNodeMap ()`
Creates an empty node map usable as placeholder, e.g.
- `static bool ClearCache ()`
Deletes all preprocessed camera description files from the cache.
- `static CNodeDataMap * CreateNodeDataFromNodeMap (INodeMap *pNodeMap)`

15.41.1 Detailed Description

The node map factory is used for creating node maps from camera description files.

Examples

```
// Simple node map creation from buffer, downloaded from a device for instance.
CNodeMapFactory cameraNodeMapFactory( ContentType_ZippedXml, buffer,
    bufferSize);

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// Node map creation with injecting additional xml fragments.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer,
    bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename2));

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// Node map creation and additional extraction of a category subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer,
    bufferSize);
// Extract a subtree for later chunk parsing.
CNodeMapFactory chunkDataNodeMapFactory = cameraParameters.ExtractSubtree("ChunkData");

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraParameters.CreateNodeMap();
// The next step is attaching the device port (not shown).
```

```

// Node map creation with injecting additional xml fragments and additional extraction of a category
subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer,
    bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename2));
CNodeMapFactory chunkDataNodeMapFactory = cameraNodeMapFactory.ExtractSubtree("ChunkData");

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// A node map factory can create multiple node maps from the provided camera description file(s).
for(int i = 0; i < 20; ++i)
{
    INodeMap* pNodeMapChunks = chunkDataNodeMapFactory.CreateNodeMap();
    //...
}

```

Attention

The is [CNodeMapFactory](#) not thread-safe.

You need to take care when camera description file data can be actually be freed, see method documentation of the node map factory for more detail.

15.41.2 Constructor & Destructor Documentation

15.41.2.1 CNodeMapFactory() [1/5]

```
CNodeMapFactory ( )
```

Creates an empty node map factory for assigning a non-empty node map factory later.

15.41.2.2 ~CNodeMapFactory()

```
virtual ~CNodeMapFactory ( ) [virtual]
```

Destroys the node map factory data if all references to the data have been released.

15.41.2.3 CNodeMapFactory() [2/5]

```
CNodeMapFactory (
    const CNodeMapFactory & )
```

Creates another reference to the node map factory data.

No data is copied.

15.41.2.4 CNodeMapFactory() [3/5]

```
CNodeMapFactory (
    EContentType_t FileType,
    const GenICam::gcstring & FileName,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and simply stores the full path to the provided camera description file data.

Parameters

in	<i>FileType</i>	Defines how the camera description file is stored, e.g. as zipped XML text.
in	<i>FileName</i>	The full path of the camera description file to process.
in	<i>CacheUsage</i>	Defines if and how to use the cache for preprocessed camera description files.
in	<i>SuppressStringsOnLoad</i>	Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint.

Throws an invalid argument exception if *FileName* is empty. Throws if environment variables in *FileName* cannot be resolved.

Attention

The given file must be readable until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

15.41.2.5 CNodeMapFactory() [4/5]

```
CNodeMapFactory (
    EContentType_t ContentType,
    const void * pData,
    size_t DataSize,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and simply stores the pointer and the size of the provided camera description file data.

Parameters

in	<i>ContentType</i>	Defines how the camera description file is stored, e.g. as zipped XML text.
in	<i>pData</i>	The pointer to the camera description file data.
in	<i>DataSize</i>	The size of the camera description file data.
in	<i>CacheUsage</i>	Defines if and how to use the cache for preprocessed camera description files.
in	<i>SuppressStringsOnLoad</i>	Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint.

Throws an invalid argument exception if *pData* is NULL or *DataSize* is 0.

Attention

The given buffer must not be freed or changed until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

15.41.2.6 CNodeMapFactory() [5/5]

```
CNodeMapFactory (
    const GenICam::gcstring & XmlData,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and copies the provided camera description file string.

Parameters

in	<i>XmlData</i>	The camera description file data as XML text. The provided text is copied. You can use the overloaded constructor accepting a buffer to avoid that. <code>gcstring cdfData; //... fill cdfData ... CNodeMapFactory factory(ContentType_Xml, cfdData.c_str(), cfdData.size()); // Create the node map. The node map can be destroyed using the IDestroy interface later. INodeMap* pNodeMap = factory.CreateNodeMap(); // The next step is attaching the device port (not shown).</code>
in	<i>CacheUsage</i>	Defines if and how to use the cache for preprocessed camera description files.
in	<i>SuppressStringsOnLoad</i>	Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint.

Throws an invalid argument exception if XmlData is empty.

15.41.3 Member Function Documentation

15.41.3.1 AddInjectionData()

```
void AddInjectionData (
    CNodeMapFactory & injectionData )
```

Adds a node map factory representing a camera description file to inject.

Parameters

in	<i>injectionData</i>	A node map factory representing a camera description file to inject.

The injected files are injected in the order they are added. InjectionData must not be preprocessed. The [IsPreprocessed\(\)](#) method can be used to check if preprocessing has been done before. The cache usage of injection data is ignored.

15.41.3.2 ApplyStyleSheet()

```
const GenICam::gcstring ApplyStyleSheet (
    const GenICam::gcstring & StyleSheetFileName )
```

Applies a style sheet to the pre-processed node map.

15.41.3.3 ClearCache()

```
static bool ClearCache ( ) [static]
```

Deletes all preprocessed camera description files from the cache.

15.41.3.4 CreateEmptyNodeMap()

```
static INodeMap* CreateEmptyNodeMap ( ) [static]
```

Creates an empty node map usable as placeholder, e.g.

if certain features are not supported by a module.

15.41.3.5 CreateNodeDataFromNodeMap()

```
static CNodeDataMap* CreateNodeDataFromNodeMap (
    INodeMap * pNodeMap ) [static]
```

15.41.3.6 CreateNodeMap() [1/2]

```
INodeMap* CreateNodeMap (
    const GenICam::gcstring & DeviceName = "Device",
    bool DoReleaseCameraDescriptionFileData = true )
```

Creates a node map from the preprocessed memory internal representation of the camera description file(s).

[Preprocess\(\)](#) is automatically called if needed. The preprocess step can be omitted by the factory depending on the cache mode setting when a cache file is available, then the cache file is read and converted directly into a node map. [ReleaseCameraDescriptionFileData\(\)](#) is called if `DoReleaseCameraDescriptionFileData` is true. This method can be called multiple times to create multiple instances of a node map.

15.41.3.7 CreateNodeMap() [2/2]

```
INodeMap* CreateNodeMap (
    CLock & UserProvidedLock,
    const GenICam::gcstring & DeviceName = "Device",
    bool DoReleaseCameraDescriptionFileData = true )
```

Creates a node map from the preprocessed memory internal representation of the camera description file(s).

[Preprocess\(\)](#) is automatically called if needed. The preprocess step can be omitted by the factory depending on the cache mode setting when a cache file is available, then the cache file is read and converted directly into a node map. [ReleaseCameraDescriptionFileData\(\)](#) is called if `DoReleaseCameraDescriptionFileData` is true. This method can be called multiple times to create multiple instances of a node map. This method allows to provide an external lock to avoid using too many locks in an application.

Attention

The provided lock must not be destroyed before the created node map.

15.41.3.8 ExtractSubtree()

```
CNodeMapFactory ExtractSubtree (
    const GenICam::gcstring & SubTreeRootNodeName,
    bool doRenameToRoot = false )
```

The name of the node that represents the root of the subtree that shall be extracted.

Parameters

in	<i>SubTreeRootNodeName</i>	The root of the branch to extract, e.g. "ChunkData".
in	<i>doRenameToRoot</i>	Renames the extracted subtree root node SubTreeRootNodeName to "Root", sets the IsFeature property. Preprocess() is automatically called if needed to create the memory internal representation of the camera description file(s). The preprocessed result can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting.

15.41.3.9 GetNodeStatistics()

```
void GetNodeStatistics (
    NodeStatistics_t & NodeStatistics )
```

15.41.3.10 GetSupportedSchemaVersions()

```
void GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) const
```

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers. Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only.

15.41.3.11 IsCameraDescriptionFileDataReleased()

```
bool IsCameraDescriptionFileDataReleased ( ) const
```

Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.

15.41.3.12 IsEmpty()

```
bool IsEmpty ( ) const
```

Returns true if nothing is loaded ([IsLoaded\(\)](#)) and no source data is available, e.g.

when the node map factory has been created with the default constructor.

15.41.3.13 IsLoaded()

```
bool IsLoaded ( ) const
```

Can be used to check whether the [LoadAndInject\(\)](#) processing step has been performed.

Returns true if [IsPreprocessed\(\)](#) returns true (Preprocessed Data has been loaded from cache).

15.41.3.14 IsPreprocessed()

```
bool IsPreprocessed ( ) const
```

Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.

15.41.3.15 LoadAndInject()

```
void LoadAndInject ( )
```

Advanced: Loads, Parses, and Injects the camera description files recursively.

The result is a memory internal representation of the camera description file(s), the CNodeDataMap (not part of the public interface).

This step is usually done automatically. Prevents cache read if called manually.

15.41.3.16 operator=()

```
CNodeMapFactory& operator= (
    const CNodeMapFactory & )
```

Creates another reference to the assigned node map factory data.

Destroys the "overwritten" node map factory data if all references to the data have been released.

15.41.3.17 Preprocess()

```
void Preprocess ( )
```

Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the CNodeDataMap (not part of the public interface).

This step is usually done automatically. Preprocessed data can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting. By calling this method directly direct cache load is suppressed, see [CreateNodeMap\(\)](#) for more information.

15.41.3.18 ReleaseCameraDescriptionFileData()

```
void ReleaseCameraDescriptionFileData ( )
```

Advanced: Releases any in constructors provided camera description file data buffers or files.

This step is usually done automatically. All references to added injection data are dropped in this step to free the data. After this step any in constructors provided buffers can be freed or any in constructors given files can be deleted.

15.41.3.19 ToString()

```
GenICam::gcstring ToString ( ) const
```

Outputs the pre-processed node map in string form (for debug purpose)

15.41.3.20 ToXml()

```
GenICam::gcstring ToXml ( ) const
```

Outputs the pre-processed node map in XML form (mainly for debug purpose)

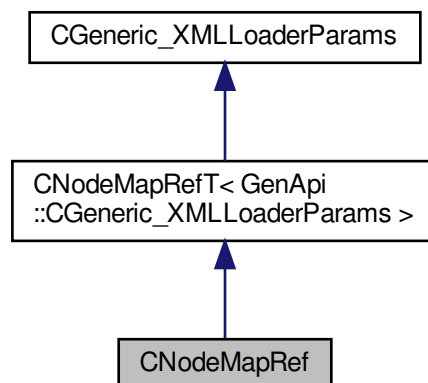
The documentation for this class was generated from the following file:

- [include/SpinGenApi/NodeMapFactory.h](#)

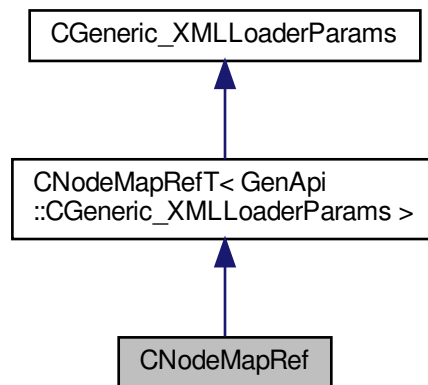
15.42 CNodeMapRef Class Reference

Smartpointer for NodeMaps with create function.

Inheritance diagram for CNodeMapRef:



Collaboration diagram for CNodeMapRef:



Public Member Functions

- **CNodeMapRef** (const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- **CNodeMapRef** (INodeMap *pNodeMap, const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- **CNodeMapRef** (const **CNodeMapRef** &Them)
Copy constructor.
- **CNodeMapRef** & **operator=** (const **CNodeMapRef** &Them)
Assignment.
- **CNodeMapRef** & **operator=** (INodeMap *pNodeMap)
Assignment of an INodeMap.*

Additional Inherited Members

15.42.1 Detailed Description

Smartpointer for NodeMaps with create function.

Note

This class is a simple typedef definition. The class syntax is only used, because Doxygen has to generate a useful documentation.

15.42.2 Constructor & Destructor Documentation

15.42.2.1 CNodeMapRef() [1/3]

```
CNodeMapRef (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

15.42.2.2 CNodeMapRef() [2/3]

```
CNodeMapRef (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

15.42.2.3 CNodeMapRef() [3/3]

```
CNodeMapRef (
    const CNodeMapRef & Them ) [inline]
```

Copy constructor.

15.42.3 Member Function Documentation

15.42.3.1 operator=() [1/2]

```
CNodeMapRef& operator= (
    const CNodeMapRef & Them ) [inline]
```

Assignment.

15.42.3.2 operator=() [2/2]

```
CNodeMapRef& operator= (
    INodeMap * pNodeMap ) [inline]
```

Assignment of an INodeMap*.

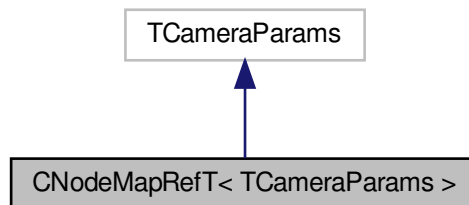
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

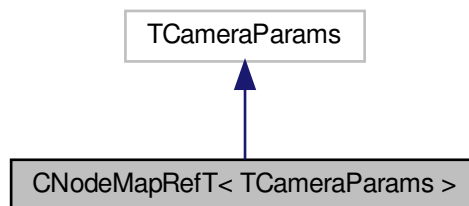
15.43 CNodeMapRefT< TCameraParams > Class Template Reference

Smartpointer template for NodeMaps with create function.

Inheritance diagram for CNodeMapRefT< TCameraParams >:



Collaboration diagram for CNodeMapRefT< TCameraParams >:



Public Member Functions

- `CNodeMapRefT` (const `GenICam::gcstring` &DeviceName="Device")
Constructor.
- `CNodeMapRefT` (`INodeMap` *pNodeMap, const `GenICam::gcstring` &DeviceName="Device")
Constructor.
- `CNodeMapRefT` (const `CNodeMapRefT` &Them)
Copy constructor.
- `CNodeMapRefT` & `operator=` (const `CNodeMapRefT` &Them)
Assignment.
- `CNodeMapRefT` & `operator=` (`INodeMap` *pNodeMap)
Assignment of an INodeMap.*
- virtual `~CNodeMapRefT` ()
Destructor.
- void `_Destroy` ()

- Destroys the node map.*
- void `_LoadXMLFromFile` (const `GenICam::gcstring` &FileName)
Creates the object from a XML file with given file name.
- void `_LoadXMLFromZIPFile` (const `GenICam::gcstring` &ZipFileName)
Creates the object from a ZIP'd XML file with given file name.
- void `_LoadXMLFromZIPData` (const void *zipData, size_t zipSize)
Creates the object from a ZIP'd XML file given in a string.
- void `_LoadXMLFromFileInject` (const `GenICam::gcstring` &TargetFileName, const `GenICam::gcstring` &InjectFileName)
Creates the object from a XML target and an inject file with given file name.
- void `_LoadXMLFromString` (const `GenICam::gcstring` &XMLData)
Creates the object from XML data given in a string.
- void `_LoadXMLFromStringInject` (const `GenICam::gcstring` &TargetXMLDataconst, const `GenICam::gcstring` &InjectXMLData)
Creates the object from XML data given in a string with injection.
- virtual void `_GetSupportedSchemaVersions` (`GenICam::gcstring_vector` &SchemaVersions)
Gets a list of supported schema versions.
- virtual `GenICam::gcstring` `_GetDeviceName` ()
Get device name.
- virtual void `_Poll` (int64_t ElapsedTime)
Fires nodes which have a polling time.
- virtual void `_GetNodes` (`NodeList_t` &Nodes)
Retrieves all nodes in the node map.
- virtual `Inode *` `_GetNode` (const `GenICam::gcstring` &key)
Retrieves the node from the central map by name.
- virtual void `_InvalidateNodes` ()
Invalidates all nodes.
- virtual bool `_Connect` (`IPort` *pPort, const `GenICam::gcstring` &PortName)
Connects a port to a port node with given name.
- virtual bool `_Connect` (`IPort` *pPort)
Connects a port to the standard port "Device".

Static Public Member Functions

- static bool `_ClearXMLCache` ()
Clears the cache of the camera description files.

Public Attributes

- `InodeMap *` `_Ptr`
Pointer to the `NodeMap`.

15.43.1 Detailed Description

```
template<class TCameraParams>
class Spinnaker::GenApi::CNodeMapRefT< TCameraParams >
```

Smartpointer template for NodeMaps with create function.

Parameters

<i>TCameraParams</i>	The camera specific parameter class (auto generated from camera xml file)

15.43.2 Member Function Documentation

15.43.2.1 _ClearXMLCache()

```
static bool _ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

15.43.2.2 _Connect() [1/2]

```
virtual bool _Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) [virtual]
```

Connects a port to a port node with given name.

15.43.2.3 _Connect() [2/2]

```
virtual bool _Connect (
    IPort * pPort ) [virtual]
```

Connects a port to the standard port "Device".

15.43.2.4 _GetDeviceName()

```
virtual GenICam::gcstring _GetDeviceName ( ) [virtual]
```

Get device name.

15.43.2.5 _GetNode()

```
virtual INode\* _GetNode (
    const GenICam::gcstring & key ) [virtual]
```

Retrieves the node from the central map by name.

15.43.2.6 _GetNodes()

```
virtual void _GetNodes (
    NodeList\_t & Nodes ) [virtual]
```

Retrieves all nodes in the node map.

15.43.2.7 _GetSupportedSchemaVersions()

```
virtual void _GetSupportedSchemaVersions (
    GenICam::gcstring\_vector & SchemaVersions ) [virtual]
```

Gets a list of supported schema versions.

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers. Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only.

15.43.2.8 _InvalidateNodes()

```
virtual void _InvalidateNodes ( ) [virtual]
```

Invalidates all nodes.

15.43.2.9 _LoadXMLFromFile()

```
void _LoadXMLFromFile (
    const GenICam::gcstring & FileName )
```

Creates the object from a XML file with given file name.

15.43.2.10 _LoadXMLFromFileInject()

```
void _LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

15.43.2.11 _LoadXMLFromString()

```
void _LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

15.43.2.12 _LoadXMLFromStringInject()

```
void _LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

15.43.2.13 _LoadXMLFromZIPData()

```
void _LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

15.43.2.14 _LoadXMLFromZIPFile()

```
void _LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

15.43.2.15 _Poll()

```
virtual void _Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

15.43.3 Member Data Documentation

15.43.3.1 _Ptr

```
INodeMap* _Ptr
```

Pointer to the [NodeMap](#).

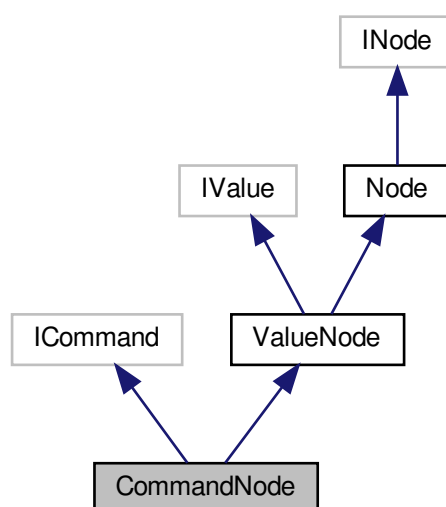
The documentation for this class was generated from the following file:

- [include/SpinGenApi/NodeMapRef.h](#)

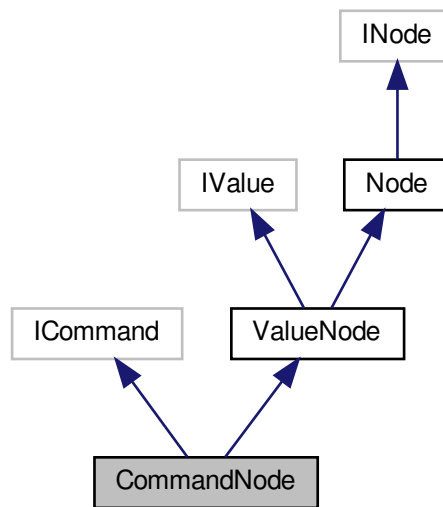
15.44 CommandNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for CommandNode:



Collaboration diagram for CommandNode:



Public Member Functions

- [CommandNode](#) ()
- [CommandNode](#) (std::shared_ptr< Node::NodeImpl > pCommand)
- virtual [~CommandNode](#) ()
- virtual void [Execute](#) (bool [Verify](#)=true)
Execute the command.
- virtual void [operator\(\)](#) ()
Execute the command.
- virtual bool [IsDone](#) (bool [Verify](#)=true)
Query whether the command is executed.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.44.1 Detailed Description

[Interface](#) for string properties.

15.44.2 Constructor & Destructor Documentation

15.44.2.1 CommandNode() [1/2]

```
CommandNode ( )
```

15.44.2.2 CommandNode() [2/2]

```
CommandNode (
    std::shared_ptr< Node::NodeImpl > pCommand )
```

15.44.2.3 ~CommandNode()

```
virtual ~CommandNode ( ) [virtual]
```

15.44.3 Member Function Documentation

15.44.3.1 Execute()

```
virtual void Execute (
    bool Verify = true ) [virtual]
```

Execute the command.

Parameters

<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.44.3.2 IsDone()

```
virtual bool IsDone (
    bool Verify = true ) [virtual]
```

Query whether the command is executed.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked

Returns

True if the Execute command has finished; false otherwise

15.44.3.3 operator()

```
virtual void operator() ( ) [virtual]
```

Execute the command.

15.44.3.4 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

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

- include/SpinGenApi/[CommandNode.h](#)

15.45 CompressedImageInfo Struct Reference

Public Member Functions

- [CompressedImageInfo](#) (const string &[fileName](#), size_t imageSize, size_t width, size_t height, size_t xOffset, size_t yOffset, [PixelFormatEnums](#) pixelFormat)

Public Attributes

- string [fileName](#)
- size_t [compressedImageSize](#)
- size_t [imageWidth](#)
- size_t [imageHeight](#)
- size_t [imageXOffset](#)
- size_t [imageYOffset](#)
- [PixelFormatEnums](#) [imagePixelFormat](#)

15.45.1 Constructor & Destructor Documentation

15.45.1.1 CompressedImageInfo()

```
CompressedImageInfo (
    const string & fileName,
    size_t imageSize,
    size_t width,
    size_t height,
    size_t xOffset,
    size_t yOffset,
    PixelFormatEnums pixelFormat ) [inline]
```

15.45.2 Member Data Documentation

15.45.2.1 compressedImageSize

size_t compressedImageSize

15.45.2.2 fileName

string fileName

15.45.2.3 imageHeight

size_t imageHeight

15.45.2.4 imagePixelFormat

PixelFormatEnums imagePixelFormat

15.45.2.5 imageWidth

size_t imageWidth

15.45.2.6 imageXOffset

```
size_t imageXOffset
```

15.45.2.7 imageYOffset

```
size_t imageYOffset
```

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

- src/Compression/[Compression.cpp](#)

15.46 Counter Class Reference

Definition of a simple [Counter](#) class.

Public Member Functions

- [Counter](#) ()
- unsigned int [GetValue](#) () const
- unsigned int [operator++](#) ()
- unsigned int [operator++](#) (int)
- unsigned int [operator--](#) (int)
- unsigned int [operator--](#) ()
- [operator unsigned int](#) ()
- bool [IsZero](#) ()

15.46.1 Detailed Description

Definition of a simple [Counter](#) class.

15.46.2 Constructor & Destructor Documentation

15.46.2.1 Counter()

```
Counter ( ) [inline]
```

15.46.3 Member Function Documentation

15.46.3.1 GetValue()

```
unsigned int GetValue ( ) const [inline]
```

15.46.3.2 IsZero()

```
bool IsZero ( ) [inline]
```

15.46.3.3 operator unsigned int()

```
operator unsigned int ( ) [inline]
```

15.46.3.4 operator++() [1/2]

```
unsigned int operator++ ( ) [inline]
```

15.46.3.5 operator++() [2/2]

```
unsigned int operator++ (
    int ) [inline]
```

15.46.3.6 operator--() [1/2]

```
unsigned int operator-- (
    int ) [inline]
```

15.46.3.7 operator--() [2/2]

```
unsigned int operator-- ( ) [inline]
```

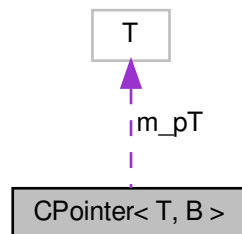
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Counter.h](#)

15.47 CPointer< T, B > Class Template Reference

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

Collaboration diagram for CPointer< T, B >:



Public Member Functions

- [CPointer](#) (void)
Default constructor.
- [CPointer](#) (B *pB)
Constructor from INode pointer type.
- virtual [~CPointer](#) (void)
- void [operator=](#) (B *pB)
Assign INode Pointer.
- [operator T*](#) (void) const
Dereferencing.
- T & [operator*](#) (void) const
Dereferencing.
- T & [operator\(\)](#) (void) const
Dereferencing.
- T * [operator->](#) (void) const
Dereferencing.
- bool [IsValid](#) () const throw ()
true if the pointer is valid
- [operator bool](#) (void) const throw ()
true if the pointer is valid
- bool [operator==](#) (T *pT) const
pointer equal
- bool [operator==](#) (const [CPointer](#)< T, B > &rT) const
pointer equal
- bool [operator==](#) (int nMustBeNull) const
pointer equal
- bool [operator!=](#) (const [CPointer](#)< T, B > &rT) const
pointer unequal
- bool [operator!=](#) (T *pT) const

- pointer unequal*
- bool `operator!=` (const long int nMustBeNull) const
- pointer unequal*
- bool `operator!=` (const int nMustBeNull) const
- pointer unequal*
- bool `operator!=` (const std::nullptr_t nullPtr) const
- pointer unequal*

Protected Attributes

- T * `m_pT`
Underlying raw pointer.

15.47.1 Detailed Description

```
template<class T, class B = IBase>
class Spinnaker::GenApi::CPointer< T, B >
```

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

15.47.2 Constructor & Destructor Documentation

15.47.2.1 CPointer() [1/2]

```
CPointer (
    void ) [inline]
```

Default constructor.

15.47.2.2 CPointer() [2/2]

```
CPointer (
    B * pB ) [inline]
```

Constructor from INode pointer type.

15.47.2.3 ~CPointer()

```
virtual ~CPointer (
    void ) [inline], [virtual]
```

15.47.3 Member Function Documentation

15.47.3.1 IsValid()

```
bool IsValid ( ) const throw )    [inline]
```

true if the pointer is valid

15.47.3.2 operator bool()

```
operator bool (
    void ) const throw )    [inline]
```

true if the pointer is valid

15.47.3.3 operator T*()

```
operator T* (
    void ) const    [inline]
```

Dereferencing.

15.47.3.4 operator!=() [1/5]

```
bool operator!= (
    const CPointer< T, B > & rT ) const    [inline]
```

pointer inequal

15.47.3.5 operator!=() [2/5]

```
bool operator!= (
    T * pT ) const    [inline]
```

pointer inequal

15.47.3.6 operator!=() [3/5]

```
bool operator!= (
    const long int nMustBeNull ) const [inline]
```

pointer unequal

15.47.3.7 operator!=() [4/5]

```
bool operator!= (
    const int nMustBeNull ) const [inline]
```

pointer unequal

15.47.3.8 operator!=() [5/5]

```
bool operator!= (
    const std::nullptr_t nullPtr ) const [inline]
```

pointer unequal

15.47.3.9 operator()

```
T& operator() (
    void ) const [inline]
```

Dereferencing.

15.47.3.10 operator*()

```
T& operator* (
    void ) const [inline]
```

Dereferencing.

15.47.3.11 operator->()

```
T* operator-> (
    void ) const [inline]
```

Dereferencing.

15.47.3.12 operator=()

```
void operator= (
    B * pB ) [inline]
```

Assign INode Pointer.

15.47.3.13 operator==() [1/3]

```
bool operator== (
    T * pT ) const [inline]
```

pointer equal

15.47.3.14 operator==() [2/3]

```
bool operator== (
    const CPointer< T, B > & rT ) const [inline]
```

pointer equal

15.47.3.15 operator==() [3/3]

```
bool operator== (
    int nMustBeNull ) const [inline]
```

pointer equal

15.47.4 Member Data Documentation

15.47.4.1 m_pT

```
T* m_pT [protected]
```

Underlying raw pointer.

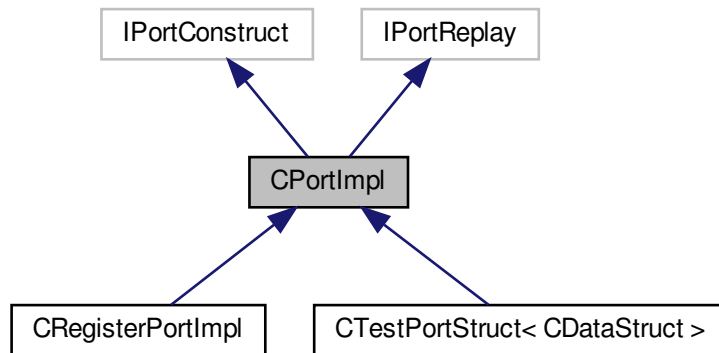
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Pointer.h](#)

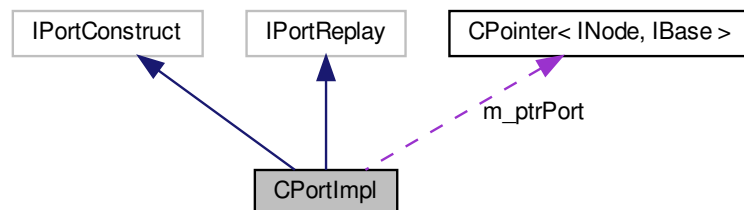
15.48 CPortImpl Class Reference

Standard implementation for a port.

Inheritance diagram for CPortImpl:



Collaboration diagram for CPortImpl:



Public Member Functions

- [CPortImpl \(\)](#)
Constructor.
- virtual [~CPortImpl \(\)](#)
Destructor.
- virtual [EAccessMode GetAccessMode \(\)](#) const =0
Get the access mode of the node.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))=0
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))=0
Writes a chunk of bytes to the port.

- virtual void [SetPortImpl](#) ([IPort](#) *pPort)
Sets pointer the real port implementation; this function may called only once.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- virtual void [Replay](#) ([IPortWriteList](#) *pPortRecorder, bool [Invalidate](#)=true)
sends the commands to the camera.
- void [InvalidateNode](#) ()

Protected Attributes

- [CNodePtr](#) [m_ptrPort](#)
Pointer to the node holding a reference to this implementation.

15.48.1 Detailed Description

Standard implementation for a port.

15.48.2 Constructor & Destructor Documentation

15.48.2.1 CPortImpl()

```
CPortImpl ( ) [inline]
```

Constructor.

15.48.2.2 ~CPortImpl()

```
virtual ~CPortImpl ( ) [inline], [virtual]
```

Destructor.

15.48.3 Member Function Documentation

15.48.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [pure virtual]
```

Get the access mode of the node.

Driver closed => NI, Driver open => RW, analyzing a struct, RO

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

15.48.3.2 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.48.3.3 InvalidateNode()

```
void InvalidateNode ( ) [inline]
```

15.48.3.4 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Reads a chunk of bytes from the port.

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

15.48.3.5 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [inline], [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

15.48.3.6 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented in [CRegisterPortImpl](#).

15.48.3.7 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes a chunk of bytes to the port.

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

15.48.4 Member Data Documentation

15.48.4.1 m_ptrPort

```
CNodePtr m_ptrPort [protected]
```

Pointer to the node holding a reference to this implementation.

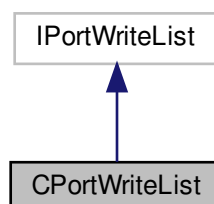
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortImpl.h](#)

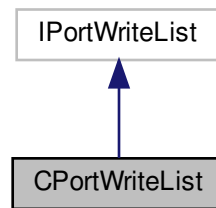
15.49 CPortWriteList Class Reference

Container holding a list of port write commands.

Inheritance diagram for CPortWriteList:



Collaboration diagram for CPortWriteList:



Public Member Functions

- [CPortWriteList](#) ()
Constructor.
- [~CPortWriteList](#) ()
Destructor.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- virtual void [Replay](#) (IPort *pPort)
Replays the write command to the given port interface.
- virtual void [SetCookie](#) (const int64_t Value)
Sets a cookie in case the port implementation want to cache a command list.
- virtual int64_t [GetCookie](#) ()
Gets the cookie a port implementation may have set for caching a command list.
- void * [GetPortWriteListHandle](#) ()

Protected Attributes

- void * [m_pWriteList](#)

15.49.1 Detailed Description

Container holding a list of port write commands.

15.49.2 Constructor & Destructor Documentation

15.49.2.1 CPortWriteList()

[CPortWriteList](#) ()

Constructor.

15.49.2.2 ~CPortWriteList()

```
~CPortWriteList ( )
```

Destructor.

15.49.3 Member Function Documentation

15.49.3.1 GetCookie()

```
virtual int64_t GetCookie ( ) [virtual]
```

Gets the cookie a port implementation may have set for caching a command list.

15.49.3.2 GetPortWriteListHandle()

```
void* GetPortWriteListHandle ( )
```

15.49.3.3 Replay()

```
virtual void Replay (
    IPort * pPort ) [virtual]
```

Replays the write command to the given port interface.

15.49.3.4 SetCookie()

```
virtual void SetCookie (
    const int64_t Value ) [virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

15.49.3.5 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

15.49.4 Member Data Documentation

15.49.4.1 m_pWriteList

```
void* m_pWriteList [protected]
```

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

- include/SpinGenApi/[PortWriteList.h](#)

15.50 CpuUsageInfo Struct Reference

Public Attributes

- bool [dummy](#)

15.50.1 Member Data Documentation

15.50.1.1 dummy

```
bool dummy
```

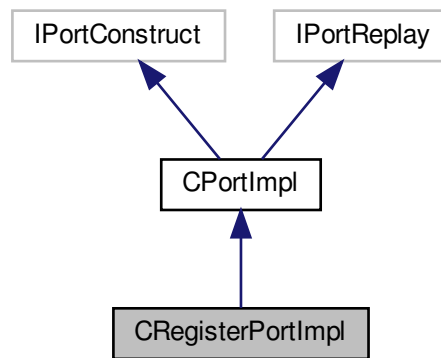
The documentation for this struct was generated from the following file:

- src/GigEVisionPerformance/[CpuUtil.h](#)

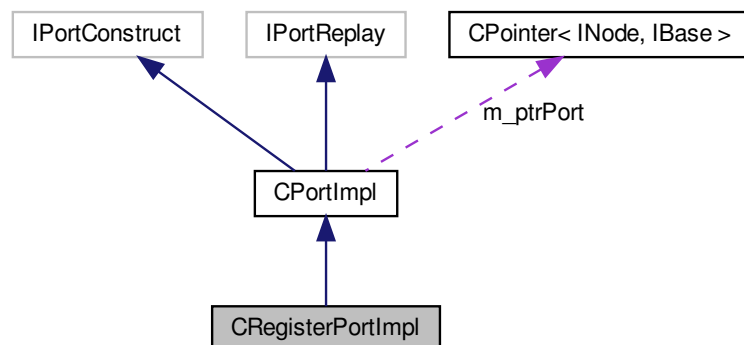
15.51 CRegisterPortImpl Class Reference

Standard implementation for a port using a register based transport layer.

Inheritance diagram for CRegisterPortImpl:



Collaboration diagram for CRegisterPortImpl:



Public Member Functions

- [CRegisterPortImpl](#) (int MaxNumQuadlets=1, bool TransportLayerSwapsEndianess=false)
Constructor.
- virtual [~CRegisterPortImpl](#) ()
Destructor.
- virtual [EAccessMode GetAccessMode](#) () const =0
Get the access mode of the node.

- virtual void [ReadRegister](#) (uint32_t *pRegisters, int64_t [Address](#), int64_t [Length](#))=0
Reads an array of quadlets from the port.
- virtual void [WriteRegister](#) (const uint32_t *pRegisters, int64_t [Address](#), int64_t [Length](#))=0
Writes an array of quadlets to the port.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- virtual void [SetPortImpl](#) (IPort *pPort)
Sets pointer the real port implementation; this function may called only once.

Additional Inherited Members

15.51.1 Detailed Description

Standard implementation for a port using a register based transport layer.

15.51.2 Constructor & Destructor Documentation

15.51.2.1 CRegisterPortImpl()

```
CRegisterPortImpl (
    int MaxNumQuadlets = 1,
    bool TransportLayerSwapsEndianness = false ) [inline]
```

Constructor.

15.51.2.2 ~CRegisterPortImpl()

```
virtual ~CRegisterPortImpl ( ) [inline], [virtual]
```

Destructor.

15.51.3 Member Function Documentation

15.51.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [pure virtual]
```

Get the access mode of the node.

Driver closed => NI, Driver open => RW, analyzing a struct, RO

Implements [CPortImpl](#).

15.51.3.2 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

15.51.3.3 ReadRegister()

```
virtual void ReadRegister (
    uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Reads an array of quadlets from the port.

15.51.3.4 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented from [CPortImpl](#).

15.51.3.5 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Implements [CPortImpl](#).

15.51.3.6 WriteRegister()

```
virtual void WriteRegister (
    const uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes an array of quadlets to the port.

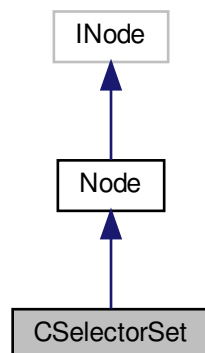
The documentation for this class was generated from the following file:

- include/SpinGenApi/[RegisterPortImpl.h](#)

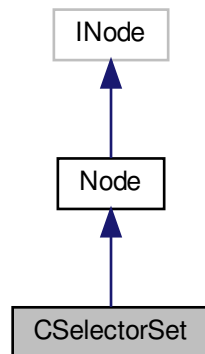
15.52 CSelectorSet Class Reference

The set of selectors selecting a given node.

Inheritance diagram for CSelectorSet:



Collaboration diagram for CSelectorSet:



Public Member Functions

- [CSelectorSet](#) ([IBase](#) *pBase)
Constructor.
- [~CSelectorSet](#) ()
Destructor.
- bool [IsEmpty](#) ()
returns true if no selectors are present
- virtual bool [SetFirst](#) ()
- virtual bool [SetNext](#) (bool Tick=true)
- virtual void [Restore](#) ()
- virtual [GenICam::gcstring ToString](#) ()
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)

Additional Inherited Members

15.52.1 Detailed Description

The set of selectors selecting a given node.

15.52.2 Constructor & Destructor Documentation

15.52.2.1 CSelectorSet()

```

CSelectorSet (
    IBase * pBase )
  
```

Constructor.

Parameters

<i>pBase</i>	Feature selected by the selector set

15.52.2.2 ~CSelectorSet()

`~CSelectorSet ( )`

Destructor.

15.52.3 Member Function Documentation**15.52.3.1 GetSelectorList()**

```
virtual void GetSelectorList (
    FeatureList_t & SelectorList,
    bool Incremental = false ) [virtual]
```

15.52.3.2 IsEmpty()

`bool IsEmpty ( )`

returns true if no selectors are present

15.52.3.3 Restore()

```
virtual void Restore ( ) [virtual]
```

15.52.3.4 SetFirst()

```
virtual bool SetFirst ( ) [virtual]
```

15.52.3.5 SetNext()

```
virtual bool SetNext (
    bool Tick = true ) [virtual]
```

15.52.3.6 ToString()

```
virtual GenICam::gcstring ToString ( ) [virtual]
```

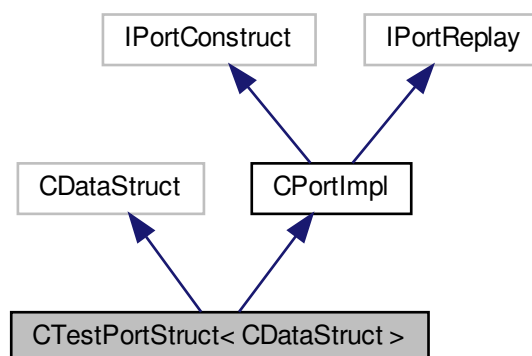
The documentation for this class was generated from the following file:

- [include/SpinGenApi/SelectorSet.h](#)

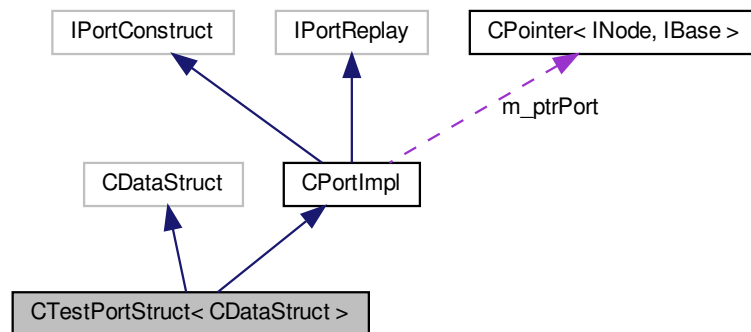
15.53 CTestPortStruct< CDataStruct > Class Template Reference

Implements a register spaces based on a C++ struct.

Inheritance diagram for CTestPortStruct< CDataStruct >:



Collaboration diagram for CTestPortStruct< CDataStruct >:



Public Member Functions

- [CTestPortStruct](#) (int64_t BaseAddress=0)
- virtual [EAccessMode GetAccessMode](#) () const
Get the access mode of the node.
- virtual [EInterfaceType GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual void [Read](#) (void *pBuffer, int64_t Address, int64_t Length)
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t Address, int64_t Length)
Writes a chunk of bytes to the port.
- void [MemSet](#) (const char FillValue)
- void [ResetStatistics](#) ()
Resets the read/write statistics.
- int64_t [GetNumReads](#) ()
Returns the number of reads since lastReset Statistics.
- int64_t [GetNumWrites](#) ()
Returns the number of writes since lastReset Statistics.

Protected Attributes

- int64_t [m_NumReads](#)
Number of reads since last reset.
- int64_t [m_NumWrites](#)
Number of writes since last reset.
- int64_t [m_BaseAddress](#)
the base address used for the struct

15.53.1 Detailed Description

```
template<class CDataStruct>
class Spinnaker::GenApi::CTestPortStruct< CDataStruct >
```

Implements a register spaces based on a C++ struct.

15.53.2 Constructor & Destructor Documentation

15.53.2.1 CTestPortStruct()

```
CTestPortStruct (
    int64_t BaseAddress = 0 ) [inline]
```

15.53.3 Member Function Documentation

15.53.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]
```

Get the access mode of the node.

Implements [CPortImpl](#).

15.53.3.2 GetNumReads()

```
int64_t GetNumReads ( ) [inline]
```

Returns the number of reads since lastReset Statistics.

15.53.3.3 GetNumWrites()

```
int64_t GetNumWrites ( ) [inline]
```

Returns the number of writes since lastReset Statistics.

15.53.3.4 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [inline], [virtual]
```

Get the type of the main interface of a node.

15.53.3.5 MemSet()

```
void MemSet (
    const char FillValue ) [inline]
```

15.53.3.6 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

15.53.3.7 ResetStatistics()

```
void ResetStatistics ( ) [inline]
```

Resets the read/write statistics.

15.53.3.8 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Implements [CPortImpl](#).

15.53.4 Member Data Documentation

15.53.4.1 m_BaseAddress

```
int64_t m_BaseAddress [protected]
```

the base address used for the struct

15.53.4.2 m_NumReads

```
int64_t m_NumReads [protected]
```

Number of reads since last reset.

15.53.4.3 m_NumWrites

```
int64_t m_NumWrites [protected]
```

Number of writes since last reset.

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

- include/SpinGenApi/[StructPort.h](#)

15.54 DCAM_CHECKSUM Struct Reference

Public Attributes

- uint32_t [CRCChecksum](#)

15.54.1 Member Data Documentation

15.54.1.1 CRCChecksum

```
uint32_t CRCChecksum
```

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

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

15.55 DCAM_CHUNK_TRAILER Struct Reference

Public Attributes

- SPIN_GUID [ChunkID](#)
- uint32_t [ChunkLength](#)
- uint32_t [InverseChunkLength](#)

15.55.1 Member Data Documentation

15.55.1.1 ChunkID

`SPIN_GUID` `ChunkID`

15.55.1.2 ChunkLength

`uint32_t` `ChunkLength`

15.55.1.3 InverseChunkLength

`uint32_t` `InverseChunkLength`

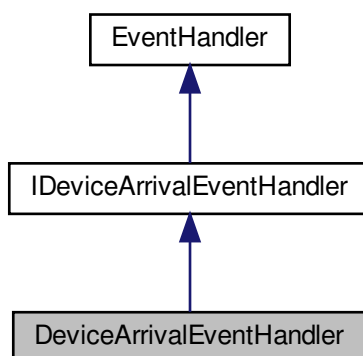
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/ChunkAdapterDcam.h`

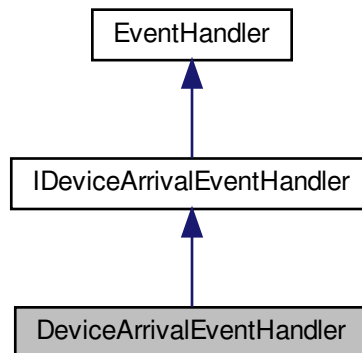
15.56 DeviceArrivalEventHandler Class Reference

An event handler for capturing the device arrival event.

Inheritance diagram for DeviceArrivalEventHandler:



Collaboration diagram for DeviceArrivalEventHandler:



Public Member Functions

- [DeviceArrivalEventHandler](#) ()
Default constructor.
- virtual [~DeviceArrivalEventHandler](#) ()
Virtual destructor.
- virtual void [OnDeviceArrival](#) (uint64_t serialNumber)=0
Callback to the device arrival event.

Protected Member Functions

- [DeviceArrivalEventHandler](#) & [operator=](#) (const [DeviceArrivalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.56.1 Detailed Description

An event handler for capturing the device arrival event.

15.56.2 Constructor & Destructor Documentation

15.56.2.1 DeviceArrivalEventHandler()

```
DeviceArrivalEventHandler ( )
```

Default constructor.

15.56.2.2 ~DeviceArrivalEventHandler()

```
virtual ~DeviceArrivalEventHandler ( ) [virtual]
```

Virtual destructor.

15.56.3 Member Function Documentation

15.56.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device arrival event.

Implements [IDeviceArrivalEventHandler](#).

15.56.3.2 operator=()

```
DeviceArrivalEventHandler& operator= (
    const DeviceArrivalEventHandler & ) [protected]
```

Assignment operator.

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

- [include/DeviceArrivalEventHandler.h](#)

15.57 DeviceEventExposureEndData Struct Reference

Data Fields for Device Event payload for EventExposureEnd.

Public Attributes

- uint64_t [frameID](#)

15.57.1 Detailed Description

Data Fields for Device Event payload for EventExposureEnd.

15.57.2 Member Data Documentation

15.57.2.1 frameID

uint64_t frameID

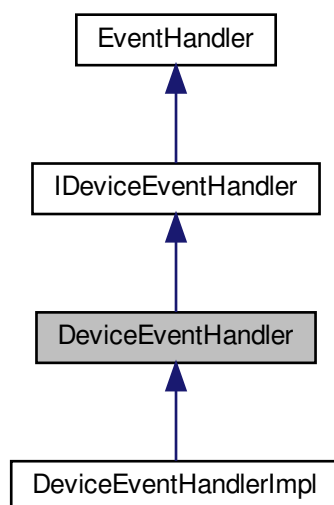
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

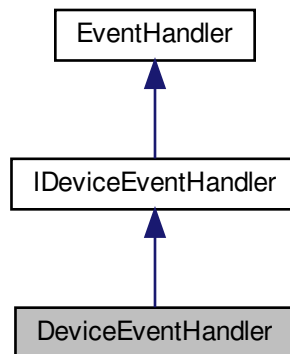
15.58 DeviceEventHandler Class Reference

A handler to device events.

Inheritance diagram for DeviceEventHandler:



Collaboration diagram for DeviceEventHandler:



Public Member Functions

- [DeviceEventHandler \(\)](#)
Default constructor.
- virtual [~DeviceEventHandler \(\)](#)
Virtual destructor.
- virtual void [OnDeviceEvent \(Spinnaker::GenICam::gcstring eventName\)=0](#)
Device event callback.
- uint64_t [GetDeviceEventId \(\)](#) const
Get the ID of the device event.
- [GenICam::gcstring GetDeviceEventName \(\)](#) const
Get the name of the device event.

Protected Member Functions

- [DeviceEventHandler & operator= \(const DeviceEventHandler &\)](#)
Assignment operator.

Additional Inherited Members

15.58.1 Detailed Description

A handler to device events.

15.58.2 Constructor & Destructor Documentation

15.58.2.1 DeviceEventHandler()

`DeviceEventHandler ( )`

Default constructor.

15.58.2.2 ~DeviceEventHandler()

`virtual ~DeviceEventHandler ( ) [virtual]`

Virtual destructor.

15.58.3 Member Function Documentation

15.58.3.1 GetDeviceEventId()

`uint64_t GetDeviceEventId ( ) const [virtual]`

Get the ID of the device event.

Returns

The device event ID

Implements [IDeviceEventHandler](#).

15.58.3.2 GetDeviceEventName()

`GenICam::gcstring GetDeviceEventName ( ) const [virtual]`

Get the name of the device event.

Returns

The device event name

Implements [IDeviceEventHandler](#).

15.58.3.3 OnDeviceEvent()

`virtual void OnDeviceEvent (
 Spinnaker::GenICam::gcstring eventName) [pure virtual]`

Device event callback.

Parameters

<i>eventName</i>	The name of the event

Implements [IDeviceEventHandler](#).

Implemented in [DeviceEventHandlerImpl](#).

15.58.3.4 operator=()

```
DeviceEventHandler& operator= (  
    const DeviceEventHandler & ) [protected]
```

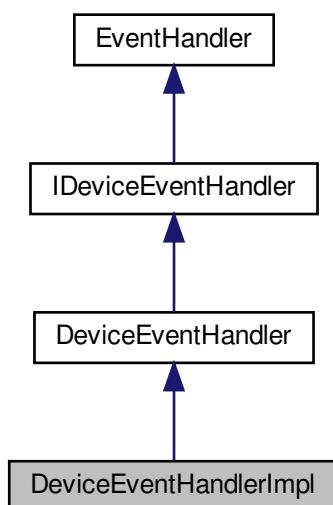
Assignment operator.

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

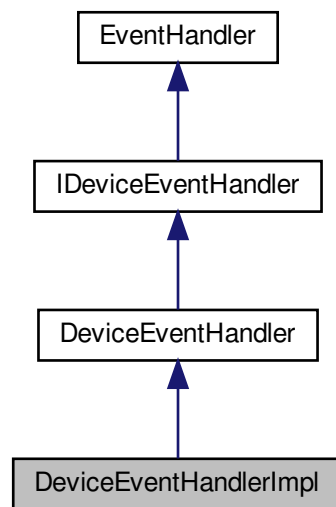
- include/[DeviceEventHandler.h](#)

15.59 DeviceEventHandlerImpl Class Reference

Inheritance diagram for DeviceEventHandlerImpl:



Collaboration diagram for DeviceEventHandlerImpl:



Public Member Functions

- [DeviceEventHandlerImpl](#) ([gcstring](#) eventName)
- [~DeviceEventHandlerImpl](#) ()
- void [OnDeviceEvent](#) ([gcstring](#) eventName)
Device event callback.

Additional Inherited Members

15.59.1 Constructor & Destructor Documentation

15.59.1.1 DeviceEventHandlerImpl()

```
DeviceEventHandlerImpl (  
    gcstring eventName ) [inline]
```

15.59.1.2 ~DeviceEventHandlerImpl()

```
~DeviceEventHandlerImpl ( ) [inline]
```

15.59.2 Member Function Documentation

15.59.2.1 OnDeviceEvent()

```
void OnDeviceEvent (
    gcstring eventName ) [inline], [virtual]
```

Device event callback.

Parameters

<i>eventName</i>	The name of the event

Implements [DeviceEventHandler](#).

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

- src/DeviceEvents/[DeviceEvents.cpp](#)

15.60 DeviceEventInferenceData Struct Reference

Data Fields for Device Event payload for EventInference.

Public Attributes

- uint32_t [result](#)
- float [confidence](#)
- uint64_t [frameID](#)

15.60.1 Detailed Description

Data Fields for Device Event payload for EventInference.

15.60.2 Member Data Documentation

15.60.2.1 confidence

```
float confidence
```

15.60.2.2 frameID

```
uint64_t frameID
```

15.60.2.3 result

```
uint32_t result
```

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

- [include/SpinnakerDefs.h](#)

15.61 DeviceEventUtility Class Reference

Static Public Member Functions

- static void [ParseDeviceEventInference](#) (const uint8_t *payloadData, const size_t payloadSize, [DeviceEventInferenceData](#) &eventData)
Parse the EventInference device event payload data.
- static void [ParseDeviceEventExposureEnd](#) (const uint8_t *payloadData, const size_t payloadSize, [DeviceEventExposureEndData](#) &eventData)
Parse the EventExposureEnd device event payload data.

15.61.1 Member Function Documentation

15.61.1.1 ParseDeviceEventExposureEnd()

```
static void ParseDeviceEventExposureEnd (  
    const uint8_t * payloadData,  
    const size_t payloadSize,  
    DeviceEventExposureEndData & eventData ) [static]
```

Parse the EventExposureEnd device event payload data.

Parameters

in	<i>payloadData</i>	Event payload data
in	<i>payloadSize</i>	Event payload data size
out	<i>eventData</i>	Parsed ExposureEnd payload data returned by reference

See also

[DeviceEventExposureEndData](#)

15.61.1.2 ParseDeviceEventInference()

```
static void ParseDeviceEventInference (
    const uint8_t * payloadData,
    const size_t payloadSize,
    DeviceEventInferenceData & eventData ) [static]
```

Parse the EventInference device event payload data.

Parameters

in	<i>payloadData</i>	Event payload data
in	<i>payloadSize</i>	Event payload data size
out	<i>eventData</i>	Parsed EventInference payload data

See also

[DeviceEventInferenceData](#)

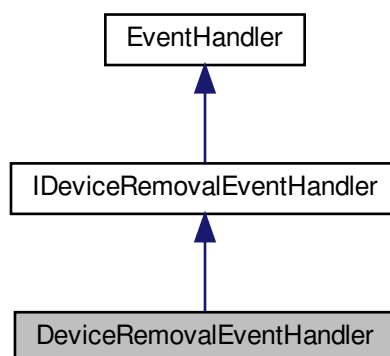
The documentation for this class was generated from the following file:

- include/[DeviceEventUtility.h](#)

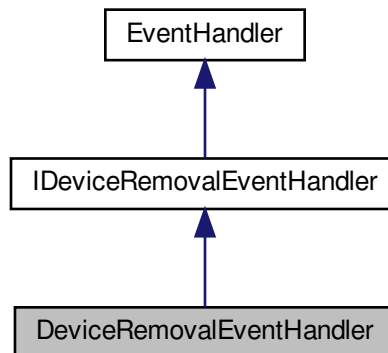
15.62 DeviceRemovalEventHandler Class Reference

An event handler for capturing the device removal event.

Inheritance diagram for DeviceRemovalEventHandler:



Collaboration diagram for DeviceRemovalEventHandler:



Public Member Functions

- [DeviceRemovalEventHandler](#) ()
Default Constructor.
- virtual [~DeviceRemovalEventHandler](#) ()
Virtual Destructor.
- virtual void [OnDeviceRemoval](#) (uint64_t serialNumber)=0
Device removal event callback.

Protected Member Functions

- [DeviceRemovalEventHandler](#) & [operator=](#) (const [DeviceRemovalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.62.1 Detailed Description

An event handler for capturing the device removal event.

15.62.2 Constructor & Destructor Documentation

15.62.2.1 DeviceRemovalEventHandler()

`DeviceRemovalEventHandler ( )`

Default Constructor.

15.62.2.2 ~DeviceRemovalEventHandler()

`virtual ~DeviceRemovalEventHandler ( ) [virtual]`

Virtual Destructor.

15.62.3 Member Function Documentation

15.62.3.1 OnDeviceRemoval()

`virtual void OnDeviceRemoval (
 uint64_t serialNumber) [pure virtual]`

Device removal event callback.

Parameters

<i>serialNumber</i>	The serial number of the device removed

Implements [IDeviceRemovalEventHandler](#).

15.62.3.2 operator=()

`DeviceRemovalEventHandler& operator= (
 const DeviceRemovalEventHandler &) [protected]`

Assignment operator.

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

- include/[DeviceRemovalEventHandler.h](#)

15.63 double_autovector_t Class Reference

Vector of doubles with reference counting.

Public Member Functions

- `double_autovector_t` ()
- `double_autovector_t` (const `double_autovector_t` &obj)
- `double_autovector_t` (size_t n)
- virtual `~double_autovector_t` (void)
- `double_autovector_t` & `operator=` (const `double_autovector_t` &obj)
- void `operator delete` (void *pWhere)
- void * `operator new` (size_t uiSize)
- double & `operator[]` (size_t uiIndex)
- const double & `operator[]` (size_t uiIndex) const
- size_t `size` () const

Protected Attributes

- std::vector< double > * `_pv`
- ATOMIC_VARIABLE * `_pCount`

15.63.1 Detailed Description

Vector of doubles with reference counting.

15.63.2 Constructor & Destructor Documentation

15.63.2.1 `double_autovector_t()` [1/3]

```
double_autovector_t ( )
```

15.63.2.2 `double_autovector_t()` [2/3]

```
double_autovector_t (
    const double_autovector_t & obj )
```

15.63.2.3 `double_autovector_t()` [3/3]

```
double_autovector_t (
    size_t n ) [explicit]
```

15.63.2.4 ~double_autovector_t()

```
virtual ~double_autovector_t (
    void ) [virtual]
```

15.63.3 Member Function Documentation

15.63.3.1 operator delete()

```
void operator delete (
    void * pWhere )
```

15.63.3.2 operator new()

```
void* operator new (
    size_t uiSize )
```

15.63.3.3 operator=()

```
double_autovector_t& operator= (
    const double_autovector_t & obj )
```

15.63.3.4 operator[]() [1/2]

```
double& operator[] (
    size_t uiIndex )
```

15.63.3.5 operator[]() [2/2]

```
const double& operator[] (
    size_t uiIndex ) const
```

15.63.3.6 size()

```
size_t size ( ) const
```

15.63.4 Member Data Documentation

15.63.4.1 _pCount

```
ATOMIC_VARIABLE* _pCount [protected]
```

15.63.4.2 _pv

```
std::vector<double>* _pv [protected]
```

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

- include/SpinGenApi/[Autovector.h](#)

15.64 EAccessModeClass Class Reference

Holds conversion methods for the access mode enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EAccessMode](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EAccessMode](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) ([EAccessMode](#) Value)
Converts a string to an int32_t property.

15.64.1 Detailed Description

Holds conversion methods for the access mode enumeration.

15.64.2 Member Function Documentation

15.64.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EAccessMode * pValue ) [static]
```

Converts a string to enum value.

15.64.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EAccessMode * pValue ) [static]
```

Converts a string to an int32_t property.

15.64.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EAccessMode Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.65 ECachingModeClass Class Reference

Holds conversion methods for the caching mode enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ECachingMode](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ECachingMode](#) *pValue)
- static [GenICam::gcstring](#) [ToString](#) ([ECachingMode](#) Value)
Converts a string to an int32_t property.

15.65.1 Detailed Description

Holds conversion methods for the caching mode enumeration.

15.65.2 Member Function Documentation

15.65.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

Converts a string to enum value.

15.65.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

15.65.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ECachingMode Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.66 EDisplayNotationClass Class Reference

Holds conversion methods for the notation type of floats.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EDisplayNotation](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EDisplayNotation](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([EDisplayNotation](#) Value)
Converts a string to an int32_t property.

15.66.1 Detailed Description

Holds conversion methods for the notation type of floats.

15.66.2 Member Function Documentation

15.66.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EDisplayNotation * pValue ) [static]
```

Converts a string to enum value.

15.66.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EDisplayNotation * pValue ) [static]
```

Converts a string to an int32_t property.

15.66.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EDisplayNotation Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.67 EEndianessClass Class Reference

Holds conversion methods for the endianess enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EEndianess](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EEndianess](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([EEndianess](#) Value)
Converts a string to an int32_t property.

15.67.1 Detailed Description

Holds conversion methods for the endianess enumeration.

15.67.2 Member Function Documentation

15.67.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EEndianess * pValue ) [static]
```

Converts a string to enum value.

15.67.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EEndianess * pValue ) [static]
```

Converts a string to an int32_t property.

15.67.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EEndianess Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.68 EGenApiSchemaVersionClass Class Reference

helper class converting EGenApiSchemaVersion from and to string

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EGenApiSchemaVersion](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EGenApiSchemaVersion](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([EGenApiSchemaVersion](#) Value)
Converts a string to an int32_t property.

15.68.1 Detailed Description

helper class converting EGenApiSchemaVersion from and to string

15.68.2 Member Function Documentation

15.68.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to enum value.

15.68.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to an int32_t property.

15.68.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EGenApiSchemaVersion Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.69 EInputDirectionClass Class Reference

Holds conversion methods for the notation type of floats.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EInputDirection](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EInputDirection](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([EInputDirection](#) Value)
Converts a string to an int32_t property.

15.69.1 Detailed Description

Holds conversion methods for the notation type of floats.

15.69.2 Member Function Documentation

15.69.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to enum value.

15.69.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to an int32_t property.

15.69.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EInputDirection Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.70 ENameSpaceClass Class Reference

Holds conversion methods for the namespace enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ENameSpace](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ENameSpace](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) ([ENameSpace](#) Value)
Converts a string to an int32_t property.

15.70.1 Detailed Description

Holds conversion methods for the namespace enumeration.

15.70.2 Member Function Documentation

15.70.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ENameSpace * pValue ) [static]
```

Converts a string to enum value.

15.70.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ENameSpace * pValue ) [static]
```

Converts a string to an int32_t property.

15.70.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ENameSpace Value ) [static]
```

Converts a string to an int32_t property.

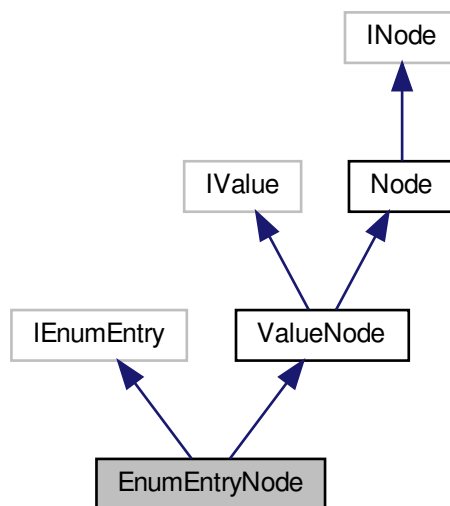
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

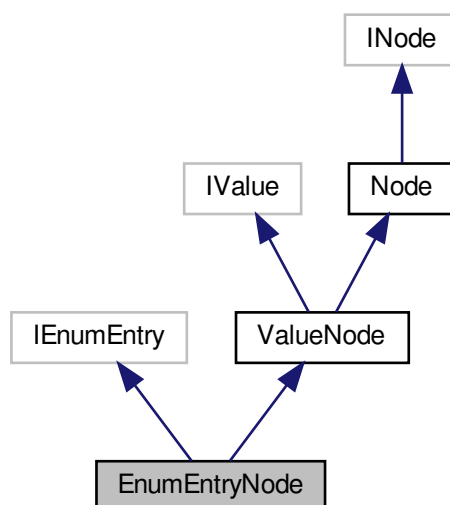
15.71 EnumEntryNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for EnumEntryNode:



Collaboration diagram for EnumEntryNode:



Public Member Functions

- [EnumEntryNode](#) ()
- [EnumEntryNode](#) (std::shared_ptr< Node::NodeImpl > pEnumEntry)
- virtual [~EnumEntryNode](#) ()
- virtual int64_t [GetValue](#) ()
Get numeric enum value.
- virtual [GenlCam::gcstring](#) [GetSymbolic](#) () const
Get symbolic enum value.
- virtual double [GetNumericValue](#) ()
Get double number associated with the entry.
- virtual bool [IsSelfClearing](#) ()
Indicates if the corresponding EnumEntry is self clearing.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for EnumEntry

Additional Inherited Members

15.71.1 Detailed Description

[Interface](#) for string properties.

15.71.2 Constructor & Destructor Documentation

15.71.2.1 EnumEntryNode() [1/2]

```
EnumEntryNode ( )
```

15.71.2.2 EnumEntryNode() [2/2]

```
EnumEntryNode (
    std::shared_ptr< Node::NodeImpl > pEnumEntry )
```

15.71.2.3 ~EnumEntryNode()

```
virtual ~EnumEntryNode ( ) [virtual]
```

15.71.3 Member Function Documentation

15.71.3.1 GetNumericValue()

```
virtual double GetNumericValue ( ) [virtual]
```

Get double number associated with the entry.

15.71.3.2 GetSymbolic()

```
virtual GenICam::gcstring GetSymbolic ( ) const [virtual]
```

Get symbolic enum value.

15.71.3.3 GetValue()

```
virtual int64_t GetValue ( ) [virtual]
```

Get numeric enum value.

15.71.3.4 IsSelfClearing()

```
virtual bool IsSelfClearing ( ) [virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

15.71.3.5 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for EnumEntry

Reimplemented from [ValueNode](#).

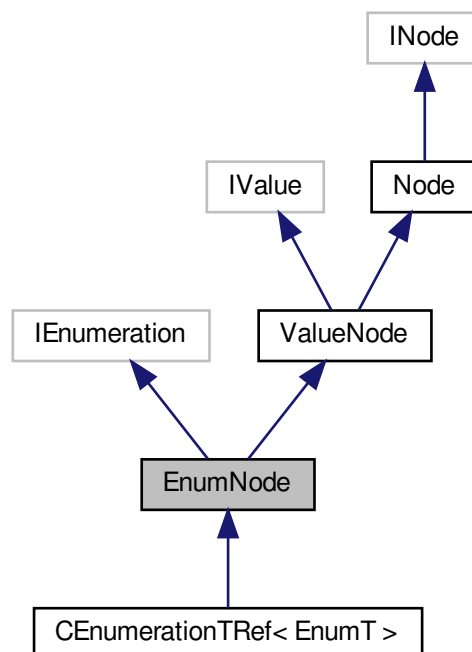
The documentation for this class was generated from the following file:

- [include/SpinGenApi/EnumEntryNode.h](#)

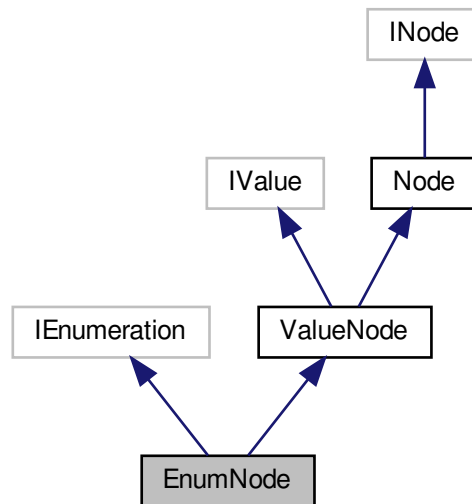
15.72 EnumNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for EnumNode:



Collaboration diagram for EnumNode:



Public Member Functions

- [EnumNode](#) ()
- [EnumNode](#) (std::shared_ptr< Node::NodeImpl > pEnumeration)
- virtual [~EnumNode](#) ()
- virtual void [GetSymbolics](#) (StringList_t &Symbolics)
Get list of symbolic Values.
- virtual void [GetEntries](#) (NodeList_t &Entries)
Get list of entry nodes.
- virtual [IEnumeration](#) & [operator=](#) (const [GenICam::gcstring](#) &ValueStr)
Set string node value.
- virtual void [SetIntValue](#) (int64_t Value, bool [Verify](#)=true)
Set integer node value.
- virtual [GenICam::gcstring](#) [operator*](#) ()
Get string node value.
- virtual int64_t [GetIntValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get integer node value.
- virtual [IEnumEntry](#) * [GetEntryByName](#) (const [GenICam::gcstring](#) &Symbolic)
Get an entry node by name.
- virtual [IEnumEntry](#) * [GetEntry](#) (const int64_t IntValue)
Get an entry node by its IntValue.
- virtual [IEnumEntry](#) * [GetCurrentEntry](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get the current entry.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Enumeration

Protected Attributes

- `std::shared_ptr< Node::NodeImpl > m_pEnumeration`

15.72.1 Detailed Description

[Interface](#) for string properties.

15.72.2 Constructor & Destructor Documentation

15.72.2.1 EnumNode() [1/2]

```
EnumNode ( )
```

15.72.2.2 EnumNode() [2/2]

```
EnumNode (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

15.72.2.3 ~EnumNode()

```
virtual ~EnumNode ( ) [virtual]
```

15.72.3 Member Function Documentation

15.72.3.1 GetCurrentEntry()

```
virtual IEnumEntry* GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get the current entry.

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.72.3.2 GetEntries()

```
virtual void GetEntries (
    NodeList_t & Entries ) [virtual]
```

Get list of entry nodes.

15.72.3.3 GetEntry()

```
virtual IEnumEntry* GetEntry (
    const int64_t IntValue ) [virtual]
```

Get an entry node by its IntValue.

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.72.3.4 GetEntryByName()

```
virtual IEnumEntry* GetEntryByName (
    const GenICam::gcstring & Symbolic ) [virtual]
```

Get an entry node by name.

15.72.3.5 GetIntValue()

```
virtual int64_t GetIntValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get integer node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.72.3.6 GetSymbolics()

```
virtual void GetSymbolics (
    StringList_t & Symbolics ) [virtual]
```

Get list of symbolic Values.

15.72.3.7 operator*()

```
virtual GenICam::gcstring operator* ( ) [virtual]
```

Get string node value.

15.72.3.8 operator=()

```
virtual IEnumeration& operator= (
    const GenICam::gcstring & ValueStr ) [virtual]
```

Set string node value.

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.72.3.9 SetIntValue()

```
virtual void SetIntValue (
    int64_t Value,
    bool Verify = true ) [virtual]
```

Set integer node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.72.3.10 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Enumeration

Reimplemented from [ValueNode](#).

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.72.4 Member Data Documentation

15.72.4.1 m_pEnumeration

```
std::shared_ptr<Node::NodeImpl> m_pEnumeration [protected]
```

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

- include/SpinGenApi/[EnumNode.h](#)

15.73 ERepresentationClass Class Reference

Holds conversion methods for the representation enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ERepresentation](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ERepresentation](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) ([ERepresentation](#) Value)
Converts a string to an int32_t property.

15.73.1 Detailed Description

Holds conversion methods for the representation enumeration.

15.73.2 Member Function Documentation

15.73.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to enum value.

15.73.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to an int32_t property.

15.73.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ERepresentation Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.74 ESignClass Class Reference

Holds conversion methods for the sign enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ESign](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ESign](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([ESign](#) Value)
Converts a string to an int32_t property.

15.74.1 Detailed Description

Holds conversion methods for the sign enumeration.

15.74.2 Member Function Documentation

15.74.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to enum value.

15.74.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to an int32_t property.

15.74.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ESign Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.75 ESlopeClass Class Reference

Holds conversion methods for the converter formulas.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ESlope](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ESlope](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([ESlope](#) Value)
Converts a string to an int32_t property.

15.75.1 Detailed Description

Holds conversion methods for the converter formulas.

15.75.2 Member Function Documentation

15.75.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESlope * pValue ) [static]
```

Converts a string to enum value.

15.75.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ESlope * pValue ) [static]
```

Converts a string to an int32_t property.

15.75.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ESlope Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.76 EStandardNameSpaceClass Class Reference

Holds conversion methods for the standard namespace enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EStandardNameSpace](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EStandardNameSpace](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([EStandardNameSpace](#) Value)
Converts a string to an int32_t property.

15.76.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

15.76.2 Member Function Documentation

15.76.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    EStandardNameSpace * pValue ) [static]
```

Converts a string to enum value.

15.76.2.2 ToString() [1/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    EStandardNameSpace * pValue ) [static]
```

Converts a string to an int32_t property.

15.76.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (  
    EStandardNameSpace Value ) [static]
```

Converts a string to an int32_t property.

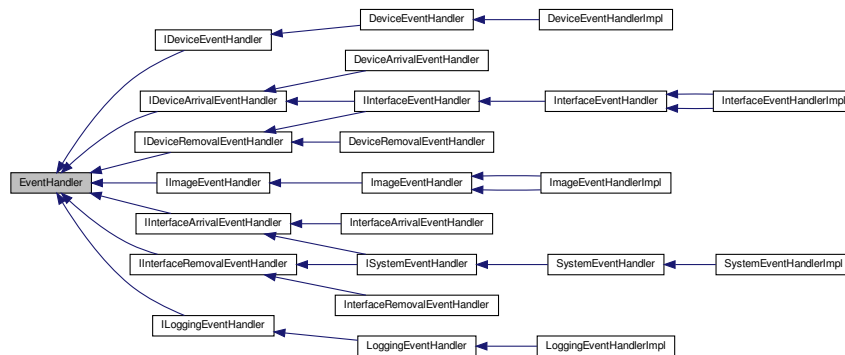
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

15.77 EventHandler Class Reference

The base class for all event handler types.

Inheritance diagram for EventHandler:



Public Member Functions

- virtual [~EventHandler](#) ()
Virtual Destructor.
- void [SetEventType](#) (EventType eventType)
Sets the event type.
- EventType [GetEventType](#) ()
Gets the event type.
- const uint8_t * [GetEventPayloadData](#) ()
Gets the event payload data.
- const size_t [GetEventPayloadDataSize](#) ()
Gets the event payload data size.

Protected Member Functions

- [EventHandler](#) ()
- [EventHandler](#) & [operator=](#) (const [EventHandler](#) &)
- void [SetEventPayload](#) (uint8_t *offset, size_t length)

Protected Attributes

- EventData * [m_pEventData](#)

Friends

- class [EventProcessor](#)
- class [IDataStream](#)
- class [Stream](#)

15.77.1 Detailed Description

The base class for all event handler types.

15.77.2 Constructor & Destructor Documentation

15.77.2.1 ~EventHandler()

```
virtual ~EventHandler ( ) [virtual]
```

Virtual Destructor.

15.77.2.2 EventHandler()

```
EventHandler ( ) [protected]
```

15.77.3 Member Function Documentation

15.77.3.1 GetEventPayloadData()

```
const uint8_t* GetEventPayloadData ( )
```

Gets the event payload data.

Returns

The event payload data

15.77.3.2 GetEventPayloadDataSize()

```
const size_t GetEventPayloadDataSize ( )
```

Gets the event payload data size.

Returns

The event payload data size

15.77.3.3 GetEventType()

```
EventType GetEventType ( )
```

Gets the event type.

Returns

The event type

15.77.3.4 operator=()

```
EventHandler& operator= (
    const EventHandler & ) [protected]
```

15.77.3.5 SetEventPayload()

```
void SetEventPayload (
    uint8_t * offset,
    size_t length ) [protected]
```

15.77.3.6 SetEventType()

```
void SetEventType (
    EventType eventType )
```

Sets the event type.

Parameters

<i>eventType</i>	The event type

15.77.4 Friends And Related Function Documentation

15.77.4.1 EventProcessor

```
friend class EventProcessor [friend]
```

15.77.4.2 IDataStream

```
friend class IDataStream [friend]
```

15.77.4.3 Stream

```
friend class Stream [friend]
```

15.77.5 Member Data Documentation

15.77.5.1 m_pEventData

```
EventData* m_pEventData [protected]
```

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

- include/[EventHandler.h](#)

15.78 EVisibilityClass Class Reference

Holds conversion methods for the visibility enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EVisibility](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EVisibility](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) ([EVisibility](#) Value)
Converts a string to an int32_t property.

15.78.1 Detailed Description

Holds conversion methods for the visibility enumeration.

15.78.2 Member Function Documentation

15.78.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to enum value.

15.78.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to an int32_t property.

15.78.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EVisibility Value ) [static]
```

Converts a string to an int32_t property.

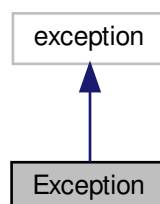
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

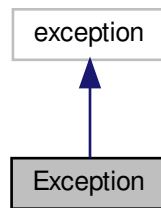
15.79 Exception Class Reference

The [Exception](#) object represents an error that is returned from the library.

Inheritance diagram for Exception:



Collaboration diagram for Exception:



Public Member Functions

- [Exception](#) ()
Default constructor.
- [Exception](#) (int line, const char *fileName, const char *funcName, const char *errMsg, [Error](#) err)
Message constructor.
- [Exception](#) (int line, const char *fileName, const char *funcName, const char *buildDate, const char *buildTime, const char *errMsg, [Error](#) err)
Message constructor.
- [Exception](#) (const [Exception](#) &except)
Copy constructor.
- virtual [~Exception](#) () throw ()
Default destructor.
- [Exception](#) & [operator=](#) (const [Exception](#) &except)
Assignment operator.
- bool [operator==](#) (const [Error](#) err) const
Equality operator.
- bool [operator!=](#) (const [Error](#) err) const
Inequality operator.
- virtual const char * [what](#) () const throw ()
virtual override for what().
- const char * [GetFullErrorMessage](#) () const
Gets the error code and full error message including the line, file, function, build date, and time.
- const char * [GetErrorMessage](#) () const
Accessor Functions.
- const char * [GetFileName](#) () const
- const char * [GetFunctionName](#) () const
- const char * [GetBuildDate](#) () const
- const char * [GetBuildTime](#) () const
- int [GetLineNumber](#) () const
- [Error](#) [GetError](#) () const

15.79.1 Detailed Description

The [Exception](#) object represents an error that is returned from the library.

Overloaded operators allow comparisons against other [Exception](#) objects.

15.79.2 Constructor & Destructor Documentation

15.79.2.1 Exception() [1/4]

Exception ()

Default constructor.

15.79.2.2 Exception() [2/4]

```
Exception (
    int line,
    const char * fileName,
    const char * funcName,
    const char * errMsg,
    Error err )
```

Message constructor.

Parameters

<i>line</i>	Line number where the exception is thrown
<i>fileName</i>	Name of the file called
<i>funcName</i>	Name of the function called
<i>errMsg</i>	A pointer to the exception message string
<i>err</i>	Error code

15.79.2.3 Exception() [3/4]

```
Exception (
    int line,
    const char * fileName,
    const char * funcName,
    const char * buildDate,
    const char * buildTime,
    const char * errMsg,
    Error err )
```

Message constructor.

Parameters

<i>line</i>	Line number where the exception is thrown
<i>fileName</i>	Name of the file called

Parameters

<i>funcName</i>	Name of the function called
<i>buildDate</i>	Build date
<i>buildTime</i>	Build time
<i>errMsg</i>	A pointer to the exception message string
<i>err</i>	Error code

15.79.2.4 Exception() [4/4]

```
Exception (
    const Exception & except )
```

Copy constructor.

15.79.2.5 ~Exception()

```
virtual ~Exception ( ) throw ( ) [virtual]
```

Default destructor.

15.79.3 Member Function Documentation

15.79.3.1 GetBuildDate()

```
const char* GetBuildDate ( ) const
```

15.79.3.2 GetBuildTime()

```
const char* GetBuildTime ( ) const
```

15.79.3.3 GetError()

```
Error GetError ( ) const
```

15.79.3.4 GetErrorMessage()

```
const char* GetErrorMessage ( ) const
```

Accessor Functions.

15.79.3.5 GetFileName()

```
const char* GetFileName ( ) const
```

15.79.3.6 GetFullErrorMessage()

```
const char* GetFullErrorMessage ( ) const
```

Gets the error code and full error message including the line, file, function, build date, and time.

15.79.3.7 GetFunctionName()

```
const char* GetFunctionName ( ) const
```

15.79.3.8 GetLineNumber()

```
int GetLineNumber ( ) const
```

15.79.3.9 operator!=()

```
bool operator!= (
    const Error err ) const
```

Inequality operator.

15.79.3.10 operator=()

```
Exception& operator= (
    const Exception & except )
```

Assignment operator.

15.79.3.11 operator==()

```
bool operator== (
    const Error err ) const
```

Equality operator.

15.79.3.12 what()

```
virtual const char* what ( ) const throw ( ) [virtual]
```

virtual override for [what\(\)](#).

Gets the error code and error message.

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

- include/[Exception.h](#)

15.80 EYesNoClass Class Reference

Holds conversion methods for the standard namespace enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EYesNo](#) *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EYesNo](#) *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) ([EYesNo](#) Value)
Converts a string to an int32_t property.

15.80.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

15.80.2 Member Function Documentation

15.80.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EYesNo * pValue ) [static]
```

Converts a string to enum value.

15.80.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EYesNo * pValue ) [static]
```

Converts a string to an int32_t property.

15.80.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EYesNo Value ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.81 FileProtocolAdapter Class Reference

Adapter between the std::iostreambuf and the SFNC Features representing the device file system.

Public Member Functions

- [FileProtocolAdapter](#) ()
Constructor.
- virtual [~FileProtocolAdapter](#) ()
- bool [attach](#) (::Spinnaker::GenApi::INodeMap *pInterface)
attach file protocol adapter to [NodeMap](#)
- bool [openFile](#) (const char *pFileName, std::ios_base::openmode mode)
open a file on the device
- bool [closeFile](#) (const char *pFileName)
close a file on the device
- std::streamsize [write](#) (const char *buf, int64_t offs, int64_t len, const char *pFileName)
writes data into a file.
- std::streamsize [read](#) (char *buf, int64_t offs, std::streamsize len, const char *pFileName)
read data from the device into a buffer
- int64_t [getBufSize](#) (const char *pFileName, std::ios_base::openmode mode)
fetch max FileAccessBuffer length for a file
- bool [deleteFile](#) (const char *pFileName)
Delete the content of the file.

15.81.1 Detailed Description

Adapter between the std::iostreambuf and the SFNC Features representing the device file system.

The adapter assumes, that the features provide stdio file access compatible semantic

15.81.2 Constructor & Destructor Documentation

15.81.2.1 FileProtocolAdapter()

```
FileProtocolAdapter ( )
```

Constructor.

15.81.2.2 ~FileProtocolAdapter()

```
virtual ~FileProtocolAdapter ( ) [virtual]
```

15.81.3 Member Function Documentation

15.81.3.1 attach()

```
bool attach (
    ::Spinnaker::GenApi::INodeMap * pInterface )
```

attach file protocol adapter to [NodeMap](#)

Parameters

<i>pInterface</i>	NodeMap of the device to which the FileProtocolAdapter is attached

Returns

true if attach was successful, false if not

15.81.3.2 closeFile()

```
bool closeFile (
    const char * pFileName )
```

close a file on the device

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector

Returns

true on success, false on error

15.81.3.3 deleteFile()

```
bool deleteFile (
    const char * pFileName )
```

Delete the content of the file.

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector

Returns

true on success, false on error

15.81.3.4 getBufSize()

```
int64_t getBufSize (
    const char * pFileName,
    std::ios_base::openmode mode )
```

fetch max FileAccessBuffer length for a file

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector
<i>mode</i>	mode to open the file. The mode must exist in the Enumeration FileOpenMode

Returns

max length of FileAccessBuffer in the given mode on the given file

15.81.3.5 openFile()

```
bool openFile (
    const char * pFileName,
    std::ios_base::openmode mode )
```

open a file on the device

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector
<i>mode</i>	mode to open the file. The mode must exist in the Enumeration FileOpenMode

Returns

true on success, false on error

15.81.3.6 read()

```
std::streamsize read (
    char * buf,
    int64_t offs,
    std::streamsize len,
    const char * pFileName )
```

read data from the device into a buffer

Parameters

<i>buf</i>	target buffer
<i>offs</i>	offset in the device file to read from
<i>len</i>	count of bytes to read
<i>pFileName</i>	filename of the file to write into The filename must exist in the Enumeration FileSelector

Returns

count of bytes successfully read

15.81.3.7 write()

```
std::streamsize write (
    const char * buf,
    int64_t offs,
    int64_t len,
    const char * pFileName )
```

writes data into a file.

Parameters

<i>buf</i>	source buffer
<i>offs</i>	offset into the device file
<i>len</i>	count of bytes to write
<i>pFileName</i>	filename of the file to write into The filename must exist in the Enumeration FileSelector

Returns

count of bytes written

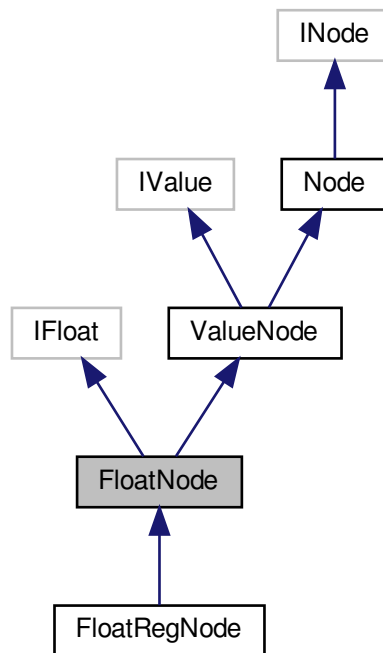
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Filestream.h](#)

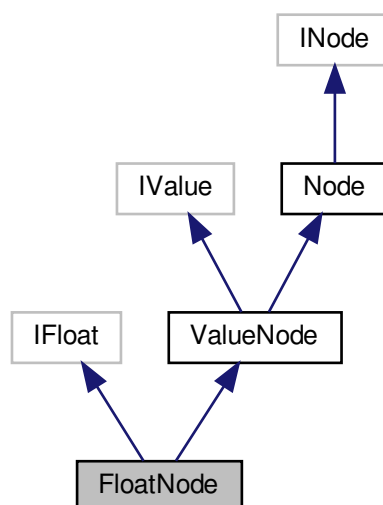
15.82 FloatNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for FloatNode:



Collaboration diagram for FloatNode:



Public Member Functions

- [FloatNode](#) ()
- [FloatNode](#) (std::shared_ptr< Node::NodeImpl > pFloat)
- virtual [~FloatNode](#) ()
- virtual void [SetValue](#) (double Value, bool [Verify](#)=true)
Set node value.
- virtual [IFloat](#) & [operator=](#) (double Value)
Set node value.
- virtual double [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get node value.
- virtual double [operator\(\)](#) ()
Get node value.
- virtual double [operator*](#) ()
Get node value.
- virtual double [GetMin](#) ()
Get minimum value allowed.
- virtual double [GetMax](#) ()
Get maximum value allowed.
- virtual bool [HasInc](#) ()
True if the float has a constant increment.
- virtual [EIncMode](#) [GetIncMode](#) ()
Get increment mode.
- virtual double [GetInc](#) ()
Get the constant increment if there is any.
- virtual [double_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)
Get list of valid value.
- virtual [ERepresentation](#) [GetRepresentation](#) ()
Get recommended representation.
- virtual [GenlCam::gcstring](#) [GetUnit](#) () const
Get the physical unit name.
- virtual [EDisplayNotation](#) [GetDisplayNotation](#) () const
Get the way the float should be converted to a string.
- virtual int64_t [GetDisplayPrecision](#) () const
Get the precision to be used when converting the float to a string.
- [Integer](#) * [GetIntAlias](#) ()
gets the interface of an alias node.
- [IEnumeration](#) * [GetEnumAlias](#) ()
gets the interface of an alias node.
- virtual void [ImposeMin](#) (double Value)
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)
Restrict maximum value.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Float

Additional Inherited Members

15.82.1 Detailed Description

[Interface](#) for string properties.

15.82.2 Constructor & Destructor Documentation

15.82.2.1 FloatNode() [1/2]

```
FloatNode ( )
```

15.82.2.2 FloatNode() [2/2]

```
FloatNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

15.82.2.3 ~FloatNode()

```
virtual ~FloatNode ( ) [virtual]
```

15.82.3 Member Function Documentation

15.82.3.1 GetDisplayNotation()

```
virtual EDisplayNotation GetDisplayNotation ( ) const [virtual]
```

Get the way the float should be converted to a string.

15.82.3.2 GetDisplayPrecision()

```
virtual int64_t GetDisplayPrecision ( ) const [virtual]
```

Get the precision to be used when converting the float to a string.

15.82.3.3 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( )
```

gets the interface of an alias node.

15.82.3.4 GetInc()

```
virtual double GetInc ( ) [virtual]
```

Get the constant increment if there is any.

15.82.3.5 GetIncMode()

```
virtual EIncMode GetIncMode ( ) [virtual]
```

Get increment mode.

15.82.3.6 GetIntAlias()

```
IInteger* GetIntAlias ( )
```

gets the interface of an alias node.

15.82.3.7 GetListOfValidValues()

```
virtual double_autovector_t GetListOfValidValues (
    bool bounded = true ) [virtual]
```

Get list of valid value.

15.82.3.8 GetMax()

```
virtual double GetMax ( ) [virtual]
```

Get maximum value allowed.

15.82.3.9 GetMin()

```
virtual double GetMin ( ) [virtual]
```

Get minimum value allowed.

15.82.3.10 GetRepresentation()

```
virtual ERepresentation GetRepresentation ( ) [virtual]
```

Get recommended representation.

15.82.3.11 GetUnit()

```
virtual GenICam::gcstring GetUnit ( ) const [virtual]
```

Get the physical unit name.

15.82.3.12 GetValue()

```
virtual double GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.82.3.13 HasInc()

```
virtual bool HasInc ( ) [virtual]
```

True if the float has a constant increment.

15.82.3.14 ImposeMax()

```
virtual void ImposeMax (
    double Value ) [virtual]
```

Restrict maximum value.

15.82.3.15 `ImposeMin()`

```
virtual void ImposeMin (
    double Value ) [virtual]
```

Restrict minimum value.

15.82.3.16 `operator()`

```
virtual double operator() ( ) [virtual]
```

Get node value.

15.82.3.17 `operator*()`

```
virtual double operator* ( ) [virtual]
```

Get node value.

15.82.3.18 `operator=()`

```
virtual IFloat& operator= (
    double Value ) [virtual]
```

Set node value.

15.82.3.19 `SetReference()`

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Float

Reimplemented from [ValueNode](#).

Reimplemented in [FloatRegNode](#).

15.82.3.20 `SetValue()`

```
virtual void SetValue (
    double Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

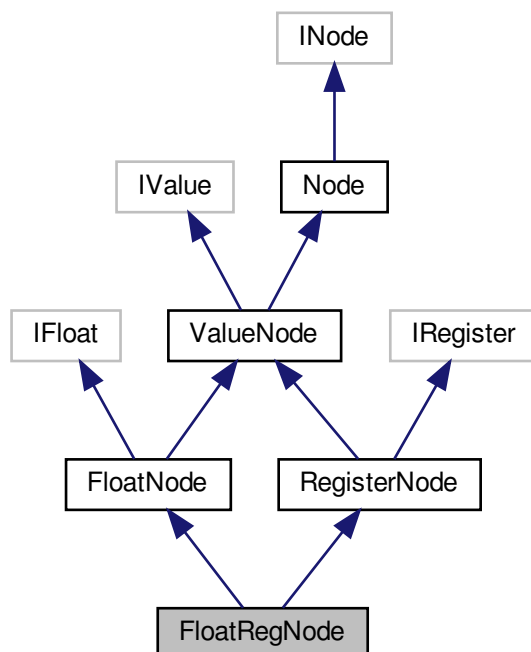
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatNode.h](#)

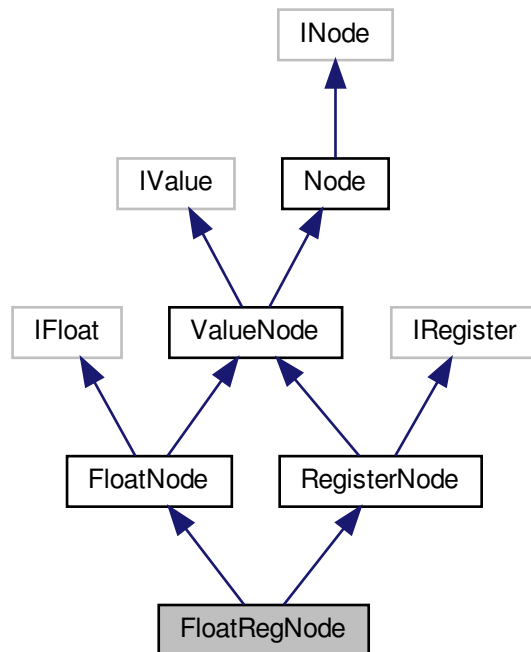
15.83 FloatRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for FloatRegNode:



Collaboration diagram for FloatRegNode:



Public Member Functions

- [FloatRegNode](#) ()
- [FloatRegNode](#) (std::shared_ptr< Node::NodeImpl > pFloat)
- virtual [~FloatRegNode](#) ()
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.83.1 Detailed Description

[Interface](#) for string properties.

15.83.2 Constructor & Destructor Documentation

15.83.2.1 FloatRegNode() [1/2]

```
FloatRegNode ( )
```

15.83.2.2 FloatRegNode() [2/2]

```
FloatRegNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

15.83.2.3 ~FloatRegNode()

```
virtual ~FloatRegNode ( ) [virtual]
```

15.83.3 Member Function Documentation

15.83.3.1 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [FloatNode](#).

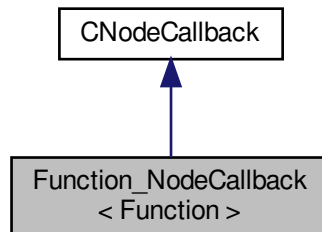
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatRegNode.h](#)

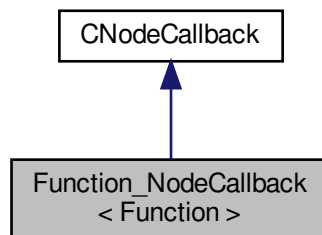
15.84 Function_NodeCallback< Function > Class Template Reference

Container for a function pointer.

Inheritance diagram for Function_NodeCallback< Function >:



Collaboration diagram for Function_NodeCallback< Function >:



Public Member Functions

- `Function_NodeCallback` (`INode *pNode`, `const Function &function`, `ECallbackType CallbackType`)
Constructor.
- `virtual void operator()` (`ECallbackType CallbackType`) `const`
execute operation: call the function
- `virtual void Destroy` ()
destroys teh object

Additional Inherited Members

15.84.1 Detailed Description

```
template<class Function>
```

```
class Spinnaker::GenApi::Function_NodeCallback< Function >
```

Container for a function pointer.

15.84.2 Constructor & Destructor Documentation

15.84.2.1 `Function_NodeCallback()`

```
Function_NodeCallback (
    INode * pNode,
    const Function & function,
    ECallbackType CallbackType ) [inline]
```

Constructor.

15.84.3 Member Function Documentation

15.84.3.1 `Destroy()`

```
virtual void Destroy ( ) [inline], [virtual]
```

destroys teh object

Implements [CNodeCallback](#).

15.84.3.2 `operator>()()`

```
virtual void operator() (
    ECallbackType CallbackType ) const [inline], [virtual]
```

execute operation: call the function

Implements [CNodeCallback](#).

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

- `include/SpinGenApi/NodeCallback.h`

15.85 gcstring Class Reference

Public Member Functions

- [gcstring](#) ()
- [gcstring](#) (const char *pc)
- [gcstring](#) (const char *pc, size_t n)
- [gcstring](#) (size_t count, char ch)
- [gcstring](#) (const [gcstring](#) &str)
- virtual [~gcstring](#) (void)
- virtual [gcstring](#) & [append](#) (const [gcstring](#) &str)
- virtual [gcstring](#) & [append](#) (size_t count, char ch)
- virtual [gcstring](#) & [assign](#) (const [gcstring](#) &str)
- virtual [gcstring](#) & [assign](#) (size_t count, char ch)
- virtual [gcstring](#) & [assign](#) (const char *pc)
- virtual [gcstring](#) & [assign](#) (const char *pc, size_t n)
- virtual int [compare](#) (const [gcstring](#) &str) const
- virtual const char * [c_str](#) (void) const
- virtual bool [empty](#) (void) const
- virtual size_t [find](#) (char ch, size_t offset=0) const
- virtual size_t [find](#) (const [gcstring](#) &str, size_t offset=0) const
- virtual size_t [find](#) (const [gcstring](#) &str, size_t offset, size_t count) const
- virtual size_t [find](#) (const char *pc, size_t offset=0) const
- virtual size_t [find](#) (const char *pc, size_t offset, size_t count) const
- virtual size_t [length](#) (void) const
- virtual size_t [size](#) (void) const
- virtual void [resize](#) (size_t n)
- virtual size_t [max_size](#) () const
- virtual [gcstring](#) [substr](#) (size_t offset=0, size_t count=GCSTRING_NPOS) const
- virtual size_t [find_first_of](#) (const [gcstring](#) &str, size_t offset=0) const
- virtual size_t [find_first_not_of](#) (const [gcstring](#) &str, size_t offset=0) const
- virtual void [swap](#) ([gcstring](#) &Right)
- bool [operator!=](#) (const [gcstring](#) &str) const
- bool [operator!=](#) (const char *pc) const
- [gcstring](#) & [operator+=](#) (const [gcstring](#) &str)
- [gcstring](#) [operator+=](#) (const [gcstring](#) &str) const
- [gcstring](#) & [operator+=](#) (const char *pc)
- [gcstring](#) & [operator+=](#) (char ch)
- [gcstring](#) [operator+=](#) (char ch) const
- [gcstring](#) & [operator=](#) (const [gcstring](#) &str)
- bool [operator==](#) (const [gcstring](#) &str) const
- bool [operator==](#) (const char *pc) const
- bool [operator<](#) (const [gcstring](#) &str) const
- bool [operator>](#) (const [gcstring](#) &str) const
- [operator](#) const char * (void) const
- void [operator delete](#) (void *pWhere)
- void [operator delete](#) (void *pWhere, void *pNewWhere)
- void * [operator new](#) (size_t uiSize)
- void * [operator new](#) (size_t uiSize, void *pWhere)

Static Public Member Functions

- static size_t [_npos](#) (void)

Static Public Attributes

- static const size_t [npos](#)

Friends

- [SPINNAKER_API](#) friend [gcstring operator+](#) (const [gcstring](#) &left, const [gcstring](#) &right)
- [SPINNAKER_API](#) friend [gcstring operator+](#) (const [gcstring](#) &left, const char *right)
- [SPINNAKER_API](#) friend [gcstring operator+](#) (const char *left, const [gcstring](#) &right)

15.85.1 Constructor & Destructor Documentation

15.85.1.1 [gcstring\(\)](#) [1/5]

```
gcstring ( )
```

15.85.1.2 [gcstring\(\)](#) [2/5]

```
gcstring (  
    const char * pc )
```

15.85.1.3 [gcstring\(\)](#) [3/5]

```
gcstring (  
    const char * pc,  
    size_t n )
```

15.85.1.4 [gcstring\(\)](#) [4/5]

```
gcstring (  
    size_t count,  
    char ch )
```

15.85.1.5 gcstring() [5/5]

```
gcstring (
    const gcstring & str )
```

15.85.1.6 ~gcstring()

```
virtual ~gcstring (
    void ) [virtual]
```

15.85.2 Member Function Documentation**15.85.2.1 _npos()**

```
static size_t _npos (
    void ) [static]
```

15.85.2.2 append() [1/2]

```
virtual gcstring& append (
    const gcstring & str ) [virtual]
```

15.85.2.3 append() [2/2]

```
virtual gcstring& append (
    size_t count,
    char ch ) [virtual]
```

15.85.2.4 assign() [1/4]

```
virtual gcstring& assign (
    const gcstring & str ) [virtual]
```

15.85.2.5 assign() [2/4]

```
virtual gcstring& assign (  
    size_t count,  
    char ch ) [virtual]
```

15.85.2.6 assign() [3/4]

```
virtual gcstring& assign (  
    const char * pc ) [virtual]
```

15.85.2.7 assign() [4/4]

```
virtual gcstring& assign (  
    const char * pc,  
    size_t n ) [virtual]
```

15.85.2.8 c_str()

```
virtual const char* c_str (  
    void ) const [virtual]
```

15.85.2.9 compare()

```
virtual int compare (  
    const gcstring & str ) const [virtual]
```

15.85.2.10 empty()

```
virtual bool empty (  
    void ) const [virtual]
```

15.85.2.11 find() [1/5]

```
virtual size_t find (  
    char ch,  
    size_t offset = 0 ) const [virtual]
```

15.85.2.12 find() [2/5]

```
virtual size_t find (  
    const gcstring & str,  
    size_t offset = 0 ) const [virtual]
```

15.85.2.13 find() [3/5]

```
virtual size_t find (  
    const gcstring & str,  
    size_t offset,  
    size_t count ) const [virtual]
```

15.85.2.14 find() [4/5]

```
virtual size_t find (  
    const char * pc,  
    size_t offset = 0 ) const [virtual]
```

15.85.2.15 find() [5/5]

```
virtual size_t find (  
    const char * pc,  
    size_t offset,  
    size_t count ) const [virtual]
```

15.85.2.16 find_first_not_of()

```
virtual size_t find_first_not_of (  
    const gcstring & str,  
    size_t offset = 0 ) const [virtual]
```

15.85.2.17 find_first_of()

```
virtual size_t find_first_of (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

15.85.2.18 length()

```
virtual size_t length (
    void ) const [virtual]
```

15.85.2.19 max_size()

```
virtual size_t max_size ( ) const [virtual]
```

15.85.2.20 operator const char *()

```
operator const char * (
    void ) const
```

15.85.2.21 operator delete() [1/2]

```
void operator delete (
    void * pWhere )
```

15.85.2.22 operator delete() [2/2]

```
void operator delete (
    void * pWhere,
    void * pNewWhere )
```

15.85.2.23 operator new() [1/2]

```
void* operator new (
    size_t uiSize )
```

15.85.2.24 operator new() [2/2]

```
void* operator new (
    size_t uiSize,
    void * pWhere )
```

15.85.2.25 operator!=() [1/2]

```
bool operator!= (
    const gcstring & str ) const
```

15.85.2.26 operator!=() [2/2]

```
bool operator!= (
    const char * pc ) const
```

15.85.2.27 operator+=() [1/5]

```
gcstring& operator+= (
    const gcstring & str )
```

15.85.2.28 operator+=() [2/5]

```
gcstring operator+= (
    const gcstring & str ) const
```

15.85.2.29 operator+=() [3/5]

```
gcstring& operator+= (
    const char * pc )
```

15.85.2.30 operator+=() [4/5]

```
gcstring& operator+= (
    char ch )
```


15.85.2.31 operator+=() [5/5]

```
gcstring operator+= (
    char ch ) const
```

15.85.2.32 operator<()

```
bool operator< (
    const gcstring & str ) const
```

15.85.2.33 operator=()

```
gcstring& operator= (
    const gcstring & str )
```

15.85.2.34 operator==() [1/2]

```
bool operator== (
    const gcstring & str ) const
```

15.85.2.35 operator==() [2/2]

```
bool operator== (
    const char * pc ) const
```

15.85.2.36 operator>()

```
bool operator> (
    const gcstring & str ) const
```

15.85.2.37 resize()

```
virtual void resize (
    size_t n ) [virtual]
```

15.85.2.38 size()

```
virtual size_t size (
    void ) const [virtual]
```

15.85.2.39 substr()

```
virtual gcstring substr (
    size_t offset = 0,
    size_t count = GCSTRING_NPOS ) const [virtual]
```

15.85.2.40 swap()

```
virtual void swap (
    gcstring & Right ) [virtual]
```

15.85.3 Friends And Related Function Documentation**15.85.3.1 operator+ [1/3]**

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const gcstring & right ) [friend]
```

15.85.3.2 operator+ [2/3]

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const char * right ) [friend]
```

15.85.3.3 operator+ [3/3]

```
SPINNAKER_API friend gcstring operator+ (
    const char * left,
    const gcstring & right ) [friend]
```

15.85.4 Member Data Documentation

15.85.4.1 npos

```
const size_t npos [static]
```

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

- [include/SpinGenApi/GCString.h](#)

15.86 GrabInfo Struct Reference

Public Member Functions

- [GrabInfo](#) (const string &deviceSerial)

Public Attributes

- unsigned int [numImagesGrabbed](#)
- unsigned int [numIncompleteImages](#)
- unsigned int [numRemovals](#)
- std::shared_ptr< [ImageEventHandlerImpl](#) > [imageEventHandler](#)

15.86.1 Constructor & Destructor Documentation

15.86.1.1 GrabInfo()

```
GrabInfo (  
    const string & deviceSerial ) [inline]
```

15.86.2 Member Data Documentation

15.86.2.1 imageEventHandler

```
std::shared_ptr<ImageEventHandlerImpl> imageEventHandler
```

15.86.2.2 numImagesGrabbed

`unsigned int numImagesGrabbed`

15.86.2.3 numIncompleteImages

`unsigned int numIncompleteImages`

15.86.2.4 numRemovals

`unsigned int numRemovals`

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

- [src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp](#)

15.87 GVCP_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- `uint32_t` [ChunkID](#)
- `uint32_t` [ChunkLength](#)

15.87.1 Detailed Description

header of a GVCP request packet

15.87.2 Member Data Documentation

15.87.2.1 ChunkID

`uint32_t` [ChunkID](#)

15.87.2.2 ChunkLength

uint32_t ChunkLength

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

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

15.88 GVCP_EVENT_ITEM Struct Reference

layout of a GVCP event item (Extended ID flag not set)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)
- uint16_t [StreamChannelId](#)
- uint16_t [BlockId](#)
- uint32_t [TimestampHigh](#)
- uint32_t [TimestampLow](#)

15.88.1 Detailed Description

layout of a GVCP event item (Extended ID flag not set)

15.88.2 Member Data Documentation

15.88.2.1 BlockId

uint16_t BlockId

15.88.2.2 EventId

uint16_t EventId

15.88.2.3 ReservedOrEventSize

uint16_t ReservedOrEventSize

15.88.2.4 StreamChannelId

uint16_t StreamChannelId

15.88.2.5 TimestampHigh

uint32_t TimestampHigh

15.88.2.6 TimestampLow

uint32_t TimestampLow

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.89 GVCP_EVENT_ITEM_BASIC Struct Reference

layout of a GVCP event item (common to all types)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)

15.89.1 Detailed Description

layout of a GVCP event item (common to all types)

15.89.2 Member Data Documentation

15.89.2.1 EventId

uint16_t EventId

15.89.2.2 ReservedOrEventSize

uint16_t ReservedOrEventSize

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.90 GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference

layout of a GVCP event item (Extended ID flag set)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)
- uint16_t [StreamChannelId](#)
- uint16_t [BlockId](#)
- uint32_t [BlockId64High](#)
- uint32_t [BlockId64Low](#)
- uint32_t [TimestampHigh](#)
- uint32_t [TimestampLow](#)

15.90.1 Detailed Description

layout of a GVCP event item (Extended ID flag set)

15.90.2 Member Data Documentation

15.90.2.1 BlockId

uint16_t BlockId

15.90.2.2 BlockId64High

`uint32_t BlockId64High`

15.90.2.3 BlockId64Low

`uint32_t BlockId64Low`

15.90.2.4 EventId

`uint16_t EventId`

15.90.2.5 ReservedOrEventSize

`uint16_t ReservedOrEventSize`

15.90.2.6 StreamChannelId

`uint16_t StreamChannelId`

15.90.2.7 TimestampHigh

`uint32_t TimestampHigh`

15.90.2.8 TimestampLow

`uint32_t TimestampLow`

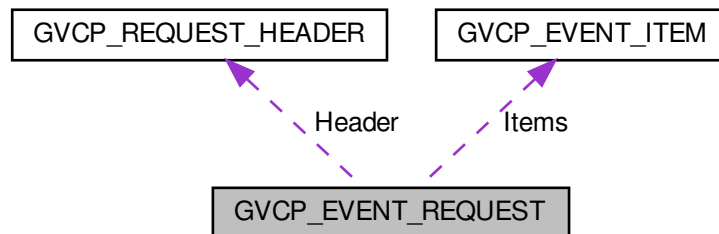
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterGEV.h`

15.91 GVCP_EVENT_REQUEST Struct Reference

Layout of a GVCP event request packet (Extended ID flag not set)

Collaboration diagram for GVCP_EVENT_REQUEST:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM](#) Items [1]

15.91.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag not set)

15.91.2 Member Data Documentation

15.91.2.1 Header

[GVCP_REQUEST_HEADER](#) Header

15.91.2.2 Items

[GVCP_EVENT_ITEM](#) Items [1]

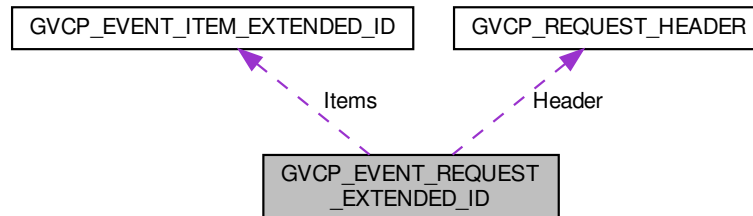
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.92 GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference

Layout of a GVCP event request packet (Extended ID flag set)

Collaboration diagram for GVCP_EVENT_REQUEST_EXTENDED_ID:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM_EXTENDED_ID](#) Items [1]

15.92.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag set)

15.92.2 Member Data Documentation

15.92.2.1 Header

[GVCP_REQUEST_HEADER](#) Header

15.92.2.2 Items

[GVCP_EVENT_ITEM_EXTENDED_ID](#) Items [1]

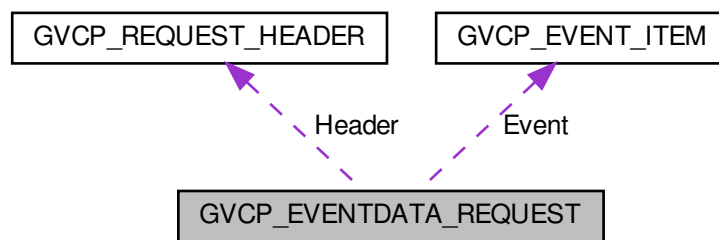
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.93 GVCP_EVENTDATA_REQUEST Struct Reference

Layout of a GVCP event data request packet (Extended ID flag not set)

Collaboration diagram for GVCP_EVENTDATA_REQUEST:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM](#) Event
- `uint32_t` [Data](#) [1]

15.93.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag not set)

15.93.2 Member Data Documentation

15.93.2.1 Data

`uint32_t` `Data`[1]

15.93.2.2 Event

[GVCP_EVENT_ITEM](#) Event

15.93.2.3 Header

[GVCP_REQUEST_HEADER](#) Header

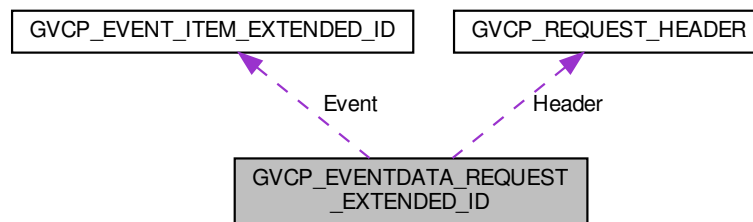
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.94 GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference

Layout of a GVCP event data request packet (Extended ID flag set)

Collaboration diagram for GVCP_EVENTDATA_REQUEST_EXTENDED_ID:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM_EXTENDED_ID](#) Event
- uint32_t [Data](#) [1]

15.94.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag set)

15.94.2 Member Data Documentation

15.94.2.1 Data

uint32_t [Data](#)[1]

15.94.2.2 Event

[GVCP_EVENT_ITEM_EXTENDED_ID](#) Event

15.94.2.3 Header

[GVCP_REQUEST_HEADER](#) Header

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.95 GVCP_REQUEST_HEADER Struct Reference

header of a GVCP request packet

Public Attributes

- uint8_t [Magic](#)
- uint8_t [Flags](#)
- uint16_t [Command](#)
- uint16_t [Length](#)
- uint16_t [ReqId](#)

15.95.1 Detailed Description

header of a GVCP request packet

15.95.2 Member Data Documentation

15.95.2.1 Command

uint16_t Command

15.95.2.2 Flags

uint8_t Flags

15.95.2.3 Length

`uint16_t` Length

15.95.2.4 Magic

`uint8_t` Magic

15.95.2.5 ReqId

`uint16_t` ReqId

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

- `include/SpinGenApi/EventAdapterGEV.h`

15.96 H264Option Struct Reference

Options for saving H264 files.

Public Member Functions

- [H264Option](#) ()

Public Attributes

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [width](#)
Width of source image.
- unsigned int [height](#)
Height of source image.
- unsigned int [bitrate](#)
Bit-rate to encode at.
- unsigned int [reserved](#) [256]
Reserved for future use.

15.96.1 Detailed Description

Options for saving H264 files.

15.96.2 Constructor & Destructor Documentation

15.96.2.1 H264Option()

```
H264Option ( ) [inline]
```

15.96.3 Member Data Documentation

15.96.3.1 bitrate

```
unsigned int bitrate
```

Bit-rate to encode at.

15.96.3.2 frameRate

```
float frameRate
```

Frame rate of the stream.

15.96.3.3 height

```
unsigned int height
```

Height of source image.

15.96.3.4 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

15.96.3.5 width

unsigned int width

Width of source image.

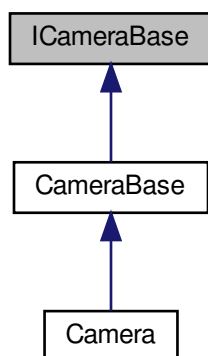
The documentation for this struct was generated from the following file:

- include/[SpinVideoDefs.h](#)

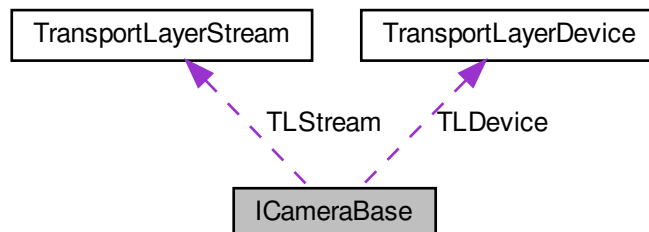
15.97 ICameraBase Class Reference

The interface file for base class for the camera object.

Inheritance diagram for ICameraBase:



Collaboration diagram for ICameraBase:



Public Member Functions

- virtual [~ICameraBase](#) (void)
- virtual void [Init](#) ()=0
- virtual void [Delinit](#) ()=0
- virtual bool [IsInitialized](#) ()=0
- virtual bool [IsValid](#) ()=0
- virtual [GenApi::INodeMap](#) & [GetNodeMap](#) () const =0
- virtual [GenApi::INodeMap](#) & [GetTLDeviceNodeMap](#) () const =0
- virtual [GenApi::INodeMap](#) & [GetTLStreamNodeMap](#) () const =0
- virtual [GenApi::EAccessMode](#) [GetAccessMode](#) () const =0
- virtual void [ReadPort](#) (uint64_t iAddress, void *pBuffer, size_t iSize)=0
- virtual void [WritePort](#) (uint64_t iAddress, const void *pBuffer, size_t iSize)=0
- virtual void [BeginAcquisition](#) ()=0
- virtual void [EndAcquisition](#) ()=0
- virtual [BufferOwnership](#) [GetBufferOwnership](#) () const =0
- virtual void [SetBufferOwnership](#) (const [BufferOwnership](#) mode)=0
- virtual uint64_t [GetUserBufferCount](#) () const =0
- virtual uint64_t [GetUserBufferSize](#) () const =0
- virtual uint64_t [GetUserBufferTotalSize](#) () const =0
- virtual void [SetUserBuffers](#) (void *const pMemBuffers, uint64_t totalSize)=0
- virtual void [SetUserBuffers](#) (void **const ppMemBuffers, const uint64_t bufferCount, const uint64_t [bufferSize](#))=0
- virtual [ImagePtr](#) [GetNextImage](#) (uint64_t grabTimeout=[EVENT_TIMEOUT_INFINITE](#), uint64_t streamID=0)=0
- virtual [GenICam::gcstring](#) [GetUniqueID](#) ()=0
- virtual bool [IsStreaming](#) () const =0
- virtual [GenICam::gcstring](#) [GetGuiXml](#) () const =0
- virtual void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)=0
- virtual void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister, const [GenICam::gcstring](#) &eventName)=0
- virtual void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)=0
- virtual unsigned int [GetNumImagesInUse](#) ()=0
- virtual unsigned int [GetNumDataStreams](#) ()=0
- virtual unsigned int [DiscoverMaxPacketSize](#) ()=0
- virtual void [ForceIP](#) ()=0

Public Attributes

- [TransportLayerDevice](#) [TLDevice](#)
Gets vital camera information by connecting to the camera's bootstrap registers.
- [TransportLayerStream](#) [TLStream](#)
Gets information about the stream data by connecting to the camera's bootstrap registers.

Protected Member Functions

- [ICameraBase](#) ()
- [ICameraBase](#) (const [ICameraBase](#) &)
- [ICameraBase](#) & [operator=](#) (const [ICameraBase](#) &)

Protected Attributes

- CameraBaseData * [m_pCameraBaseData](#)

Friends

- class [CameraInternal](#)
- class [InterfaceImpl](#)

15.97.1 Detailed Description

The interface file for base class for the camera object.

15.97.2 Constructor & Destructor Documentation

15.97.2.1 `~ICameraBase()`

```
virtual ~ICameraBase (
    void ) [inline], [virtual]
```

15.97.2.2 `ICameraBase()` [1/2]

```
ICameraBase ( ) [inline], [protected]
```

15.97.2.3 `ICameraBase()` [2/2]

```
ICameraBase (
    const ICameraBase & ) [inline], [protected]
```

15.97.3 Member Function Documentation

15.97.3.1 `BeginAcquisition()`

```
virtual void BeginAcquisition ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.2 DeInit()

```
virtual void DeInit ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.3 DiscoverMaxPacketSize()

```
virtual unsigned int DiscoverMaxPacketSize ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.4 EndAcquisition()

```
virtual void EndAcquisition ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.5 ForceIP()

```
virtual void ForceIP ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.6 GetAccessMode()

```
virtual GenApi::EAccessMode GetAccessMode ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.7 GetBufferOwnership()

```
virtual BufferOwnership GetBufferOwnership ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.8 GetGuiXml()

```
virtual GenICam::gcstring GetGuiXml ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.9 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT\_TIMEOUT\_INFINITE,
    uint64_t streamID = 0 ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.10 GetNodeMap()

```
virtual GenApi::INodeMap& GetNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.11 GetNumDataStreams()

```
virtual unsigned int GetNumDataStreams ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.12 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.13 GetTLDeviceNodeMap()

```
virtual GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.14 GetTLStreamNodeMap()

```
virtual GenApi::INodeMap& GetTLStreamNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.15 GetUniqueID()

```
virtual GenICam::gcstring GetUniqueID ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.16 GetUserBufferCount()

```
virtual uint64_t GetUserBufferCount ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.17 GetUserBufferSize()

```
virtual uint64_t GetUserBufferSize ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.18 GetUserBufferTotalSize()

```
virtual uint64_t GetUserBufferTotalSize ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.19 Init()

```
virtual void Init ( ) [pure virtual]
```

Implemented in [CameraBase](#), and [Camera](#).

15.97.3.20 IsInitialized()

```
virtual bool IsInitialized ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.21 IsStreaming()

```
virtual bool IsStreaming ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.22 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.23 operator=()

```
ICameraBase& operator= (
    const ICameraBase & ) [protected]
```

15.97.3.24 ReadPort()

```
virtual void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.25 RegisterEventHandler() [1/2]

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.26 RegisterEventHandler() [2/2]

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister,
    const GenICam::gcstring & eventName ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.27 SetBufferOwnership()

```
virtual void SetBufferOwnership (
    const BufferOwnership mode ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.28 SetUserBuffers() [1/2]

```
virtual void SetUserBuffers (
    void *const pMemBuffers,
    uint64_t totalSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.29 SetUserBuffers() [2/2]

```
virtual void SetUserBuffers (
    void **const ppMemBuffers,
    const uint64_t bufferCount,
    const uint64_t bufferSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.30 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.3.31 WritePort()

```
virtual void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.97.4 Friends And Related Function Documentation

15.97.4.1 CameraInternal

```
friend class CameraInternal [friend]
```

15.97.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

15.97.5 Member Data Documentation

15.97.5.1 m_pCameraBaseData

```
CameraBaseData* m_pCameraBaseData [protected]
```

15.97.5.2 TLDevice

[TransportLayerDevice](#) TLDevice

Gets vital camera information by connecting to the camera's bootstrap registers.

These nodes also access host software modules and the nodes can be used without having to call [Init\(\)](#) on the camera.

15.97.5.3 TLStream

[TransportLayerStream](#) TLStream

Gets information about the stream data by connecting to the camera's bootstrap registers.

These nodes also access host software modules and the nodes can be used without having to call [Init\(\)](#) on the camera.

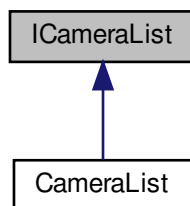
The documentation for this class was generated from the following file:

- include/Interface/[ICameraBase.h](#)

15.98 ICameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for ICameraList:



Public Member Functions

- virtual [~ICameraList](#) ()
- virtual [CameraPtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [CameraPtr GetByIndex](#) (unsigned int index) const =0
- virtual [CameraPtr GetBySerial](#) (std::string serialNumber) const =0
- virtual [CameraPtr GetByDeviceID](#) (std::string deviceID) const =0
- virtual void [Clear](#) ()=0
- virtual void [RemoveBySerial](#) (std::string serialNumber)=0
- virtual void [RemoveByIndex](#) (unsigned int index)=0
- virtual void [RemoveByDeviceID](#) (std::string deviceID)=0
- virtual void [Append](#) (const [CameraList](#) &list)=0

Protected Member Functions

- [ICameraList](#) ()
- [ICameraList](#) (const [ICameraList](#) &)
- [ICameraList](#) & [operator=](#) (const [ICameraList](#) &)

Protected Attributes

- CameraListData * [m_pCameraListData](#)

Friends

- class [InterfaceImpl](#)
- class [CameraListImpl](#)

15.98.1 Detailed Description

Used to hold a list of camera objects.

15.98.2 Constructor & Destructor Documentation

15.98.2.1 ~ICameraList()

```
virtual ~ICameraList ( ) [inline], [virtual]
```

15.98.2.2 ICameraList() [1/2]

```
ICameraList ( ) [inline], [protected]
```

15.98.2.3 ICameraList() [2/2]

```
ICameraList (
    const ICameraList & ) [inline], [protected]
```

15.98.3 Member Function Documentation

15.98.3.1 Append()

```
virtual void Append (
    const CameraList & list ) [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.2 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.3 GetByDeviceID()

```
virtual CameraPtr GetByDeviceID (
    std::string deviceID ) const [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.4 GetByIndex()

```
virtual CameraPtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.5 GetBySerial()

```
virtual CameraPtr GetBySerial (
    std::string serialNumber ) const [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.6 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.7 operator=()

```
ICameraList& operator= (
    const ICameraList & ) [protected]
```

15.98.3.8 operator[]()

```
virtual CameraPtr operator[] (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.9 RemoveByDeviceID()

```
virtual void RemoveByDeviceID (
    std::string deviceID ) [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.10 RemoveByIndex()

```
virtual void RemoveByIndex (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

15.98.3.11 RemoveBySerial()

```
virtual void RemoveBySerial (
    std::string serialNumber ) [pure virtual]
```

Implemented in [CameraList](#).

15.98.4 Friends And Related Function Documentation**15.98.4.1 CameraListImpl**

```
friend class CameraListImpl [friend]
```

15.98.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

15.98.5 Member Data Documentation

15.98.5.1 m_pCameraListData

```
CameraListData* m_pCameraListData [protected]
```

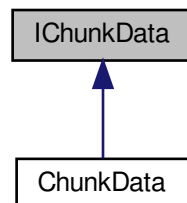
The documentation for this class was generated from the following file:

- [include/Interface/ICameraList.h](#)

15.99 IChunkData Class Reference

The [Interface](#) file for [ChunkData](#).

Inheritance diagram for IChunkData:



Public Member Functions

- virtual [~IChunkData](#) ()
- virtual void [SetChunks](#) ([GenApi::INodeMap](#) &pNodeMap)=0
- virtual [float64_t](#) [GetBlackLevel](#) () const =0
- virtual [int64_t](#) [GetFrameID](#) () const =0
- virtual [float64_t](#) [GetExposureTime](#) () const =0
- virtual [int64_t](#) [GetCompressionMode](#) () const =0
- virtual [float64_t](#) [GetCompressionRatio](#) () const =0
- virtual [int64_t](#) [GetTimestamp](#) () const =0
- virtual [int64_t](#) [GetExposureEndLineStatusAll](#) () const =0
- virtual [int64_t](#) [GetWidth](#) () const =0
- virtual [int64_t](#) [GetImage](#) () const =0
- virtual [int64_t](#) [GetHeight](#) () const =0
- virtual [float64_t](#) [GetGain](#) () const =0
- virtual [int64_t](#) [GetSequencerSetActive](#) () const =0
- virtual [int64_t](#) [GetCRC](#) () const =0

- virtual int64_t [GetOffsetX](#) () const =0
- virtual int64_t [GetOffsetY](#) () const =0
- virtual int64_t [GetSerialDataLength](#) () const =0
- virtual int64_t [GetPartSelector](#) () const =0
- virtual int64_t [GetPixelDynamicRangeMin](#) () const =0
- virtual int64_t [GetPixelDynamicRangeMax](#) () const =0
- virtual int64_t [GetTimestampLatchValue](#) () const =0
- virtual int64_t [GetLineStatusAll](#) () const =0
- virtual int64_t [GetCounterValue](#) () const =0
- virtual float64_t [GetTimerValue](#) () const =0
- virtual int64_t [GetScanLineSelector](#) () const =0
- virtual int64_t [GetEncoderValue](#) () const =0
- virtual int64_t [GetLinePitch](#) () const =0
- virtual int64_t [GetTransferBlockID](#) () const =0
- virtual int64_t [GetTransferQueueCurrentBlockCount](#) () const =0
- virtual int64_t [GetStreamChannelID](#) () const =0
- virtual float64_t [GetScan3dCoordinateScale](#) () const =0
- virtual float64_t [GetScan3dCoordinateOffset](#) () const =0
- virtual float64_t [GetScan3dInvalidDataValue](#) () const =0
- virtual float64_t [GetScan3dAxisMin](#) () const =0
- virtual float64_t [GetScan3dAxisMax](#) () const =0
- virtual float64_t [GetScan3dTransformValue](#) () const =0
- virtual float64_t [GetScan3dCoordinateReferenceValue](#) () const =0
- virtual int64_t [GetInferenceFrameId](#) () const =0
- virtual int64_t [GetInferenceResult](#) () const =0
- virtual float64_t [GetInferenceConfidence](#) () const =0
- virtual [InferenceBoundingBoxResult](#) [GetInferenceBoundingBoxResult](#) () const =0

Protected Member Functions

- [IChunkData](#) ()

15.99.1 Detailed Description

The [Interface](#) file for [ChunkData](#).

15.99.2 Constructor & Destructor Documentation

15.99.2.1 ~IChunkData()

```
virtual ~IChunkData ( ) [inline], [virtual]
```

15.99.2.2 IChunkData()

```
IChunkData ( ) [inline], [protected]
```

15.99.3 Member Function Documentation

15.99.3.1 GetBlackLevel()

```
virtual float64_t GetBlackLevel ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.2 GetCompressionMode()

```
virtual int64_t GetCompressionMode ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.3 GetCompressionRatio()

```
virtual float64_t GetCompressionRatio ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.4 GetCounterValue()

```
virtual int64_t GetCounterValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.5 GetCRC()

```
virtual int64_t GetCRC ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.6 GetEncoderValue()

```
virtual int64_t GetEncoderValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.7 GetExposureEndLineStatusAll()

```
virtual int64_t GetExposureEndLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.8 GetExposureTime()

```
virtual float64_t GetExposureTime ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.9 GetFrameID()

```
virtual int64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.10 GetGain()

```
virtual float64_t GetGain ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.11 GetHeight()

```
virtual int64_t GetHeight ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.12 GetImage()

```
virtual int64_t GetImage ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.13 GetInferenceBoundingBoxResult()

```
virtual InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.14 GetInferenceConfidence()

```
virtual float64_t GetInferenceConfidence ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.15 GetInferenceFrameId()

```
virtual int64_t GetInferenceFrameId ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.16 GetInferenceResult()

```
virtual int64_t GetInferenceResult ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.17 GetLinePitch()

```
virtual int64_t GetLinePitch ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.18 GetLineStatusAll()

```
virtual int64_t GetLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.19 GetOffsetX()

```
virtual int64_t GetOffsetX ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.20 GetOffsetY()

```
virtual int64_t GetOffsetY ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.21 GetPartSelector()

```
virtual int64_t GetPartSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.22 GetPixelDynamicRangeMax()

```
virtual int64_t GetPixelDynamicRangeMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.23 GetPixelDynamicRangeMin()

```
virtual int64_t GetPixelDynamicRangeMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.24 GetScan3dAxisMax()

```
virtual float64_t GetScan3dAxisMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.25 GetScan3dAxisMin()

```
virtual float64_t GetScan3dAxisMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.26 GetScan3dCoordinateOffset()

```
virtual float64_t GetScan3dCoordinateOffset ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.27 GetScan3dCoordinateReferenceValue()

```
virtual float64_t GetScan3dCoordinateReferenceValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.28 GetScan3dCoordinateScale()

```
virtual float64_t GetScan3dCoordinateScale ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.29 GetScan3dInvalidDataValue()

```
virtual float64_t GetScan3dInvalidDataValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.30 GetScan3dTransformValue()

```
virtual float64_t GetScan3dTransformValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.31 GetScanLineSelector()

```
virtual int64_t GetScanLineSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.32 GetSequencerSetActive()

```
virtual int64_t GetSequencerSetActive ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.33 GetSerialDataLength()

```
virtual int64_t GetSerialDataLength ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.34 GetStreamChannelID()

```
virtual int64_t GetStreamChannelID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.35 GetTimerValue()

```
virtual float64_t GetTimerValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.36 GetTimestamp()

```
virtual int64_t GetTimestamp ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.37 GetTimestampLatchValue()

```
virtual int64_t GetTimestampLatchValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.38 GetTransferBlockID()

```
virtual int64_t GetTransferBlockID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.39 GetTransferQueueCurrentBlockCount()

```
virtual int64_t GetTransferQueueCurrentBlockCount ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.40 GetWidth()

```
virtual int64_t GetWidth ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.99.3.41 SetChunks()

```
virtual void SetChunks (
    GenApi::INodeMap & pNodeMap ) [pure virtual]
```

Implemented in [ChunkData](#).

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

- [include/Interface/IChunkData.h](#)

15.100 IDataStream Class Reference

Public Member Functions

- virtual [~IDataStream](#) ()
- virtual void [ResetStreamHandle](#) (std::shared_ptr< Spinnaker::IGenTLDevice > pDevice)=0
- virtual [StreamTypeEnum](#) [GetStreamType](#) () const =0
- virtual void [AnnouncelImage](#) (size_t size)=0
- virtual void [AnnouncelImage](#) (size_t size, void *pPrivate)=0
- virtual void [AnnouncelImage](#) (size_t size, void *pData, void *pPrivate)=0
- virtual void [RevokelImages](#) ()=0
- virtual void [StartStream](#) (const unsigned int stream_index=0)=0
- virtual void [StopStream](#) ()=0
- virtual [ImagePtr](#) [GetNextImage](#) (uint64_t grabTimeout)=0
- virtual [ImagePtr](#) [GetNextImageInternal](#) (void **ppPrivate, uint64_t grabTimeout)=0
- virtual void [ReleaseImage](#) (const uint64_t imageID)=0
- virtual void [FlushQueueAllDiscard](#) ()=0
- virtual bool [IsStreaming](#) ()=0
- virtual void [KillBufferEvent](#) ()=0
- virtual bool [IsImageInUse](#) (const uint64_t imageID)=0
- virtual unsigned int [GetNumImagesInUse](#) () const =0
- virtual size_t [GetStreamInfoSizeType](#) (GenTL::STREAM_INFO_CMD iInfoCmd)=0
- virtual bool [GetStreamInfoBool8Type](#) (GenTL::STREAM_INFO_CMD iInfoCmd)=0
- virtual void * [GetBufferInfoPtrType](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual size_t [GetBufferInfoSizeType](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual uint64_t [GetBufferInfoUInt64Type](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual bool [GetBufferInfoBool8Type](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual void [RegisterImageEventHandler](#) (IImageEventHandler &imageEventHandler, EventPollingOptions pollingOption)=0
- virtual void [UnregisterImageEventHandler](#) (IImageEventHandler &imageEventHandler)=0
- virtual void [WaitOnImageEvent](#) (uint64_t timeout)=0
- virtual void [InitChunkAdapter](#) (GenApi::INodeMap &nodemap)=0
- virtual void [CleanupChunkAdapter](#) ()=0
- virtual GenTL::GC_ERROR [GetBufferChunkData](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::SINGLE_CHUNK_DATA *pChunkData, size_t *piNumChunks)=0
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, GenApi::SingleChunkData_t *ChunkData, int64_t NumChunks)=0
- virtual bool [IsCRCCheckEnabled](#) () const =0
- virtual [GenApi::INodeMap](#) & [GetNodeMap](#) () const =0
- virtual [GenApi::INodeMap](#) * [GetDeviceNodeMap](#) () const =0
- virtual Port & [GetPort](#) () const =0
- virtual const [TransportLayerStream](#) & [TransportLayerStreamInfo](#) () const =0

Protected Member Functions

- [IDataStream](#) ()

15.100.1 Constructor & Destructor Documentation

15.100.1.1 ~IDataStream()

```
virtual ~IDataStream ( ) [inline], [virtual]
```

15.100.1.2 IDataStream()

```
IDataStream ( ) [inline], [protected]
```

15.100.2 Member Function Documentation

15.100.2.1 AnnounceImage() [1/3]

```
virtual void AnnounceImage (
    size_t size ) [pure virtual]
```

15.100.2.2 AnnounceImage() [2/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pPrivate ) [pure virtual]
```

15.100.2.3 AnnounceImage() [3/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pData,
    void * pPrivate ) [pure virtual]
```

15.100.2.4 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    GenApi::SingleChunkData_t * ChunkData,
    int64_t NumChunks ) [pure virtual]
```

15.100.2.5 CleanupChunkAdapter()

```
virtual void CleanupChunkAdapter ( ) [pure virtual]
```

15.100.2.6 FlushQueueAllDiscard()

```
virtual void FlushQueueAllDiscard ( ) [pure virtual]
```

15.100.2.7 GetBufferChunkData()

```
virtual GenTL::GC_ERROR GetBufferChunkData (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::SINGLE_CHUNK_DATA * pChunkData,
    size_t * piNumChunks ) [pure virtual]
```

15.100.2.8 GetBufferInfoBool8Type()

```
virtual bool GetBufferInfoBool8Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

15.100.2.9 GetBufferInfoPtrType()

```
virtual void* GetBufferInfoPtrType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

15.100.2.10 GetBufferInfoSizeType()

```
virtual size_t GetBufferInfoSizeType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```


15.100.2.11 GetBufferInfoUInt64Type()

```
virtual uint64_t GetBufferInfoUInt64Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

15.100.2.12 GetDeviceNodeMap()

```
virtual GenApi::INodeMap* GetDeviceNodeMap ( ) const [pure virtual]
```

15.100.2.13 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout ) [pure virtual]
```

15.100.2.14 GetNextImageInternal()

```
virtual ImagePtr GetNextImageInternal (
    void ** ppPrivate,
    uint64_t grabTimeout ) [pure virtual]
```

15.100.2.15 GetNodeMap()

```
virtual GenApi::INodeMap& GetNodeMap ( ) const [pure virtual]
```

15.100.2.16 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) const [pure virtual]
```

15.100.2.17 GetPort()

```
virtual Port& GetPort ( ) const [pure virtual]
```

15.100.2.18 GetStreamInfoBool8Type()

```
virtual bool GetStreamInfoBool8Type (
    GenTL::STREAM_INFO_CMD iInfoCmd ) [pure virtual]
```

15.100.2.19 GetStreamInfoSizeType()

```
virtual size_t GetStreamInfoSizeType (
    GenTL::STREAM_INFO_CMD iInfoCmd ) [pure virtual]
```

15.100.2.20 GetStreamType()

```
virtual StreamTypeEnum GetStreamType ( ) const [pure virtual]
```

15.100.2.21 InitChunkAdapter()

```
virtual void InitChunkAdapter (
    GenApi::INodeMap & nodemap ) [pure virtual]
```

15.100.2.22 IsCRCCheckEnabled()

```
virtual bool IsCRCCheckEnabled ( ) const [pure virtual]
```

15.100.2.23 IsImageInUse()

```
virtual bool IsImageInUse (
    const uint64_t imageID ) [pure virtual]
```

15.100.2.24 IsStreaming()

```
virtual bool IsStreaming ( ) [pure virtual]
```

15.100.2.25 KillBufferEvent()

```
virtual void KillBufferEvent ( ) [pure virtual]
```

15.100.2.26 RegisterImageEventHandler()

```
virtual void RegisterImageEventHandler (
    IImageEventHandler & imageEventHandler,
    EventPollingOptions pollingOption ) [pure virtual]
```

15.100.2.27 ReleaseImage()

```
virtual void ReleaseImage (
    const uint64_t imageID ) [pure virtual]
```

15.100.2.28 ResetStreamHandle()

```
virtual void ResetStreamHandle (
    std::shared_ptr< Spinnaker::IGenTLDevice > pDevice ) [pure virtual]
```

15.100.2.29 RevokeImages()

```
virtual void RevokeImages ( ) [pure virtual]
```

15.100.2.30 StartStream()

```
virtual void StartStream (
    const unsigned int stream_index = 0 ) [pure virtual]
```

15.100.2.31 StopStream()

```
virtual void StopStream ( ) [pure virtual]
```

15.100.2.32 TransportLayerStreamInfo()

```
virtual const TransportLayerStream& TransportLayerStreamInfo ( ) const [pure virtual]
```

15.100.2.33 UnregisterImageEventHandler()

```
virtual void UnregisterImageEventHandler (
    IImageEventHandler & imageEventHandler ) [pure virtual]
```

15.100.2.34 WaitOnImageEvent()

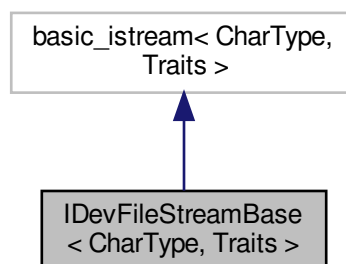
```
virtual void WaitOnImageEvent (
    uint64_t timeout ) [pure virtual]
```

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

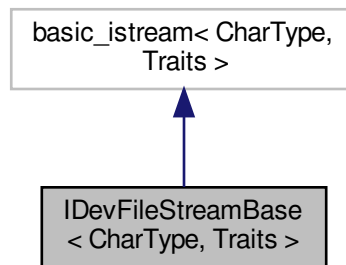
- [include/Interface/IStream.h](#)

15.101 IDevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for IDevFileStreamBase< CharType, Traits >:



Collaboration diagram for IDevFileStreamBase< CharType, Traits >:



Public Types

- typedef `IDevFileStreamBuf< CharType, Traits >` `filebuf_type`
- typedef `std::basic_ios< CharType, Traits >` `ios_type`
- typedef `std::basic_istream< CharType, Traits >` `istream_type`

Public Member Functions

- `filebuf_type * rdbuf () const`
- `bool is_open () const`
- `void open (Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::in)`
Open file on device in write mode.
- `void close ()`
Close the file on the device.

15.101.1 Member Typedef Documentation

15.101.1.1 filebuf_type

```
typedef IDevFileStreamBuf<CharType, Traits> filebuf_type
```

15.101.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

15.101.1.3 istream_type

```
typedef std::basic_istream<CharType, Traits> istream_type
```

15.101.2 Member Function Documentation

15.101.2.1 close()

```
void close ( ) [inline]
```

Close the file on the device.

15.101.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.101.2.3 open()

```
void open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

Open file on device in write mode.

Parameters

<i>pInterface</i>	NodeMap of the device to which the FileProtocolAdapter is attached
<i>pFileName</i>	Name of the file to open
<i>mode</i>	open mode

15.101.2.4 rdbuf()

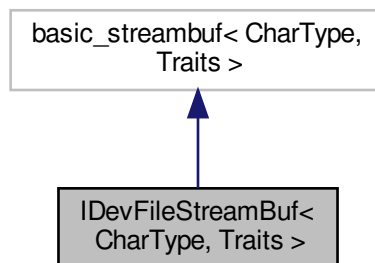
```
filebuf_type* rdbuf ( ) const [inline]
```

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

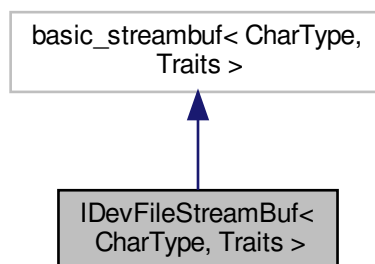
- include/SpinGenApi/[Filestream.h](#)

15.102 IDevFileStreamBuf< CharType, Traits > Class Template Reference

Inheritance diagram for IDevFileStreamBuf< CharType, Traits >:



Collaboration diagram for IDevFileStreamBuf< CharType, Traits >:



Public Member Functions

- [IDevFileStreamBuf](#) ()
- [~IDevFileStreamBuf](#) ()
- [filebuf_type](#) * [open](#) ([Spinnaker::GenApi::INodeMap](#) *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::in)
- bool [is_open](#) () const
- [filebuf_type](#) * [close](#) ()

Protected Member Functions

- int_type [underflow](#) ()
- int_type [pbackfail](#) (int_type c)

15.102.1 Constructor & Destructor Documentation

15.102.1.1 IDevFileStreamBuf()

```
IDevFileStreamBuf ( ) [inline]
```

15.102.1.2 ~IDevFileStreamBuf()

```
~IDevFileStreamBuf ( ) [inline]
```

15.102.2 Member Function Documentation

15.102.2.1 close()

```
filebuf_type* close ( ) [inline]
```

15.102.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.102.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

15.102.2.4 pbackfail()

```
int_type pbackfail (
    int_type c ) [inline], [protected]
```


15.102.2.5 underflow()

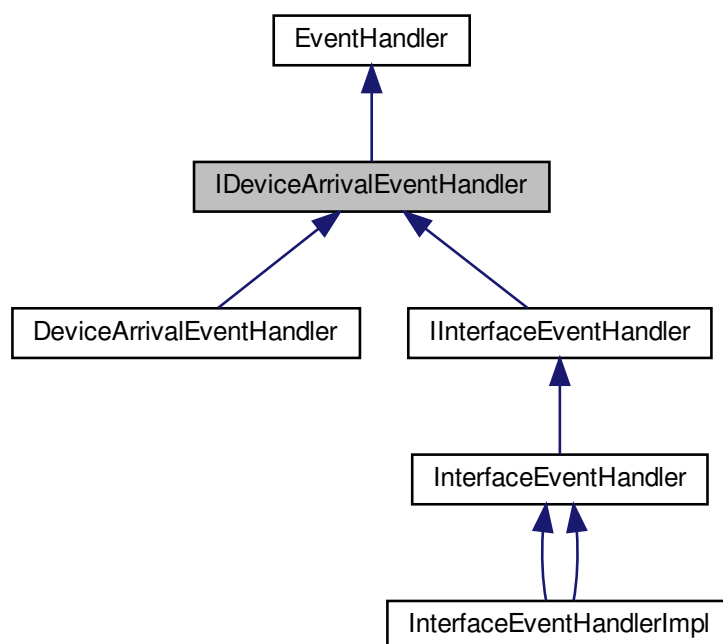
```
int_type underflow ( ) [inline], [protected]
```

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

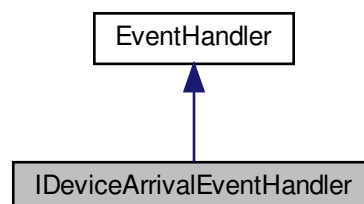
- include/SpinGenApi/[Filestream.h](#)

15.103 IDeviceArrivalEventHandler Class Reference

Inheritance diagram for IDeviceArrivalEventHandler:



Collaboration diagram for IDeviceArrivalEventHandler:



Public Member Functions

- virtual [~IDeviceArrivalEventHandler](#) ()
- virtual void [OnDeviceArrival](#) (uint64_t serialNumber)=0

Protected Member Functions

- [IDeviceArrivalEventHandler](#) ()
- [IDeviceArrivalEventHandler](#) (const [IDeviceArrivalEventHandler](#) &)
- [IDeviceArrivalEventHandler](#) & [operator=](#) (const [IDeviceArrivalEventHandler](#) &)

Additional Inherited Members

15.103.1 Constructor & Destructor Documentation

15.103.1.1 [~IDeviceArrivalEventHandler](#)()

[virtual](#) [~IDeviceArrivalEventHandler](#) () [\[inline\]](#), [\[virtual\]](#)

15.103.1.2 [IDeviceArrivalEventHandler](#)() [\[1/2\]](#)

[IDeviceArrivalEventHandler](#) () [\[inline\]](#), [\[protected\]](#)

15.103.1.3 [IDeviceArrivalEventHandler](#)() [\[2/2\]](#)

[IDeviceArrivalEventHandler](#) (
 const [IDeviceArrivalEventHandler](#) &) [\[inline\]](#), [\[protected\]](#)

15.103.2 Member Function Documentation

15.103.2.1 [OnDeviceArrival](#)()

[virtual](#) void [OnDeviceArrival](#) (
 uint64_t *serialNumber*) [\[pure virtual\]](#)

Implemented in [InterfaceEventHandlerImpl](#), [InterfaceEventHandlerImpl](#), [DeviceArrivalEventHandler](#), [InterfaceEventHandler](#), and [IInterfaceEventHandler](#).

15.103.2.2 operator=()

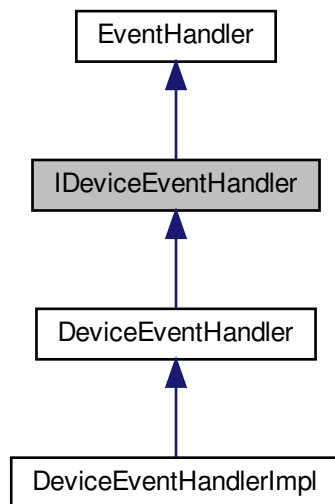
```
IDeviceArrivalEventHandler& operator= (
    const IDeviceArrivalEventHandler & ) [protected]
```

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

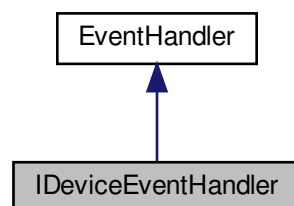
- include/Interface/[IDeviceArrivalEventHandler.h](#)

15.104 IDeviceEventHandler Class Reference

Inheritance diagram for IDeviceEventHandler:



Collaboration diagram for IDeviceEventHandler:



Public Member Functions

- virtual [~IDeviceEventHandler](#) ()
- virtual void [OnDeviceEvent](#) ([Spinnaker::GenICam::gcstring](#) eventName)=0
- virtual uint64_t [GetDeviceEventId](#) () const =0
- virtual [GenICam::gcstring](#) [GetDeviceEventName](#) () const =0

Protected Member Functions

- [IDeviceEventHandler](#) ()
- [IDeviceEventHandler](#) (const [IDeviceEventHandler](#) &)
- [IDeviceEventHandler](#) & [operator=](#) (const [IDeviceEventHandler](#) &)

Additional Inherited Members

15.104.1 Constructor & Destructor Documentation

15.104.1.1 ~IDeviceEventHandler()

[virtual ~IDeviceEventHandler](#) () [inline], [virtual]

15.104.1.2 IDeviceEventHandler() [1/2]

[IDeviceEventHandler](#) () [inline], [protected]

15.104.1.3 IDeviceEventHandler() [2/2]

[IDeviceEventHandler](#) (
 const [IDeviceEventHandler](#) &) [inline], [protected]

15.104.2 Member Function Documentation

15.104.2.1 GetDeviceEventId()

[virtual uint64_t](#) [GetDeviceEventId](#) () const [pure virtual]

Implemented in [DeviceEventHandler](#).

15.104.2.2 GetDeviceEventName()

```
virtual GenICam::gcstring GetDeviceEventName ( ) const [pure virtual]
```

Implemented in [DeviceEventHandler](#).

15.104.2.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (   
    Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Implemented in [DeviceEventHandlerImpl](#), and [DeviceEventHandler](#).

15.104.2.4 operator=()

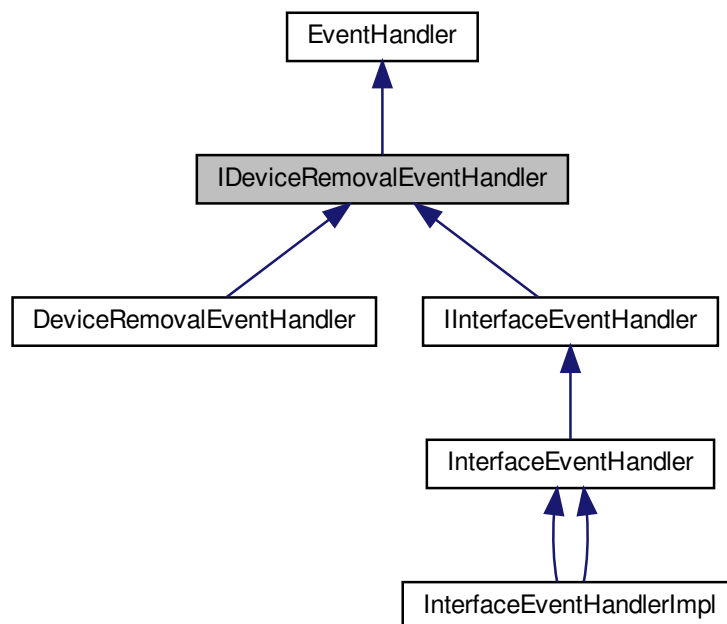
```
IDeviceEventHandler& operator= (   
    const IDeviceEventHandler & ) [protected]
```

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

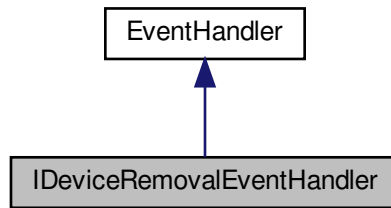
- [include/Interface/IDeviceEventHandler.h](#)

15.105 IDeviceRemovalEventHandler Class Reference

Inheritance diagram for IDeviceRemovalEventHandler:



Collaboration diagram for IDeviceRemovalEventHandler:



Public Member Functions

- virtual [~IDeviceRemovalEventHandler](#) ()
- virtual void [OnDeviceRemoval](#) (uint64_t serialNumber)=0

Protected Member Functions

- [IDeviceRemovalEventHandler](#) ()
- [IDeviceRemovalEventHandler](#) (const [IDeviceRemovalEventHandler](#) &)
- [IDeviceRemovalEventHandler](#) & [operator=](#) (const [IDeviceRemovalEventHandler](#) &)

Additional Inherited Members

15.105.1 Constructor & Destructor Documentation

15.105.1.1 [~IDeviceRemovalEventHandler](#)()

```
virtual ~IDeviceRemovalEventHandler ( ) [inline], [virtual]
```

15.105.1.2 [IDeviceRemovalEventHandler](#)() [1/2]

```
IDeviceRemovalEventHandler ( ) [inline], [protected]
```

15.105.1.3 IDeviceRemovalEventHandler() [2/2]

```
IDeviceRemovalEventHandler (
    const IDeviceRemovalEventHandler & ) [inline], [protected]
```

15.105.2 Member Function Documentation

15.105.2.1 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implemented in [InterfaceEventHandlerImpl](#), [InterfaceEventHandlerImpl](#), [InterfaceEventHandler](#), [DeviceRemovalEventHandler](#), and [IInterfaceEventHandler](#).

15.105.2.2 operator=()

```
IDeviceRemovalEventHandler& operator= (
    const IDeviceRemovalEventHandler & ) [protected]
```

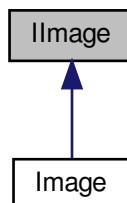
The documentation for this class was generated from the following file:

- [include/Interface/IDeviceRemovalEventHandler.h](#)

15.106 IImage Class Reference

The interface file for [Image](#).

Inheritance diagram for IImage:



Public Member Functions

- virtual [~Image](#) ()
- virtual [ColorProcessingAlgorithm](#) [GetColorProcessing](#) () const =0
- virtual [ImagePtr](#) [Convert](#) ([PixelFormatEnums](#) format, [ColorProcessingAlgorithm](#) colorAlgorithm=DEFAULT) const =0
- virtual void [Convert](#) ([ImagePtr](#) destinationImage, [PixelFormatEnums](#) format, [ColorProcessingAlgorithm](#) colorAlgorithm=DEFAULT) const =0
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat)=0
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData)=0
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData, [PayloadTypeInfoIds](#) dataPayloadType, size_t dataSize)=0
- virtual void [Release](#) ()=0
- virtual uint64_t [GetID](#) () const =0
- virtual void * [GetData](#) () const =0
- virtual void * [GetPrivateData](#) () const =0
- virtual float [GetDataAbsoluteMax](#) () const =0
- virtual float [GetDataAbsoluteMin](#) () const =0
- virtual size_t [GetBufferSize](#) () const =0
- virtual void [DeepCopy](#) (const [ImagePtr](#) pSrcImage)=0
- virtual size_t [GetWidth](#) () const =0
- virtual size_t [GetHeight](#) () const =0
- virtual size_t [GetStride](#) () const =0
- virtual size_t [GetBitsPerPixel](#) () const =0
- virtual size_t [GetNumChannels](#) () const =0
- virtual size_t [GetXOffset](#) () const =0
- virtual size_t [GetYOffset](#) () const =0
- virtual size_t [GetXPadding](#) () const =0
- virtual size_t [GetYPadding](#) () const =0
- virtual uint64_t [GetFrameID](#) () const =0
- virtual size_t [GetPayloadType](#) () const =0
- virtual [PayloadTypeInfoIds](#) [GetTLPayloadType](#) () const =0
- virtual uint64_t [GetTLPixelFormat](#) () const =0
- virtual [PixelFormatNamespaceID](#) [GetTLPixelFormatNamespace](#) () const =0
- virtual [GenlCam::gcstring](#) [GetPixelFormatName](#) () const =0
- virtual [PixelFormatEnums](#) [GetPixelFormat](#) () const =0
- virtual [PixelFormatIntType](#) [GetPixelFormatIntType](#) () const =0
- virtual bool [IsIncomplete](#) () const =0
- virtual size_t [GetValidPayloadSize](#) () const =0
- virtual uint64_t [GetChunkLayoutId](#) () const =0
- virtual uint64_t [GetTimeStamp](#) () const =0
- virtual void [Save](#) (const char *pFilename, [ImageFileFormat](#) format=FROM_FILE_EXT) const =0
- virtual void [Save](#) (const char *pFilename, [PNGOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [PPMOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [PGMOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [TIFFOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [JPEGOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [JPG2Option](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [BMPOption](#) &pOption) const =0
- virtual const [ChunkData](#) & [GetChunkData](#) () const =0
- virtual void [CalculateStatistics](#) ([ImageStatistics](#) &pStatistics)=0
- virtual bool [HasCRC](#) () const =0
- virtual bool [CheckCRC](#) () const =0
- virtual size_t [GetImageSize](#) () const =0
- virtual bool [IsInUse](#) ()=0
- virtual [ImageStatus](#) [GetImageStatus](#) () const =0
- virtual bool [IsCompressed](#) () const =0

Protected Member Functions

- [IImage](#) ()
- virtual ImageData * [GetImageData](#) () const =0

Friends

- class [Stream](#)

15.106.1 Detailed Description

The interface file for [Image](#).

15.106.2 Constructor & Destructor Documentation

15.106.2.1 ~IImage()

```
virtual ~IImage ( ) [inline], [virtual]
```

15.106.2.2 IImage()

```
IImage ( ) [inline], [protected]
```

15.106.3 Member Function Documentation

15.106.3.1 CalculateStatistics()

```
virtual void CalculateStatistics (
    ImageStatistics & pStatistics ) [pure virtual]
```

Implemented in [Image](#).

15.106.3.2 CheckCRC()

```
virtual bool CheckCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.3 Convert() [1/2]

```
virtual ImagePtr Convert (  
    PixelFormatEnums format,  
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.4 Convert() [2/2]

```
virtual void Convert (  
    ImagePtr destinationImage,  
    PixelFormatEnums format,  
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.5 DeepCopy()

```
virtual void DeepCopy (  
    const ImagePtr pSrcImage ) [pure virtual]
```

Implemented in [Image](#).

15.106.3.6 GetBitsPerPixel()

```
virtual size_t GetBitsPerPixel ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.7 GetBufferSize()

```
virtual size_t GetBufferSize ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.8 GetChunkData()

```
virtual const ChunkData& GetChunkData ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.9 GetChunkLayoutId()

```
virtual uint64_t GetChunkLayoutId ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.10 GetColorProcessing()

```
virtual ColorProcessingAlgorithm GetColorProcessing ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.11 GetData()

```
virtual void* GetData ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.12 GetDataAbsoluteMax()

```
virtual float GetDataAbsoluteMax ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.13 GetDataAbsoluteMin()

```
virtual float GetDataAbsoluteMin ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.14 GetFrameID()

```
virtual uint64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.15 GetHeight()

```
virtual size_t GetHeight ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.16 GetID()

```
virtual uint64_t GetID ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.17 GetImageData()

```
virtual ImageData* GetImageData ( ) const [protected], [pure virtual]
```

Implemented in [Image](#).

15.106.3.18 GetImageSize()

```
virtual size_t GetImageSize ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.19 GetImageStatus()

```
virtual ImageStatus GetImageStatus ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.20 GetNumChannels()

```
virtual size_t GetNumChannels ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.21 GetPayloadType()

```
virtual size_t GetPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.22 GetPixelFormat()

```
virtual PixelFormatEnums GetPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.23 GetPixelFormatIntType()

```
virtual PixelFormatIntType GetPixelFormatIntType ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.24 GetPixelFormatName()

```
virtual GenICam::gcstring GetPixelFormatName ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.25 GetPrivateData()

```
virtual void* GetPrivateData ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.26 GetStride()

```
virtual size_t GetStride ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.27 GetTimeStamp()

```
virtual uint64_t GetTimeStamp ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.28 GetTLPayloadType()

```
virtual PayloadTypeInfoIDs GetTLPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.29 GetTLPixelFormat()

```
virtual uint64_t GetTLPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.30 GetTLPixelFormatNamespace()

```
virtual PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.31 GetValidPayloadSize()

```
virtual size_t GetValidPayloadSize ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.32 GetWidth()

```
virtual size_t GetWidth ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.33 GetXOffset()

```
virtual size_t GetXOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.34 GetXPadding()

```
virtual size_t GetXPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.35 GetYOffset()

```
virtual size_t GetYOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.36 GetYPadding()

```
virtual size_t GetYPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.37 HasCRC()

```
virtual bool HasCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.38 IsCompressed()

```
virtual bool IsCompressed ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.39 IsIncomplete()

```
virtual bool IsIncomplete ( ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.40 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [Image](#).

15.106.3.41 Release()

```
virtual void Release ( ) [pure virtual]
```

Implemented in [Image](#).

15.106.3.42 ResetImage() [1/3]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat ) [pure virtual]
```

Implemented in [Image](#).

15.106.3.43 ResetImage() [2/3]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData ) [pure virtual]
```

Implemented in [Image](#).

15.106.3.44 ResetImage() [3/3]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData,
    PayloadTypeInfoIDs dataPayloadType,
    size_t dataSize ) [pure virtual]
```

Implemented in [Image](#).

15.106.3.45 Save() [1/8]

```
virtual void Save (
    const char * pFilename,
    ImageFileFormat format = FROM_FILE_EXT ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.46 Save() [2/8]

```
virtual void Save (
    const char * pFilename,
    PNGOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.47 Save() [3/8]

```
virtual void Save (  
    const char * pFilename,  
    PPMOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.48 Save() [4/8]

```
virtual void Save (  
    const char * pFilename,  
    PGMOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.49 Save() [5/8]

```
virtual void Save (  
    const char * pFilename,  
    TIFFOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.50 Save() [6/8]

```
virtual void Save (  
    const char * pFilename,  
    JPEGOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.51 Save() [7/8]

```
virtual void Save (  
    const char * pFilename,  
    JPG2Option & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.106.3.52 Save() [8/8]

```
virtual void Save (  
    const char * pFilename,  
    BMPOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.106.4 Friends And Related Function Documentation

15.106.4.1 Stream

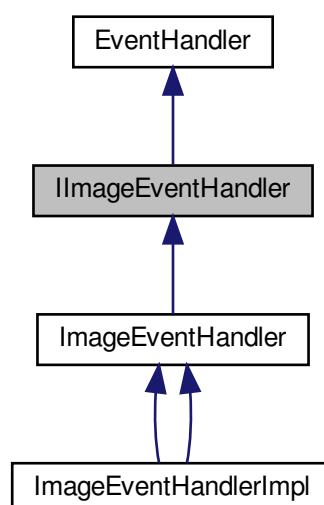
```
friend class Stream [friend]
```

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

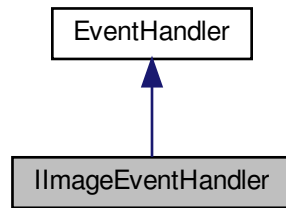
- [include/Interface/IImage.h](#)

15.107 IImageEventHandler Class Reference

Inheritance diagram for IImageEventHandler:



Collaboration diagram for `IImageEventHandler`:



Public Member Functions

- virtual `~IImageEventHandler()`
- virtual void `OnImageEvent(ImagePtr image)=0`

Protected Member Functions

- `IImageEventHandler()`
- `IImageEventHandler(const IImageEventHandler &)`
- `IImageEventHandler & operator=(const IImageEventHandler &)`

Additional Inherited Members

15.107.1 Constructor & Destructor Documentation

15.107.1.1 `~IImageEventHandler()`

```
virtual ~IImageEventHandler() [inline], [virtual]
```

15.107.1.2 `IImageEventHandler()` [1/2]

```
IImageEventHandler() [inline], [protected]
```

15.107.1.3 IImageEventHandler() [2/2]

```
IImageEventHandler (
    const IImageEventHandler & ) [inline], [protected]
```

15.107.2 Member Function Documentation

15.107.2.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

Implemented in [ImageEventHandlerImpl](#), [ImageEventHandlerImpl](#), and [ImageEventHandler](#).

15.107.2.2 operator=()

```
IImageEventHandler& operator= (
    const IImageEventHandler & ) [protected]
```

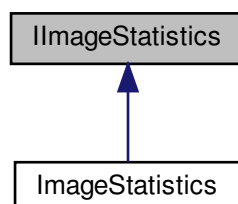
The documentation for this class was generated from the following file:

- [include/Interface/IImageEventHandler.h](#)

15.108 IImageStatistics Class Reference

The interface file for image statistics.

Inheritance diagram for IImageStatistics:



Public Member Functions

- virtual [~IImageStatistics](#) ()
- virtual void [EnableAll](#) ()=0
- virtual void [DisableAll](#) ()=0
- virtual void [EnableGreyOnly](#) ()=0
- virtual void [EnableRGBOnly](#) ()=0
- virtual void [EnableHSLOnly](#) ()=0
- virtual void [GetChannelStatus](#) ([StatisticsChannel](#) channel, bool *pEnabled) const =0
- virtual void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)=0
- virtual void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int *pMin, unsigned int *pMax) const =0
- virtual void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int *pPixelValueMin, unsigned int *pPixelValueMax) const =0
- virtual void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int *pNumPixelValues) const =0
- virtual void [GetMean](#) ([StatisticsChannel](#) channel, float *pPixelValueMean) const =0
- virtual void [GetHistogram](#) ([StatisticsChannel](#) channel, int **ppHistogram) const =0
- virtual void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int *pRangeMin=NULL, unsigned int *pRangeMax=NULL, unsigned int *pPixelValueMin=NULL, unsigned int *pPixelValueMax=NULL, unsigned int *pNumPixelValues=NULL, float *pPixelValueMean=NULL, int **ppHistogram=NULL) const =0

Protected Member Functions

- [IImageStatistics](#) ()
- [IImageStatistics](#) (const [IImageStatistics](#) &)

15.108.1 Detailed Description

The interface file for image statistics.

15.108.2 Constructor & Destructor Documentation

15.108.2.1 ~IImageStatistics()

```
virtual ~IImageStatistics ( ) [inline], [virtual]
```

15.108.2.2 IImageStatistics() [1/2]

```
IImageStatistics ( ) [inline], [protected]
```

15.108.2.3 IImageStatistics() [2/2]

```
IImageStatistics (
    const IImageStatistics & ) [inline], [protected]
```

15.108.3 Member Function Documentation

15.108.3.1 DisableAll()

```
virtual void DisableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.2 EnableAll()

```
virtual void EnableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
    StatisticsChannel channel,
    bool * pEnabled ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.7 GetHistogram()

```
virtual void GetHistogram (
    StatisticsChannel channel,
    int ** ppHistogram ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.8 GetMean()

```
virtual void GetMean (
    StatisticsChannel channel,
    float * pPixelValueMean ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
    StatisticsChannel channel,
    unsigned int * pNumPixelValues ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.10 GetPixelValueRange()

```
virtual void GetPixelValueRange (
    StatisticsChannel channel,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.11 GetRange()

```
virtual void GetRange (
    StatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.12 GetStatistics()

```
virtual void GetStatistics (
    StatisticsChannel channel,
    unsigned int * pRangeMin = NULL,
    unsigned int * pRangeMax = NULL,
    unsigned int * pPixelValueMin = NULL,
    unsigned int * pPixelValueMax = NULL,
    unsigned int * pNumPixelValues = NULL,
    float * pPixelValueMean = NULL,
    int ** ppHistogram = NULL ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.108.3.13 SetChannelStatus()

```
virtual void SetChannelStatus (
    StatisticsChannel channel,
    bool enabled ) [pure virtual]
```

Implemented in [ImageStatistics](#).

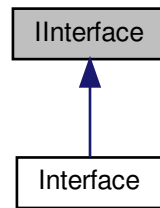
The documentation for this class was generated from the following file:

- [include/Interface/IImageStatistics.h](#)

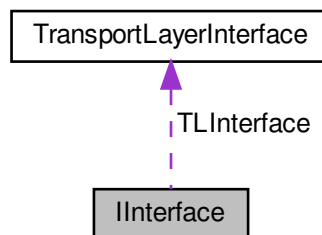
15.109 Interface Class Reference

The interface file for [Interface](#).

Inheritance diagram for IInterface:



Collaboration diagram for IInterface:



Public Member Functions

- virtual [~IInterface](#) ()
- virtual [CameraList GetCameras](#) (bool updateCameras=true) const =0
- virtual bool [UpdateCameras](#) ()=0
- virtual [GenApi::INodeMap & GetTLNodeMap](#) () const =0
- virtual void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)=0
- virtual void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)=0
- virtual bool [IsInUse](#) () const =0
- virtual void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL) const =0
- virtual bool [IsValid](#) ()=0

Public Attributes

- [TransportLayerInterface TLInterface](#)

Protected Member Functions

- [IInterface](#) ()
- [IInterface](#) (const [IInterface](#) &)
- [IInterface](#) & [operator=](#) (const [IInterface](#) &)

Protected Attributes

- InterfaceData * [m_pInterfaceData](#)

Friends

- class [InterfaceInternal](#)
- class [ProducerImpl](#)

15.109.1 Detailed Description

The interface file for [Interface](#).

15.109.2 Constructor & Destructor Documentation

15.109.2.1 ~IInterface()

```
virtual ~IInterface ( ) [inline], [virtual]
```

15.109.2.2 IInterface() [1/2]

```
IInterface ( ) [inline], [protected]
```

15.109.2.3 IInterface() [2/2]

```
IInterface (
    const IInterface & ) [inline], [protected]
```

15.109.3 Member Function Documentation

15.109.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateCameras = true ) const [pure virtual]
```

Implemented in [Interface](#).

15.109.3.2 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in [Interface](#).

15.109.3.3 IsInUse()

```
virtual bool IsInUse ( ) const [pure virtual]
```

Implemented in [Interface](#).

15.109.3.4 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in [Interface](#).

15.109.3.5 operator=()

```
IInterface& operator= (
    const IInterface & ) [protected]
```

15.109.3.6 RegisterEventHandler()

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [Interface](#).

15.109.3.7 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) const [pure virtual]
```

Implemented in [Interface](#).

15.109.3.8 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [Interface](#).

15.109.3.9 UpdateCameras()

```
virtual bool UpdateCameras ( ) [pure virtual]
```

Implemented in [Interface](#).

15.109.4 Friends And Related Function Documentation

15.109.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

15.109.4.2 ProducerImpl

```
friend class ProducerImpl [friend]
```

15.109.5 Member Data Documentation

15.109.5.1 m_pInterfaceData

```
InterfaceData* m_pInterfaceData [protected]
```

15.109.5.2 TLInterface

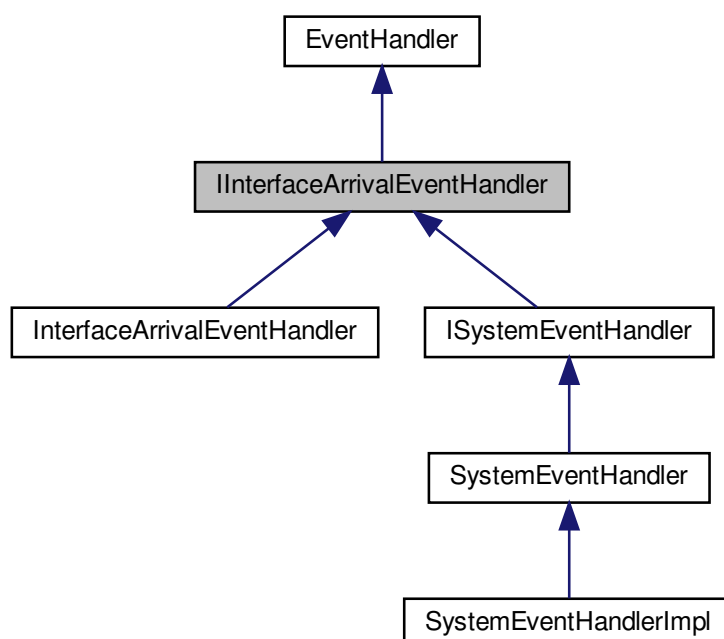
```
TransportLayerInterface TLInterface
```

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

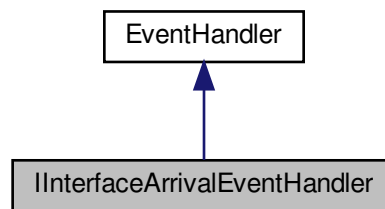
- include/Interface/[IInterface.h](#)

15.110 IInterfaceArrivalEventHandler Class Reference

Inheritance diagram for IInterfaceArrivalEventHandler:



Collaboration diagram for IInterfaceArrivalEventHandler:



Public Member Functions

- virtual [~IInterfaceArrivalEventHandler](#) ()
- virtual void [OnInterfaceArrival](#) (std::string interfaceID)=0

Protected Member Functions

- [IInterfaceArrivalEventHandler](#) ()
- [IInterfaceArrivalEventHandler](#) (const [IInterfaceArrivalEventHandler](#) &)
- [IInterfaceArrivalEventHandler](#) & [operator=](#) (const [IInterfaceArrivalEventHandler](#) &)

Additional Inherited Members

15.110.1 Constructor & Destructor Documentation

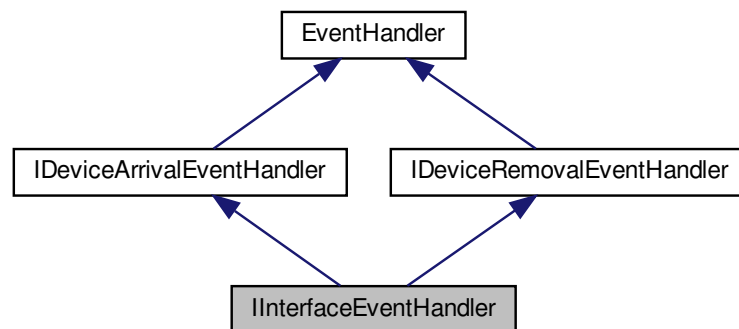
15.110.1.1 ~IInterfaceArrivalEventHandler()

```
virtual ~IInterfaceArrivalEventHandler ( ) [inline], [virtual]
```

15.110.1.2 IInterfaceArrivalEventHandler() [1/2]

```
IInterfaceArrivalEventHandler ( ) [inline], [protected]
```


Collaboration diagram for IInterfaceEventHandler:



Public Member Functions

- virtual [~IInterfaceEventHandler](#) ()
- virtual void [OnDeviceArrival](#) (uint64_t serialNumber)=0
- virtual void [OnDeviceRemoval](#) (uint64_t serialNumber)=0

Protected Member Functions

- [IInterfaceEventHandler](#) ()
- [IInterfaceEventHandler](#) (const [IInterfaceEventHandler](#) &)
- [IInterfaceEventHandler](#) & [operator=](#) (const [IInterfaceEventHandler](#) &)

Additional Inherited Members

15.111.1 Constructor & Destructor Documentation

15.111.1.1 ~IInterfaceEventHandler()

```
virtual ~IInterfaceEventHandler ( ) [inline], [virtual]
```

15.111.1.2 IInterfaceEventHandler() [1/2]

```
IInterfaceEventHandler ( ) [inline], [protected]
```

15.111.1.3 InterfaceEventHandler() [2/2]

```
IInterfaceEventHandler (
    const IInterfaceEventHandler & ) [inline], [protected]
```

15.111.2 Member Function Documentation

15.111.2.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IDeviceArrivalEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandler](#).

15.111.2.2 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IDeviceRemovalEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandler](#).

15.111.2.3 operator=()

```
IInterfaceEventHandler& operator= (
    const IInterfaceEventHandler & ) [protected]
```

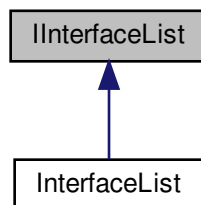
The documentation for this class was generated from the following file:

- [include/Interface/IInterfaceEventHandler.h](#)

15.112 IInterfaceList Class Reference

The interface file for [InterfaceList](#) class.

Inheritance diagram for IInterfaceList:



Public Member Functions

- virtual [~IInterfaceList](#) (void)
- virtual [InterfacePtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [InterfacePtr GetByIndex](#) (unsigned int index) const =0
- virtual void [Clear](#) ()=0
- virtual void [Append](#) (const [InterfaceList](#) *list)=0

Protected Member Functions

- [IInterfaceList](#) (void)
- [IInterfaceList](#) (const [IInterfaceList](#) &)
- [IInterfaceList](#) & [operator=](#) (const [IInterfaceList](#) &)

Protected Attributes

- [InterfaceListData](#) * [m_pInterfaceListData](#)

Friends

- class [InterfaceListImpl](#)

15.112.1 Detailed Description

The interface file for [InterfaceList](#) class.

15.112.2 Constructor & Destructor Documentation

15.112.2.1 ~IInterfaceList()

```
virtual ~IInterfaceList (
    void ) [inline], [virtual]
```

15.112.2.2 IInterfaceList() [1/2]

```
IInterfaceList (
    void ) [inline], [protected]
```

15.112.2.3 IInterfaceList() [2/2]

```
IInterfaceList (
    const IInterfaceList & ) [inline], [protected]
```

15.112.3 Member Function Documentation

15.112.3.1 Append()

```
virtual void Append (
    const IInterfaceList * list ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.112.3.2 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.112.3.3 GetByIndex()

```
virtual InterfacePtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [InterfaceList](#).

15.112.3.4 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [InterfaceList](#).

15.112.3.5 operator=()

```
IInterfaceList& operator= (
    const IInterfaceList & ) [protected]
```

15.112.3.6 operator[]()

```
virtual InterfacePtr operator[] (
    unsigned int index ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.112.4 Friends And Related Function Documentation

15.112.4.1 InterfaceListImpl

```
friend class InterfaceListImpl [friend]
```

15.112.5 Member Data Documentation

15.112.5.1 m_pInterfaceListData

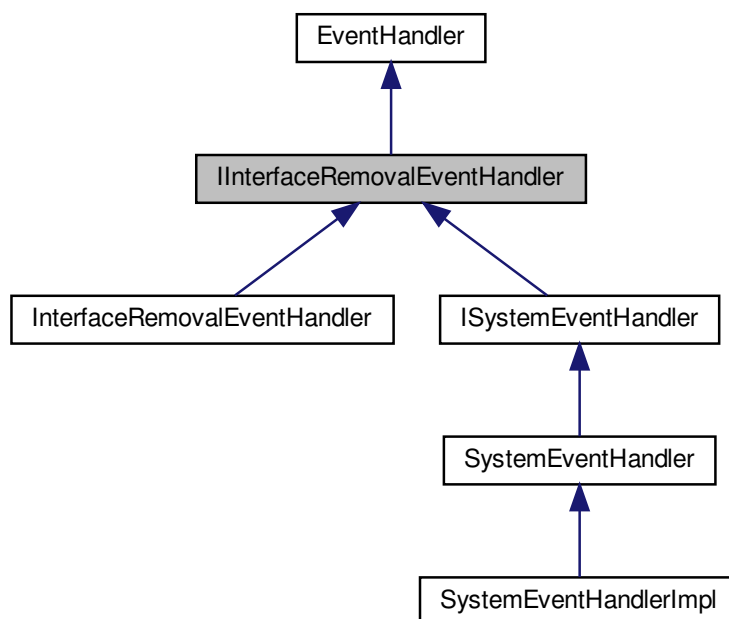
```
InterfaceListData* m_pInterfaceListData [protected]
```

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

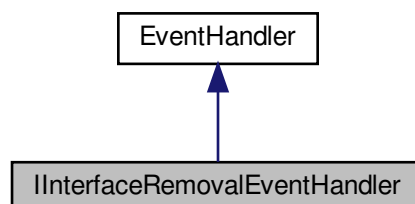
- include/Interface/[IInterfaceList.h](#)

15.113 IInterfaceRemovalEventHandler Class Reference

Inheritance diagram for IInterfaceRemovalEventHandler:



Collaboration diagram for IInterfaceRemovalEventHandler:



Public Member Functions

- virtual [~IInterfaceRemovalEventHandler](#) ()
- virtual void [OnInterfaceRemoval](#) (std::string interfaceID)=0

Protected Member Functions

- [IInterfaceRemovalEventHandler](#) ()
- [IInterfaceRemovalEventHandler](#) (const [IInterfaceRemovalEventHandler](#) &)
- [IInterfaceRemovalEventHandler](#) & [operator=](#) (const [IInterfaceRemovalEventHandler](#) &)

Additional Inherited Members

15.113.1 Constructor & Destructor Documentation

15.113.1.1 ~IInterfaceRemovalEventHandler()

```
virtual ~IInterfaceRemovalEventHandler ( ) [inline], [virtual]
```

15.113.1.2 IInterfaceRemovalEventHandler() [1/2]

```
IInterfaceRemovalEventHandler ( ) [inline], [protected]
```

15.113.1.3 IInterfaceRemovalEventHandler() [2/2]

```
IInterfaceRemovalEventHandler (
    const IInterfaceRemovalEventHandler & ) [inline], [protected]
```

15.113.2 Member Function Documentation

15.113.2.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

Implemented in [SystemEventHandlerImpl](#), [SystemEventHandler](#), [InterfaceRemovalEventHandler](#), and [ISystemEventHandler](#).

15.113.2.2 operator=()

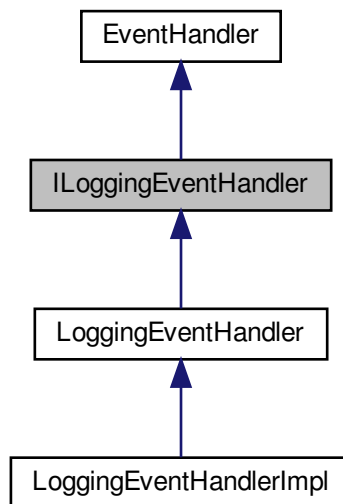
```
IInterfaceRemovalEventHandler& operator= (  
    const IInterfaceRemovalEventHandler & ) [protected]
```

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

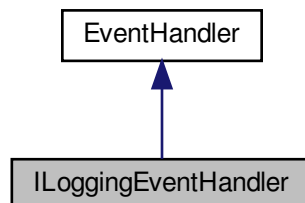
- include/Interface/IInterfaceRemovalEventHandler.h

15.114 ILoggingEventHandler Class Reference

Inheritance diagram for ILoggingEventHandler:



Collaboration diagram for ILoggingEventHandler:



Public Member Functions

- virtual [~ILoggerEventHandler](#) ()
- virtual void [OnLogEvent](#) ([LoggingEventDataPtr](#) eventPtr)=0

Protected Member Functions

- [ILoggerEventHandler](#) ()
- [ILoggerEventHandler](#) (const [ILoggerEventHandler](#) &)
- [ILoggerEventHandler](#) & [operator=](#) (const [ILoggerEventHandler](#) &)

Additional Inherited Members

15.114.1 Constructor & Destructor Documentation

15.114.1.1 ~ILoggerEventHandler()

[virtual ~ILoggerEventHandler](#) () [inline], [virtual]

15.114.1.2 ILoggerEventHandler() [1/2]

[ILoggerEventHandler](#) () [inline], [protected]

15.114.1.3 ILoggerEventHandler() [2/2]

[ILoggerEventHandler](#) (
 const [ILoggerEventHandler](#) &) [inline], [protected]

15.114.2 Member Function Documentation

15.114.2.1 OnLogEvent()

[virtual void OnLogEvent](#) (
 [LoggingEventDataPtr](#) eventPtr) [pure virtual]

Implemented in [LoggingEventHandler](#).

15.114.2.2 operator=()

```
ILoggingEventHandler& operator= (
    const ILoggingEventHandler & ) [protected]
```

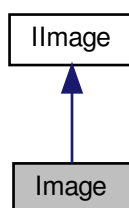
The documentation for this class was generated from the following file:

- [include/Interface/ILoggingEventHandler.h](#)

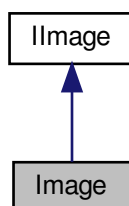
15.115 Image Class Reference

The image object class.

Inheritance diagram for Image:



Collaboration diagram for Image:



Public Member Functions

- virtual `~Image ()`
Virtual destructor.
- `ColorProcessingAlgorithm GetColorProcessing () const`
Gets the color algorithm used to produce the image.
- `ImagePtr Convert (Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const`
Converts the current image buffer to the specified output pixel format and stores the result in the specified image.
- `void Convert (ImagePtr destinationImage, Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const`
Converts the current image buffer to the specified output pixel format and stores the result in the specified destination image.
- `void ResetImage (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat)`
Sets new dimensions of the image object and allocates memory.
- `void ResetImage (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void *pData)`
Sets new dimensions of the image object.
- `void ResetImage (size_t width, size_t height, size_t offsetX, size_t offsetY, PixelFormatEnums pixelFormat, void *pData, PayloadTypeInfoIds dataPayloadType, size_t dataSize)`
Sets new dimensions of the image object.
- `void Release ()`
- `uint64_t GetID () const`
Gets a unique ID for this image.
- `void * GetData () const`
Gets a pointer to the data associated with the image.
- `float GetDataAbsoluteMax () const`
Get the value for which no image data will exceed.
- `float GetDataAbsoluteMin () const`
Get the value for which no image data will be less than.
- `void * GetPrivateData () const`
Gets a pointer to the user passed data associated with the image.
- `size_t GetBufferSize () const`
Gets the size of the buffer associated with the image in bytes.
- `void DeepCopy (const ImagePtr pSrcImage)`
*Performs a deep copy of the *Image*.*
- `size_t GetWidth () const`
Gets the width of the image in pixels.
- `size_t GetHeight () const`
Gets the height of the image in pixels.
- `size_t GetStride () const`
Gets the stride of the image in bytes.
- `size_t GetBitsPerPixel () const`
Gets the number of bits used per pixel in the image.
- `size_t GetNumChannels () const`
Gets the number of channels (depth) used in the image.
- `size_t GetXOffset () const`
Gets the ROI x offset in pixels for this image.
- `size_t GetYOffset () const`
Gets the ROI y offset in pixels for this image.
- `size_t GetXPadding () const`

- Gets the x padding in bytes for this image.*

 - `size_t GetYPadding () const`
- Gets the y padding in bytes for this image.*

 - `uint64_t GetFrameID () const`
- Gets the frame ID for this image.*

 - `size_t GetPayloadType () const`
- Gets the payload type that was transmitted.*

 - `PayloadTypeInfoIDs GetTLPayloadType () const`
- Gets the GenTL specific payload type that was transmitted.*

 - `uint64_t GetTLPixelFormat () const`
- Gets the pixel format of the image.*

 - `PixelFormatNamespaceID GetTLPixelFormatNamespace () const`
- Returns an enum value that represents the namespace in which this image's TL specific pixel format resides.*

 - `GenICam::gcstring GetPixelFormatName () const`
- Returns a string value that represents this image's pixel format.*

 - `Spinnaker::PixelFormatEnums GetPixelFormat () const`
- Returns an enum value that represents the pixel format of this image.*

 - `Spinnaker::PixelFormatIntType GetPixelFormatIntType () const`
- Returns an enum value that represents the integer type used in the pixel format of this image.*

 - `bool IsIncomplete () const`
- Returns a boolean value indicating if this image was incomplete.*

 - `size_t GetValidPayloadSize () const`
- Returns the size of valid data in the image payload.*

 - `uint64_t GetChunkLayoutId () const`
- Returns the id of the chunk data layout.*

 - `uint64_t GetTimeStamp () const`
- Gets the time stamp for the image in nanoseconds.*

 - `void Save (const char *pFilename, ImageFileFormat format=FROM_FILE_EXT) const`
- Saves the image to the specified file name with the file format specified.*

 - `void Save (const char *pFilename, PNGOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, PPMOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, PGMOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, TIFFOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, JPEGOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, JPG2Option &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, BMPOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `const ChunkData & GetChunkData () const`
- Returns a pointer to a chunk data interface.*

 - `void CalculateStatistics (ImageStatistics &pStatistics)`
- Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.*

 - `bool HasCRC () const`
- Checks if the image contains ImageCRC checksum from chunk data.*

 - `bool CheckCRC () const`
- Checks if the computed checksum matches with chunk data's ImageCRC.*

- `size_t GetImageSize ()` const
Returns the size of the image.
- `bool IsInUse ()`
Returns true if the image is still in use by the stream.
- `ImageStatus GetImageStatus ()` const
Returns data integrity status of the image returned from `GetNextImage()`
- `bool IsCompressed ()` const
Returns a boolean value indicating whether this image is compressed.

Static Public Member Functions

- static `ImagePtr Create ()`
Create an image object.
- static `ImagePtr Create (const ImagePtr image)`
Create an image object that is a deep copy of the input image.
- static `ImagePtr Create (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormat↔ Enums pixelFormat, void *pData)`
Create an image object with the specified parameters.
- static `ImagePtr Create (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormat↔ Enums pixelFormat, void *pData, PayloadTypeInfoIds dataPayloadType, size_t dataSize)`
Create an image object with the specified parameters.
- static void `SetDefaultColorProcessing (ColorProcessingAlgorithm colorAlgorithm)`
Sets the default color processing algorithm.
- static `ColorProcessingAlgorithm GetDefaultColorProcessing ()`
Gets the default color processing algorithm.
- static void `SetNumDecompressionThreads (unsigned int numThreads)`
Sets the default number of threads used for image decompression during `Convert()`.
- static unsigned int `GetNumDecompressionThreads ()`
Gets the number of threads used for image decompression during `Convert()`.
- static const char * `GetImageStatusDescription (ImageStatus status)`
Returns a string describing the meaning of the status enum.

Protected Member Functions

- `ImageData * GetImageData ()` const
- `Image ()`
- `Image (const ImagePtr image)`
- `Image (size_t width, size_t height, size_t offsetX, size_t offsetY, PixelFormatEnums pixelFormat, void *pData)`
- `Image (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void *pData, PayloadTypeInfoIds payloadType, size_t payloadSize)`
- `ImagePtr CreateShared ()` const
- void `DeepCopy (const Image &pSrcImage)`
- void `Convert (PixelFormatEnums format, Image &pDestImage, ColorProcessingAlgorithm color↔ Algorithm=DEFAULT)` const

Friends

- class [IDataStream](#)
- class [Stream](#)
- class [ImageConverter](#)
- class [ImageConverterIpp](#)
- class [ImageFiler](#)
- class [ImageStatsCalculator](#)
- class [ImageUtilityImpl](#)
- class [ImageUtilityPolarizationImpl](#)

15.115.1 Detailed Description

The image object class.

15.115.2 Constructor & Destructor Documentation

15.115.2.1 `~Image()`

```
virtual ~Image ( ) [virtual]
```

Virtual destructor.

15.115.2.2 `Image()` [1/4]

```
Image ( ) [protected]
```

15.115.2.3 `Image()` [2/4]

```
Image (  
    const ImagePtr image ) [protected]
```

15.115.2.4 `Image()` [3/4]

```
Image (  
    size_t width,  
    size_t height,  
    size_t offsetX,  
    size_t offsetY,  
    PixelFormatEnums pixelFormat,  
    void * pData ) [protected]
```

15.115.2.5 Image() [4/4]

```
Image (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat,
    void * pData,
    PayloadTypeInfoIDs payloadType,
    size_t payloadSize ) [protected]
```

15.115.3 Member Function Documentation

15.115.3.1 CalculateStatistics()

```
void CalculateStatistics (
    ImageStatistics & pStatistics ) [virtual]
```

Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.

Parameters

<i>pStatistics</i>	The statistics of an image.

Implements [IImage](#).

15.115.3.2 CheckCRC()

```
bool CheckCRC ( ) const [virtual]
```

Checks if the computed checksum matches with chunk data's ImageCRC.

Returns

Returns true if computed checksum matches with the chunk data's CRC and false otherwise.

Implements [IImage](#).

15.115.3.3 Convert() [1/3]

```
ImagePtr Convert (
    Spinnaker::PixelFormatEnums format,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [virtual]
```

Converts the current image buffer to the specified output pixel format and stores the result in the specified image.

The destination image does not need to be configured in any way before the call is made. Note that compressed images are decompressed before any further color processing or conversion during this call. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [SetNumDecompressionThreads\(\)](#).

See also

[PixelFormatEnums](#)
[SetNumDecompressionThreads\(unsigned int numThreads\)](#)

Parameters

<i>format</i>	Output format of the converted image.
<i>colorAlgorithm</i>	Optional color processing algorithm for producing the converted image

Returns

The converted image.

Implements [IImage](#).

15.115.3.4 Convert() [2/3]

```
void Convert (
    ImagePtr destinationImage,
    Spinnaker::PixelFormatEnums format,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [virtual]
```

Converts the current image buffer to the specified output pixel format and stores the result in the specified destination image.

The destination image buffer size must be sufficient to store the converted image data. Note that compressed images are decompressed before any further color processing or conversion during this call. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [SetNumDecompressionThreads\(\)](#).

See also

[Create](#)(size_t width, size_t height, size_t offsetX, size_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat, void* pData)
[SetNumDecompressionThreads\(unsigned int numThreads\)](#)

Parameters

<i>destinationImage</i>	Destination image where the converted output result will be stored.
<i>format</i>	Output format of the converted image.
<i>colorAlgorithm</i>	Optional color processing algorithm for producing the converted image.

Implements [IImage](#).

15.115.3.5 Convert() [3/3]

```
void Convert (
    PixelFormatEnums format,
    Image & pDestImage,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [protected]
```

15.115.3.6 Create() [1/4]

```
static ImagePtr Create ( ) [static]
```

Create an image object.

15.115.3.7 Create() [2/4]

```
static ImagePtr Create (
    const ImagePtr image ) [static]
```

Create an image object that is a deep copy of the input image.

Parameters

<i>image</i>	The input image to copy

15.115.3.8 Create() [3/4]

```
static ImagePtr Create (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
```

```
Spinnaker::PixelFormatEnums pixelFormat,
void * pData ) [static]
```

Create an image object with the specified parameters.

Parameters

<i>width</i>	The image width in pixels
<i>height</i>	The image height in pixels
<i>offsetX</i>	The image X offset
<i>offsetY</i>	The image Y offset
<i>pixelFormat</i>	The image pixel format
<i>pData</i>	The image data

15.115.3.9 Create() [4/4]

```
static ImagePtr Create (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat,
    void * pData,
    PayloadTypeInfoIDs dataPayloadType,
    size_t dataSize ) [static]
```

Create an image object with the specified parameters.

This function is used to create an image from existing image data with a specific payload type, such as a compressed image.

Note that images with chunk payload types are saved with only the image data preserved. Remember to specify the non-chunk equivalent payload type when creating images with these chunk payload types. For example, images need to be created with PAYLOAD_TYPE_IMAGE payload type if the original image had PAYLOAD_TYPE_EX-
TENDED_CHUNK payload type.

Parameters

<i>width</i>	The image width in pixels
<i>height</i>	The image height in pixels
<i>offsetX</i>	The image X offset
<i>offsetY</i>	The image Y offset
<i>pixelFormat</i>	The image pixel format
<i>pData</i>	The compressed image data
<i>dataPayloadType</i>	The payload type of the data. This value can be retrieved from an existing image by using the GetTLPayloadType() function call.
<i>dataSize</i>	The size of the provided data in bytes

See also

[GetTLPayloadType\(\)](#)

15.115.3.10 CreateShared()

```
ImagePtr CreateShared ( ) const [protected]
```

15.115.3.11 DeepCopy() [1/2]

```
void DeepCopy (
    const ImagePtr pSrcImage ) [virtual]
```

Performs a deep copy of the [Image](#).

After this operation, the image contents and member variables will be the same. The Images will not share a buffer. The [Image](#)'s current buffer will not be released.

Parameters

<i>pSrcImage</i>	The Image to copy the data from.

Implements [IImage](#).

15.115.3.12 DeepCopy() [2/2]

```
void DeepCopy (
    const Image & pSrcImage ) [protected]
```

15.115.3.13 GetBitsPerPixel()

```
size_t GetBitsPerPixel ( ) const [virtual]
```

Gets the number of bits used per pixel in the image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The number of bits used per pixel.

Implements [IImage](#).

15.115.3.14 GetBufferSize()

```
size_t GetBufferSize ( ) const [virtual]
```

Gets the size of the buffer associated with the image in bytes.

For user created images, this function returns the size of the user provided data if the data size was provided. If the data size was not provided, the buffer size is calculated based on the image dimensions and pixel format.

Returns

The size of the buffer, in bytes.

Implements [IImage](#).

15.115.3.15 GetChunkData()

```
const ChunkData& GetChunkData ( ) const [virtual]
```

Returns a pointer to a chunk data interface.

No ownership is transferred, the chunk data interface reference is valid until [Image::Release\(\)](#) is called on this image.

Returns

[ChunkData](#) interface that provides access to image chunks.

Implements [IImage](#).

15.115.3.16 GetChunkLayoutId()

```
uint64_t GetChunkLayoutId ( ) const [virtual]
```

Returns the id of the chunk data layout.

Returns

uint64_t value representing the id of the chunk data layout.

Implements [IImage](#).

15.115.3.17 GetColorProcessing()

```
ColorProcessingAlgorithm GetColorProcessing ( ) const [virtual]
```

Gets the color algorithm used to produce the image.

See also

[Convert\(\)](#)

Returns

The color processing algorithm used to produce the image.

Implements [IImage](#).

15.115.3.18 GetData()

```
void* GetData ( ) const [virtual]
```

Gets a pointer to the data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is released. The pointer may also be invalidated if the [Image](#) object is passed to [Image::Release\(\)](#).

Returns

A pointer to the image data.

Implements [IImage](#).

15.115.3.19 GetDataAbsoluteMax()

```
float GetDataAbsoluteMax ( ) const [virtual]
```

Get the value for which no image data will exceed.

Returns

the maximim theoretical image data value

Implements [IImage](#).

15.115.3.20 GetDataAbsoluteMin()

```
float GetDataAbsoluteMin ( ) const [virtual]
```

Get the value for which no image data will be less than.

Returns

the minimum theoretical image data value

Implements [IImage](#).

15.115.3.21 GetDefaultColorProcessing()

```
static ColorProcessingAlgorithm GetDefaultColorProcessing ( ) [static]
```

Gets the default color processing algorithm.

See also

[SetDefaultColorProcessing\(\)](#)

Returns

The default color processing algorithm.

15.115.3.22 GetFrameID()

```
uint64_t GetFrameID ( ) const [virtual]
```

Gets the frame ID for this image.

Returns

The frame ID.

Implements [IImage](#).

15.115.3.23 GetHeight()

```
size_t GetHeight ( ) const [virtual]
```

Gets the height of the image in pixels.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The height in pixels.

Implements [IImage](#).

15.115.3.24 GetID()

```
uint64_t GetID ( ) const [virtual]
```

Gets a unique ID for this image.

Each image in a steam will have a unique ID to help identify it.

Returns

The 64 bit unique id for this image.

Implements [IImage](#).

15.115.3.25 GetImageData()

```
ImageData* GetImageData ( ) const [protected], [virtual]
```

Implements [IImage](#).

15.115.3.26 GetImageSize()

```
size_t GetImageSize ( ) const [virtual]
```

Returns the size of the image.

For chunk images, only the size of chunk image portion is reported here. The entire chunk data payload can be queried by [GetValidPayloadSize\(\)](#). For compressed images, this value may be different than the image size once decompressed.

See also

[GetBufferSize\(\)](#)
[GetValidPayloadSize\(\)](#)

Returns

The image size in bytes.

Implements [IImage](#).

15.115.3.27 GetImageStatus()

```
ImageStatus GetImageStatus ( ) const [virtual]
```

Returns data integrity status of the image returned from GetNextImage()

Returns

Returns whether image has any data integrity issues.

Implements [IImage](#).

15.115.3.28 GetImageStatusDescription()

```
static const char* GetImageStatusDescription (
    ImageStatus status ) [static]
```

Returns a string describing the meaning of the status enum.

Returns

Returns the meaning of the status enum.

15.115.3.29 GetNumChannels()

```
size_t GetNumChannels ( ) const [virtual]
```

Gets the number of channels (depth) used in the image.

Returns 0 if the number of channels for the given pixel format is unknown.

Returns

The number of channels per pixel.

Implements [IImage](#).

15.115.3.30 GetNumDecompressionThreads()

```
static unsigned int GetNumDecompressionThreads ( ) [static]
```

Gets the number of threads used for image decompression during [Convert\(\)](#).

See also

[SetNumDecompressionThreads\(\)](#)

Returns

Number of parallel image decompression threads set to run.

15.115.3.31 GetPayloadType()

```
size_t GetPayloadType ( ) const [virtual]
```

Gets the payload type that was transmitted.

This is a device types specific value that identifies how the image was transmitted. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

Device types specific payload type.

Implements [IImage](#).

15.115.3.32 GetPixelFormat()

```
Spinnaker::PixelFormatEnums GetPixelFormat ( ) const [virtual]
```

Returns an enum value that represents the pixel format of this image.

The enum can be used with the easy access [GenICam](#) features available through the [Camera.h](#) header file. This easy access enum can also be used in the [Convert\(\)](#) function.

See also

[Convert\(\)](#)

Returns

enum value representing the PixelFormat.

Implements [IImage](#).

15.115.3.33 GetPixelFormatIntType()

```
Spinnaker::PixelFormatIntType GetPixelFormatIntType ( ) const [virtual]
```

Returns an enum value that represents the integer type used in the pixel format of this image.

Returns

enum value representing the integer type used.

Implements [IImage](#).

15.115.3.34 GetPixelFormatName()

```
GenICam::gcstring GetPixelFormatName ( ) const [virtual]
```

Returns a string value that represents this image's pixel format.

The string is a valid SFNC name that maps to the underlying TL specific pixel format. This is the most generic way to identify the pixel format of the image.

Returns

string value representing the PixelFormat.

Implements [IImage](#).

15.115.3.35 GetPrivateData()

```
void* GetPrivateData ( ) const [virtual]
```

Gets a pointer to the user passed data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is released. The pointer may also be invalidated if the [Image](#) object is passed to [Image::Release\(\)](#).

TODO: no way to set private data for image yet.

Returns

A pointer to the user passed data pointer.

Implements [IImage](#).

15.115.3.36 GetStride()

```
size_t GetStride ( ) const [virtual]
```

Gets the stride of the image in bytes.

The stride of an image is how many bytes are in each row. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The stride in bytes.

Implements [Image](#).

15.115.3.37 GetTimeStamp()

```
uint64_t GetTimeStamp ( ) const [virtual]
```

Gets the time stamp for the image in nanoseconds.

Returns

The time stamp of the image.

Implements [Image](#).

15.115.3.38 GetTLPayloadType()

```
PayloadTypeInfoIDs GetTLPayloadType ( ) const [virtual]
```

Gets the GenTL specific payload type that was transmitted.

This is a Transport Layer specific value that identifies how the image was transmitted. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

Transport Layer specific payload type.

Implements [Image](#).

15.115.3.39 GetTLPixelFormat()

```
uint64_t GetTLPixelFormat ( ) const [virtual]
```

Gets the pixel format of the image.

This is a Transport Layer specific pixel format that identifies how the pixels in the image should be interpreted. To understand how to interpret this value it is necessary to know what the transport layer namespace is. This can be retrieved through a call to [GetTLPixelFormatNamespace\(\)](#). This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

See also

[GetTLPixelFormatNamespace\(\)](#)

Returns

Transport Layer specific pixel format.

Implements [Image](#).

15.115.3.40 GetTLPixelFormatNamespace()

```
PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const [virtual]
```

Returns an enum value that represents the namespace in which this image's TL specific pixel format resides.

This information is important to properly interpret the value returned by [GetTLPixelFormat\(\)](#)

See also

[GetTLPixelFormat\(\)](#)

Returns

enum value representing the PixelFormatNamespace.

Implements [Image](#).

15.115.3.41 GetValidPayloadSize()

```
size_t GetValidPayloadSize ( ) const [virtual]
```

Returns the size of valid data in the image payload.

This is the actual amount of data read from the device. A user created image has a payload size of zero. The value returned here can be equal to the value returned by [GetImageSize\(\)](#) if image data is the only payload. Note that [GetBufferSize\(\)](#) returns the total size of bytes allocated for the image and could be equal to or greater than the size returned by this function.

See also

[GetBufferSize\(\)](#)

[GetImageSize\(\)](#)

Returns

size_t value representing valid payload.

Implements [IImage](#).

15.115.3.42 GetWidth()

```
size_t GetWidth ( ) const [virtual]
```

Gets the width of the image in pixels.

This information is retrieved from the Transport Layer image format headers. It is retrieved on a per image basis.

Returns

The width in pixels.

Implements [IImage](#).

15.115.3.43 GetXOffset()

```
size_t GetXOffset ( ) const [virtual]
```

Gets the ROI x offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The x offset in pixels.

Implements [IImage](#).

15.115.3.44 GetXPadding()

```
size_t GetXPadding ( ) const [virtual]
```

Gets the x padding in bytes for this image.

This is the number of bytes at the end of each line to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The x padding in bytes.

Implements [Image](#).

15.115.3.45 GetYOffset()

```
size_t GetYOffset ( ) const [virtual]
```

Gets the ROI y offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The y offset in pixels.

Implements [Image](#).

15.115.3.46 GetYPadding()

```
size_t GetYPadding ( ) const [virtual]
```

Gets the y padding in bytes for this image.

This is the number of bytes at the end of each image to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The y padding in bytes.

Implements [Image](#).

15.115.3.47 HasCRC()

```
bool HasCRC ( ) const [virtual]
```

Checks if the image contains ImageCRC checksum from chunk data.

Returns

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

Implements [IImage](#).

15.115.3.48 IsCompressed()

```
bool IsCompressed ( ) const [virtual]
```

Returns a boolean value indicating whether this image is compressed.

Returns

Returns true if image is compressed, false otherwise.

Implements [IImage](#).

15.115.3.49 IsIncomplete()

```
bool IsIncomplete ( ) const [virtual]
```

Returns a boolean value indicating if this image was incomplete.

An image is marked as incomplete if the transport layer received less data then it requested.

Returns

Returns true if image is incomplete, false otherwise.

Implements [IImage](#).

15.115.3.50 IsInUse()

```
bool IsInUse ( ) [virtual]
```

Returns true if the image is still in use by the stream.

Returns

Returns true if the image is in use and false otherwise.

Implements [IImage](#).

15.115.3.51 Release()

```
void Release ( ) [virtual]
```

Implements [IImage](#).

15.115.3.52 ResetImage() [1/3]

```
void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat ) [virtual]
```

Sets new dimensions of the image object and allocates memory.

Parameters

<i>width</i>	The width of image in pixels to set.
<i>height</i>	The height of image in pixels to set.
<i>offsetX</i>	The x offset in pixels to set.
<i>offsetY</i>	The y offset in pixels to set.
<i>pixelFormat</i>	Pixel format to set.

Implements [IImage](#).

15.115.3.53 ResetImage() [2/3]

```
void ResetImage (
    size_t width,
```



```

    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat,
    void * pData ) [virtual]

```

Sets new dimensions of the image object.

Parameters

<i>width</i>	The width of image in pixels to set.
<i>height</i>	The height of image in pixels to set.
<i>offsetX</i>	The x offset in pixels to set.
<i>offsetY</i>	The y offset in pixels to set.
<i>pixelFormat</i>	Pixel format to set.
<i>pData</i>	Pointer to the image buffer.

Implements [Image](#).

15.115.3.54 ResetImage() [3/3]

```

void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData,
    PayloadTypeInfoIDs dataPayloadType,
    size_t dataSize ) [virtual]

```

Sets new dimensions of the image object.

This function is used to create an image from existing image data with a specific payload type, such as a compressed image.

Parameters

<i>width</i>	The width of image in pixels to set.
<i>height</i>	The height of image in pixels to set.
<i>offsetX</i>	The x offset in pixels to set.
<i>offsetY</i>	The y offset in pixels to set.
<i>pixelFormat</i>	Pixel format to set.
<i>pData</i>	Pointer to the image buffer.
<i>dataPayloadType</i>	The payload type of the data. This value can be retrieved from an existing image by using the GetTLPayloadType() function call.
<i>dataSize</i>	The size of the provided data in bytes

See also

[GetTLPayloadType\(\)](#)

Implements [IImage](#).

15.115.3.55 Save() [1/8]

```
void Save (
    const char * pFilename,
    ImageFileFormat format = FROM_FILE_EXT ) const [virtual]
```

Saves the image to the specified file name with the file format specified.

Note that only the image data is saved regardless of the payload type used to transmit the image from camera to host. Saving a RAW image that was transmitted using the chunk image payload type will only contain the image data portion and none of the other chunks are preserved.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>format</i>	File format to save in.

Implements [IImage](#).

15.115.3.56 Save() [2/8]

```
void Save (
    const char * pFilename,
    PNGOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.115.3.57 Save() [3/8]

```
void Save (
    const char * pFilename,
    PPMOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.115.3.58 Save() [4/8]

```
void Save (
    const char * pFilename,
    PGMOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.115.3.59 Save() [5/8]

```
void Save (
    const char * pFilename,
    TIFFOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.115.3.60 Save() [6/8]

```
void Save (
    const char * pFilename,
    JPEGOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.115.3.61 Save() [7/8]

```
void Save (
    const char * pFilename,
    JPG2Option & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.115.3.62 Save() [8/8]

```
void Save (
    const char * pFilename,
    BMPOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.115.3.63 SetDefaultColorProcessing()

```
static void SetDefaultColorProcessing (
    ColorProcessingAlgorithm colorAlgorithm ) [static]
```

Sets the default color processing algorithm.

This method will be used for any image with the DEFAULT algorithm set. The method used is determined at the time of the [Convert\(\)](#) call, therefore the most recent execution of this function will take precedence. The default setting is shared within the current process.

Parameters

<i>colorAlgorithm</i>	The color processing algorithm to set.

See also

[GetDefaultColorProcessing\(\)](#)

15.115.3.64 SetNumDecompressionThreads()

```
static void SetNumDecompressionThreads (
    unsigned int numThreads ) [static]
```

Sets the default number of threads used for image decompression during [Convert\(\)](#).

The number of threads used is defaulted to be equal to one less than the number of concurrent threads supported by the system.

Parameters

<i>numThreads</i>	Number of parallel image decompression threads set to run

See also

[Convert\(\)](#)

15.115.4 Friends And Related Function Documentation

15.115.4.1 IDataStream

```
friend class IDataStream [friend]
```

15.115.4.2 ImageConverter

```
friend class ImageConverter [friend]
```

15.115.4.3 ImageConverterIpp

```
friend class ImageConverterIpp [friend]
```

15.115.4.4 ImageFiler

```
friend class ImageFiler [friend]
```

15.115.4.5 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

15.115.4.6 ImageUtilityImpl

```
friend class ImageUtilityImpl [friend]
```

15.115.4.7 ImageUtilityPolarizationImpl

```
friend class ImageUtilityPolarizationImpl [friend]
```

15.115.4.8 Stream

```
friend class Stream [friend]
```

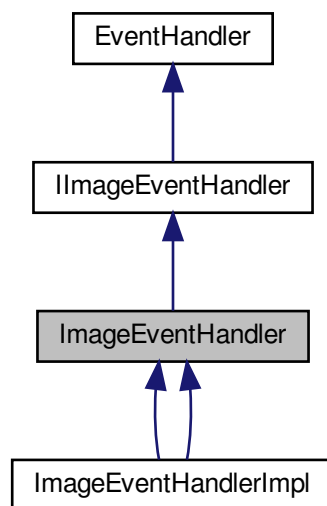
The documentation for this class was generated from the following file:

- [include/Image.h](#)

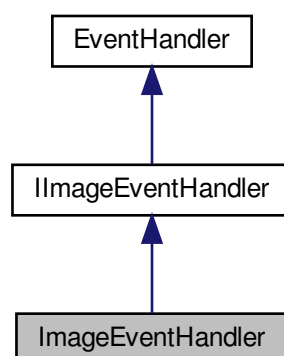
15.116 ImageEventHandler Class Reference

A handler for capturing image arrival events.

Inheritance diagram for ImageEventHandler:



Collaboration diagram for ImageEventHandler:



Public Member Functions

- [ImageEventHandler](#) ()

Default Constructor.

- virtual `~ImageEventHandler()`

Virtual Destructor.

- virtual void `OnImageEvent(ImagePtr image)=0`
Image event callback.

Protected Member Functions

- `ImageEventHandler & operator= (const ImageEventHandler &)`
Assignment operator.

Additional Inherited Members

15.116.1 Detailed Description

A handler for capturing image arrival events.

15.116.2 Constructor & Destructor Documentation

15.116.2.1 ImageEventHandler()

```
ImageEventHandler ( )
```

Default Constructor.

15.116.2.2 ~ImageEventHandler()

```
virtual ~ImageEventHandler ( ) [virtual]
```

Virtual Destructor.

15.116.3 Member Function Documentation

15.116.3.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

Image event callback.

Parameters

<i>image</i>	The ImagePtr object

Implements [IImageEventHandler](#).

Implemented in [ImageEventHandlerImpl](#), and [ImageEventHandlerImpl](#).

15.116.3.2 operator=()

```
ImageEventHandler& operator= (
    const ImageEventHandler & ) [protected]
```

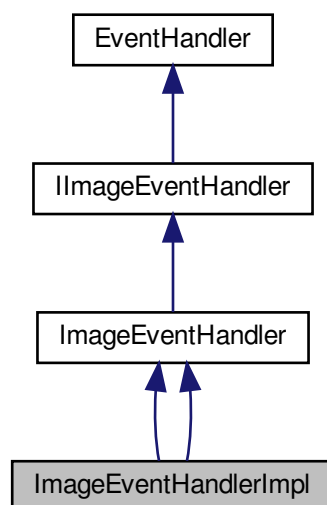
Assignment operator.

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

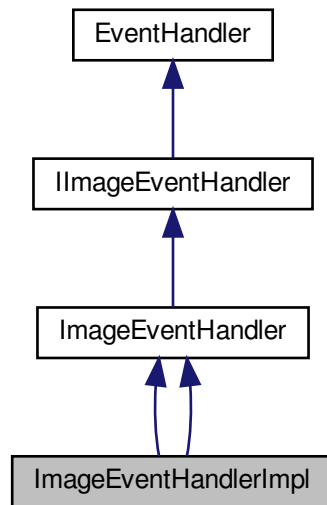
- [include/ImageEventHandler.h](#)

15.117 ImageEventHandlerImpl Class Reference

Inheritance diagram for ImageEventHandlerImpl:



Collaboration diagram for ImageEventHandlerImpl:



Public Member Functions

- [ImageEventHandlerImpl](#) (string deviceSerial)
- [~ImageEventHandlerImpl](#) ()
- void [OnImageEvent](#) ([ImagePtr](#) image)
Image event callback.
- [ImageEventHandlerImpl](#) ([CameraPtr](#) pCam)
- [~ImageEventHandlerImpl](#) ()
- void [OnImageEvent](#) ([ImagePtr](#) image)
Image event callback.
- int [getImageCount](#) ()
- int [getMaxImages](#) ()

Additional Inherited Members

15.117.1 Constructor & Destructor Documentation

15.117.1.1 ImageEventHandlerImpl() [1/2]

```
ImageEventHandlerImpl (
    string deviceSerial ) [inline]
```

15.117.1.2 ~ImageEventHandlerImpl() [1/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

15.117.1.3 ImageEventHandlerImpl() [2/2]

```
ImageEventHandlerImpl (
    CameraPtr pCam ) [inline]
```

15.117.1.4 ~ImageEventHandlerImpl() [2/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

15.117.2 Member Function Documentation

15.117.2.1 getImageCount()

```
int getImageCount ( ) [inline]
```

15.117.2.2 getMaxImages()

```
int getMaxImages ( ) [inline]
```

15.117.2.3 OnImageEvent() [1/2]

```
void OnImageEvent (
    ImagePtr image ) [virtual]
```

Image event callback.

Parameters

<i>image</i>	The ImagePtr object

Implements [ImageEventHandler](#).

15.117.2.4 OnImageEvent() [2/2]

```
void OnImageEvent (
    ImagePtr image ) [inline], [virtual]
```

Image event callback.

Parameters

<i>image</i>	The ImagePtr object

Implements [ImageEventHandler](#).

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

- [src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp](#)
- [src/ImageEvents/ImageEvents.cpp](#)

15.118 ImageInfo Struct Reference

Public Member Functions

- [ImageInfo](#) (string filename)

Public Attributes

- `size_t` [imageWidth](#)
- `size_t` [imageHeight](#)
- [PixelFormatEnums](#) [pixelFormat](#)
- `string` [imageFileName](#)
- `std::shared_ptr< fstream >` [imageFile](#)

15.118.1 Constructor & Destructor Documentation

15.118.1.1 ImageInfo()

```
ImageInfo (
    string filename ) [inline]
```

15.118.2 Member Data Documentation

15.118.2.1 imageFile

```
std::shared_ptr<fstream> imageFile
```

15.118.2.2 imageFileName

```
string imageFileName
```

15.118.2.3 imageHeight

```
size_t imageHeight
```

15.118.2.4 imageWidth

```
size_t imageWidth
```

15.118.2.5 pixelFormat

```
PixelFormatEnums pixelFormat
```

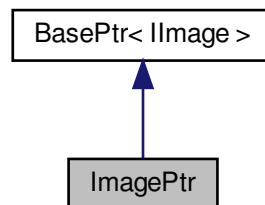
The documentation for this struct was generated from the following file:

- [src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp](#)

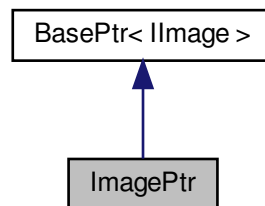
15.119 ImagePtr Class Reference

A reference tracked pointer to an image object.

Inheritance diagram for ImagePtr:



Collaboration diagram for ImagePtr:



Public Member Functions

- [ImagePtr](#) ()
Default constructor.
- [ImagePtr](#) (const int)
Default constructor with argument.
- [ImagePtr](#) (const long)
Default constructor with argument.
- [ImagePtr](#) (const std::nullptr_t)
Default constructor with argument.
- virtual [~ImagePtr](#) (void)
Virtual destructor.
- virtual [ImagePtr](#) & [operator=](#) (const [ImagePtr](#) &)
Assignment operator.

Additional Inherited Members

15.119.1 Detailed Description

A reference tracked pointer to an image object.

When the [ImagePtr](#) goes out of scope, it will trigger an auto release of the image from the stream.

15.119.2 Constructor & Destructor Documentation

15.119.2.1 ImagePtr() [1/4]

```
ImagePtr ( )
```

Default constructor.

15.119.2.2 ImagePtr() [2/4]

```
ImagePtr (
    const int )
```

Default constructor with argument.

15.119.2.3 ImagePtr() [3/4]

```
ImagePtr (
    const long )
```

Default constructor with argument.

15.119.2.4 ImagePtr() [4/4]

```
ImagePtr (
    const std::nullptr_t )
```

Default constructor with argument.

15.119.2.5 ~ImagePtr()

```
virtual ~ImagePtr (
    void ) [virtual]
```

Virtual destructor.

15.119.3 Member Function Documentation

15.119.3.1 operator=()

```
virtual ImagePtr& operator= (
    const ImagePtr & ) [virtual]
```

Assignment operator.

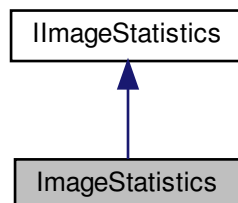
The documentation for this class was generated from the following file:

- include/[ImagePtr.h](#)

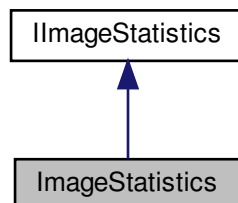
15.120 ImageStatistics Class Reference

Represents image statistics for an image.

Inheritance diagram for ImageStatistics:



Collaboration diagram for ImageStatistics:



Public Member Functions

- [ImageStatistics](#) ()
Default constructor.
- virtual [~ImageStatistics](#) ()
Default destructor.
- [ImageStatistics](#) (const [ImageStatistics](#) &other)
Copy constructor.
- [ImageStatistics](#) & [operator=](#) (const [ImageStatistics](#) &other)
Assignment operator.
- virtual void [EnableAll](#) ()
Enable all channels.
- virtual void [DisableAll](#) ()
Disable all channels.
- virtual void [EnableGreyOnly](#) ()
Enable only the grey channel.
- virtual void [EnableRGBOnly](#) ()
Enable only the RGB channels.
- virtual void [EnableHSLOnly](#) ()
Enable only the HSL channels.
- virtual void [GetChannelStatus](#) ([StatisticsChannel](#) channel, bool *pEnabled) const
Gets the status of a statistics channel.
- virtual void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)
Sets the status of a statistics channel.
- virtual void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int *pMin, unsigned int *pMax) const
Gets the range of a statistics channel.
- virtual void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int *pPixelValueMin, unsigned int *pPixelValueMax) const
Gets the range of a statistics channel.
- virtual void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int *pNumPixelValues) const
Gets the number of unique pixel values in the image.
- virtual void [GetMean](#) ([StatisticsChannel](#) channel, float *pPixelValueMean) const
Gets the mean of the image.
- virtual void [GetHistogram](#) ([StatisticsChannel](#) channel, int **ppHistogram) const
Gets the histogram for the image.
- virtual void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int *pRangeMin=NULL, unsigned int *pRangeMax=NULL, unsigned int *pPixelValueMin=NULL, unsigned int *pPixelValueMax=NULL, unsigned int *pNumPixelValues=NULL, float *pPixelValueMean=NULL, int **ppHistogram=NULL) const
Gets all statistics for the image.

Friends

- class [ImageStatsCalculator](#)

Additional Inherited Members

15.120.1 Detailed Description

Represents image statistics for an image.

15.120.2 Constructor & Destructor Documentation

15.120.2.1 ImageStatistics() [1/2]

```
ImageStatistics ( )
```

Default constructor.

15.120.2.2 ~ImageStatistics()

```
virtual ~ImageStatistics ( ) [virtual]
```

Default destructor.

15.120.2.3 ImageStatistics() [2/2]

```
ImageStatistics (
    const ImageStatistics & other )
```

Copy constructor.

15.120.3 Member Function Documentation

15.120.3.1 DisableAll()

```
virtual void DisableAll ( ) [virtual]
```

Disable all channels.

Implements [IImageStatistics](#).

15.120.3.2 EnableAll()

```
virtual void EnableAll ( ) [virtual]
```

Enable all channels.

Implements [IImageStatistics](#).

15.120.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [virtual]
```

Enable only the grey channel.

Implements [IImageStatistics](#).

15.120.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [virtual]
```

Enable only the HSL channels.

Implements [IImageStatistics](#).

15.120.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [virtual]
```

Enable only the RGB channels.

Implements [IImageStatistics](#).

15.120.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
    StatisticsChannel channel,
    bool * pEnabled ) const [virtual]
```

Gets the status of a statistics channel.

Parameters

<i>channel</i>	The statistics channel.
<i>pEnabled</i>	Whether the channel is enabled.

See also

[SetChannelStatus\(\)](#)

Implements [IImageStatistics](#).

15.120.3.7 GetHistogram()

```
virtual void GetHistogram (
    StatisticsChannel channel,
    int ** ppHistogram ) const [virtual]
```

Gets the histogram for the image.

Parameters

<i>channel</i>	The statistics channel.
<i>ppHistogram</i>	Pointer to an array containing the histogram.

Implements [IImageStatistics](#).

15.120.3.8 GetMean()

```
virtual void GetMean (
    StatisticsChannel channel,
    float * pPixelValueMean ) const [virtual]
```

Gets the mean of the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pPixelValueMean</i>	The mean of the image.

Implements [IImageStatistics](#).

15.120.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
    StatisticsChannel channel,
    unsigned int * pNumPixelValues ) const [virtual]
```

Gets the number of unique pixel values in the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pNumPixelValues</i>	The number of unique pixel values.

Implements [IImageStatistics](#).

15.120.3.10 GetPixelValueRange()

```
virtual void GetPixelValueRange (
    StatisticsChannel channel,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax ) const [virtual]
```

Gets the range of a statistics channel.

The values returned are the maximum values recorded for all pixels in the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pPixelValueMin</i>	The minimum pixel value.
<i>pPixelValueMax</i>	The maximum pixel value.

Implements [IImageStatistics](#).

15.120.3.11 GetRange()

```
virtual void GetRange (
    StatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax ) const [virtual]
```

Gets the range of a statistics channel.

The values returned are the maximum possible values for any given pixel in the image. This is generally 0-255 for 8 bit images, and 0-65535 for 16 bit images.

Parameters

<i>channel</i>	The statistics channel.
<i>pMin</i>	The minimum possible value.
<i>pMax</i>	The maximum possible value.

Implements [IImageStatistics](#).

15.120.3.12 GetStatistics()

```
virtual void GetStatistics (
    StatisticsChannel channel,
    unsigned int * pRangeMin = NULL,
    unsigned int * pRangeMax = NULL,
    unsigned int * pPixelValueMin = NULL,
```

```

unsigned int * pPixelValueMax = NULL,
unsigned int * pNumPixelValues = NULL,
float * pPixelValueMean = NULL,
int ** ppHistogram = NULL ) const [virtual]

```

Gets all statistics for the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pRangeMin</i>	The minimum possible value.
<i>pRangeMax</i>	The maximum possible value.
<i>pPixelValueMin</i>	The minimum pixel value.
<i>pPixelValueMax</i>	The maximum pixel value.
<i>pNumPixelValues</i>	The number of unique pixel values.
<i>pPixelValueMean</i>	The mean of the image.
<i>ppHistogram</i>	Pointer to an array containing the histogram.

Implements [IImageStatistics](#).

15.120.3.13 operator=()

```

ImageStatistics& operator= (
    const ImageStatistics & other )

```

Assignment operator.

Parameters

<i>other</i>	The ImageStatistics object to copy from.

15.120.3.14 SetChannelStatus()

```

virtual void SetChannelStatus (
    StatisticsChannel channel,
    bool enabled ) [virtual]

```

Sets the status of a statistics channel.

Parameters

<i>channel</i>	The statistics channel.
<i>enabled</i>	Whether the channel should be enabled.

See also

[GetChannelStatus\(\)](#)

Implements [IImageStatistics](#).

15.120.4 Friends And Related Function Documentation

15.120.4.1 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

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

- include/[ImageStatistics.h](#)

15.121 ImageUtility Class Reference

Static helper functions for the image object class.

Public Types

- enum [ImageScalingAlgorithm](#) { NEAREST_NEIGHBOR }
Image scaling algorithms.
- enum [SourceDataRange](#) {
IMAGE_DATA_RANGE,
ABSOLUTE_DATA_RANGE,
IMAGE_MIN_ABSOLUTE_MAX,
ABSOLUTE_MIN_IMAGE_MAX }
Image normalization source data options.

Static Public Member Functions

- static [ImagePtr](#) [CreateScaled](#) (const [ImagePtr](#) &srcImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static void [CreateScaled](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=IMAGE_DATA_RANGE)
Computes a normalized image.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const double min, const double max, [SourceDataRange](#) srcDataRange=IMAGE_DATA_RANGE)
Computes a normalized image.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const double min, const double max, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=IMAGE_DATA_RANGE)
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, [SourceDataRange](#) srcDataRange=IMAGE_DATA_RANGE)
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, const double min, const double max, [SourceDataRange](#) srcDataRange=IMAGE_DATA_RANGE)
Computes a normalized image.

15.121.1 Detailed Description

Static helper functions for the image object class.

15.121.2 Member Enumeration Documentation

15.121.2.1 ImageScalingAlgorithm

enum [ImageScalingAlgorithm](#)

[Image](#) scaling algorithms.

Enumerator

NEAREST_NEIGHBOR	

15.121.2.2 SourceDataRange

enum [SourceDataRange](#)

[Image](#) normalization source data options.

Options to normalize the source data based on the max and min values present in the specific image (image data) or the theoretical absolute max and min image data values for the image type (absolute data). By default the absolute max and min values for an image are the max and min values allowable for the image's pixel format. An exception to this is for some computed image data formats such as AoLP, DoLP and Stokes, where the absolute max and min are dependant on the algorithm used.

For a given pixel, normalization is done by: $\text{NormalizedValue} = ((\text{maxDest} - \text{minDest}) * (\text{PixelValue} - \text{minSrc}) / (\text{maxSrc} - \text{minSrc})) + \text{minDest}$

Enumerator

IMAGE_DATA_RANGE	Normalize based on the actual max and min values for the source image.
ABSOLUTE_DATA_RANGE	Normalize based on the theoretical max and min values for the source image.
IMAGE_MIN_ABSOLUTE_MAX	Normalize based on the actual min and theoretical max values for the source image.
ABSOLUTE_MIN_IMAGE_MAX	Normalize based on the theoretical min and actual max values for the source image.

15.121.3 Member Function Documentation

15.121.3.1 CreateNormalized() [1/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const PixelFormatEnums destPixelFormat,
    SourceDataRange srcDataRange = IMAGE_DATA_RANGE ) [static]
```

Computes a normalized image.

The full range of the destination pixel format data type will be used as the min and max range for normalization. The destination pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>destPixelFormat</i>	The desired pixel format for the normalized image
<i>srcDataRange</i>	The desired option for the source data range to normalize from

Returns

The normalized image

15.121.3.2 CreateNormalized() [2/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max,
    SourceDataRange srcDataRange = IMAGE_DATA_RANGE ) [static]
```

Computes a normalized image.

The min and max must be within range of the destination pixel format data type. The normalized image pixel format will be the same as the source image.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>min</i>	The lower bound of the normalization range
<i>max</i>	The upper bound of the normalization range
<i>srcDataRange</i>	The desired option for the source data range to normalize from

Returns

The normalized image

15.121.3.3 CreateNormalized() [3/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max,
    const PixelFormatEnums destPixelFormat,
    SourceDataRange srcDataRange = IMAGE_DATA_RANGE ) [static]
```

Computes a normalized image.

The min and max must be within range of the destination pixel format data type. The destination pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>min</i>	The lower bound of the normalization range
<i>max</i>	The upper bound of the normalization range
<i>destPixelFormat</i>	The desired pixel format for the normalized image
<i>srcDataRange</i>	The desired option for the source data range to normalize from

Returns

The normalized image

15.121.3.4 CreateNormalized() [4/5]

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    SourceDataRange srcDataRange = IMAGE_DATA_RANGE ) [static]
```

Computes a normalized image.

The full range of the destination pixel format data type will be used as the min and max range for normalization. The destination image must be initialized and have the same width and height as the source image. The destination image pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>destImage</i>	The destination image in which to store the normalized image
<i>srcDataRange</i>	The desired option for the source data range to normalize from

15.121.3.5 CreateNormalized() [5/5]

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    const double min,
    const double max,
    SourceDataRange srcDataRange = IMAGE_DATA_RANGE ) [static]
```

Computes a normalized image.

The min and max must be within range of the destination pixel format data type. The destination image must be initialized and have the same width and height as the source image. The destination image pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>destImage</i>	The destination image in which to store the normalized image
<i>min</i>	The lower bound of the normalization range
<i>max</i>	The upper bound of the normalization range
<i>srcDataRange</i>	The desired option for the source data range to normalize from

15.121.3.6 CreateScaled() [1/2]

```
static ImagePtr CreateScaled (
    const ImagePtr & srcImage,
    ImageScalingAlgorithm scalingAlg,
    double scalingFactor ) [static]
```

Computes a scaled image using the specified parameters.

Does not support scaling of raw bayer images.

Parameters

<i>srcImage</i>	The source image from which to create scaled image
<i>scalingAlg</i>	The desired image scaling algorithm to use
<i>scalingFactor</i>	The desired image scaling factor to use

Returns

The scaled image

15.121.3.7 CreateScaled() [2/2]

```
static void CreateScaled (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    ImageScalingAlgorithm scalingAlg,
    double scalingFactor ) [static]
```

Computes a scaled image using the specified parameters.

Does not support scaling of raw bayer images. The destination image height and width must be sufficient to store the calculated data. The destination image pixel format must be the same as the source image.

Parameters

<i>srcImage</i>	The source image from which to create scaled image
<i>destImage</i>	An image object in which to store the scaled data
<i>scalingAlg</i>	The desired image scaling algorithm to use
<i>scalingFactor</i>	The desired image scaling factor to use

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

- [include/ImageUtility.h](#)

15.122 ImageUtilityCCM Class Reference

Static function to create color corrected images from an image object.

Static Public Member Functions

- static [ImagePtr](#) [CreateColorCorrected](#) (const [ImagePtr](#) &srcImage, const [CCMSettings](#) &settings)
Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.
- static void [CreateColorCorrected](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, const [CCMSettings](#) &settings)
Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.

15.122.1 Detailed Description

Static function to create color corrected images from an image object.

15.122.2 Member Function Documentation

15.122.2.1 CreateColorCorrected() [1/2]

```
static ImagePtr CreateColorCorrected (
    const ImagePtr & srcImage,
    const CCMSettings & settings ) [static]
```

Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.

When using [ImageUtilityCCM](#), users are advised to disable CCM on the camera before capturing source images. This can be done through the camera node "ColorTransformationEnable".

Color correction is currently supported for the following pixel formats:

- PixelFormat_BGR8
- PixelFormat_BGRa8
- PixelFormat_RGBa8
- PixelFormat_RGB8
- PixelFormat_BGR16
- PixelFormat_BGRa16
- PixelFormat_RGBa16
- PixelFormat_RGB16 The output image will have the same pixel format as the source image.

Parameters

<i>srcImage</i>	The source image to which the CCM is applied
<i>settings</i>	Selected CCM settings including CCMColorTemperature, CCMTType, CCMSensor, etc

Returns

The color corrected image

See also

[CCMSettings](#)

15.122.2.2 CreateColorCorrected() [2/2]

```
static void CreateColorCorrected (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    const CCMSettings & settings ) [static]
```

Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.

When using [ImageUtilityCCM](#), users are advised to disable CCM on the camera before capturing source images. This can be done through the camera node "ColorTransformationEnable".

Color correction is currently supported for the following pixel formats:

- PixelFormat_BGR8
- PixelFormat_BGRa8
- PixelFormat_RGBa8
- PixelFormat_RGB8
- PixelFormat_BGR16
- PixelFormat_BGRa16
- PixelFormat_RGBa16
- PixelFormat_RGB16

The destination image height and width must be the same as the source image.

Parameters

<i>srcImage</i>	The source image to which the CCM is applied
<i>destImage</i>	The destination image in which to store the color corrected image
<i>settings</i>	Selected CCM settings including CCMColorTemperature, CCMTType, CCMSensor, etc

See also

[CCMSettings](#)

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

- include/[ImageUtilityCCM.h](#)

15.123 ImageUtilityHeatmap Class Reference

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

Public Types

- enum [HeatmapColor](#) {
[HEATMAP_BLACK](#) = 1,
[HEATMAP_BLUE](#) = 2,
[HEATMAP_CYAN](#) = 3,
[HEATMAP_GREEN](#) = 4,
[HEATMAP_YELLOW](#) = 5,
[HEATMAP_RED](#) = 6,
[HEATMAP_WHITE](#) = 7 }

Color specifiers for the heatmap color gradient.

Static Public Member Functions

- static [ImagePtr CreateHeatmap](#) (const [ImagePtr](#) &srcImage)
Computes a heatmap image.
- static void [CreateHeatmap](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage)
Computes a heatmap image.
- static void [SetHeatmapColorGradient](#) (const [HeatmapColor](#) newLowColor, const [HeatmapColor](#) newHighColor)
Sets the heatmap gradient color vector to the new desired range between HEATMAP_BLACK and HEATMAP_WHITE.
- static void [GetHeatmapColorGradient](#) ([HeatmapColor](#) &currentLowColor, [HeatmapColor](#) &currentHighColor)
Returns the current heatmap gradient color range.
- static void [SetHeatmapRange](#) (const unsigned int newLowValue, const unsigned int newHighValue)
Sets the high and low values used to determine which grayscale values are converted to a color 'heatmap' representation.
- static void [GetHeatmapRange](#) (unsigned int &currentLowValue, unsigned int &currentHighValue)
Returns the current high and low values used in heatmap representations.

15.123.1 Detailed Description

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

15.123.2 Member Enumeration Documentation

15.123.2.1 HeatmapColor

enum [HeatmapColor](#)

Color specifiers for the heatmap color gradient.

Enumerator

HEATMAP_BLACK	
HEATMAP_BLUE	
HEATMAP_CYAN	
HEATMAP_GREEN	
HEATMAP_YELLOW	
HEATMAP_RED	
HEATMAP_WHITE	

15.123.3 Member Function Documentation

15.123.3.1 CreateHeatmap() [1/2]

```
static ImagePtr CreateHeatmap (  
    const ImagePtr & srcImage ) [static]
```

Computes a heatmap image.

A heatmap image reinterprets monochrome data by mapping the luminosity of each pixel to a color value defined in the heatmap color gradient. The created image can be modified by changing the color gradient and heatmap range from the accompanying functions. The source image is required to be Mono8 or Mono16 pixel format.

Parameters

<i>srcImage</i>	The source image from which to create the heatmap

See also

[SetHeatmapRange\(\)](#)
[SetHeatmapColorGradient\(\)](#)

Returns

The heatmap image

15.123.3.2 CreateHeatmap() [2/2]

```
static void CreateHeatmap (  
    const ImagePtr & srcImage,  
    ImagePtr & destImage ) [static]
```

Computes a heatmap image.

A heatmap image reinterprets monochrome data by mapping the luminosity of each pixel to a color value defined in the heatmap color gradient. The created image can be modified by changing the color gradient and heatmap range from the accompanying functions. The source image is required to be Mono8 or Mono16 pixel format. The destination is required to be initialized, RGB8 or RGB16 pixel format, and have the same width, height, x offset, and y offset as the source image.

Parameters

<i>srcImage</i>	The source image from which to create the heatmap
<i>destImage</i>	The destination image in which to store the created heatmap

See also

[SetHeatmapRange\(\)](#)
[SetHeatmapColorGradient\(\)](#)

15.123.3.3 GetHeatmapColorGradient()

```
static void GetHeatmapColorGradient (
    HeatmapColor & currentLowColor,
    HeatmapColor & currentHighColor ) [static]
```

Returns the current heatmap gradient color range.

Parameters

<i>currentLowColor</i>	Current color at which the gradient begins.
<i>currentHighColor</i>	Current color at which the gradient ends.

See also

[SetHeatmapColorGradient\(\)](#)

15.123.3.4 GetHeatmapRange()

```
static void GetHeatmapRange (
    unsigned int & currentLowValue,
    unsigned int & currentHighValue ) [static]
```

Returns the current high and low values used in heatmap representations.

Parameters

<i>currentLowValue</i>	Current value at which color representation begins.
<i>currentHighValue</i>	Current value at which color representation ends.

See also

[SetHeatmapRange\(\)](#)

15.123.3.5 SetHeatmapColorGradient()

```
static void SetHeatmapColorGradient (
    const HeatmapColor newLowColor,
    const HeatmapColor newHighColor ) [static]
```

Sets the heatmap gradient color vector to the new desired range between HEATMAP_BLACK and HEATMAP_↔WHITE.

Parameters

<i>newLowColor</i>	New color at which to begin the gradient.
<i>newHighColor</i>	New color at which to end the gradient.

15.123.3.6 SetHeatmapRange()

```
static void SetHeatmapRange (
    const unsigned int newLowValue,
    const unsigned int newHighValue ) [static]
```

Sets the high and low values used to determine which grayscale values are converted to a color 'heatmap' representation.

Acceptable values range from 0 to 100.

Parameters

<i>newLowValue</i>	New value at which to begin color representation.
<i>newHighValue</i>	New value at which to end color representation.

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

- [include/ImageUtilityHeatmap.h](#)

15.124 ImageUtilityPolarization Class Reference

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

Public Types

- enum [PolarizationQuadrant](#) {
[QUADRANT_I0](#),
[QUADRANT_I45](#),
[QUADRANT_I90](#),
[QUADRANT_I135](#) }

Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

Static Public Member Functions

- static [ImagePtr](#) [ExtractPolarQuadrant](#) (const [ImagePtr](#) &srcImage, const [PolarizationQuadrant](#) desired↔
Quadrant)

Extracts all pixels of a specified degree of linear polarization into a new image object.

- static void [ExtractPolarQuadrant](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destQuadImage, const [PolarizationQuadrant](#) desiredQuadrant)
Extracts all pixels of a specified degree of linear polarization into the provided image object.
- static [ImagePtr](#) [CreateGlareReduced](#) (const [ImagePtr](#) &srcImage)
*Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant
The source image pixel format must be Polarized8 or BayerRGPolarized8.*
- static void [CreateGlareReduced](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destGlareReducedImage)
*Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant
The source image pixel format must be Polarized8 or BayerRGPolarized8.*
- static [ImagePtr](#) [CreateStokesS0](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the overall intensity of light from a polarized image.
- static void [CreateStokesS0](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS0Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the overall intensity of light from a polarized image.
- static [ImagePtr](#) [CreateStokesS1](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.
- static void [CreateStokesS1](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS1Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.
- static [ImagePtr](#) [CreateStokesS2](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the difference in intensity accepted through the polarizers at 45 and -45 to the horizontal.
- static void [CreateStokesS2](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS2Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the difference in intensity accepted through the polarizers.
- static [ImagePtr](#) [CreateDolp](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the fraction of incident light intensity in the linear polarization states.
- static void [CreateDolp](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destDolpImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the fraction of incident light intensity in the linear polarization states.
- static [ImagePtr](#) [CreateAolp](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.
- static void [CreateAolp](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destAolpImg, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

15.124.1 Detailed Description

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

15.124.2 Member Enumeration Documentation

15.124.2.1 PolarizationQuadrant

enum `PolarizationQuadrant`

Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

Enumerator

QUADRANT_I0	The 0 degree of polarization.
QUADRANT_I45	The 45 degree of polarization.
QUADRANT_I90	The 90 degree of polarization.
QUADRANT_I135	The 135 degree of polarization.

15.124.3 Member Function Documentation

15.124.3.1 `CreateAolp()` [1/2]

```
static ImagePtr CreateAolp (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono32f or RGB32f respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The angle of linear polarization (aolp) image

15.124.3.2 `CreateAolp()` [2/2]

```
static void CreateAolp (
    const ImagePtr & srcImage,
    ImagePtr & destAolpImg,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono32f or RGB32f respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destAolpImg</i>	The destination image in which to store the angle of linear polarization (aolp) image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.124.3.3 CreateDolp() [1/2]

```
static ImagePtr CreateDolp (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the fraction of incident light intensity in the linear polarization states.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono32f or RGB32f respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The degree of linear polarization (dolp) image

15.124.3.4 CreateDolp() [2/2]

```
static void CreateDolp (
    const ImagePtr & srcImage,
    ImagePtr & destDolpImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the fraction of incident light intensity in the linear polarization states.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono32f or RGB32f respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destDolpImage</i>	The destination image in which to store the degree of linear polarization (dolp) image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.124.3.5 CreateGlareReduced() [1/2]

```
static ImagePtr CreateGlareReduced (
    const ImagePtr & srcImage ) [static]
```

Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant. The source image pixel format must be Polarized8 or BayerRGPolarized8.

The destination image pixel format will be Mono8 or BayerRG8 respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data

Returns

The reduced glare image

15.124.3.6 CreateGlareReduced() [2/2]

```
static void CreateGlareReduced (
    const ImagePtr & srcImage,
    ImagePtr & destGlareReducedImage ) [static]
```

Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant. The source image pixel format must be Polarized8 or BayerRGPolarized8.

The destination image pixel format will be Mono8 or BayerRG8 respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to apply glare reduction
<i>destGlareReducedImage</i>	The destination image in which to store the image with reduced glare

15.124.3.7 CreateStokesS0() [1/2]

```
static ImagePtr CreateStokesS0 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the overall intensity of light from a polarized image.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono16s or RGB16s respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The Stokes' S0 image

15.124.3.8 CreateStokesS0() [2/2]

```
static void CreateStokesS0 (
    const ImagePtr & srcImage,
    ImagePtr & destStokesS0Image,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the overall intensity of light from a polarized image.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono16s or RGB16s respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destStokesS0Image</i>	The destination image in which to store the Stokes' S0 image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.124.3.9 CreateStokesS1() [1/2]

```
static ImagePtr CreateStokesS1 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono16s or RGB16s respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The Stokes' S1 image

15.124.3.10 CreateStokesS1() [2/2]

```
static void CreateStokesS1 (  
    const ImagePtr & srcImage,  
    ImagePtr & destStokesS1Image,  
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono16s or RGB16s respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destStokesS1Image</i>	The destination image in which to store the Stokes' S1 image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.124.3.11 CreateStokesS2() [1/2]

```
static ImagePtr CreateStokesS2 (  
    const ImagePtr & srcImage,  
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the difference in intensity accepted through the polarizers at 45 and -45 to the horizontal.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono16s or RGB16s respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The Stokes' S2 image

15.124.3.12 CreateStokesS2() [2/2]

```
static void CreateStokesS2 (
    const ImagePtr & srcImage,
    ImagePtr & destStokesS2Image,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [static]
```

Computes an image representing the difference in intensity accepted through the polarizers.

at 45 and -45 to the horizontal. The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono16s or RGB16s respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destStokesS2Image</i>	The destination image in which to store the Stokes' S2 image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.124.3.13 ExtractPolarQuadrant() [1/2]

```
static ImagePtr ExtractPolarQuadrant (
    const ImagePtr & srcImage,
    const PolarizationQuadrant desiredQuadrant ) [static]
```

Extracts all pixels of a specified degree of linear polarization into a new image object.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono8 or BayerRG8 respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>desiredQuadrant</i>	The polarization quadrant to extract

Returns

The specified polarization quadrant image

15.124.3.14 ExtractPolarQuadrant() [2/2]

```
static void ExtractPolarQuadrant (
    const ImagePtr & srcImage,
    ImagePtr & destQuadImage,
    const PolarizationQuadrant desiredQuadrant ) [static]
```

Extracts all pixels of a specified degree of linear polarization into the provided image object.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono8 or BayerRG8 respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destQuadImage</i>	The destination image in which to store the extracted polarization quadrant
<i>desiredQuadrant</i>	The polarization quadrant to extract

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

- [include/ImageUtilityPolarization.h](#)

15.125 InferenceBoundingBox Struct Reference

Inference Bounding Boxes data structure.

Public Attributes

- InferenceBoxType [boxType](#)
- int16_t [classId](#)
- float32_t [confidence](#)
- InferenceBoxRect [rect](#)
- InferenceBoxCircle [circle](#)
- InferenceBoxRotatedRect [rotatedRect](#)

15.125.1 Detailed Description

Inference Bounding Boxes data structure.

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

- [include/ChunkDataInference.h](#)

15.126 InferenceBoundingBoxResult Class Reference

An inference bounding boxes object which holds information about the detected bounding boxes.

Public Member Functions

- [InferenceBoundingBoxResult](#) ()
Default Constructor.
- [~InferenceBoundingBoxResult](#) ()
Destructor.
- [InferenceBoundingBoxResult](#) (const uint8_t *data, const int64_t lengthInBytes)
Default Constructor with arguments.
- [InferenceBoundingBoxResult](#) (const [InferenceBoundingBoxResult](#) &other)
Copy Constructor.
- [InferenceBoundingBoxResult](#) & [operator=](#) (const [InferenceBoundingBoxResult](#) &rhs)
Assignment Operator.
- int8_t [GetVersion](#) () const
Returns the bounding box format version number.
- int16_t [GetBoxCount](#) () const
Returns the number of bounding boxes.
- int8_t [GetBoxSize](#) () const
Returns the number of bytes allocated for one bounding box.
- [InferenceBoundingBox](#) [GetBoxAt](#) (const uint16_t index) const
Returns the bounding box at specified index.

15.126.1 Detailed Description

An inference bounding boxes object which holds information about the detected bounding boxes.

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

- include/[ChunkDataInference.h](#)

15.127 InferenceBoxCircle Struct Reference

Public Attributes

- int16_t [centerXCoord](#)
- int16_t [centerYCoord](#)
- int16_t [radius](#)

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

- include/[ChunkDataInference.h](#)

15.128 InferenceBoxRect Struct Reference

Inference Bounding Box Type Data Structures.

Public Attributes

- `int16_t` [topLeftXCoord](#)
- `int16_t` [topLeftYCoord](#)
- `int16_t` [bottomRightXCoord](#)
- `int16_t` [bottomRightYCoord](#)

15.128.1 Detailed Description

Inference Bounding Box Type Data Structures.

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

- `include/ChunkDataInference.h`

15.129 InferenceBoxRotatedRect Struct Reference

Public Attributes

- `int16_t` [topLeftXCoord](#)
- `int16_t` [topLeftYCoord](#)
- `int16_t` [bottomRightXCoord](#)
- `int16_t` [bottomRightYCoord](#)
- `short` [rotationAngle](#)

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

- `include/ChunkDataInference.h`

15.130 `int64_autovector_t` Class Reference

Vector of integers with reference counting.

Public Member Functions

- [int64_autovector_t](#) ()
- [int64_autovector_t](#) (const [int64_autovector_t](#) &obj)
- [int64_autovector_t](#) (size_t n)
- virtual [~int64_autovector_t](#) (void)
- [int64_autovector_t](#) & [operator=](#) (const [int64_autovector_t](#) &obj)
- void [operator delete](#) (void *pWhere)
- void * [operator new](#) (size_t uiSize)
- [int64_t](#) & [operator\[\]](#) (size_t uiIndex)
- const [int64_t](#) & [operator\[\]](#) (size_t uiIndex) const
- size_t [size](#) () const

Protected Attributes

- `std::vector< int64_t > * _pv`
- `ATOMIC_VARIABLE * _pCount`

15.130.1 Detailed Description

Vector of integers with reference counting.

15.130.2 Constructor & Destructor Documentation

15.130.2.1 `int64_autovector_t()` [1/3]

```
int64_autovector_t ( )
```

15.130.2.2 `int64_autovector_t()` [2/3]

```
int64_autovector_t (
    const int64_autovector_t & obj )
```

15.130.2.3 `int64_autovector_t()` [3/3]

```
int64_autovector_t (
    size_t n ) [explicit]
```

15.130.2.4 `~int64_autovector_t()`

```
virtual ~int64_autovector_t (
    void ) [virtual]
```

15.130.3 Member Function Documentation

15.130.3.1 operator delete()

```
void operator delete (
    void * pWhere )
```

15.130.3.2 operator new()

```
void* operator new (
    size_t uiSize )
```

15.130.3.3 operator=()

```
int64_autovector_t& operator= (
    const int64_autovector_t & obj )
```

15.130.3.4 operator[]() [1/2]

```
int64_t& operator[] (
    size_t uiIndex )
```

15.130.3.5 operator[]() [2/2]

```
const int64_t& operator[] (
    size_t uiIndex ) const
```

15.130.3.6 size()

```
size_t size ( ) const
```

15.130.4 Member Data Documentation

15.130.4.1 _pCount

```
ATOMIC_VARIABLE* _pCount [protected]
```

15.130.4.2 _pv

```
std::vector<int64_t>* _pv [protected]
```

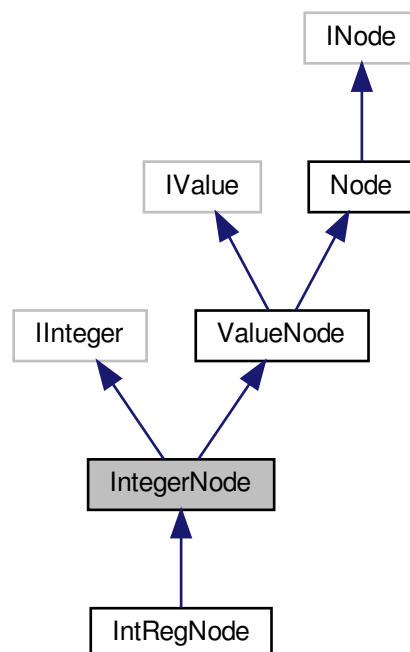
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Autovector.h](#)

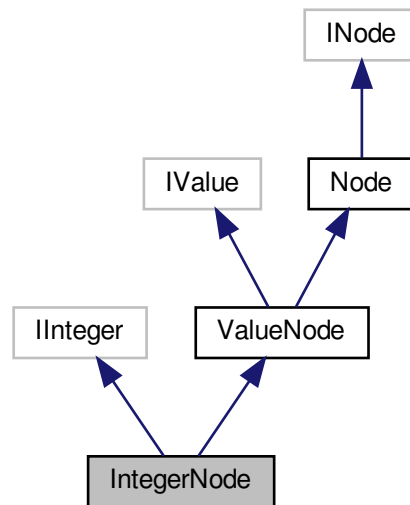
15.131 IntegerNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for IntegerNode:



Collaboration diagram for IntegerNode:



Public Member Functions

- [IntegerNode](#) ()
- [IntegerNode](#) (std::shared_ptr< Node::NodeImpl > pInteger)
- virtual [~IntegerNode](#) ()
- virtual void [SetValue](#) (int64_t Value, bool [Verify](#)=true)
 - Set node value.*
- virtual [IInteger](#) & [operator=](#) (int64_t Value)
 - Set node value.*
- virtual int64_t [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
 - Get node value.*
- virtual int64_t [operator\(\)](#) ()
 - Get node value.*
- virtual int64_t [operator*](#) ()
 - Get node value.*
- virtual int64_t [GetMin](#) ()
 - Get minimum value allowed.*
- virtual int64_t [GetMax](#) ()
 - Get maximum value allowed.*
- virtual [EIncMode](#) [GetIncMode](#) ()
 - Get increment mode.*
- virtual int64_t [GetInc](#) ()
 - Get increment.*
- virtual [int64_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)
 - Get list of valid value.*
- virtual [ERepresentation](#) [GetRepresentation](#) ()
 - Get recommended representation.*

- virtual [GenICam::gcstring](#) [GetUnit](#) ()
Get the physical unit name.
- virtual [IFloat](#) * [GetFloatAlias](#) ()
gets the interface of an alias node.
- virtual void [ImposeMin](#) (int64_t Value)
Restrict minimum value.
- virtual void [ImposeMax](#) (int64_t Value)
Restrict maximum value.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Integer

Additional Inherited Members

15.131.1 Detailed Description

[Interface](#) for string properties.

15.131.2 Constructor & Destructor Documentation

15.131.2.1 IntegerNode() [1/2]

[IntegerNode](#) ()

15.131.2.2 IntegerNode() [2/2]

[IntegerNode](#) (
 std::shared_ptr< [Node::NodeImpl](#) > pInteger)

15.131.2.3 ~IntegerNode()

virtual ~[IntegerNode](#) () [virtual]

15.131.3 Member Function Documentation

15.131.3.1 GetFloatAlias()

```
virtual IFloat* GetFloatAlias ( ) [virtual]
```

gets the interface of an alias node.

15.131.3.2 GetInc()

```
virtual int64_t GetInc ( ) [virtual]
```

Get increment.

15.131.3.3 GetIncMode()

```
virtual EIncMode GetIncMode ( ) [virtual]
```

Get increment mode.

15.131.3.4 GetListOfValidValues()

```
virtual int64_autovector_t GetListOfValidValues (
    bool bounded = true ) [virtual]
```

Get list of valid value.

15.131.3.5 GetMax()

```
virtual int64_t GetMax ( ) [virtual]
```

Get maximum value allowed.

15.131.3.6 GetMin()

```
virtual int64_t GetMin ( ) [virtual]
```

Get minimum value allowed.

15.131.3.7 GetRepresentation()

```
virtual ERepresentation GetRepresentation ( ) [virtual]
```

Get recommended representation.

15.131.3.8 GetUnit()

```
virtual GenICam::gcstring GetUnit ( ) [virtual]
```

Get the physical unit name.

15.131.3.9 GetValue()

```
virtual int64_t GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.131.3.10 ImposeMax()

```
virtual void ImposeMax (
    int64_t Value ) [virtual]
```

Restrict maximum value.

15.131.3.11 ImposeMin()

```
virtual void ImposeMin (
    int64_t Value ) [virtual]
```

Restrict minimum value.

15.131.3.12 operator()()

```
virtual int64_t operator() ( ) [virtual]
```

Get node value.

15.131.3.13 operator*()

```
virtual int64_t operator* ( ) [virtual]
```

Get node value.

15.131.3.14 operator=()

```
virtual Integer& operator= (
    int64_t Value ) [virtual]
```

Set node value.

15.131.3.15 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Integer

Reimplemented from [ValueNode](#).

Reimplemented in [IntRegNode](#).

15.131.3.16 SetValue()

```
virtual void SetValue (
    int64_t Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

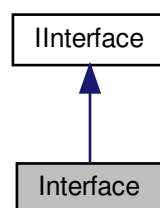
The documentation for this class was generated from the following file:

- `include/SpinGenApi/IntegerNode.h`

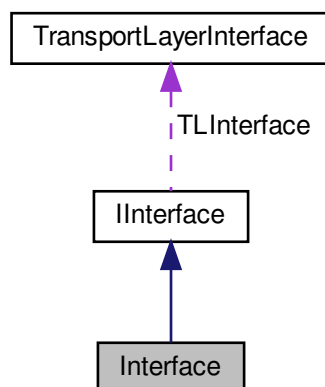
15.132 Interface Class Reference

An interface object which holds a list of cameras.

Inheritance diagram for Interface:



Collaboration diagram for Interface:



Public Member Functions

- virtual `~Interface` (void)
Virtual Destructor.
- `CameraList GetCameras` (bool updateCameras=true) const

- Returns a list of cameras available on this interface.*

 - bool [UpdateCameras](#) ()

Updates the list of cameras on this interface.
- [GenApi::INodeMap](#) & [GetTLNodeMap](#) () const

Gets a nodeMap that is generated from a [GenICam](#) XML file for the GenTL interface Module.
- void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)

Registers an event handler for the interface Event handlers are automatically cleaned up when an interface is removed, and must be registered to interfaces as they arrive.
- void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)

Unregisters an event handler for the interface.
- bool [IsInUse](#) () const

Checks if the interface is in use by any camera objects.
- void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL) const

Broadcast an Action Command to all devices on interface.
- bool [IsValid](#) ()

IsValid Checks a flag to determine if interface is still valid for use.

Friends

- class [InterfaceInternal](#)

Additional Inherited Members

15.132.1 Detailed Description

An interface object which holds a list of cameras.

15.132.2 Constructor & Destructor Documentation

15.132.2.1 ~Interface()

```
virtual ~Interface (
    void ) [virtual]
```

Virtual Destructor.

15.132.3 Member Function Documentation

15.132.3.1 GetCameras()

```
CameraList GetCameras (
    bool updateCameras = true ) const [virtual]
```

Returns a list of cameras available on this interface.

This call returns either usb3 vision or gige vision cameras depending on the underlying transport layer of this interface. The camera list object will reference count the cameras that it holds. It is important that the [CameraList](#) is destroyed or is cleared before [System::ReleaseInstance\(\)](#) can be called or an [InterfaceList](#) that holds this interface can be cleared.

See also

[System::ReleaseInstance\(\)](#)
[InterfaceList::Clear\(\)](#)
[CameraList::Clear\(\)](#)

Parameters

<i>updateCameras</i>	A flag used to issue an updateCameras() call internally before getting the camera list

Returns

An [CameraList](#) object that contains a list of cameras on this interface.

Implements [IInterface](#).

15.132.3.2 GetTLNodeMap()

```
GenApi::INodeMap& GetTLNodeMap ( ) const [virtual]
```

Gets a nodeMap that is generated from a [GenICam](#) XML file for the GenTL interface Module.

Returns

A reference to a [INodeMap](#) object.

Implements [IInterface](#).

15.132.3.3 IsInUse()

```
bool IsInUse ( ) const [virtual]
```

Checks if the interface is in use by any camera objects.

Returns

Returns true if the interface is in use and false otherwise.

Implements [IInterface](#).

15.132.3.4 IsValid()

```
bool IsValid ( ) [virtual]
```

IsValid Checks a flag to determine if interface is still valid for use.

Returns

If interface is valid or not

Implements [IInterface](#).

15.132.3.5 RegisterEventHandler()

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [virtual]
```

Registers an event handler for the interface Event handlers are automatically cleaned up when an interface is removed, and must be registered to interfaces as they arrive.

Note that GEV interfaces experience arrival/removal events when the IP information changes, similar to GEV cameras. Please refer to the EnumerationEvents example for recommended use.

See also

[InterfaceEventHandler](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the interface

Implements [IInterface](#).

15.132.3.6 SendActionCommand()

```
void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) const [virtual]
```

Broadcast an Action Command to all devices on interface.

Parameters

<i>deviceKey</i>	The Action Command's device key
<i>groupKey</i>	The Action Command's group key
<i>groupMask</i>	The Action Command's group mask
<i>actionTime</i>	(Optional) Time when to assert a future action. Zero means immediate action.
<i>pResultSize</i>	(Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results. If this parameter is 0 or NULL, the function will return as soon as the command has been broadcasted.
<i>results</i>	(Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

Implements [IInterface](#).

15.132.3.7 UnregisterEventHandler()

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters an event handler for the interface.

See also

[InterfaceEventHandler](#)

Parameters

<i>evtHandlerToUnregister</i>	The event handler to unregister from the interface

Implements [IInterface](#).

15.132.3.8 UpdateCameras()

```
bool UpdateCameras ( ) [virtual]
```

Updates the list of cameras on this interface.

This function needs to be called before any cameras can be discovered using [GetCameras\(\)](#). [System::GetCameras\(\)](#) will automatically call this function for each interface it enumerates. If the list changed after the last time [System::GetCameras\(\)](#) or [UpdateCameras\(\)](#) was called then the return value will be true, otherwise it is false.

See also

[System::GetCameras\(\)](#)
[GetCameras\(\)](#)

Returns

true if cameras changed on interface and false otherwise.

Implements [Interface](#).

15.132.4 Friends And Related Function Documentation

15.132.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

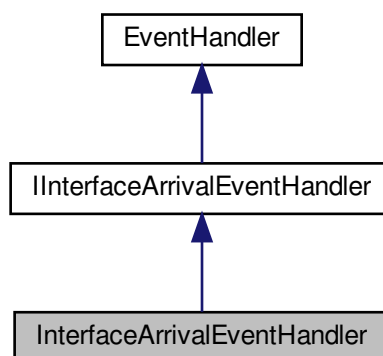
The documentation for this class was generated from the following file:

- include/[Interface.h](#)

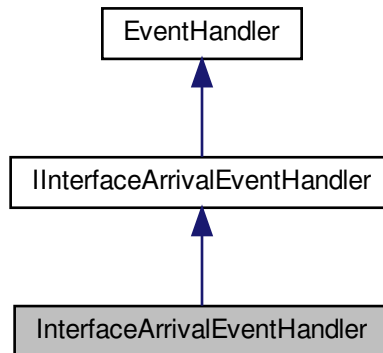
15.133 InterfaceArrivalEventHandler Class Reference

An event handler for capturing the interface arrival event.

Inheritance diagram for InterfaceArrivalEventHandler:



Collaboration diagram for InterfaceArrivalEventHandler:



Public Member Functions

- [InterfaceArrivalEventHandler](#) ()
Default constructor.
- virtual [~InterfaceArrivalEventHandler](#) ()
Virtual destructor.
- virtual void [OnInterfaceArrival](#) (std::string interfaceID)=0
[Interface](#) arrival event callback.

Protected Member Functions

- [InterfaceArrivalEventHandler](#) & [operator=](#) (const [InterfaceArrivalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.133.1 Detailed Description

An event handler for capturing the interface arrival event.

Note that only GEV interface arrivals are currently handled.

15.133.2 Constructor & Destructor Documentation

15.133.2.1 InterfaceArrivalEventHandler()

```
InterfaceArrivalEventHandler ( )
```

Default constructor.

15.133.2.2 ~InterfaceArrivalEventHandler()

```
virtual ~InterfaceArrivalEventHandler ( ) [virtual]
```

Virtual destructor.

15.133.3 Member Function Documentation

15.133.3.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    std::string interfaceID ) [pure virtual]
```

[Interface](#) arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

<i>interfaceID</i>	The ID of the interface that arrived

Implements [IInterfaceArrivalEventHandler](#).

15.133.3.2 operator=()

```
InterfaceArrivalEventHandler& operator= (
    const InterfaceArrivalEventHandler & ) [protected]
```

Assignment operator.

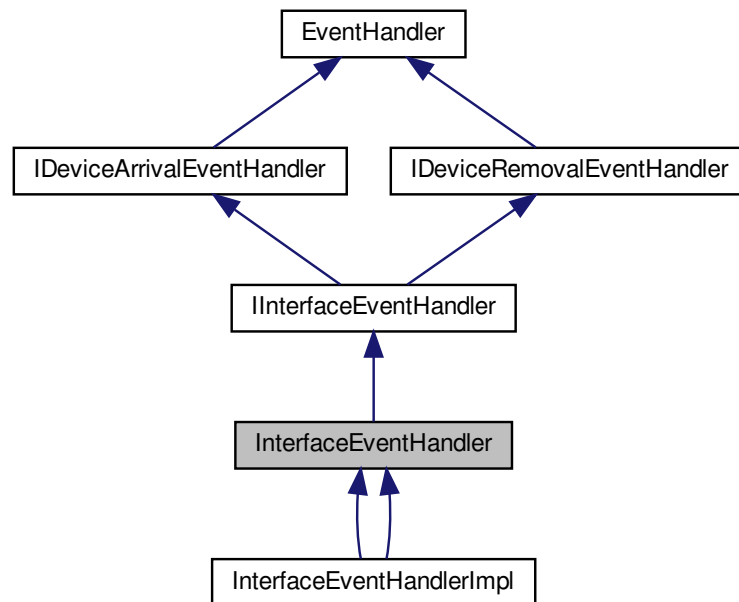
The documentation for this class was generated from the following file:

- include/[InterfaceArrivalEventHandler.h](#)

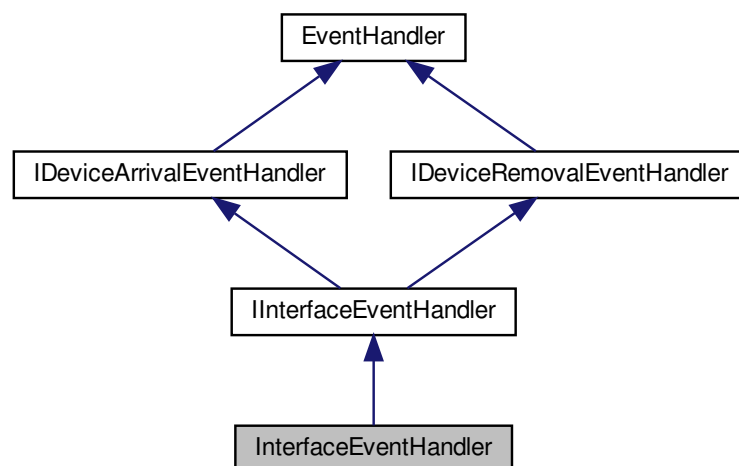
15.134 InterfaceEventHandler Class Reference

A handler to device arrival and removal events on all interfaces.

Inheritance diagram for InterfaceEventHandler:



Collaboration diagram for InterfaceEventHandler:



Public Member Functions

- [InterfaceEventHandler](#) ()
Default constructor.
- virtual [~InterfaceEventHandler](#) ()
Virtual destructor.
- virtual void [OnDeviceArrival](#) (uint64_t serialNumber)=0
Device arrival event callback.
- virtual void [OnDeviceRemoval](#) (uint64_t serialNumber)=0
Callback to the device removal event.

Protected Member Functions

- [InterfaceEventHandler](#) & [operator=](#) (const [InterfaceEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.134.1 Detailed Description

A handler to device arrival and removal events on all interfaces.

15.134.2 Constructor & Destructor Documentation

15.134.2.1 [InterfaceEventHandler\(\)](#)

```
InterfaceEventHandler ( )
```

Default constructor.

15.134.2.2 [~InterfaceEventHandler\(\)](#)

```
virtual ~InterfaceEventHandler ( ) [virtual]
```

Virtual destructor.

15.134.3 Member Function Documentation

15.134.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Device arrival event callback.

Implements [InterfaceEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.134.3.2 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device removal event.

Parameters

<i>serialNumber</i>	The serial number of the removed device

Implements [InterfaceEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.134.3.3 operator=()

```
InterfaceEventHandler& operator= (
    const InterfaceEventHandler & ) [protected]
```

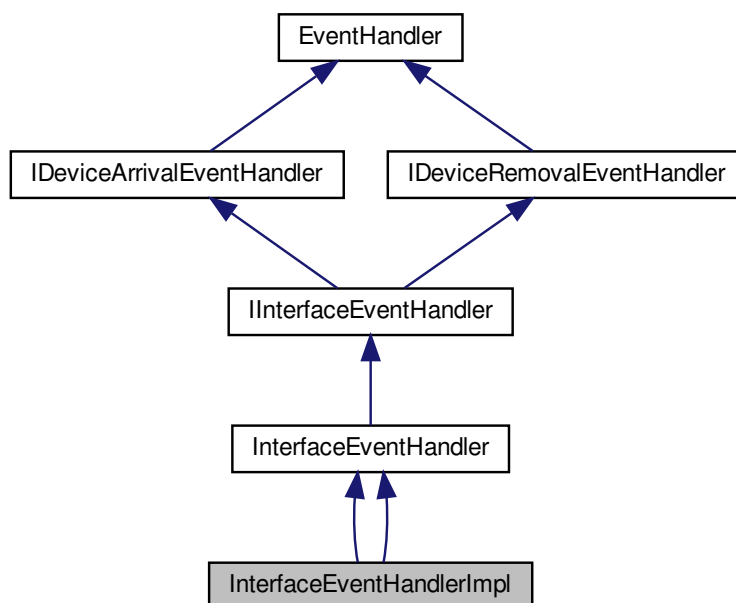
Assignment operator.

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

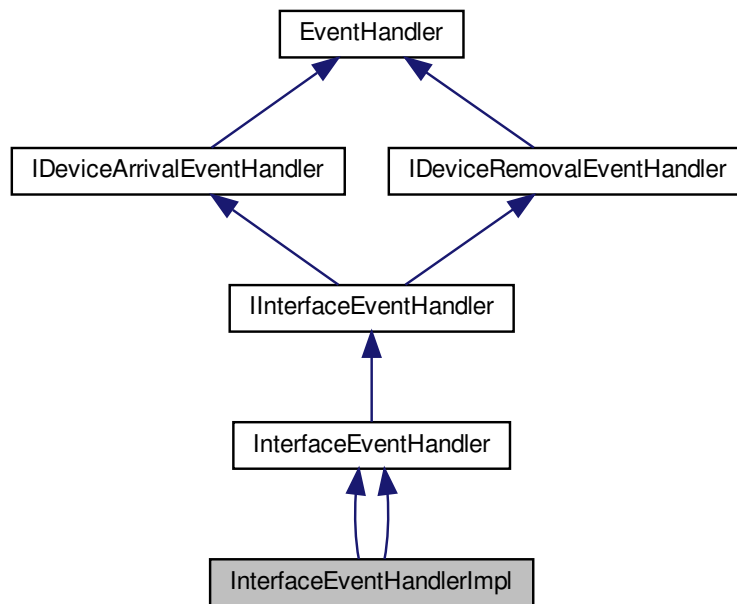
- [include/InterfaceEventHandler.h](#)

15.135 InterfaceEventHandlerImpl Class Reference

Inheritance diagram for InterfaceEventHandlerImpl:



Collaboration diagram for InterfaceEventHandlerImpl:



Public Member Functions

- [InterfaceEventHandlerImpl](#) ([SystemPtr](#) system)
- [~InterfaceEventHandlerImpl](#) ()
- void [OnDeviceArrival](#) (uint64_t deviceSerialNumber)
Device arrival event callback.
- void [OnDeviceRemoval](#) (uint64_t deviceSerialNumber)
Callback to the device removal event.
- [InterfaceEventHandlerImpl](#) ([SystemPtr](#) system)
- [InterfaceEventHandlerImpl](#) ([InterfacePtr](#) iface, std::string interfaceID)
- [~InterfaceEventHandlerImpl](#) ()
- void [PrintGenericHandlerMessage](#) (const unsigned long long deviceCount)
- void [OnDeviceArrival](#) (uint64_t deviceSerialNumber)
Device arrival event callback.
- void [OnDeviceRemoval](#) (uint64_t deviceSerialNumber)
Callback to the device removal event.
- std::string [GetInterfaceId](#) ()

Additional Inherited Members

15.135.1 Constructor & Destructor Documentation

15.135.1.1 InterfaceEventHandlerImpl() [1/3]

```
InterfaceEventHandlerImpl (
    SystemPtr system ) [inline]
```

15.135.1.2 ~InterfaceEventHandlerImpl() [1/2]

```
~InterfaceEventHandlerImpl ( ) [inline]
```

15.135.1.3 InterfaceEventHandlerImpl() [2/3]

```
InterfaceEventHandlerImpl (
    SystemPtr system ) [inline]
```

15.135.1.4 InterfaceEventHandlerImpl() [3/3]

```
InterfaceEventHandlerImpl (
    InterfacePtr iface,
    std::string interfaceID ) [inline]
```

15.135.1.5 ~InterfaceEventHandlerImpl() [2/2]

```
~InterfaceEventHandlerImpl ( ) [inline]
```

15.135.2 Member Function Documentation**15.135.2.1 GetInterfaceId()**

```
std::string GetInterfaceId ( ) [inline]
```

15.135.2.2 OnDeviceArrival() [1/2]

```
void OnDeviceArrival (
    uint64_t serialNumber ) [inline], [virtual]
```

Device arrival event callback.

Implements [InterfaceEventHandler](#).

15.135.2.3 OnDeviceArrival() [2/2]

```
void OnDeviceArrival (
    uint64_t serialNumber ) [inline], [virtual]
```

Device arrival event callback.

Implements [InterfaceEventHandler](#).

15.135.2.4 OnDeviceRemoval() [1/2]

```
void OnDeviceRemoval (
    uint64_t serialNumber ) [inline], [virtual]
```

Callback to the device removal event.

Parameters

<i>serialNumber</i>	The serial number of the removed device

Implements [InterfaceEventHandler](#).

15.135.2.5 OnDeviceRemoval() [2/2]

```
void OnDeviceRemoval (
    uint64_t serialNumber ) [inline], [virtual]
```

Callback to the device removal event.

Parameters

<i>serialNumber</i>	The serial number of the removed device

Implements [InterfaceEventHandler](#).

15.135.2.6 PrintGenericHandlerMessage()

```
void PrintGenericHandlerMessage (
    const unsigned long long deviceCount ) [inline]
```

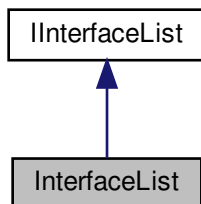
The documentation for this class was generated from the following files:

- [src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp](#)
- [src/EnumerationEvents/EnumerationEvents.cpp](#)

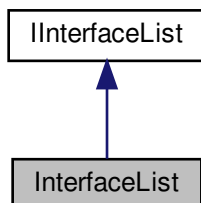
15.136 InterfaceList Class Reference

A list of the available interfaces on the system.

Inheritance diagram for InterfaceList:



Collaboration diagram for InterfaceList:



Public Member Functions

- [InterfaceList](#) (void)
- virtual [~InterfaceList](#) (void)
- [InterfaceList](#) (const [InterfaceList](#) &iface)
- [InterfaceList](#) & [operator=](#) (const [InterfaceList](#) &iface)
Assignment operator.
- [InterfacePtr](#) [operator\[\]](#) (unsigned int index)
Array subscription operators.
- unsigned int [GetSize](#) () const
Returns the size of the interface list.
- [InterfacePtr](#) [GetByIndex](#) (unsigned int index) const
Returns a pointer to an [Interface](#) object at the "index".
- void [Clear](#) ()
Clears the list of interfaces and destroys their corresponding objects.
- void [Append](#) (const [InterfaceList](#) *list)
Appends a copy of the interfact list.

Friends

- class [SystemImpl](#)
- class [ProducerImpl](#)

Additional Inherited Members

15.136.1 Detailed Description

A list of the available interfaces on the system.

15.136.2 Constructor & Destructor Documentation

15.136.2.1 [InterfaceList\(\)](#) [1/2]

```
InterfaceList (
    void )
```

15.136.2.2 [~InterfaceList\(\)](#)

```
virtual ~InterfaceList (
    void ) [virtual]
```

15.136.2.3 InterfaceList() [2/2]

```
InterfaceList (
    const InterfaceList & iface )
```

15.136.3 Member Function Documentation

15.136.3.1 Append()

```
void Append (
    const InterfaceList * list ) [virtual]
```

Appends a copy of the interface list.

Parameters

<i>list</i>	Another InterfaceList object, whose elements are added to this list.

Implements [InterfaceList](#).

15.136.3.2 Clear()

```
void Clear ( ) [virtual]
```

Clears the list of interfaces and destroys their corresponding objects.

It is important to first make sure there are no referenced cameras still in use before calling [Clear\(\)](#). If a camera on any of the interfaces is still in use this function will throw an exception.

Implements [InterfaceList](#).

15.136.3.3 GetByIndex()

```
InterfacePtr GetByIndex (
    unsigned int index ) const [virtual]
```

Returns a pointer to an [Interface](#) object at the "index".

Parameters

<i>index</i>	The index at which to retrieve the Interface object

Returns

A pointer to an [Interface](#) object.

Implements [IInterfaceList](#).

15.136.3.4 GetSize()

```
unsigned int GetSize ( ) const [virtual]
```

Returns the size of the interface list.

The size is the number of [Interface](#) objects stored in the list.

Returns

An integer that represents the list size.

Implements [IInterfaceList](#).

15.136.3.5 operator=()

```
InterfaceList& operator= (
    const InterfaceList & iface )
```

Assignment operator.

15.136.3.6 operator[]()

```
InterfacePtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [IInterfaceList](#).

15.136.4 Friends And Related Function Documentation

15.136.4.1 ProducerImpl

```
friend class ProducerImpl [friend]
```

15.136.4.2 SystemImpl

```
friend class SystemImpl [friend]
```

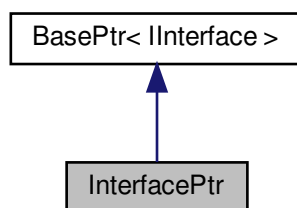
The documentation for this class was generated from the following file:

- include/[InterfaceList.h](#)

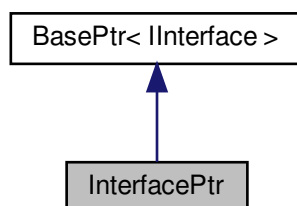
15.137 InterfacePtr Class Reference

A reference tracked pointer to the interface object.

Inheritance diagram for InterfacePtr:



Collaboration diagram for InterfacePtr:



Public Member Functions

- [InterfacePtr](#) () throw ()
Default Constructor.
- [InterfacePtr](#) (const int) throw ()
Default Constructor with argument.
- [InterfacePtr](#) (const long) throw ()
- [InterfacePtr](#) (const std::nullptr_t) throw ()

Additional Inherited Members

15.137.1 Detailed Description

A reference tracked pointer to the interface object.

15.137.2 Constructor & Destructor Documentation

15.137.2.1 [InterfacePtr\(\)](#) [1/4]

```
InterfacePtr ( ) throw ( ) [inline]
```

Default Constructor.

15.137.2.2 [InterfacePtr\(\)](#) [2/4]

```
InterfacePtr (  
    const int ) throw ( ) [inline]
```

Default Constructor with argument.

15.137.2.3 [InterfacePtr\(\)](#) [3/4]

```
InterfacePtr (  
    const long ) throw ( ) [inline]
```

15.137.2.4 InterfacePtr() [4 / 4]

```
InterfacePtr (
    const std::nullptr_t ) throw )    [inline]
```

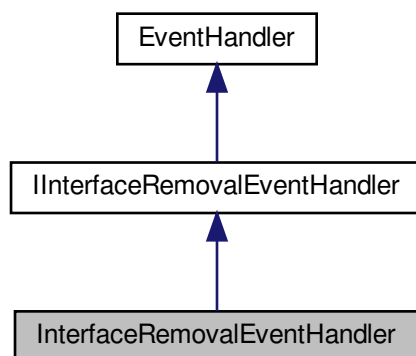
The documentation for this class was generated from the following file:

- include/[InterfacePtr.h](#)

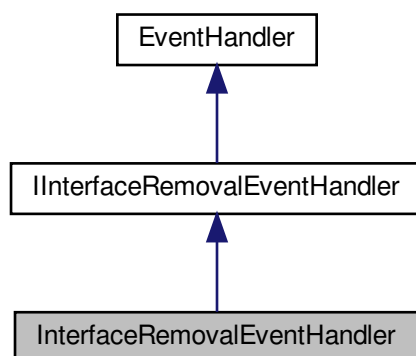
15.138 InterfaceRemovalEventHandler Class Reference

An event handler for capturing the interface removal event.

Inheritance diagram for InterfaceRemovalEventHandler:



Collaboration diagram for InterfaceRemovalEventHandler:



Public Member Functions

- [InterfaceRemovalEventHandler](#) ()
Default Constructor.
- virtual [~InterfaceRemovalEventHandler](#) ()
Virtual Destructor.
- virtual void [OnInterfaceRemoval](#) (std::string interfaceID)=0
Interface removal event callback.

Protected Member Functions

- [InterfaceRemovalEventHandler](#) & [operator=](#) (const [InterfaceRemovalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.138.1 Detailed Description

An event handler for capturing the interface removal event.

Note that only GEV interface removals are currently handled.

15.138.2 Constructor & Destructor Documentation

15.138.2.1 InterfaceRemovalEventHandler()

```
InterfaceRemovalEventHandler ( )
```

Default Constructor.

15.138.2.2 ~InterfaceRemovalEventHandler()

```
virtual ~InterfaceRemovalEventHandler ( ) [virtual]
```

Virtual Destructor.

15.138.3 Member Function Documentation

15.138.3.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (  
    std::string interfaceID ) [pure virtual]
```

[Interface](#) removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

<i>interfaceID</i>	The ID of the interface removed

Implements [InterfaceRemovalEventHandler](#).

15.138.3.2 operator=()

```
InterfaceRemovalEventHandler& operator= (
    const InterfaceRemovalEventHandler & ) [protected]
```

Assignment operator.

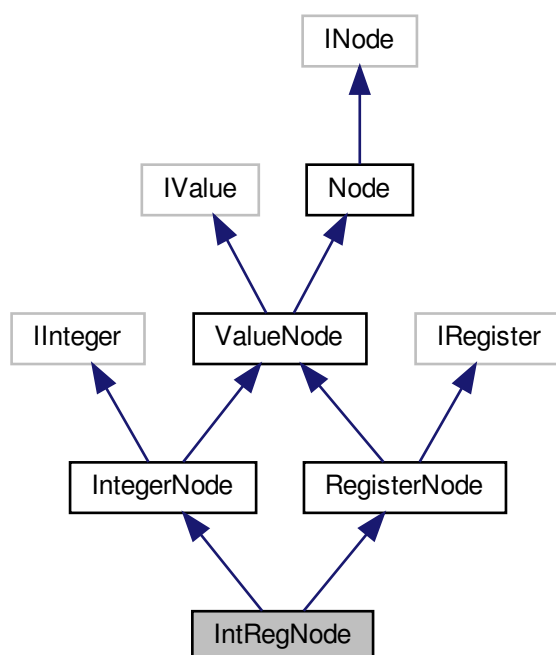
The documentation for this class was generated from the following file:

- include/[InterfaceRemovalEventHandler.h](#)

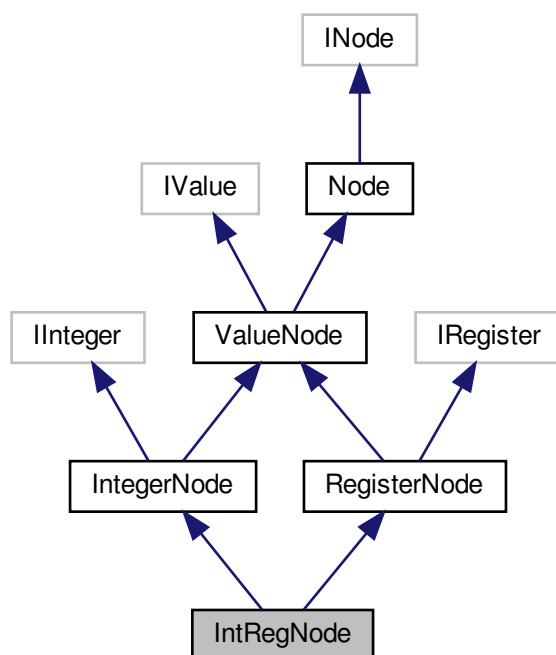
15.139 IntRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for IntRegNode:



Collaboration diagram for IntRegNode:



Public Member Functions

- [IntRegNode](#) ()
- [IntRegNode](#) (std::shared_ptr< Node::NodeImpl > pInteger)
- virtual [~IntRegNode](#) ()
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.139.1 Detailed Description

[Interface](#) for string properties.

15.139.2 Constructor & Destructor Documentation

15.139.2.1 IntRegNode() [1/2]

[IntRegNode](#) ()

15.139.2.2 IntRegNode() [2/2]

[IntRegNode](#) (
 std::shared_ptr< Node::NodeImpl > *pInteger*)

15.139.2.3 ~IntRegNode()

virtual ~[IntRegNode](#) () [virtual]

15.139.3 Member Function Documentation

15.139.3.1 SetReference()

virtual void SetReference (
 [INode](#) * *pBase*) [virtual]

overload SetReference for Value

Reimplemented from [IntegerNode](#).

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

- include/SpinGenApi/[IntRegNode.h](#)

15.140 IpInfo Struct Reference

Public Member Functions

- [IpInfo](#) ()

Public Attributes

- std::string [ipAddress](#)
- std::string [subnetMask](#)
- std::string [gateway](#)
- unsigned int [subnetLength](#)

15.140.1 Constructor & Destructor Documentation

15.140.1.1 IpInfo()

```
IpInfo ( ) [inline]
```

15.140.2 Member Data Documentation

15.140.2.1 gateway

```
std::string gateway
```

15.140.2.2 ipAddress

```
std::string ipAddress
```

15.140.2.3 subnetLength

```
unsigned int subnetLength
```

15.140.2.4 subnetMask

```
std::string subnetMask
```

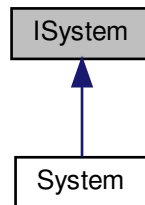
The documentation for this struct was generated from the following file:

- include/[AdapterConfig.h](#)

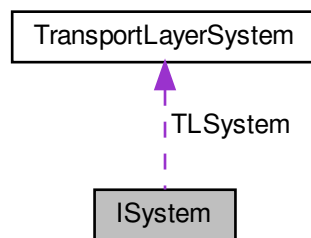
15.141 ISystem Class Reference

The interface file for [System](#).

Inheritance diagram for ISystem:



Collaboration diagram for ISystem:



Public Member Functions

- virtual [~ISystem](#) ()
- virtual void [ReleaseInstance](#) ()=0
- virtual [InterfaceList GetInterfaces](#) (bool updateInterface=true)=0
- virtual void [UpdateInterfaceList](#) ()=0
- virtual [CameraList GetCameras](#) (bool updateInterfaces=true, bool updateCameras=true)=0
- virtual bool [UpdateCameras](#) (bool updateInterfaces=true)=0
- virtual void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)=0
- virtual void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)=0
- virtual void [RegisterInterfaceEventHandler](#) ([EventHandler](#) &evtHandlerToRegister, bool updateInterface=true)=0
- virtual void [UnregisterInterfaceEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)=0
- virtual void [RegisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)=0
- virtual void [UnregisterAllLoggingEventHandlers](#) ()=0

- virtual void [UnregisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)=0
- virtual void [SetLoggingEventPriorityLevel](#) ([SpinnakerLogLevel](#) level)=0
- virtual [SpinnakerLogLevel](#) [GetLoggingEventPriorityLevel](#) ()=0
- virtual bool [IsInUse](#) ()=0
- virtual void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL)=0
- virtual const [LibraryVersion](#) [GetLibraryVersion](#) ()=0
- virtual [GenApi::INodeMap](#) & [GetTLNodeMap](#) () const =0

Public Attributes

- [TransportLayerSystem](#) [TLSystem](#)

Protected Member Functions

- [ISystem](#) ()
- [ISystem](#) (const [ISystem](#) &)
- [ISystem](#) & [operator=](#) (const [ISystem](#) &)

Friends

- class [SystemPtrInternal](#)

15.141.1 Detailed Description

The interface file for [System](#).

15.141.2 Constructor & Destructor Documentation

15.141.2.1 ~ISystem()

```
virtual ~ISystem ( ) [inline], [virtual]
```

15.141.2.2 ISystem() [1/2]

```
ISystem ( ) [inline], [protected]
```

15.141.2.3 ISystem() [2/2]

```
ISystem (
    const ISystem & ) [inline], [protected]
```

15.141.3 Member Function Documentation**15.141.3.1 GetCameras()**

```
virtual CameraList GetCameras (
    bool updateInterfaces = true,
    bool updateCameras = true ) [pure virtual]
```

Implemented in [System](#).

15.141.3.2 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

15.141.3.3 GetLibraryVersion()

```
virtual const LibraryVersion GetLibraryVersion ( ) [pure virtual]
```

Implemented in [System](#).

15.141.3.4 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( ) [pure virtual]
```

Implemented in [System](#).

15.141.3.5 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in [System](#).

15.141.3.6 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [System](#).

15.141.3.7 operator=()

```
ISystem& operator= (
    const ISystem & ) [protected]
```

15.141.3.8 RegisterEventHandler()

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [System](#).

15.141.3.9 RegisterInterfaceEventHandler()

```
virtual void RegisterInterfaceEventHandler (
    EventHandler & evtHandlerToRegister,
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

15.141.3.10 RegisterLoggingEventHandler()

```
virtual void RegisterLoggingEventHandler (
    LoggingEventHandler & handler ) [pure virtual]
```

Implemented in [System](#).

15.141.3.11 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [pure virtual]
```

Implemented in [System](#).

15.141.3.12 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) [pure virtual]
```

Implemented in [System](#).

15.141.3.13 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [pure virtual]
```

Implemented in [System](#).

15.141.3.14 UnregisterAllLoggingEventHandlers()

```
virtual void UnregisterAllLoggingEventHandlers ( ) [pure virtual]
```

Implemented in [System](#).

15.141.3.15 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [System](#).

15.141.3.16 UnregisterInterfaceEventHandler()

```
virtual void UnregisterInterfaceEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [System](#).

15.141.3.17 UnregisterLoggingEventHandler()

```
virtual void UnregisterLoggingEventHandler (
    LoggingEventHandler & handler ) [pure virtual]
```

Implemented in [System](#).

15.141.3.18 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [pure virtual]
```

Implemented in [System](#).

15.141.3.19 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [pure virtual]
```

Implemented in [System](#).

15.141.4 Friends And Related Function Documentation**15.141.4.1 SystemPtrInternal**

```
friend class SystemPtrInternal [friend]
```

15.141.5 Member Data Documentation

15.141.5.1 TLSystem

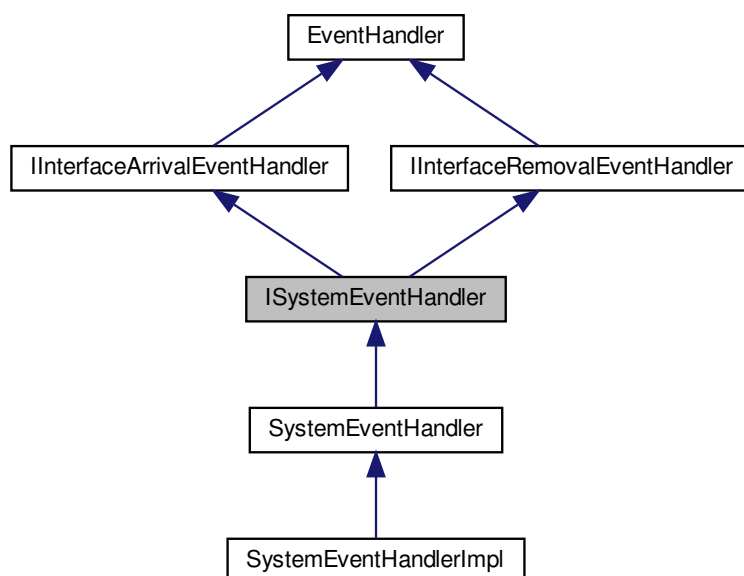
[TransportLayerSystem](#) TLSystem

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

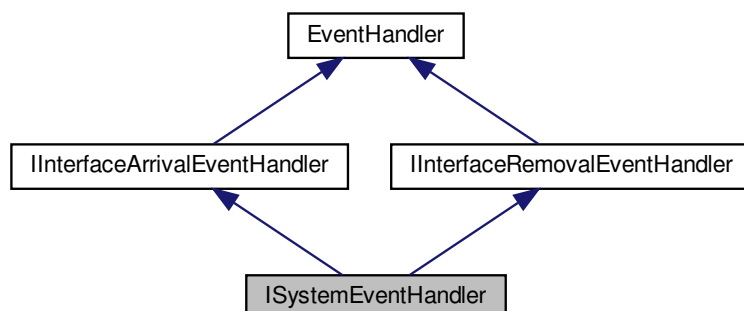
- [include/Interface/ISystem.h](#)

15.142 ISystemEventHandler Class Reference

Inheritance diagram for ISystemEventHandler:



Collaboration diagram for ISystemEventHandler:



Public Member Functions

- virtual [~ISystemEventHandler](#) ()
- virtual void [OnInterfaceArrival](#) (std::string interfaceID)=0
- virtual void [OnInterfaceRemoval](#) (std::string interfaceID)=0

Protected Member Functions

- [ISystemEventHandler](#) ()
- [ISystemEventHandler](#) (const [ISystemEventHandler](#) &)
- [ISystemEventHandler](#) & [operator=](#) (const [ISystemEventHandler](#) &)

Additional Inherited Members

15.142.1 Constructor & Destructor Documentation

15.142.1.1 ~ISystemEventHandler()

```
virtual ~ISystemEventHandler ( ) [inline], [virtual]
```

15.142.1.2 ISystemEventHandler() [1/2]

```
ISystemEventHandler ( ) [inline], [protected]
```

15.142.1.3 ISystemEventHandler() [2/2]

```
ISystemEventHandler (
    const ISystemEventHandler & ) [inline], [protected]
```

15.142.2 Member Function Documentation

15.142.2.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    std::string interfaceID ) [pure virtual]
```

Implements [IInterfaceArrivalEventHandler](#).

Implemented in [SystemEventHandlerImpl](#), and [SystemEventHandler](#).

15.142.2.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

Implements [IInterfaceRemovalEventHandler](#).

Implemented in [SystemEventHandlerImpl](#), and [SystemEventHandler](#).

15.142.2.3 operator=()

```
ISystemEventHandler& operator= (
    const ISystemEventHandler & ) [protected]
```

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

- [include/Interface/ISystemEventHandler.h](#)

15.143 JPEGOption Struct Reference

Options for saving JPEG image.

Public Member Functions

- [JPEGOption](#) ()

Public Attributes

- bool [progressive](#)
Whether to save as a progressive JPEG file.
- unsigned int [quality](#)
JPEG image quality in range (0-100).
- unsigned int [reserved](#) [16]
Reserved for future use.

15.143.1 Detailed Description

Options for saving JPEG image.

15.143.2 Constructor & Destructor Documentation

15.143.2.1 JPEGOption()

```
JPEGOption ( ) [inline]
```

15.143.3 Member Data Documentation

15.143.3.1 progressive

```
bool progressive
```

Whether to save as a progressive JPEG file.

15.143.3.2 quality

```
unsigned int quality
```

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

15.143.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/[SpinnakerDefs.h](#)

15.144 JPG2Option Struct Reference

Options for saving JPEG2000 image.

Public Member Functions

- [JPG2Option](#) ()

Public Attributes

- unsigned int [quality](#)
JPEG saving quality in range (1-512).
- unsigned int [reserved](#) [16]
Reserved for future use.

15.144.1 Detailed Description

Options for saving JPEG2000 image.

15.144.2 Constructor & Destructor Documentation

15.144.2.1 [JPG2Option](#)()

```
JPG2Option ( ) [inline]
```

15.144.3 Member Data Documentation

15.144.3.1 [quality](#)

```
unsigned int quality
```

JPEG saving quality in range (1-512).

15.144.3.2 [reserved](#)

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/[SpinnakerDefs.h](#)

15.145 LibraryVersion Struct Reference

Provides easier access to the current version of [Spinnaker](#).

Public Attributes

- unsigned int [major](#)
Major version of the library.
- unsigned int [minor](#)
Minor version of the library.
- unsigned int [type](#)
Version type of the library.
- unsigned int [build](#)
Build number of the library.

15.145.1 Detailed Description

Provides easier access to the current version of [Spinnaker](#).

15.145.2 Member Data Documentation

15.145.2.1 build

```
unsigned int build
```

Build number of the library.

15.145.2.2 major

```
unsigned int major
```

Major version of the library.

15.145.2.3 minor

```
unsigned int minor
```

Minor version of the library.

15.145.2.4 type

unsigned int type

Version type of the library.

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

- include/[SpinnakerDefs.h](#)

15.146 LockableObject< Object >::Lock Class Reference

A scopelevel [Lock](#) class.

Public Member Functions

- [Lock](#) (const [LockableObject](#)< Object > &obj)
- [~Lock](#) ()

15.146.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >::Lock
```

A scopelevel [Lock](#) class.

Automatically acquires the lock when created and releases it when destroyed.

15.146.2 Constructor & Destructor Documentation

15.146.2.1 Lock()

```
Lock (
    const LockableObject< Object > & obj ) [inline]
```

15.146.2.2 ~Lock()

```
~Lock ( ) [inline]
```

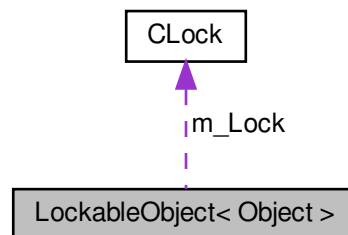
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

15.147 LockableObject< Object > Class Template Reference

Instance-Lock for an object.

Collaboration diagram for LockableObject< Object >:



Classes

- class [Lock](#)
A scopelevel [Lock](#) class.

Public Member Functions

- [Lock GetLock](#) () const
Get a new lock.

Public Attributes

- [CLock m_Lock](#)

Friends

- class [Lock](#)

15.147.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >
```

Instance-Lock for an object.

15.147.2 Member Function Documentation

15.147.2.1 GetLock()

```
Lock GetLock ( ) const [inline]
```

Get a new lock.

15.147.3 Friends And Related Function Documentation

15.147.3.1 Lock

```
friend class Lock [friend]
```

15.147.4 Member Data Documentation

15.147.4.1 m_Lock

```
CLock m_Lock [mutable]
```

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

- include/SpinGenApi/[GCSynch.h](#)

15.148 LoggingEventData Class Reference

The [LoggingEventData](#) object.

Public Member Functions

- [~LoggingEventData](#) ()
Default Destructor.
- const char * [GetCategoryName](#) ()
Gets the logging event category name.
- const char * [GetLogMessage](#) ()
Gets the logging event message.
- const char * [GetNDC](#) ()
Gets the logging event's Nested Diagnostic Context (NDC).
- const int [GetPriority](#) ()
Gets the logging event priority.
- const char * [GetThreadName](#) ()
Gets the logging event thread name.
- const char * [GetTimestamp](#) ()
Gets the logging event time stamp.
- const char * [GetPriorityName](#) ()
Gets the logging event priority name.

Protected Member Functions

- [LoggingEventData](#) (void *data)
Default Constructor.

Friends

- class [SystemImpl](#)

15.148.1 Detailed Description

The [LoggingEventData](#) object.

15.148.2 Constructor & Destructor Documentation

15.148.2.1 ~LoggingEventData()

[~LoggingEventData](#) ()

Default Destructor.

15.148.2.2 LoggingEventData()

```
LoggingEventData (
    void * data ) [protected]
```

Default Constructor.

15.148.3 Member Function Documentation

15.148.3.1 GetCategoryName()

```
const char* GetCategoryName ( )
```

Gets the logging event category name.

Returns

The category name

15.148.3.2 GetLogMessage()

```
const char* GetLogMessage ( )
```

Gets the logging event message.

Returns

The log message

15.148.3.3 GetNDC()

```
const char* GetNDC ( )
```

Gets the logging event's Nested Diagnostic Context (NDC).

Returns

The log event's NDC

15.148.3.4 GetPriority()

```
const int GetPriority ( )
```

Gets the logging event priority.

Returns

The log priority

15.148.3.5 GetPriorityName()

```
const char* GetPriorityName ( )
```

Gets the logging event priority name.

Returns

The priority name of the log

15.148.3.6 GetThreadName()

```
const char* GetThreadName ( )
```

Gets the logging event thread name.

Returns

The thread name

15.148.3.7 GetTimestamp()

```
const char* GetTimestamp ( )
```

Gets the logging event time stamp.

Returns

The time stamp of the log

15.148.4 Friends And Related Function Documentation

15.148.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

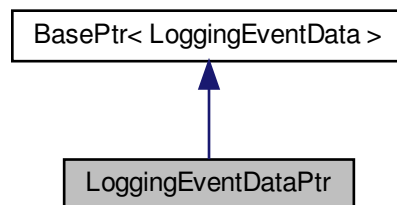
The documentation for this class was generated from the following file:

- include/[LoggingEventData.h](#)

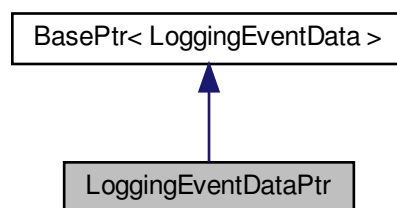
15.149 LoggingEventDataPtr Class Reference

A reference tracked pointer to the LoggingEvent object.

Inheritance diagram for LoggingEventDataPtr:



Collaboration diagram for LoggingEventDataPtr:



Public Member Functions

- [LoggingEventDataPtr](#) () throw ()
Default Constructor.
- [LoggingEventDataPtr](#) (const int) throw ()
Default Constructor with argument.
- [LoggingEventDataPtr](#) (const long) throw ()
Default Constructor with argument.
- [LoggingEventDataPtr](#) (const std::nullptr_t) throw ()
Default Constructor with argument.

Additional Inherited Members

15.149.1 Detailed Description

A reference tracked pointer to the LoggingEvent object.

15.149.2 Constructor & Destructor Documentation

15.149.2.1 LoggingEventDataPtr() [1/4]

```
LoggingEventDataPtr ( ) throw ) [inline]
```

Default Constructor.

15.149.2.2 LoggingEventDataPtr() [2/4]

```
LoggingEventDataPtr (
    const int ) throw ) [inline]
```

Default Constructor with argument.

15.149.2.3 LoggingEventDataPtr() [3/4]

```
LoggingEventDataPtr (
    const long ) throw ) [inline]
```

Default Constructor with argument.

15.149.2.4 LoggingEventDataPtr() [4/4]

```
LoggingEventDataPtr (
    const std::nullptr_t ) throw ) [inline]
```

Default Constructor with argument.

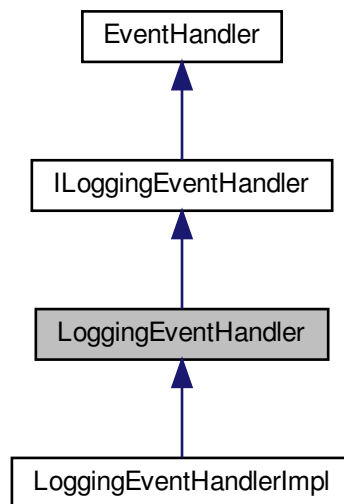
The documentation for this class was generated from the following file:

- include/LoggingEventDataPtr.h

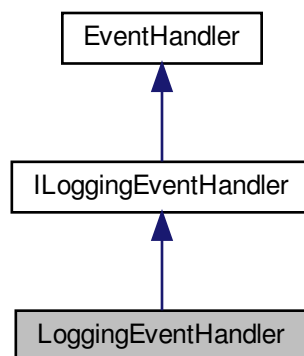
15.150 LoggingEventHandler Class Reference

An event handler for capturing the device logging event.

Inheritance diagram for LoggingEventHandler:



Collaboration diagram for LoggingEventHandler:



Public Member Functions

- [LoggingEventHandler\(\)](#)

Default constructor.

- `~LoggingEventHandler()`

Virtual destructor.

- virtual void `OnLogEvent(LoggingEventDataPtr eventPtr)=0`

The callback for the log event.

Protected Member Functions

- `LoggingEventHandler & operator= (const LoggingEventHandler &)`

Assignment operator.

Additional Inherited Members

15.150.1 Detailed Description

An event handler for capturing the device logging event.

15.150.2 Constructor & Destructor Documentation

15.150.2.1 LoggingEventHandler()

```
LoggingEventHandler ( )
```

Default constructor.

15.150.2.2 ~LoggingEventHandler()

```
~LoggingEventHandler ( )
```

Virtual destructor.

15.150.3 Member Function Documentation

15.150.3.1 OnLogEvent()

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

The callback for the log event.

Parameters

<i>eventPtr</i>	The logging event pointer

Implements [ILoggingEventHandler](#).

15.150.3.2 operator=()

```
LoggingEventHandler& operator= (
    const LoggingEventHandler & ) [protected]
```

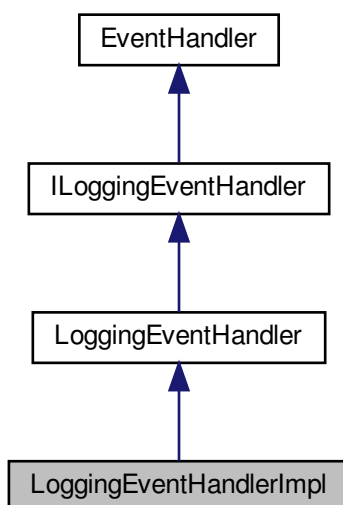
Assignment operator.

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

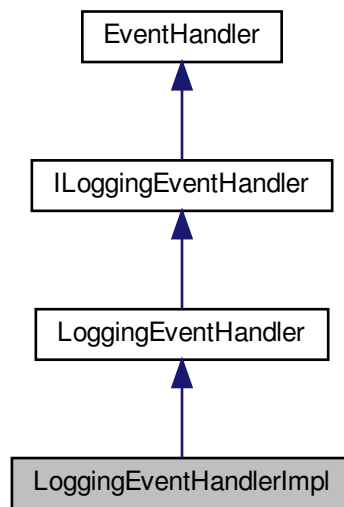
- include/[LoggingEventHandler.h](#)

15.151 LoggingEventHandlerImpl Class Reference

Inheritance diagram for LoggingEventHandlerImpl:



Collaboration diagram for LoggingEventHandlerImpl:



Additional Inherited Members

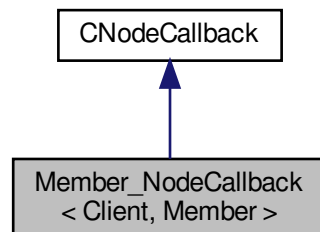
The documentation for this class was generated from the following file:

- src/Logging/[Logging.cpp](#)

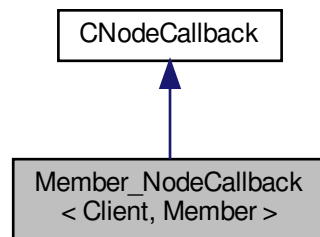
15.152 Member_NodeCallback< Client, Member > Class Template Reference

Container for a member function pointer.

Inheritance diagram for Member_NodeCallback< Client, Member >:



Collaboration diagram for Member_NodeCallback< Client, Member >:



Public Types

- `typedef void(Client::* PMEMBERFUNC) (INode *)`
Member function type.

Public Member Functions

- `Member_NodeCallback (INode *pNode, Client &client, Member member, ECallbackType CallbackType)`
Constructor.
- `virtual void operator() (ECallbackType CallbackType) const`
execute operation
- `virtual void Destroy ()`
destroys the object

Additional Inherited Members

15.152.1 Detailed Description

```
template<class Client, class Member>
class Spinnaker::GenApi::Member_NodeCallback< Client, Member >
```

Container for a member function pointer.

15.152.2 Member Typedef Documentation

15.152.2.1 PMEMBERFUNC

```
typedef void(Client::* PMEMBERFUNC) (INode *)
```

Member function type.

15.152.3 Constructor & Destructor Documentation

15.152.3.1 Member_NodeCallback()

```
Member_NodeCallback (
    INode * pNode,
    Client & client,
    Member member,
    ECallbackType CallbackType ) [inline]
```

Constructor.

15.152.4 Member Function Documentation

15.152.4.1 Destroy()

```
virtual void Destroy ( ) [inline], [virtual]
```

destroys the object

Implements [CNodeCallback](#).

15.152.4.2 operator()

```
virtual void operator() (
    ECallbackType CallbackType ) const [inline], [virtual]
```

execute operation

Implements [CNodeCallback](#).

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

- include/SpinGenApi/[NodeCallback.h](#)

15.153 MJPGOption Struct Reference

Options for saving MJPG files.

Public Member Functions

- [MJPGOption](#) ()

Public Attributes

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [quality](#)
Image quality (1-100)
- unsigned int [reserved](#) [256]

15.153.1 Detailed Description

Options for saving MJPG files.

15.153.2 Constructor & Destructor Documentation

15.153.2.1 MJPGOption()

```
MJPGOption ( ) [inline]
```

15.153.3 Member Data Documentation

15.153.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

15.153.3.2 quality

```
unsigned int quality
```

[Image](#) quality (1-100)

15.153.3.3 reserved

```
unsigned int reserved[256]
```

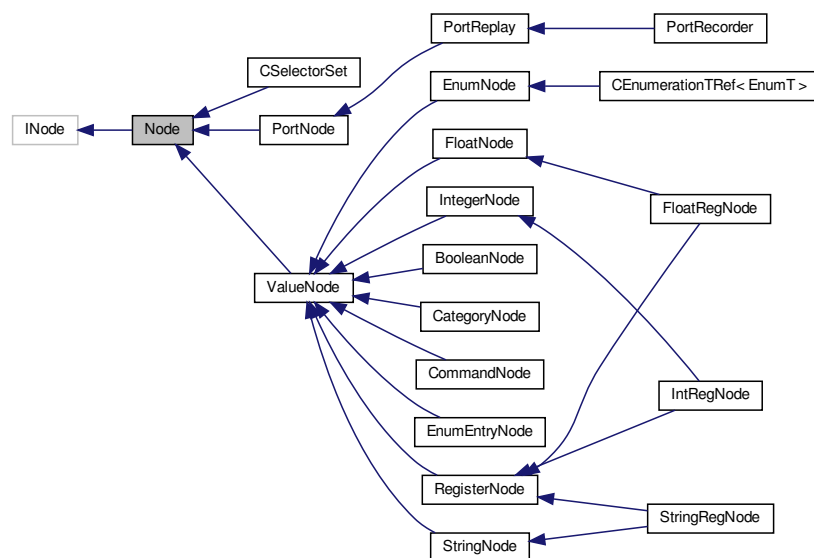
The documentation for this struct was generated from the following file:

- [include/SpinVideoDefs.h](#)

15.154 Node Class Reference

class common to all nodes

Inheritance diagram for Node:



Collaboration diagram for Node:



Public Member Functions

- [Node](#) ()
Constructor.
- [Node](#) (std::shared_ptr< Node::NodeImpl > pNodeHandle)
Constructor.
- [~Node](#) ()
Destructor.
- virtual [GenICam::gcstring](#) [GetName](#) (bool FullQualified=false) const
Get node name.
- virtual [GenApi::ENamespace](#) [GetNamespace](#) () const
Get name space.
- virtual [EVisibility](#) [GetVisibility](#) () const
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const
Is the node value cacheable.
- virtual [EYesNo](#) [IsAccessModeCacheable](#) () const
True if the AccessMode can be cached.
- virtual [ECachingMode](#) [GetCachingMode](#) () const
Get Caching Mode.
- virtual int64_t [GetPollingTime](#) () const
recommended polling time (for not cacheable nodes)
- virtual [GenICam::gcstring](#) [GetToolTip](#) () const
Get a short description of the node.
- virtual [GenICam::gcstring](#) [GetDescription](#) () const
Get a long description of the node.
- virtual [GenICam::gcstring](#) [GetDisplayName](#) () const
Get a name string for display.
- virtual [GenICam::gcstring](#) [GetDeviceName](#) () const
Get a name of the device.
- virtual void [GetChildren](#) ([GenApi::NodeList_t](#) &Children, [ELinkType](#) LinkType=ctReadingChildren) const
Get all nodes this node directly depends on.
- virtual void [GetParents](#) ([GenApi::NodeList_t](#) &Parents) const
Gets all nodes this node is directly depending on.
- virtual [CallbackHandleType](#) [RegisterCallback](#) ([CNodeCallback](#) *pCallback)
Register change callback Takes ownership of the CNodeCallback object.
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)
De register change callback Destroys CNodeCallback object.
- virtual [INodeMap](#) * [GetNodeMap](#) () const
Retrieves the central node map.
- virtual [GenICam::gcstring](#) [GetEventID](#) () const
Get the EventId of the node.
- virtual bool [IsStreamable](#) () const
True if the node is streamable.
- virtual void [GetPropertyNames](#) ([GenICam::gcstring_vector](#) &PropertyNames) const
Returns a list of the names all properties set during initialization.
- virtual bool [GetProperty](#) (const [GenICam::gcstring](#) &PropertyName, [GenICam::gcstring](#) &ValueStr, [GenICam::gcstring](#) &AttributeStr)

Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.

- virtual void [ImposeAccessMode](#) ([EAccessMode](#) ImposedAccessMode)
Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) ([EVisibility](#) ImposedVisibility)
Imposes a visibility to the natural visibility of the node.
- virtual [INode](#) * [GetAlias](#) () const
Retrieves the a node which describes the same feature in a different way.
- virtual [INode](#) * [GetCastAlias](#) () const
Retrieves the a node which describes the same feature so that it can be casted.
- virtual [GenICam::gcstring](#) [GetDocuURL](#) () const
Gets a URL pointing to the documentation of that feature.
- virtual bool [IsDeprecated](#) () const
True if the node should not be used any more.
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual bool [IsFeature](#) () const
True if the node can be reached via category nodes from a category node named "Root".
- void [SetNodeHandle](#) (std::shared_ptr< [Node::NodeImpl](#) > pNodeHandle)
Set [Node](#) handle.
- std::shared_ptr< [Node::NodeImpl](#) > [GetNodeHandle](#) () const
Get [Node](#) handle.
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Base interface overrides.
- virtual bool [IsSelector](#) () const
Selector interface overrides.
- virtual void [GetSelectedFeatures](#) ([FeatureList_t](#) &) const
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) ([FeatureList_t](#) &) const
retrieve the group of features selecting this node
- virtual void [SetReference](#) ([INode](#) *pBase)
Reference interface overrides `ingroup Spinnaker_GenApi_PublicImpl`.
- virtual void [SetReference](#) ([ISelector](#) *pBase)
- void [SetNodeMap](#) ([INodeMap](#) *pNodeMap)
- virtual bool [operator==](#) (int nullPtr) const
- virtual bool [operator!=](#) (int nullPtr) const

Protected Attributes

- std::shared_ptr< [Node::NodeImpl](#) > [m_pNodeData](#)
- std::list< [CallbackHandleType_t](#) * > [m_Callbacks](#)
List of callbacks.
- [INodeMap](#) * [m_pNodeMap](#)

15.154.1 Detailed Description

class common to all nodes

15.154.2 Constructor & Destructor Documentation

15.154.2.1 Node() [1/2]

`Node ( )`

Constructor.

15.154.2.2 Node() [2/2]

`Node (`
 `std::shared_ptr< Node::NodeImpl > pNodeHandle )`

Constructor.

15.154.2.3 ~Node()

`~Node ( )`

Destructor.

15.154.3 Member Function Documentation

15.154.3.1 DeregisterCallback()

`virtual bool DeregisterCallback (`
 `CallbackHandleType hCallback ) [virtual]`

De register change callback Destroys [CNodeCallback](#) object.

Returns

true if the callback handle was valid

15.154.3.2 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.154.3.3 GetAlias()

```
virtual INode* GetAlias ( ) const [virtual]
```

Retrieves the a node which describes the same feature in a different way.

15.154.3.4 GetCachingMode()

```
virtual ECachingMode GetCachingMode ( ) const [virtual]
```

Get Caching Mode.

15.154.3.5 GetCastAlias()

```
virtual INode* GetCastAlias ( ) const [virtual]
```

Retrieves the a node which describes the same feature so that it can be casted.

15.154.3.6 GetChildren()

```
virtual void GetChildren (
    GenApi::NodeList_t & Children,
    ELinkType LinkType = ctReadingChildren ) const [virtual]
```

Get all nodes this node directly depends on.

Parameters

out	<i>Children</i>	List of children nodes
	<i>LinkType</i>	The link type

15.154.3.7 GetDescription()

```
virtual GenICam::gcstring GetDescription ( ) const [virtual]
```

Get a long description of the node.

15.154.3.8 GetDeviceName()

```
virtual GenICam::gcstring GetDeviceName ( ) const [virtual]
```

Get a name of the device.

15.154.3.9 GetDisplayName()

```
virtual GenICam::gcstring GetDisplayName ( ) const [virtual]
```

Get a name string for display.

15.154.3.10 GetDocuURL()

```
virtual GenICam::gcstring GetDocuURL ( ) const [virtual]
```

Gets a URL pointing to the documentation of that feature.

15.154.3.11 GetEventID()

```
virtual GenICam::gcstring GetEventID ( ) const [virtual]
```

Get the EventId of the node.

15.154.3.12 GetName()

```
virtual GenICam::gcstring GetName (
    bool FullQualified = false ) const [virtual]
```

Get node name.

15.154.3.13 GetNameSpace()

```
virtual GenApi::ENamespace GetNameSpace ( ) const [virtual]
```

Get name space.

15.154.3.14 GetNodeHandle()

```
std::shared_ptr<Node::NodeImpl> GetNodeHandle ( ) const
```

Get [Node](#) handle.

15.154.3.15 GetNodeMap()

```
virtual INodeMap* GetNodeMap ( ) const [virtual]
```

Retrieves the central node map.

15.154.3.16 GetParents()

```
virtual void GetParents (
    GenApi::NodeList_t & Parents ) const [virtual]
```

Gets all nodes this node is directly depending on.

Parameters

out	<i>Parents</i>	List of parent nodes

15.154.3.17 GetPollingTime()

```
virtual int64_t GetPollingTime ( ) const [virtual]
```

recommended polling time (for not cacheable nodes)

15.154.3.18 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

15.154.3.19 GetProperty()

```
virtual bool GetProperty (
    const GenICam::gcstring & PropertyName,
    GenICam::gcstring & ValueStr,
    GenICam::gcstring & AttributeStr ) [virtual]
```

Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.

15.154.3.20 GetPropertyNames()

```
virtual void GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [virtual]
```

Returns a list of the names all properties set during initialization.

15.154.3.21 GetSelectedFeatures()

```
virtual void GetSelectedFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of selected features

15.154.3.22 GetSelectingFeatures()

```
virtual void GetSelectingFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of features selecting this node

15.154.3.23 GetToolTip()

```
virtual GenICam::gcstring GetToolTip ( ) const [virtual]
```

Get a short description of the node.

15.154.3.24 GetVisibility()

```
virtual EVisibility GetVisibility ( ) const [virtual]
```

Get the recommended visibility of the node.

15.154.3.25 ImposeAccessMode()

```
virtual void ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [virtual]
```

Imposes an access mode to the natural access mode of the node.

15.154.3.26 ImposeVisibility()

```
virtual void ImposeVisibility (
    EVisibility ImposedVisibility ) [virtual]
```

Imposes a visibility to the natural visibility of the node.

15.154.3.27 InvalidateNode()

```
virtual void InvalidateNode ( ) [virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

15.154.3.28 IsAccessModeCacheable()

```
virtual EYesNo IsAccessModeCacheable ( ) const [virtual]
```

True if the AccessMode can be cached.

15.154.3.29 IsCachable()

```
virtual bool IsCachable ( ) const [virtual]
```

Is the node value cacheable.

15.154.3.30 IsDeprecated()

```
virtual bool IsDeprecated ( ) const [virtual]
```

True if the node should not be used any more.

15.154.3.31 IsFeature()

```
virtual bool IsFeature ( ) const [virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

15.154.3.32 IsSelector()

```
virtual bool IsSelector ( ) const [virtual]
```

Selector interface overrides.

true if this feature selects a group of features

15.154.3.33 IsStreamable()

```
virtual bool IsStreamable ( ) const [virtual]
```

True if the node is streamable.

15.154.3.34 operator!=()

```
virtual bool operator!= (
    int nullptr ) const [virtual]
```

15.154.3.35 operator==()

```
virtual bool operator== (
    int nullPtr ) const [virtual]
```

15.154.3.36 RegisterCallback()

```
virtual CallbackHandleType RegisterCallback (
    CNodeCallback * pCallback ) [virtual]
```

Register change callback Takes ownership of the [CNodeCallback](#) object.

15.154.3.37 SetNodeHandle()

```
void SetNodeHandle (
    std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Set [Node](#) handle.

15.154.3.38 SetNodeMap()

```
void SetNodeMap (
    INodeMap * pNodeMap )
```

15.154.3.39 SetReference() [1/2]

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

Reference interface overrides \ingroup Spinnaker_GenApi_PublicImpl.

Reimplemented in [FloatNode](#), [PortNode](#), [IntegerNode](#), [EnumNode](#), [CEnumerationTRef< EnumT >](#), [StringNode](#), [ValueNode](#), [RegisterNode](#), [BooleanNode](#), [CommandNode](#), [EnumEntryNode](#), [CategoryNode](#), [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

15.154.3.40 SetReference() [2/2]

```
virtual void SetReference (
    ISelector * pBase ) [virtual]
```

15.154.4 Member Data Documentation

15.154.4.1 m_Callbacks

```
std::list<CallbackHandleType_t*> m_Callbacks [protected]
```

List of callbacks.

15.154.4.2 m_pNodeData

```
std::shared_ptr<Node::NodeImpl> m_pNodeData [protected]
```

15.154.4.3 m_pNodeMap

```
INodeMap* m_pNodeMap [protected]
```

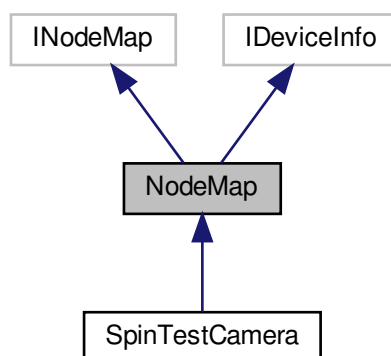
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Node.h](#)

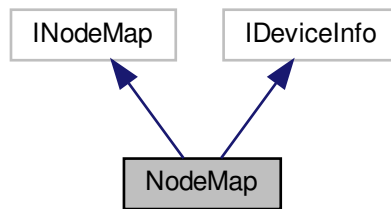
15.155 NodeMap Class Reference

Smart pointer template for NodeMaps with create function.

Inheritance diagram for NodeMap:



Collaboration diagram for NodeMap:



Public Member Functions

- `NodeMap (GenICam::gcstring DeviceName="Device")`
Constructor.
- `virtual ~NodeMap ()`
Destructor.
- `void Destroy ()`
Destroys the node map.
- `void LoadXMLFromFile (GenICam::gcstring FileName)`
Creates the object from a XML file with given file name.
- `void LoadXMLFromZIPFile (GenICam::gcstring ZipFileName)`
Creates the object from a ZIP'd XML file with given file name.
- `void LoadXMLFromZIPData (const void *zipData, size_t zipSize)`
Creates the object from a ZIP'd XML file given in a string.
- `void LoadXMLFromFileInject (GenICam::gcstring TargetFileName, GenICam::gcstring InjectFileName)`
Creates the object from a XML target and an inject file with given file name.
- `void LoadXMLFromString (const GenICam::gcstring &XMLData)`
Creates the object from XML data given in a string.
- `void LoadXMLFromStringInject (const GenICam::gcstring &TargetXMLDataconst, const GenICam::gcstring &InjectXMLData)`
Creates the object from XML data given in a string with injection.
- `virtual void GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions)`
Gets a list of supported schema versions.
- `virtual GenICam::gcstring GetDeviceName ()`
Get device name.
- `virtual void Poll (int64_t ElapsedTime)`
Fires nodes which have a polling time.
- `virtual void GetNodes (NodeList_t &Nodes) const`
Retrieves all nodes in the node map.
- `virtual INode * GetNode (const GenICam::gcstring &key) const`
Retrieves the node from the central map by name.
- `virtual void InvalidateNodes () const`
Invalidates all nodes.
- `virtual bool Connect (IPort *pPort, const GenICam::gcstring &PortName) const`
Connects a port to a port node with given name.

- virtual bool [Connect](#) ([IPort](#) *pPort) const
Connects a port to the standard port "Device".
- virtual [CLock](#) & [GetLock](#) () const
Returns the lock which guards the node map.
- virtual uint64_t [GetNumNodes](#) () const
Get the number of nodes in the map.
- void * [GetNodeMapHandle](#) () const
- virtual [GenICam::gcstring](#) [GetModelName](#) ()
Get the model name.
- virtual [GenICam::gcstring](#) [GetVendorName](#) ()
Get the vendor name.
- virtual [GenICam::gcstring](#) [GetToolTip](#) ()
Get tool tip.
- virtual [GenICam::gcstring](#) [GetStandardNameSpace](#) ()
Get the standard name space.
- virtual void [GetGenApiVersion](#) ([GenICam::Version_t](#) &Version, uint16_t &Build)
Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) ([GenICam::Version_t](#) &Version)
Get the schema version number.
- virtual void [GetDeviceVersion](#) ([GenICam::Version_t](#) &Version)
Get the version of the device description file.
- virtual [GenICam::gcstring](#) [GetProductGuid](#) ()
Get the GUID describing the product.
- virtual [GenICam::gcstring](#) [GetVersionGuid](#) ()
Get the GUID describing the product version.

Static Public Member Functions

- static bool [ClearXMLCache](#) ()
Clears the cache of the camera description files.

Public Attributes

- [INodeMap](#) * [_Ptr](#)
Pointer to the [NodeMap](#).

15.155.1 Detailed Description

Smart pointer template for NodeMaps with create function.

Parameters

<i>TCameraParams</i>	The camera specific parameter class (auto generated from camera xml file)

15.155.2 Constructor & Destructor Documentation

15.155.2.1 NodeMap()

```
NodeMap (
    GenICam::gcstring DeviceName = "Device" )
```

Constructor.

15.155.2.2 ~NodeMap()

```
virtual ~NodeMap ( ) [virtual]
```

Destructor.

15.155.3 Member Function Documentation

15.155.3.1 ClearXMLCache()

```
static bool ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

15.155.3.2 Connect() [1/2]

```
virtual bool Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) const [virtual]
```

Connects a port to a port node with given name.

15.155.3.3 Connect() [2/2]

```
virtual bool Connect (
    IPort * pPort ) const [virtual]
```

Connects a port to the standard port "Device".

15.155.3.4 Destroy()

```
void Destroy ( )
```

Destroys the node map.

15.155.3.5 GetDeviceName()

```
virtual GenICam::gcstring GetDeviceName ( ) [virtual]
```

Get device name.

15.155.3.6 GetDeviceVersion()

```
virtual void GetDeviceVersion (
    GenICam::Version\_t & Version ) [virtual]
```

Get the version of the device description file.

15.155.3.7 GetGenApiVersion()

```
virtual void GetGenApiVersion (
    GenICam::Version\_t & Version,
    uint16\_t & Build ) [virtual]
```

Get the version of the DLL's [GenApi](#) implementation.

15.155.3.8 GetLock()

```
virtual CLock& GetLock ( ) const [virtual]
```

Returns the lock which guards the node map.

15.155.3.9 GetModelName()

```
virtual GenICam::gcstring GetModelName ( ) [virtual]
```

Get the model name.

15.155.3.10 GetNode()

```
virtual INode* GetNode (
    const GenICam::gcstring & key ) const [virtual]
```

Retrieves the node from the central map by name.

15.155.3.11 GetNodeMapHandle()

```
void* GetNodeMapHandle ( ) const
```

15.155.3.12 GetNodes()

```
virtual void GetNodes (
    NodeList_t & Nodes ) const [virtual]
```

Retrieves all nodes in the node map.

15.155.3.13 GetNumNodes()

```
virtual uint64_t GetNumNodes ( ) const [virtual]
```

Get the number of nodes in the map.

15.155.3.14 GetProductGuid()

```
virtual GenICam::gcstring GetProductGuid ( ) [virtual]
```

Get the GUID describing the product.

15.155.3.15 GetSchemaVersion()

```
virtual void GetSchemaVersion (
    GenICam::Version_t & Version ) [virtual]
```

Get the schema version number.

15.155.3.16 GetStandardNameSpace()

```
virtual GenICam::gcstring GetStandardNameSpace ( ) [virtual]
```

Get the standard name space.

15.155.3.17 GetSupportedSchemaVersions()

```
virtual void GetSupportedSchemaVersions (
    GenICam::gcstring\_vector & SchemaVersions ) [virtual]
```

Gets a list of supported schema versions.

! Loads an XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromFile(const [GenICam::gcstring](#)& XMLFileName, const [GenICam::gcstring](#)& StyleSheetFileName, const [GenICam::gcstring](#)& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Loads a Zipped XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromZIPFile(const [GenICam::gcstring](#)& ZIPFileName, const [GenICam::gcstring](#)& StyleSheetFileName, const [GenICam::gcstring](#)& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Injects an XML file into a target file virtual void MergeXMLFiles(const [GenICam::gcstring](#)& TargetFileName, *< Name of the target XML file to process const [GenICam::gcstring](#)& InjectedFileName, *< Name of the Injected XML file to process const [GenICam::gcstring](#)& OutputFileName *< Name of the output file);

! Extract independent subtree virtual void ExtractIndependentSubtree(const [GenICam::gcstring](#)& XMLData, *< The XML data the subtree is extracted from. const [GenICam::gcstring](#)& InjectXMLData, *< Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed. const [GenICam::gcstring](#)& SubTreeRootNodeName,*< The name of the node that represents the root of the subtree that shall be extracted. [GenICam::gcstring](#)& ExtractedSubtree *< The returned extracted subtree as string.);

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

15.155.3.18 GetToolTip()

```
virtual GenICam::gcstring GetToolTip ( ) [virtual]
```

Get tool tip.

15.155.3.19 GetVendorName()

```
virtual GenICam::gcstring GetVendorName ( ) [virtual]
```

Get the vendor name.

15.155.3.20 GetVersionGuid()

```
virtual GenICam::gcstring GetVersionGuid ( ) [virtual]
```

Get the GUID describing the product version.

15.155.3.21 InvalidateNodes()

```
virtual void InvalidateNodes ( ) const [virtual]
```

Invalidates all nodes.

15.155.3.22 LoadXMLFromFile()

```
void LoadXMLFromFile (
    GenICam::gcstring FileName )
```

Creates the object from a XML file with given file name.

! Creates the object from the default DLL ! note Can only be used if the class TCameraParams was auto generated from a specific camera xml file void LoadDLL(void);

! Creates the object from a DLL whose name is deduced from vendor and model name void LoadDLL(GenICam::gcstring VendorName, GenICam::gcstring ModelName);

! Creates the object from a DLL with given file name void LoadDLL(GenICam::gcstring FileName);

15.155.3.23 LoadXMLFromFileInject()

```
void LoadXMLFromFileInject (
    GenICam::gcstring TargetFileName,
    GenICam::gcstring InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

15.155.3.24 LoadXMLFromString()

```
void LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

15.155.3.25 LoadXMLFromStringInject()

```
void LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

15.155.3.26 LoadXMLFromZIPData()

```
void LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

15.155.3.27 LoadXMLFromZIPFile()

```
void LoadXMLFromZIPFile (
    GenICam::gcstring ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

15.155.3.28 Poll()

```
virtual void Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

15.155.4 Member Data Documentation

15.155.4.1 _Ptr

[INodeMap](#)* _Ptr

Pointer to the [NodeMap](#).

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

- [include/SpinGenApi/NodeMap.h](#)

15.156 CNodeMapFactory::NodeStatistics_t Struct Reference

Public Attributes

- uint32_t [NumNodes](#)
- uint32_t [NumProperties](#)
- uint32_t [NumLinks](#)
- uint32_t [NumStrings](#)

15.156.1 Member Data Documentation

15.156.1.1 NumLinks

uint32_t NumLinks

15.156.1.2 NumNodes

uint32_t NumNodes

15.156.1.3 NumProperties

uint32_t NumProperties

15.156.1.4 NumStrings

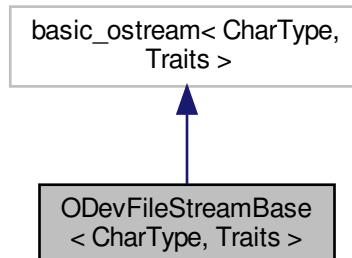
uint32_t NumStrings

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

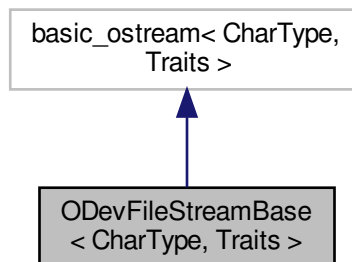
- include/SpinGenApi/[NodeMapFactory.h](#)

15.157 ODevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBase< CharType, Traits >:



Collaboration diagram for ODevFileStreamBase< CharType, Traits >:



Public Types

- typedef [ODevFileStreamBuf< CharType, Traits >](#) [filebuf_type](#)
- typedef std::basic_ios< CharType, Traits > [ios_type](#)
- typedef std::basic_ostream< CharType, Traits > [ostream_type](#)

Public Member Functions

- [filebuf_type](#) * [rdbuf](#) () const
- bool [is_open](#) () const
- void [open](#) (INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::out|std::ios_base::trunc)
Open file on device in write mode.
- void [close](#) ()
Close the file on device.

15.157.1 Member Typedef Documentation

15.157.1.1 filebuf_type

```
typedef ODevFileStreamBuf<CharType, Traits> filebuf_type
```

15.157.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

15.157.1.3 ostream_type

```
typedef std::basic_ostream<CharType, Traits> ostream_type
```

15.157.2 Member Function Documentation

15.157.2.1 close()

```
void close ( ) [inline]
```

Close the file on device.

15.157.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.157.2.3 open()

```
void open (
    INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::out | std::ios_base::trunc ) [inline]
```

Open file on device in write mode.

Parameters

<i>pInterface</i>	NodeMap of the device to which the FileProtocolAdapter is attached
<i>pFileName</i>	Name of the file to open
<i>mode</i>	open mode

15.157.2.4 rdbuf()

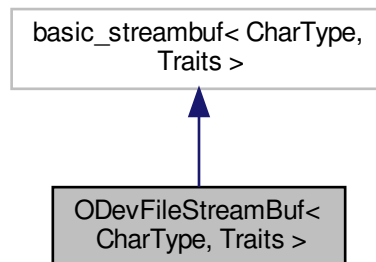
```
filebuf_type* rdbuf ( ) const [inline]
```

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

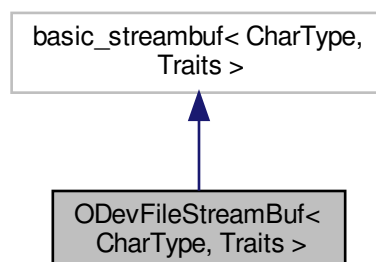
- include/SpinGenApi/[Filestream.h](#)

15.158 ODevFileStreamBuf< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBuf< CharType, Traits >:



Collaboration diagram for ODevFileStreamBuf< CharType, Traits >:



Public Member Functions

- [ODevFileStreamBuf](#) ()
- [~ODevFileStreamBuf](#) ()
- [filebuf_type](#) * [open](#) (Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode)
- bool [is_open](#) () const
- [filebuf_type](#) * [close](#) ()

Protected Member Functions

- std::streamsize [xspu](#)tn (const char_type *s, std::streamsize n)
- int_type [overflow](#) (int_type c=traits_type::eof())
- int [sync](#) ()

15.158.1 Constructor & Destructor Documentation

15.158.1.1 ODevFileStreamBuf()

```
ODevFileStreamBuf ( ) [inline]
```

15.158.1.2 ~ODevFileStreamBuf()

```
~ODevFileStreamBuf ( ) [inline]
```

15.158.2 Member Function Documentation

15.158.2.1 close()

```
filebuf_type* close ( ) [inline]
```

15.158.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.158.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode ) [inline]
```

15.158.2.4 overflow()

```
int_type overflow (
    int_type c = traits_type::eof() ) [inline], [protected]
```

15.158.2.5 sync()

```
int sync ( ) [inline], [protected]
```

15.158.2.6 xspn()

```
std::streamsize xspn (
    const char_type * s,
    std::streamsize n ) [inline], [protected]
```

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

- include/SpinGenApi/[Filestream.h](#)

15.159 PGMOption Struct Reference

Options for saving PGM images.

Public Member Functions

- [PGMOption](#) ()

Public Attributes

- bool [binaryFile](#)
Whether to save the PPM as a binary file.
- unsigned int [reserved](#) [16]
Reserved for future use.

15.159.1 Detailed Description

Options for saving PGM images.

15.159.2 Constructor & Destructor Documentation

15.159.2.1 PGMOption()

```
PGMOption ( ) [inline]
```

15.159.3 Member Data Documentation

15.159.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

15.159.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

15.160 PNGOption Struct Reference

Options for saving PNG images.

Public Member Functions

- [PNGOption \(\)](#)

Public Attributes

- bool [interlaced](#)
Whether to save the PNG as interlaced.
- unsigned int [compressionLevel](#)
Compression level (0-9).
- unsigned int [reserved](#) [16]
Reserved for future use.

15.160.1 Detailed Description

Options for saving PNG images.

15.160.2 Constructor & Destructor Documentation

15.160.2.1 PNGOption()

```
PNGOption ( ) [inline]
```

15.160.3 Member Data Documentation

15.160.3.1 [compressionLevel](#)

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

15.160.3.2 [interlaced](#)

```
bool interlaced
```

Whether to save the PNG as interlaced.

15.160.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

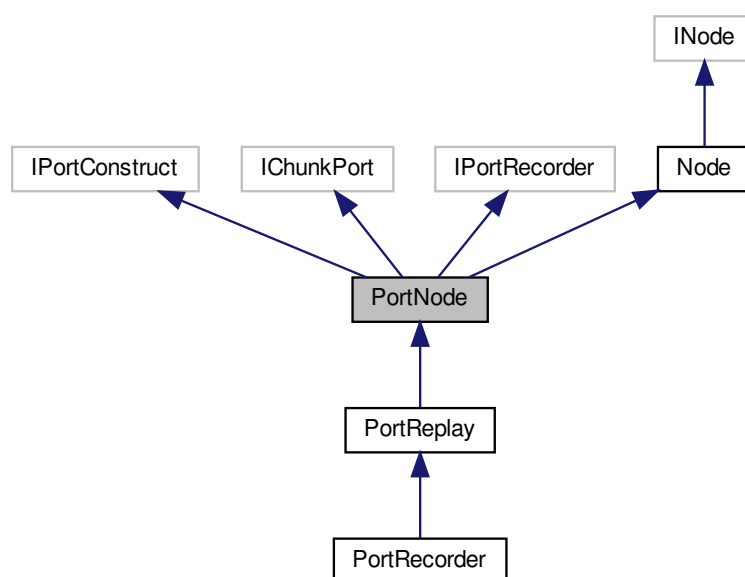
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

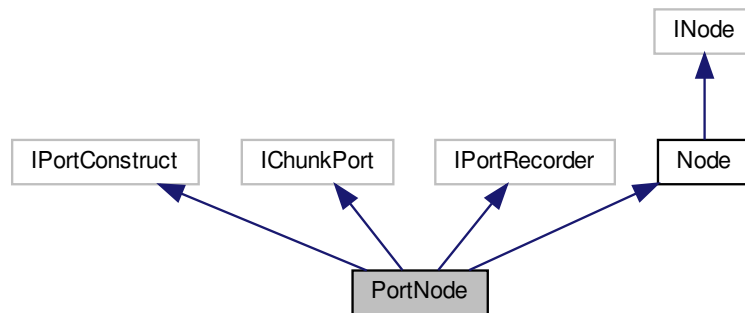
15.161 PortNode Class Reference

[Interface](#) for value properties.

Inheritance diagram for PortNode:



Collaboration diagram for PortNode:



Public Member Functions

- [PortNode](#) ()
Constructor.
- [PortNode](#) (std::shared_ptr< Node::NodeImpl > pValue)
constructor with [GenICam](#) IValue
- [~PortNode](#) ()
Destructor.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- void [SetPortImpl](#) (IPort *pPort)
Sets pointer the real port implementation; this function may called only once.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- virtual [Spinnaker::GenICam::gcstring](#) [GetChunkID](#) () const
Get the Id of the chunk the port should be attached to.
- virtual [EYesNo](#) [CacheChunkData](#) () const
Indicates if the chunk a adapter must hold a cached version of the chunk data.
- virtual void [StartRecording](#) (IPortWriteList *pPortRecorder)
Starts logging all WriteRegister commands to a list.
- virtual void [StopRecording](#) ()
Stops recording.
- virtual void [Replay](#) (IPortWriteList *pPortRecorder, bool [Invalidate](#)=true)
Sends the commands to the camera.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value
- virtual void [SetReference](#) (IPort *pBase)
overload SetReference for Value
- virtual void [SetReference](#) (IChunkPort *pBase)
overload SetReference for Value
- std::shared_ptr< Node::NodeImpl > [GetPortHandle](#) ()

Additional Inherited Members

15.161.1 Detailed Description

[Interface](#) for value properties.

15.161.2 Constructor & Destructor Documentation

15.161.2.1 PortNode() [1/2]

```
PortNode ( )
```

Constructor.

15.161.2.2 PortNode() [2/2]

```
PortNode (
    std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenlCam](#) IValue

15.161.2.3 ~PortNode()

```
~PortNode ( )
```

Destructor.

15.161.3 Member Function Documentation

15.161.3.1 CacheChunkData()

```
virtual EYesNo CacheChunkData ( ) const [virtual]
```

Indicates if the chunk a adapter must hold a cached version of the chunk data.

15.161.3.2 GetChunkID()

```
virtual Spinnaker::GenICam::gcstring GetChunkID ( ) const [virtual]
```

Get the Id of the chunk the port should be attached to.

15.161.3.3 GetPortHandle()

```
std::shared_ptr<Node::NodeImpl> GetPortHandle ( ) [inline]
```

15.161.3.4 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.161.3.5 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.161.3.6 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [virtual]
```

Sends the commands to the camera.

The default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.161.3.7 SetPortImpl()

```
void SetPortImpl (
    IPort * pPort )
```

Sets pointer the real port implementation; this function may called only once.

15.161.3.8 SetReference() [1/3]

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [Node](#).

15.161.3.9 SetReference() [2/3]

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.161.3.10 SetReference() [3/3]

```
virtual void SetReference (
    IChunkPort * pBase ) [virtual]
```

overload SetReference for Value

15.161.3.11 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

Starts logging all WriteRegister commands to a list.

Reimplemented in [PortRecorder](#).

15.161.3.12 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

Stops recording.

Reimplemented in [PortRecorder](#).

15.161.3.13 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented in [PortRecorder](#), and [PortReplay](#).

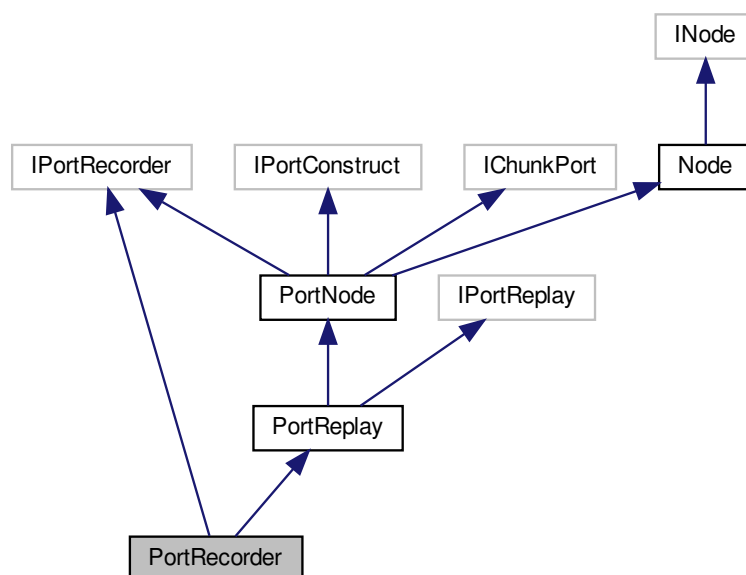
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortNode.h](#)

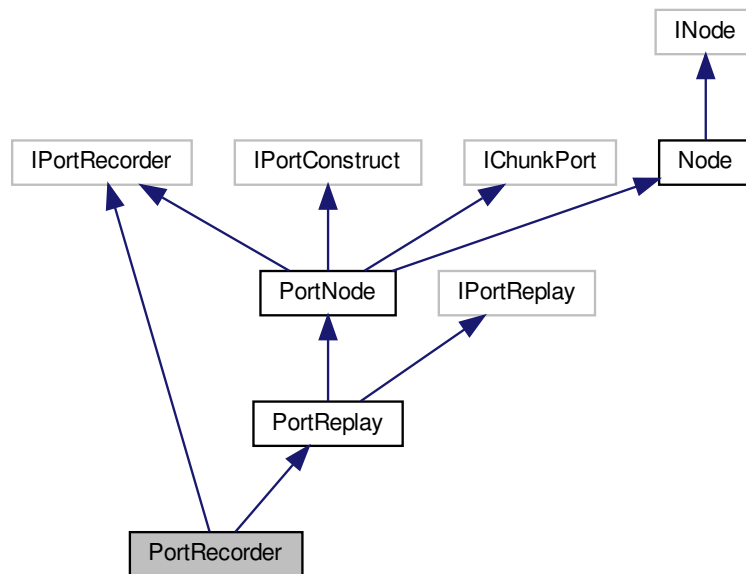
15.162 PortRecorder Class Reference

[Interface](#) for recording write commands on a port.

Inheritance diagram for PortRecorder:



Collaboration diagram for PortRecorder:



Public Member Functions

- [PortRecorder](#) ()
- virtual [~PortRecorder](#) ()
- virtual void [StartRecording](#) ([IPortWriteList](#) *pPortRecorder)
starts logging all WriteRegister commands to a list
- virtual void [StopRecording](#) ()
stops recording
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Get the access mode of the node.
- virtual void [SetReference](#) ([IPort](#) *pBase)
overload SetReference for Value
- virtual void [Replay](#) ([IPortWriteList](#) *pPortRecorder, bool [Invalidate](#)=true)
sends the commands to the camera.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.

Additional Inherited Members

15.162.1 Detailed Description

[Interface](#) for recording write commands on a port.

15.162.2 Constructor & Destructor Documentation

15.162.2.1 PortRecorder()

```
PortRecorder ( )
```

15.162.2.2 ~PortRecorder()

```
virtual ~PortRecorder ( ) [virtual]
```

15.162.3 Member Function Documentation

15.162.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

Reimplemented from [PortReplay](#).

15.162.3.2 Read()

```
virtual void Read (  
    void * pBuffer,  
    int64_t Address,  
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented from [PortReplay](#).

15.162.3.3 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [inline], [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented from [PortReplay](#).

15.162.3.4 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortReplay](#).

15.162.3.5 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

starts logging all WriteRegister commands to a list

Reimplemented from [PortNode](#).

15.162.3.6 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

stops recording

Reimplemented from [PortNode](#).

15.162.3.7 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented from [PortReplay](#).

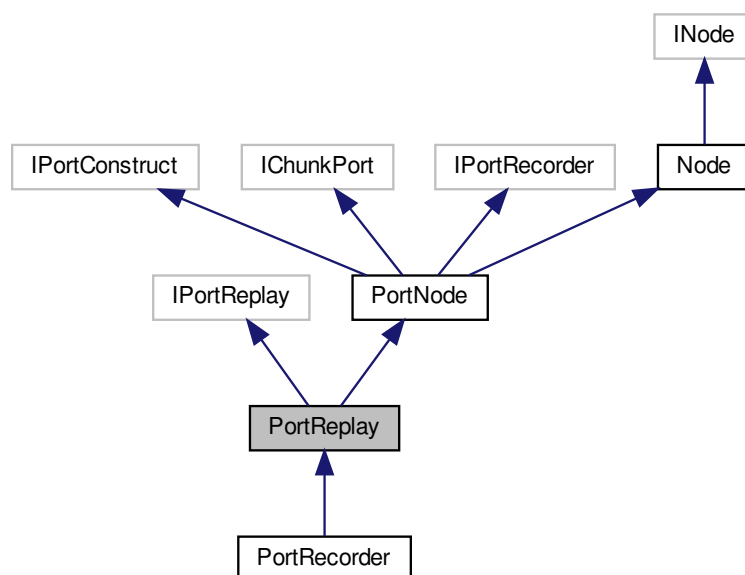
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortRecorder.h](#)

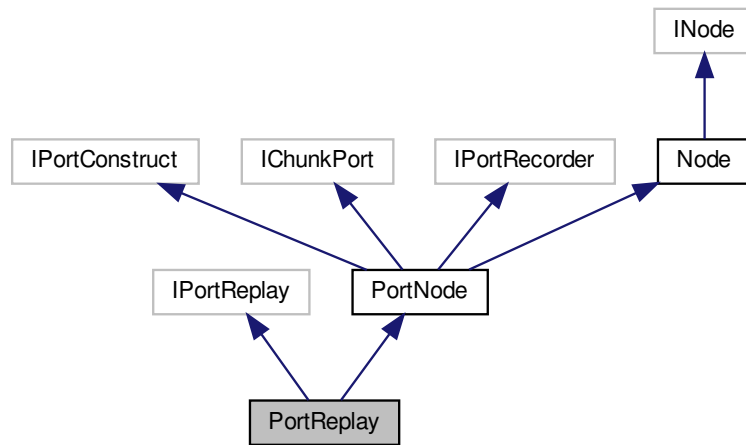
15.163 PortReplay Class Reference

[Interface](#) for replaying write commands on a port.

Inheritance diagram for PortReplay:



Collaboration diagram for PortReplay:



Public Member Functions

- [PortReplay](#) ()
- virtual [~PortReplay](#) ()
- virtual void [Replay](#) ([IPortWriteList](#) *pPortRecorder, bool [Invalidate](#)=true)
sends the commands to the camera.
- virtual void [SetReference](#) ([IPort](#) *pBase)
overload SetReference for Value
- void * [GetPortReplayHandle](#) ()
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Base interface overrides.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.

Additional Inherited Members

15.163.1 Detailed Description

[Interface](#) for replaying write commands on a port.

15.163.2 Constructor & Destructor Documentation

15.163.2.1 PortReplay()

`PortReplay ( )`

15.163.2.2 ~PortReplay()

`virtual ~PortReplay ( ) [virtual]`

15.163.3 Member Function Documentation

15.163.3.1 GetAccessMode()

`virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]`

Base interface overrides.

Get the access mode of the node

Reimplemented from [Node](#).

Reimplemented in [PortRecorder](#).

15.163.3.2 GetPortReplayHandle()

`void* GetPortReplayHandle ( )`

15.163.3.3 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

15.163.3.4 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

15.163.3.5 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

15.163.3.6 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

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

- include/SpinGenApi/[PortReplay.h](#)

15.164 PPMOption Struct Reference

Options for saving PPM images.

Public Member Functions

- [PPMOption](#) ()

Public Attributes

- bool [binaryFile](#)
Whether to save the PPM as a binary file.
- unsigned int [reserved](#) [16]
Reserved for future use.

15.164.1 Detailed Description

Options for saving PPM images.

15.164.2 Constructor & Destructor Documentation

15.164.2.1 PPMOption()

```
PPMOption ( ) [inline]
```

15.164.3 Member Data Documentation

15.164.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

15.164.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

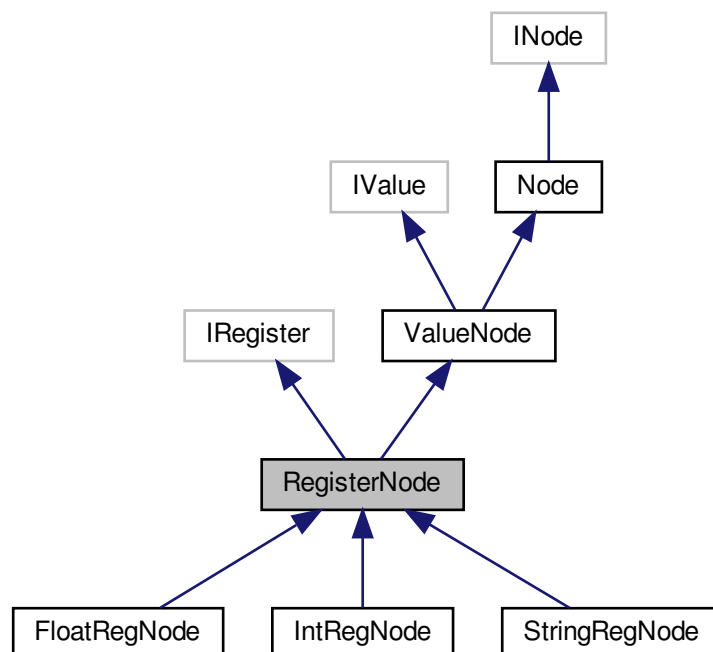
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

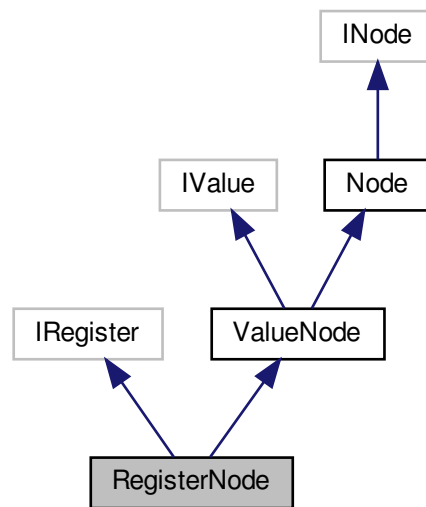
15.165 RegisterNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for RegisterNode:



Collaboration diagram for RegisterNode:



Public Member Functions

- [RegisterNode](#) ()
- [RegisterNode](#) (std::shared_ptr< Node::NodeImpl > pRegister)
- virtual [~RegisterNode](#) ()
- virtual void [Set](#) (const uint8_t *pBuffer, int64_t [Length](#), bool [Verify](#)=true)
Set the register's contents.
- virtual void [Get](#) (uint8_t *pBuffer, int64_t [Length](#), bool [Verify](#)=false, bool IgnoreCache=false)
Fills a buffer with the register's contents.
- virtual int64_t [GetLength](#) ()
Retrieves the Length of the register [Bytes].
- virtual int64_t [GetAddress](#) ()
Retrieves the Address of the register.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Register

Additional Inherited Members

15.165.1 Detailed Description

[Interface](#) for string properties.

15.165.2 Constructor & Destructor Documentation

15.165.2.1 RegisterNode() [1/2]

```
RegisterNode ( )
```

15.165.2.2 RegisterNode() [2/2]

```
RegisterNode (
    std::shared_ptr< Node::NodeImpl > pRegister )
```

15.165.2.3 ~RegisterNode()

```
virtual ~RegisterNode ( ) [virtual]
```

15.165.3 Member Function Documentation

15.165.3.1 Get()

```
virtual void Get (
    uint8_t * pBuffer,
    int64_t Length,
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Fills a buffer with the register's contents.

Parameters

<i>pBuffer</i>	The buffer receiving the data to read
<i>Length</i>	The number of bytes to retrieve
<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.165.3.2 GetAddress()

```
virtual int64_t GetAddress ( ) [virtual]
```

Retrieves the Address of the register.

15.165.3.3 GetLength()

```
virtual int64_t GetLength ( ) [virtual]
```

Retrieves the Length of the register [Bytes].

15.165.3.4 Set()

```
virtual void Set (
    const uint8_t * pBuffer,
    int64_t Length,
    bool Verify = true ) [virtual]
```

Set the register's contents.

Parameters

<i>pBuffer</i>	The buffer containing the data to set
<i>Length</i>	The number of bytes in pBuffer
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.165.3.5 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Register

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

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

- include/SpinGenApi/[RegisterNode.h](#)

15.166 SingleChunkData_t Struct Reference

Public Attributes

- uint64_t [ChunkID](#)
- ptrdiff_t [ChunkOffset](#)
- size_t [ChunkLength](#)

15.166.1 Member Data Documentation

15.166.1.1 ChunkID

uint64_t ChunkID

15.166.1.2 ChunkLength

size_t ChunkLength

15.166.1.3 ChunkOffset

ptrdiff_t ChunkOffset

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

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

15.167 SingleChunkDataStr_t Struct Reference

Public Attributes

- GenICam::gcstring [ChunkID](#)
- ptrdiff_t [ChunkOffset](#)
- size_t [ChunkLength](#)

15.167.1 Member Data Documentation

15.167.1.1 ChunkID

GenICam::gcstring ChunkID

15.167.1.2 ChunkLength

`size_t` ChunkLength

15.167.1.3 ChunkOffset

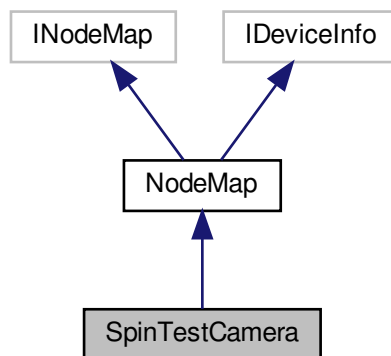
`ptrdiff_t` ChunkOffset

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

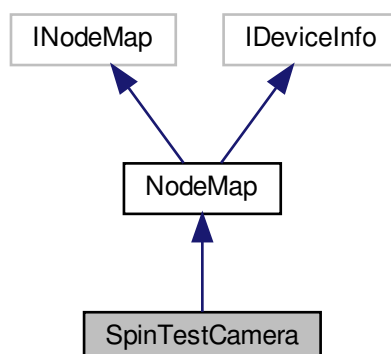
- `include/SpinGenApi/ChunkAdapterGeneric.h`

15.168 SpinTestCamera Class Reference

Inheritance diagram for SpinTestCamera:



Collaboration diagram for SpinTestCamera:



Additional Inherited Members

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

- include/SpinGenApi/[SpinTestCamera.h](#)

15.169 SpinVideo Class Reference

Provides the functionality for the user to record images to an AVI/MP4 file.

Public Member Functions

- [SpinVideo](#) ()
Default constructor.
- virtual [~SpinVideo](#) ()
Default destructor.
- virtual void [Open](#) (const char *pFileName, [AVIOption](#) &pOption)
Open an video file in preparation for writing Images to disk.
- virtual void [Open](#) (const char *pFileName, [MJPGOption](#) &pOption)
Open an MJPEG video file in preparation for writing Images to disk.
- virtual void [Open](#) (const char *pFileName, [H264Option](#) &pOption)
Open an H264 MP4 video file in preparation for writing Images to disk.
- virtual void [Append](#) ([ImagePtr](#) plmage)
Append an image to the video file.
- virtual void [Close](#) ()
Close the video file.
- virtual void [SetMaximumFileSize](#) (unsigned int size)
Set the maximum file size (in megabytes) of a AVI/MP4 file.

15.169.1 Detailed Description

Provides the functionality for the user to record images to an AVI/MP4 file.

15.169.2 Constructor & Destructor Documentation

15.169.2.1 SpinVideo()

[SpinVideo](#) ()

Default constructor.

15.169.2.2 ~SpinVideo()

```
virtual ~SpinVideo ( ) [virtual]
```

Default destructor.

15.169.3 Member Function Documentation

15.169.3.1 Append()

```
virtual void Append (
    ImagePtr pImage ) [virtual]
```

Append an image to the video file.

When using the H264 encoder, several images are required to be appended before the encoder is able to output the first encoded frame.

Parameters

<i>pImage</i>	The image to append.

15.169.3.2 Close()

```
virtual void Close ( ) [virtual]
```

Close the video file.

This function will throw an exception when the H264 encoder was unable to output any encoded frames, in which case the output video should be considered invalid.

See also

[Open\(\)](#)
[Append\(ImagePtr pImage\)](#)

15.169.3.3 Open() [1/3]

```
virtual void Open (
    const char * pFileName,
    AVIOption & pOption ) [virtual]
```

Open an video file in preparation for writing Images to disk.

The size of video files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the video file.
<i>pOption</i>	Options to apply to the video file.

See also

[Close\(\)](#)

15.169.3.4 Open() [2/3]

```
virtual void Open (  
    const char * pFileName,  
    MJPGOption & pOption ) [virtual]
```

Open an MJPEG video file in preparation for writing Images to disk.

The size of video files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the video file.
<i>pOption</i>	MJPEG options to apply to the video file.

See also

[Close\(\)](#)
[MJPGOption](#)

15.169.3.5 Open() [3/3]

```
virtual void Open (  
    const char * pFileName,  
    H264Option & pOption ) [virtual]
```

Open an H264 MP4 video file in preparation for writing Images to disk.

The size of MP4 files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the MP4 video file.
<i>pOption</i>	H264 options to apply to the MP4 video file.

See also

[Close\(\)](#)
[H264Option](#)

15.169.3.6 SetMaximumFileSize()

```
virtual void SetMaximumFileSize (  
    unsigned int size ) [virtual]
```

Set the maximum file size (in megabytes) of a AVI/MP4 file.

A new video file is created automatically when file size limit is reached. Setting a maximum size of 0 indicates no limit on file size.

Parameters

<i>size</i>	The maximum video file size in MB.

See also

[Append\(ImagePtr pImage\)](#)

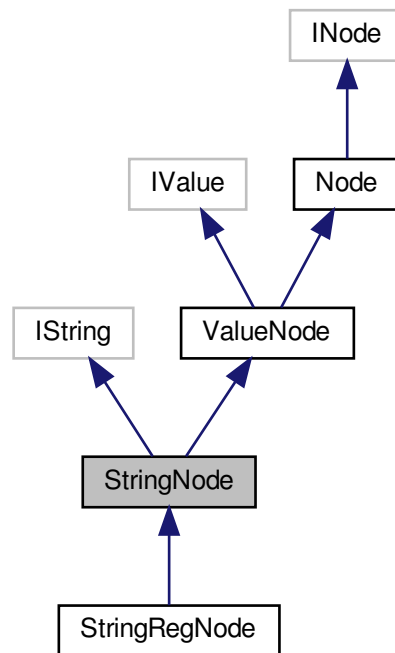
The documentation for this class was generated from the following file:

- [include/SpinVideo.h](#)

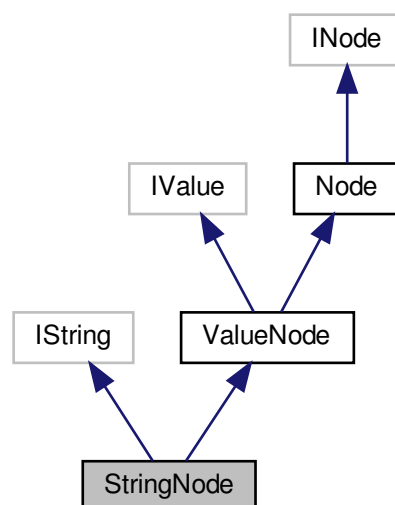
15.170 StringNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for StringNode:



Collaboration diagram for StringNode:



Public Member Functions

- [StringNode](#) ()
- [StringNode](#) (std::shared_ptr< Node::NodeImpl > pString)
- virtual [~StringNode](#) ()
- virtual void [SetValue](#) (const [GenlCam::gcstring](#) &Value, bool [Verify](#)=true)
Set node value.
- virtual [IString](#) & [operator=](#) (const [GenlCam::gcstring](#) &Value)
Set node value.
- virtual [GenlCam::gcstring](#) [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get node value.
- virtual [GenlCam::gcstring](#) [operator\(\)](#) ()
Get node value.
- virtual [GenlCam::gcstring](#) [operator*](#) ()
Get node value.
- virtual int64_t [GetMaxLength](#) ()
Retrieves the maximum length of the string in bytes.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Value

Additional Inherited Members

15.170.1 Detailed Description

[Interface](#) for string properties.

15.170.2 Constructor & Destructor Documentation

15.170.2.1 [StringNode](#)() [1/2]

[StringNode](#) ()

15.170.2.2 [StringNode](#)() [2/2]

[StringNode](#) (
std::shared_ptr< Node::NodeImpl > pString)

15.170.2.3 [~StringNode](#)()

virtual [~StringNode](#) () [virtual]

15.170.3 Member Function Documentation

15.170.3.1 GetMaxLength()

```
virtual int64_t GetMaxLength ( ) [virtual]
```

Retrieves the maximum length of the string in bytes.

15.170.3.2 GetValue()

```
virtual GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.170.3.3 operator()

```
virtual GenICam::gcstring operator() ( ) [virtual]
```

Get node value.

15.170.3.4 operator*()

```
virtual GenICam::gcstring operator* ( ) [virtual]
```

Get node value.

15.170.3.5 operator=()

```
virtual IString& operator= (
    const GenICam::gcstring & Value ) [virtual]
```

Set node value.

15.170.3.6 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#).

15.170.3.7 SetValue()

```
virtual void SetValue (
    const GenICam::gcstring & Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

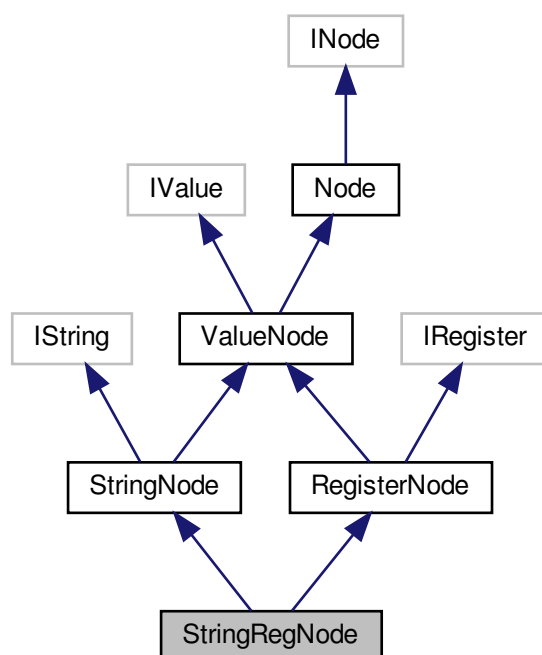
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringNode.h](#)

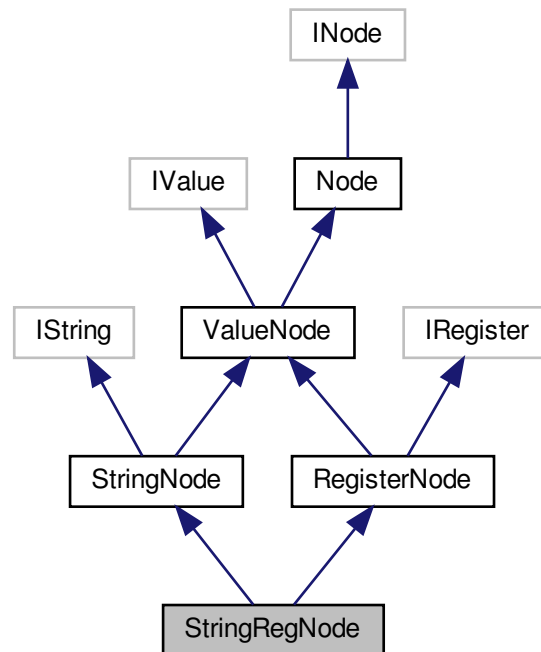
15.171 StringRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for StringRegNode:



Collaboration diagram for StringRegNode:



Public Member Functions

- [StringRegNode](#) ()
- [StringRegNode](#) (std::shared_ptr< Node::NodeImpl > pString)
- virtual [~StringRegNode](#) ()
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.171.1 Detailed Description

[Interface](#) for string properties.

15.171.2 Constructor & Destructor Documentation

15.171.2.1 StringRegNode() [1/2]

```
StringRegNode ( )
```

15.171.2.2 StringRegNode() [2/2]

```
StringRegNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

15.171.2.3 ~StringRegNode()

```
virtual ~StringRegNode ( ) [virtual]
```

15.171.3 Member Function Documentation

15.171.3.1 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [RegisterNode](#).

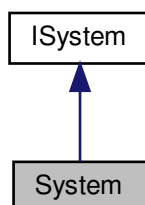
The documentation for this class was generated from the following file:

- [include/SpinGenApi/StringRegNode.h](#)

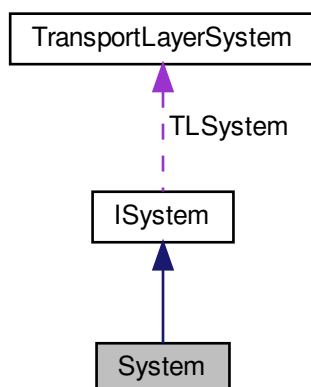
15.172 System Class Reference

The system object is used to retrieve the list of interfaces and cameras available.

Inheritance diagram for System:



Collaboration diagram for System:



Public Member Functions

- virtual `~System ()`
Default destructor.
- virtual void `ReleaseInstance ()`
This call releases the referenced instance of the `System` Singleton.
- virtual `InterfaceList GetInterfaces (bool updateInterface=true)`
Returns a list of interfaces available on the system.
- virtual void `UpdateInterfaceList ()`
Updates the list of interfaces on the system.

- virtual [CameraList GetCameras](#) (bool updateInterfaces=true, bool updateCameras=true)
Returns a list of cameras that are available on the system.
- virtual bool [UpdateCameras](#) (bool updateInterfaces=true)
Updates the list of cameras on the system.
- void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)
Registers an event handler for the system.
- void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)
Unregisters an event handler for the system.
- virtual void [RegisterInterfaceEventHandler](#) ([EventHandler](#) &evtHandlerToRegister, bool updateInterface=true)
Registers event handlers for all available interfaces that are found on the system. If new interfaces are detected by the system after [RegisterInterfaceEventHandler\(\)](#) is called, those interfaces will be automatically registered with this event.
- void [UnregisterInterfaceEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)
Unregisters event handlers for all available interfaces that are found on the system.
- virtual void [RegisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)
Registers a logging event.
- virtual void [UnregisterAllLoggingEventHandlers](#) ()
Unregisters all previously registered logging events.
- virtual void [UnregisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)
Unregisters a logging event.
- virtual void [SetLoggingEventPriorityLevel](#) ([SpinnakerLogLevel](#) level)
Sets a threshold priority level for logging event.
- virtual [SpinnakerLogLevel GetLoggingEventPriorityLevel](#) ()
Retrieves the current logging event priority level.
- virtual bool [IsInUse](#) ()
Checks if the system is in use by any interface or camera objects.
- virtual void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL)
Broadcast an Action Command to all devices on system.
- virtual const [LibraryVersion GetLibraryVersion](#) ()
Get current library version of [Spinnaker](#).
- virtual [GenApi::INodeMap & GetTLNodeMap](#) () const
Gets a reference to the system node map.

Static Public Member Functions

- static [SystemPtr GetInstance](#) ()
Returns a pointer to a Singleton instance of a [System](#) object.

Protected Member Functions

- [System](#) ()
Default constructor.

Additional Inherited Members

15.172.1 Detailed Description

The system object is used to retrieve the list of interfaces and cameras available.

15.172.2 Constructor & Destructor Documentation

15.172.2.1 ~System()

```
virtual ~System ( ) [virtual]
```

Default destructor.

15.172.2.2 System()

```
System ( ) [protected]
```

Default constructor.

15.172.3 Member Function Documentation

15.172.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateInterfaces = true,
    bool updateCameras = true ) [virtual]
```

Returns a list of cameras that are available on the system.

This call returns both GigE Vision and Usb3 Vision cameras from all interfaces. The camera list object will reference count the cameras it returns. It is important that the camera list is destroyed or is cleared before calling system->[ReleaseInstance\(\)](#) or else the call to system->[ReleaseInstance\(\)](#) will result in an error message thrown that a reference to the camera is still held.

See also

[ReleaseInstance\(\)](#)
[CameraList::Clear\(\)](#)

Parameters

<i>updateInterfaces</i>	Determines whether or not updateInterfaceList() is called before getting cameras from available interfaces on the system
<i>updateCameras</i>	Determines whether or not UpdateCameras() is called before getting cameras from available interfaces on the system

Returns

An [CameraList](#) object that contains a list of all cameras.

Implements [ISystem](#).

15.172.3.2 GetInstance()

```
static SystemPtr GetInstance ( ) [static]
```

Returns a pointer to a Singleton instance of a [System](#) object.

The [System](#) object may be used to get cameras or interfaces. When an application is done using the cameras it is necessary to free the [System](#) by calling [ReleaseInstance\(\)](#).

See also

[ReleaseInstance\(\)](#)

Returns

A const ref to a system object.

15.172.3.3 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [virtual]
```

Returns a list of interfaces available on the system.

This call returns GigE and Usb2 and Usb3 interfaces. Note that on MacOS only active GigE interfaces will be stored in the returned [InterfaceList](#).

See also

[UpdateInterfaceList\(\)](#)

Parameters

<i>updateInterface</i>	Determines whether or not UpdateInterfaceList() is called before getting available interfaces

Returns

An [InterfaceList](#) object that contains a list of all interfaces.

Implements [ISystem](#).

15.172.3.4 GetLibraryVersion()

```
virtual const LibraryVersion GetLibraryVersion ( ) [virtual]
```

Get current library version of [Spinnaker](#).

Returns

A struct containing the current version of [Spinnaker](#) (major, minor, type, build).

Implements [ISystem](#).

15.172.3.5 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( ) [virtual]
```

Retrieves the current logging event priority level.

[Spinnaker](#) uses five levels of logging:

- Error - failures that are non-recoverable without user intervention.
- Warning - failures that are recoverable without user intervention.
- Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.
- Info - information about recurring events that are generated regularly such as information on individual images.
- Debug - information that can be used to troubleshoot the system.

See also

[SpinnakerLogLevel](#)

Returns

Level The threshold level

Implements [ISystem](#).

15.172.3.6 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [virtual]
```

Gets a reference to the system node map.

The system must be initialized by a call to [System::GetInstance\(\)](#) first before a node map reference can be successfully acquired.

Returns

A reference to the [System](#) [INodeMap](#).

Implements [ISystem](#).

15.172.3.7 IsInUse()

```
virtual bool IsInUse ( ) [virtual]
```

Checks if the system is in use by any interface or camera objects.

Returns

Returns true if the system is in use and false otherwise.

Implements [ISystem](#).

15.172.3.8 RegisterEventHandler()

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [virtual]
```

Registers an event handler for the system.

See also

[SystemEventHandler](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the system

Implements [ISystem](#).

15.172.3.9 RegisterInterfaceEventHandler()

```
virtual void RegisterInterfaceEventHandler (
    EventHandler & evtHandlerToRegister,
    bool updateInterface = true ) [virtual]
```

Registers event handlers for all available interfaces that are found on the system. If new interfaces are detected by the system after [RegisterInterfaceEventHandler\(\)](#) is called, those interfaces will be automatically registered with this event.

Note that only GEV interface arrivals and removals are currently handled.

See also

[InterfaceEventHandler](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the available interfaces
<i>updateInterface</i>	Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system

Implements [ISystem](#).

15.172.3.10 RegisterLoggingEventHandler()

```
virtual void RegisterLoggingEventHandler (
    LoggingEventHandler & handler ) [virtual]
```

Registers a logging event.

Parameters

<i>handler</i>	The logging event handler to register

Implements [ISystem](#).

15.172.3.11 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [virtual]
```

This call releases the referenced instance of the [System](#) Singleton.

After successfully releasing the [System](#) instance, the pointer returned by [GetInstance\(\)](#) will be invalid. Once the final [System](#) instance is released, all remaining [Spinnaker](#) resources will be released. If the final [System](#) instance is released while an interface or camera reference is still held, this function will throw an error of type SPINNAKE←
R_ERR_RESOURCE_IN_USE.

See also

[Error](#)
[GetInstance\(\)](#)

Implements [ISystem](#).

15.172.3.12 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) [virtual]
```

Broadcast an Action Command to all devices on system.

Parameters

<i>deviceKey</i>	The Action Command's device key
<i>groupKey</i>	The Action Command's group key
<i>groupMask</i>	The Action Command's group mask
<i>actionTime</i>	(Optional) Time when to assert a future action. Zero means immediate action.
<i>pResultSize</i>	(Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results. If this parameter is 0 or NULL, the function will return as soon as the command has been broadcasted.
<i>results</i>	(Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

Implements [ISystem](#).

15.172.3.13 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [virtual]
```

Sets a threshold priority level for logging event.

Logging events below such level will not trigger callbacks.

[Spinnaker](#) uses five levels of logging:

- Error - failures that are non-recoverable without user intervention.
- Warning - failures that are recoverable without user intervention.
- Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.
- Info - information about recurring events that are generated regularly such as information on individual images.
- Debug - information that can be used to troubleshoot the system.

See also

[SpinnakerLogLevel](#)

Parameters

<i>level</i>	The threshold level

Implements [ISystem](#).

15.172.3.14 UnregisterAllLoggingEventHandlers()

```
virtual void UnregisterAllLoggingEventHandlers ( ) [virtual]
```

Unregisters all previously registered logging events.

Implements [ISystem](#).

15.172.3.15 UnregisterEventHandler()

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters an event handler for the system.

See also

[SystemEventHandler](#)

Parameters

<i>evtHandlerToUnregister</i>	The event handler to unregister from the system

Implements [ISystem](#).

15.172.3.16 UnregisterInterfaceEventHandler()

```
void UnregisterInterfaceEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters event handlers for all available interfaces that are found on the system.

See also

[InterfaceEventHandler](#)

Parameters

<i>evtHandlerToUnregister</i>	The event handler to unregister from the available interfaces

Implements [ISystem](#).

15.172.3.17 UnregisterLoggingEventHandler()

```
virtual void UnregisterLoggingEventHandler (
    LoggingEventHandler & handler ) [virtual]
```

Unregisters a logging event.

Parameters

<i>handler</i>	The logging event handler to unregister

Implements [ISystem](#).

15.172.3.18 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [virtual]
```

Updates the list of cameras on the system.

Note that [System::GetCameras\(\)](#) internally calls [UpdateCameras\(\)](#) for each interface it enumerates. If the list changed between this call and the last time [UpdateCameras](#) was called then the return value will be true, otherwise it is false.

See also

[GetCameras\(\)](#)

Parameters

<i>updateInterfaces</i>	Determines whether or not UpdateInterfaceList() is called before updating cameras for available interfaces on the system

Returns

True if cameras changed on interface and false otherwise.

Implements [ISystem](#).

15.172.3.19 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [virtual]
```

Updates the list of interfaces on the system.

If desired, local copies of [InterfaceList](#) should be updated by calling [GetInterfaces](#).

See also

[GetInterfaces\(\)](#)

Implements [ISystem](#).

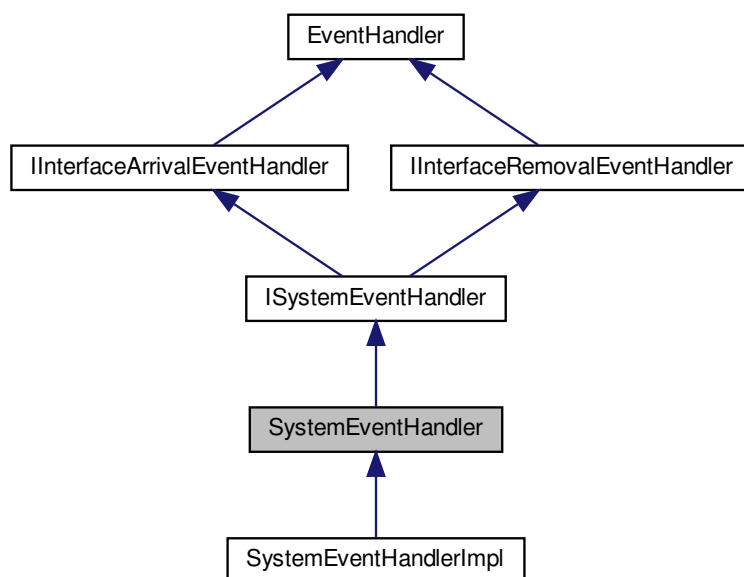
The documentation for this class was generated from the following file:

- include/[System.h](#)

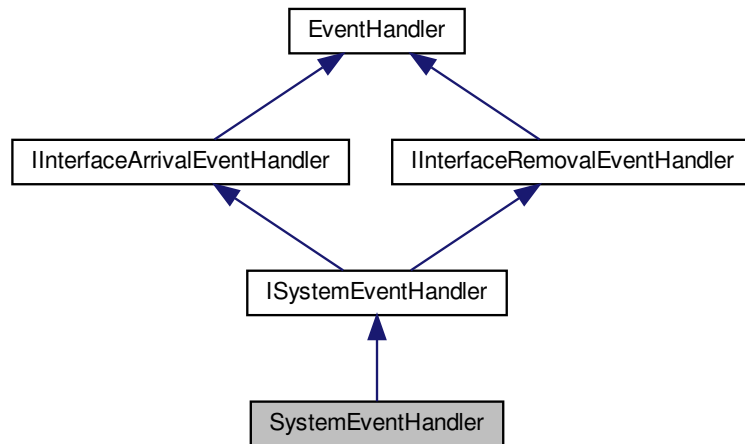
15.173 SystemEventHandler Class Reference

A handler to interface arrival and removal events on the system.

Inheritance diagram for SystemEventHandler:



Collaboration diagram for SystemEventHandler:



Public Member Functions

- [SystemEventHandler](#) ()
Default constructor.
- virtual [~SystemEventHandler](#) ()
Virtual destructor.
- virtual void [OnInterfaceArrival](#) (std::string interfaceID)=0
Interface arrival event callback.
- virtual void [OnInterfaceRemoval](#) (std::string interfaceID)=0
Interface removal event callback.

Protected Member Functions

- [SystemEventHandler](#) & [operator=](#) (const [SystemEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.173.1 Detailed Description

A handler to interface arrival and removal events on the system.

Note that only GEV interface arrivals and removals are currently handled.

15.173.2 Constructor & Destructor Documentation

15.173.2.1 SystemEventHandler()

```
SystemEventHandler ( )
```

Default constructor.

15.173.2.2 ~SystemEventHandler()

```
virtual ~SystemEventHandler ( ) [virtual]
```

Virtual destructor.

15.173.3 Member Function Documentation

15.173.3.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    std::string interfaceID ) [pure virtual]
```

[Interface](#) arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

<i>interfaceID</i>	The ID of the arrived interface

Implements [ISystemEventHandler](#).

Implemented in [SystemEventHandlerImpl](#).

15.173.3.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

[Interface](#) removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

<i>interfaceID</i>	The ID of the removed interface

Implements [ISystemEventHandler](#).

Implemented in [SystemEventHandlerImpl](#).

15.173.3.3 operator=()

```
SystemEventHandler& operator= (  
    const SystemEventHandler & ) [protected]
```

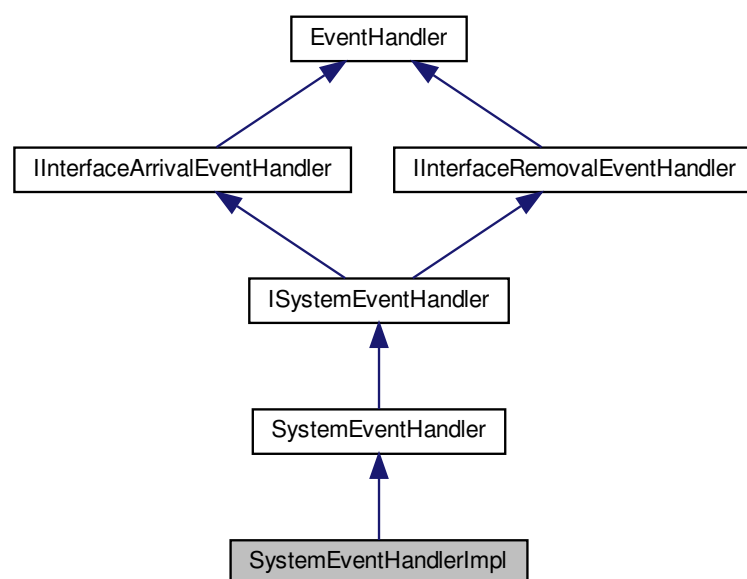
Assignment operator.

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

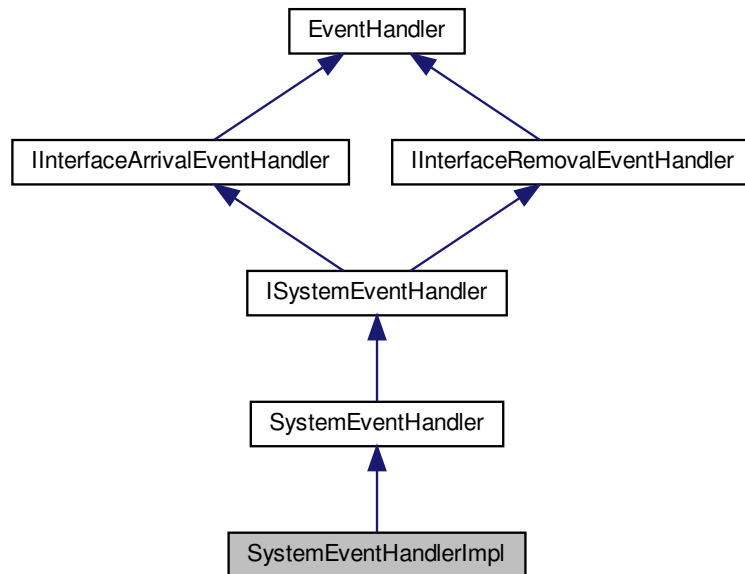
- include/[SystemEventHandler.h](#)

15.174 SystemEventHandlerImpl Class Reference

Inheritance diagram for SystemEventHandlerImpl:



Collaboration diagram for SystemEventHandlerImpl:



Public Member Functions

- [SystemEventHandlerImpl](#) ([SystemPtr](#) system)
- [~SystemEventHandlerImpl](#) ()
- void [LockEventHandlerMutex](#) ()
- void [UnlockEventHandlerMutex](#) ()
- void [OnInterfaceArrival](#) (std::string interfaceID)
Interface arrival event callback.
- void [OnInterfaceRemoval](#) (std::string interfaceID)
Interface removal event callback.
- void [RegisterInterfaceEventToSystem](#) ()
- void [UnregisterInterfaceEventFromSystem](#) ()
- void [RegisterAllInterfaceEvents](#) ()
- void [UnregisterAllInterfaceEvents](#) ()

Additional Inherited Members

15.174.1 Constructor & Destructor Documentation

15.174.1.1 SystemEventHandlerImpl()

```

SystemEventHandlerImpl (
    SystemPtr system ) [inline]

```

15.174.1.2 ~SystemEventHandlerImpl()

```
~SystemEventHandlerImpl ( ) [inline]
```

15.174.2 Member Function Documentation

15.174.2.1 LockEventHandlerMutex()

```
void LockEventHandlerMutex ( ) [inline]
```

15.174.2.2 OnInterfaceArrival()

```
void OnInterfaceArrival (
    std::string interfaceID ) [inline], [virtual]
```

Interface arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

<i>interfaceID</i>	The ID of the arrived interface

Implements [SystemEventHandler](#).

15.174.2.3 OnInterfaceRemoval()

```
void OnInterfaceRemoval (
    std::string interfaceID ) [inline], [virtual]
```

Interface removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

<i>interfaceID</i>	The ID of the removed interface

Implements [SystemEventHandler](#).

15.174.2.4 RegisterAllInterfaceEvents()

```
void RegisterAllInterfaceEvents ( ) [inline]
```

15.174.2.5 RegisterInterfaceEventToSystem()

```
void RegisterInterfaceEventToSystem ( ) [inline]
```

15.174.2.6 UnlockEventHandlerMutex()

```
void UnlockEventHandlerMutex ( ) [inline]
```

15.174.2.7 UnregisterAllInterfaceEvents()

```
void UnregisterAllInterfaceEvents ( ) [inline]
```

15.174.2.8 UnregisterInterfaceEventFromSystem()

```
void UnregisterInterfaceEventFromSystem ( ) [inline]
```

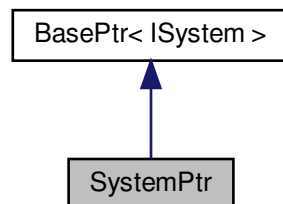
The documentation for this class was generated from the following file:

- src/EnumerationEvents/[EnumerationEvents.cpp](#)

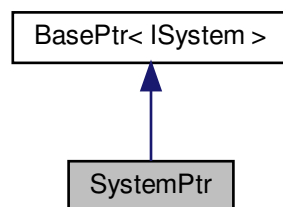
15.175 SystemPtr Class Reference

A reference tracked pointer to a system object.

Inheritance diagram for SystemPtr:



Collaboration diagram for SystemPtr:



Public Member Functions

- [SystemPtr](#) ()
Default constructor.
- [SystemPtr](#) (const int)
Copy constructor.
- [SystemPtr](#) (const long)
Copy constructor.
- [SystemPtr](#) (const std::nullptr_t)
Copy constructor.
- virtual [~SystemPtr](#) (void)
Virtual destructor.

Additional Inherited Members

15.175.1 Detailed Description

A reference tracked pointer to a system object.

15.175.2 Constructor & Destructor Documentation

15.175.2.1 SystemPtr() [1/4]

```
SystemPtr ( ) [inline]
```

Default constructor.

15.175.2.2 SystemPtr() [2/4]

```
SystemPtr (
    const int ) [inline]
```

Copy constructor.

15.175.2.3 SystemPtr() [3/4]

```
SystemPtr (
    const long ) [inline]
```

Copy constructor.

15.175.2.4 SystemPtr() [4/4]

```
SystemPtr (
    const std::nullptr_t ) [inline]
```

Copy constructor.

15.175.2.5 ~SystemPtr()

```
virtual ~SystemPtr (  
    void ) [inline], [virtual]
```

Virtual destructor.

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

- include/SystemPtr.h

15.176 TIFFOption Struct Reference

Options for saving TIFF images.

Public Types

- enum [CompressionMethod](#) {
 NONE = 1,
 PACKBITS,
 DEFLATE,
 ADOBE_DEFLATE,
 CCITTFAX3,
 CCITTFAX4,
 LZW,
 JPEG }

Public Member Functions

- [TIFFOption](#) ()

Public Attributes

- [CompressionMethod](#) *compression*
 Compression method to use for encoding TIFF images.
- unsigned int [reserved](#) [16]
 Reserved for future use.

15.176.1 Detailed Description

Options for saving TIFF images.

15.176.2 Member Enumeration Documentation

15.176.2.1 CompressionMethod

```
enum CompressionMethod
```

Enumerator

NONE	Save without any compression.
PACKBITS	Save using PACKBITS compression.
DEFLATE	Save using DEFLATE compression (ZLIB compression).
ADOBE_DEFLATE	Save using ADOBE DEFLATE compression.
CCITTFAX3	Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
CCITTFAX4	Save using CCITT Group 4 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
LZW	Save using LZW compression.
JPEG	Save using JPEG compression. This is only valid for 8-bit greyscale and 24-bit only. Default to LZW for other bit depths.

15.176.3 Constructor & Destructor Documentation

15.176.3.1 TIFFOption()

```
TIFFOption ( ) [inline]
```

15.176.4 Member Data Documentation

15.176.4.1 compression

```
CompressionMethod compression
```

Compression method to use for encoding TIFF images.

15.176.4.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

15.177 TransportLayerDevice Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerDevice](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerDevice](#) ()

Public Attributes

- [GenApi::IString](#) & [DeviceID](#)
Description: Interface-wide unique identifier of this device.
- [GenApi::IString](#) & [DeviceSerialNumber](#)
Description: Serial number of the remote device.
- [GenApi::IString](#) & [DeviceVendorName](#)
Description: Name of the remote device vendor.
- [GenApi::IString](#) & [DeviceModelName](#)
Description: Name of the remote device model.
- [GenApi::IEnumerationT](#) < [DeviceTypeEnum](#) > & [DeviceType](#)
Description: Transport layer type of the device.
- [GenApi::IString](#) & [DeviceDisplayName](#)
Description: User readable name of the device.
- [GenApi::IEnumerationT](#) < [DeviceAccessStatusEnum](#) > & [DeviceAccessStatus](#)
Description: Gets the access status the transport layer Producer has on the device.
- [GenApi::IString](#) & [DeviceVersion](#)
Description: Version of the device.
- [GenApi::IString](#) & [DeviceUserID](#)
Description: User Defined Name.
- [GenApi::IString](#) & [DeviceDriverVersion](#)
Description: Version of the device driver.
- [GenApi::IBoolean](#) & [DeviceIsUpdater](#)
Description: Indicates whether the device is in updater mode.
- [GenApi::IEnumerationT](#) < [GevCCPEnum](#) > & [GevCCP](#)
Description: Controls the device access privilege of an application.
- [GenApi::IEnumerationT](#) < [GUIXMLLocationEnum](#) > & [GUIXMLLocation](#)
Description: Sets the location to load GUI XML.
- [GenApi::IString](#) & [GUIXMLPath](#)
Description: GUI XML Path.
- [GenApi::IEnumerationT](#) < [GenICamXMLLocationEnum](#) > & [GenICamXMLLocation](#)
Description: Sets the location to load [GenICam](#) XML.
- [GenApi::IString](#) & [GenICamXMLPath](#)
Description: [GenICam](#) XML Path.
- [GenApi::Integer](#) & [GevDeviceIPAddress](#)
Description: Current IP address of the GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceSubnetMask](#)
Description: Current subnet mask of the GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceMACAddress](#)
Description: 48-bit MAC address of the GVCP interface of the selected remote device.

- [GenApi::Integer](#) & [GevDeviceGateway](#)
Description: Current gateway IP address of the GVCP interface of the remote device.
- [GenApi::Integer](#) & [DeviceLinkSpeed](#)
Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.
- [GenApi::Integer](#) & [GevVersionMajor](#)
Description: Major version of the specification.
- [GenApi::Integer](#) & [GevVersionMinor](#)
Description: Minor version of the specification.
- [GenApi::Boolean](#) & [GevDeviceModelsBigEndian](#)
Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).
- [GenApi::Integer](#) & [GevDeviceReadAndWriteTimeout](#)
Description: The timeout in us for read/write operations to the camera.
- [GenApi::Integer](#) & [GevDeviceMaximumRetryCount](#)
Description: Maximum number of times to retry a read/write operation.
- [GenApi::Integer](#) & [GevDevicePort](#)
Description: Current IP port of the GVCP interface of the selected remote device.
- [GenApi::Command](#) & [GevDeviceDiscoverMaximumPacketSize](#)
Description: Discovers and updates the maximum packet size that can be safely used by the device on the current interface.
- [GenApi::Integer](#) & [GevDeviceMaximumPacketSize](#)
Description: The maximum packet size that can be safely used by the device on the current interface.
- [GenApi::Boolean](#) & [GevDeviceWrongSubnet](#)
Description: Indicates whether the device is on the wrong subnet.
- [GenApi::Command](#) & [GevDeviceAutoForceIP](#)
Description: Forces the camera to be on the same subnet as its corresponding interface.
- [GenApi::Command](#) & [GevDeviceForceIP](#)
Description: Apply the force IP settings ([GevDeviceForceIPAddress](#), [GevDeviceForceSubnetMask](#) and [GevDeviceForceGateway](#)) in the Device using ForceIP command.
- [GenApi::Integer](#) & [GevDeviceForceIPAddress](#)
Description: Static IP address to set for the GVCP interface of the remote device.
- [GenApi::Integer](#) & [GevDeviceForceSubnetMask](#)
Description: Static subnet mask to set for GVCP interface of the remote device.
- [GenApi::Integer](#) & [GevDeviceForceGateway](#)
Description: Static gateway IP address to set for the GVCP interface of the remote device.
- [GenApi::Boolean](#) & [DeviceMulticastMonitorMode](#)
Description: Controls and indicates if the device is operating in as a Multicast Monitor.
- [GenApi::EnumerationT< DeviceEndiannessMechanismEnum >](#) & [DeviceEndiannessMechanism](#)
Description: Identifies the endianness handling mode.
- [GenApi::String](#) & [DeviceInstanceId](#)
Description: Visibility: Invisible.
- [GenApi::String](#) & [DeviceLocation](#)
Description: Device Location.
- [GenApi::EnumerationT< DeviceCurrentSpeedEnum >](#) & [DeviceCurrentSpeed](#)
Description: The USB Speed that the device is currently operating at.
- [GenApi::Boolean](#) & [DeviceU3VProtocol](#)
Description: Indicates whether the device is communicating in U3V Protocol.
- [GenApi::String](#) & [DevicePortId](#)
Description: Device Port ID.

Protected Member Functions

- [TransportLayerDevice](#) ()

Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameraInternal](#)

15.177.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.177.2 Constructor & Destructor Documentation

15.177.2.1 TransportLayerDevice() [1/2]

```
TransportLayerDevice (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.177.2.2 ~TransportLayerDevice()

```
~TransportLayerDevice ( )
```

15.177.2.3 TransportLayerDevice() [2/2]

```
TransportLayerDevice ( ) [protected]
```

15.177.3 Friends And Related Function Documentation

15.177.3.1 CameraBase

```
friend class CameraBase [friend]
```

15.177.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

15.177.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

15.177.4 Member Data Documentation

15.177.4.1 DeviceAccessStatus

```
GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus
```

Description: Gets the access status the transport layer Producer has on the device.

Visibility: Beginner

15.177.4.2 DeviceCurrentSpeed

```
GenApi::IEnumerationT<DeviceCurrentSpeedEnum>& DeviceCurrentSpeed
```

Description: The USB Speed that the device is currently operating at.

Visibility: Expert

15.177.4.3 DeviceDisplayName

```
GenApi::IString& DeviceDisplayName
```

Description: User readable name of the device.

If this is not defined in the device this should be "VENDOR MODEL (ID)". Visibility: Expert

15.177.4.4 DeviceDriverVersion

```
GenApi::IString& DeviceDriverVersion
```

Description: Version of the device driver.

Visibility: Expert

15.177.4.5 DeviceEndiannessMechanism

`GenApi::IEnumerationT<DeviceEndiannessMechanismEnum>& DeviceEndiannessMechanism`

Description: Identifies the endianness handling mode.

Visibility: Expert

15.177.4.6 DeviceID

`GenApi::IString& DeviceID`

Description: Interface-wide unique identifier of this device.

Visibility: Expert

15.177.4.7 DeviceInstanceId

`GenApi::IString& DeviceInstanceId`

Description: Visibility: Invisible.

15.177.4.8 DeviceIsUpdater

`GenApi::IBoolean& DeviceIsUpdater`

Description: Indicates whether the device is in updater mode.

Visibility: Expert

15.177.4.9 DeviceLinkSpeed

`GenApi::IInteger& DeviceLinkSpeed`

Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.

Visibility: Expert

15.177.4.10 DeviceLocation

`GenApi::IString& DeviceLocation`

Description: Device Location.

Visibility: Expert

15.177.4.11 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the remote device model.

Visibility: Beginner

15.177.4.12 DeviceMulticastMonitorMode

`GenApi::IBoolean& DeviceMulticastMonitorMode`

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

Visibility: Expert

15.177.4.13 DevicePortId

`GenApi::IString& DevicePortId`

Description: Device Port ID.

Visibility: Expert

15.177.4.14 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description: Serial number of the remote device.

Visibility: Expert

15.177.4.15 DeviceType

`GenApi::IEnumerationT<DeviceTypeEnum>& DeviceType`

Description: Transport layer type of the device.

Visibility: Expert

15.177.4.16 DeviceU3VProtocol

`GenApi::IBoolean& DeviceU3VProtocol`

Description: Indicates whether the device is communicating in U3V Protocol.

Visibility: Expert

15.177.4.17 DeviceUserID

`GenApi::IString& DeviceUserID`

Description: User Defined Name.

Visibility: Expert

15.177.4.18 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the remote device vendor.

Visibility: Beginner

15.177.4.19 DeviceVersion

`GenApi::IString& DeviceVersion`

Description: Version of the device.

Visibility: Beginner

15.177.4.20 GenICamXMLLocation

`GenApi::IEnumerationT<GenICamXMLLocationEnum>& GenICamXMLLocation`

Description: Sets the location to load [GenICam XML](#).

Visibility: Beginner

15.177.4.21 GenICamXMLPath

`GenApi::IString& GenICamXMLPath`

Description: [GenICam XML Path](#).

Visibility: Beginner

15.177.4.22 GevCCP

`GenApi::IEnumerationT<GevCCPEnum>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility: Beginner

15.177.4.23 GevDeviceAutoForceIP

`GenApi::ICommand&   GevDeviceAutoForceIP`

Description: Forces the camera to be on the same subnet as its corresponding interface.

Visibility: Expert

15.177.4.24 GevDeviceDiscoverMaximumPacketSize

`GenApi::ICommand&   GevDeviceDiscoverMaximumPacketSize`

Description: Discovers and updates the maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

15.177.4.25 GevDeviceForceGateway

`GenApi::IInteger&   GevDeviceForceGateway`

Description: Static gateway IP address to set for the GVCP interface of the remote device.

Visibility: Expert

15.177.4.26 GevDeviceForceIP

`GenApi::ICommand&   GevDeviceForceIP`

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDeviceForceGateway) in the Device using ForceIP command.

Visibility: Expert

15.177.4.27 GevDeviceForceIPAddress

`GenApi::IInteger&   GevDeviceForceIPAddress`

Description: Static IP address to set for the GVCP interface of the remote device.

Visibility: Expert

15.177.4.28 GevDeviceForceSubnetMask

`GenApi::IInteger&   GevDeviceForceSubnetMask`

Description: Static subnet mask to set for GVCP interface of the remote device.

Visibility: Expert

15.177.4.29 GevDeviceGateway

`GenApi::Integer&   GevDeviceGateway`

Description: Current gateway IP address of the GVCP interface of the remote device.

Visibility: Expert

15.177.4.30 GevDeviceIPAddress

`GenApi::Integer&   GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

15.177.4.31 GevDeviceIsWrongSubnet

`GenApi::Boolean&   GevDeviceIsWrongSubnet`

Description: Indicates whether the device is on the wrong subnet.

Visibility: Expert

15.177.4.32 GevDeviceMACAddress

`GenApi::Integer&   GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

15.177.4.33 GevDeviceMaximumPacketSize

`GenApi::Integer&   GevDeviceMaximumPacketSize`

Description: The maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

15.177.4.34 GevDeviceMaximumRetryCount

`GenApi::Integer&   GevDeviceMaximumRetryCount`

Description: Maximum number of times to retry a read/write operation.

Visibility: Expert

15.177.4.35 GevDeviceModelsBigEndian

`GenApi::IBoolean &GevDeviceModeIsBigEndian`

Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).

Visibility: Expert

15.177.4.36 GevDevicePort

`GenApi::IInteger &GevDevicePort`

Description: Current IP port of the GVCP interface of the selected remote device.

Visibility: Expert

15.177.4.37 GevDeviceReadAndWriteTimeout

`GenApi::IInteger &GevDeviceReadAndWriteTimeout`

Description: The timeout in us for read/write operations to the camera.

Visibility: Expert

15.177.4.38 GevDeviceSubnetMask

`GenApi::IInteger &GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

15.177.4.39 GevVersionMajor

`GenApi::IInteger &GevVersionMajor`

Description: Major version of the specification.

Visibility: Expert

15.177.4.40 GevVersionMinor

`GenApi::IInteger &GevVersionMinor`

Description: Minor version of the specification.

Visibility: Expert

15.177.4.41 GUIXMLLocation

`GenApi::IEnumerationT<GUIXMLLocationEnum>& GUIXMLLocation`

Description: Sets the location to load GUI XML.

Visibility: Beginner

15.177.4.42 GUIXMLPath

`GenApi::IString& GUIXMLPath`

Description: GUI XML Path.

Visibility: Beginner

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

- include/[TransportLayerDevice.h](#)

15.178 TransportLayerInterface Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerInterface](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerInterface](#) ()

Public Attributes

- [GenApi::IString](#) & [InterfaceID](#)
Description: Transport layer Producer wide unique identifier of the selected interface.
- [GenApi::IString](#) & [InterfaceDisplayName](#)
Description: User readable name of the selected interface.
- [GenApi::IEnumerationT](#) < [InterfaceTypeEnum](#) > & [InterfaceType](#)
Description: Transport layer type of the interface.
- [GenApi::Integer](#) & [GevInterfaceGatewaySelector](#)
Description: Selector for the different gateway entries for this interface.
- [GenApi::Integer](#) & [GevInterfaceGateway](#)
Description: IP address of the selected gateway entry of this interface.
- [GenApi::Integer](#) & [GevInterfaceMACAddress](#)
Description: 48-bit MAC address of this interface.
- [GenApi::Integer](#) & [GevInterfaceSubnetSelector](#)
Description: Selector for the subnet of this interface.
- [GenApi::Integer](#) & [GevInterfaceSubnetIPAddress](#)
Description: IP address of the selected subnet of this interface.
- [GenApi::Integer](#) & [GevInterfaceSubnetMask](#)

- Description: Subnet mask of the selected subnet of this interface.*

 - [GenApi::Integer](#) & [GevInterfaceTransmitLinkSpeed](#)

Description: Transmit link speed of this interface in bits per second.

 - [GenApi::Integer](#) & [GevInterfaceReceiveLinkSpeed](#)

Description: Receive link speed of this interface in bits per second.

 - [GenApi::Integer](#) & [GevInterfaceMTU](#)

Description: Maximum transmission unit of this interface.

 - [GenApi::EnumerationT](#) < [POEStatusEnum](#) > & [POEStatus](#)

Description: Reports and controls the interface's power over Ethernet status.

 - [GenApi::EnumerationT](#) < [FilterDriverStatusEnum](#) > & [FilterDriverStatus](#)

Description: Reports whether FLIR Light Weight Filter Driver is enabled or not.

 - [GenApi::Integer](#) & [GevActionDeviceKey](#)

Description: Key to authorize the action for the device.

 - [GenApi::Integer](#) & [GevActionGroupKey](#)

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

 - [GenApi::Integer](#) & [GevActionGroupMask](#)

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

 - [GenApi::Integer](#) & [GevActionTime](#)

Description: Provides the time in nanoseconds when the action is to be executed.

 - [GenApi::ICommand](#) & [ActionCommand](#)

Description: Issues an Action Command to attached GEV devices on interface.

 - [GenApi::IString](#) & [DeviceUnlock](#)

Description: Unlocks devices for internal use.

 - [GenApi::ICommand](#) & [DeviceUpdateList](#)

Description: Updates the internal device list.

 - [GenApi::Integer](#) & [DeviceCount](#)

Description: Number of compatible devices detected on current interface.

 - [GenApi::Integer](#) & [DeviceSelector](#)

Description: Selector for the different devices on this interface.

 - [GenApi::IString](#) & [DeviceID](#)

Description: [Interface](#) wide unique identifier of the selected device.

 - [GenApi::IString](#) & [DeviceVendorName](#)

Description: Name of the device vendor.

 - [GenApi::IString](#) & [DeviceModelName](#)

Description: Name of the device model.

 - [GenApi::IString](#) & [DeviceSerialNumber](#)

Description: Serial number of the selected remote device.

 - [GenApi::EnumerationT](#) < [DeviceAccessStatusEnum](#) > & [DeviceAccessStatus](#)

Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".

 - [GenApi::Integer](#) & [GevDeviceIPAddress](#)

Description: Current IP address of the GVCP interface of the selected remote device.

 - [GenApi::Integer](#) & [GevDeviceSubnetMask](#)

Description: Current subnet mask of the GVCP interface of the selected remote device.

 - [GenApi::Integer](#) & [GevDeviceGateway](#)

Description: Current gateway IP address of the GVCP interface of the selected remote device.

 - [GenApi::Integer](#) & [GevDeviceMACAddress](#)

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

 - [GenApi::Integer](#) & [IncompatibleDeviceCount](#)

Description: Number of incompatible devices detected on current interface.

- [GenApi::Integer](#) & [IncompatibleDeviceSelector](#)
Description: Selector for the devices that are not compatible with [Spinnaker](#) on this interface.
- [GenApi::IString](#) & [IncompatibleDeviceID](#)
Description: [Interface](#) wide unique identifier of the selected incompatible device.
- [GenApi::IString](#) & [IncompatibleDeviceVendorName](#)
Description: Name of the incompatible device vendor.
- [GenApi::IString](#) & [IncompatibleDeviceModelName](#)
Description: Name of the incompatible device model.
- [GenApi::Integer](#) & [IncompatibleGevDeviceIPAddress](#)
Description: Current IP address of the GVCP interface of the selected remote incompatible device.
- [GenApi::Integer](#) & [IncompatibleGevDeviceSubnetMask](#)
Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.
- [GenApi::Integer](#) & [IncompatibleGevDeviceMACAddress](#)
Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.
- [GenApi::ICommand](#) & [GevDeviceForceIP](#)
Description: Apply the force IP settings ([GevDeviceForceIPAddress](#), [GevDeviceForceSubnetMask](#) and [GevDeviceForceGateway](#)) in the selected remote device using [ForceIP](#) command.
- [GenApi::Integer](#) & [GevDeviceForceIPAddress](#)
Description: Static IP address to set for the GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceForceSubnetMask](#)
Description: Static subnet mask to set for GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceForceGateway](#)
Description: Static gateway IP address to set for the GVCP interface of the selected remote device.
- [GenApi::ICommand](#) & [GevDeviceAutoForceIP](#)
Description: Automatically forces the selected remote device to an IP Address on the same subnet as the GVCP interface.
- [GenApi::IString](#) & [HostAdapterName](#)
Description: User readable name of the interface's host adapter.
- [GenApi::IString](#) & [HostAdapterVendor](#)
Description: User readable name of the host adapter's vendor.
- [GenApi::IString](#) & [HostAdapterDriverVersion](#)
Description: Driver version of the interface's host adapter.

Protected Member Functions

- [TransportLayerInterface](#) ()

Friends

- class [Interface](#)
- class [IInterface](#)
- class [InterfaceInternal](#)

15.178.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.178.2 Constructor & Destructor Documentation

15.178.2.1 TransportLayerInterface() [1/2]

```
TransportLayerInterface (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.178.2.2 ~TransportLayerInterface()

```
~TransportLayerInterface ( )
```

15.178.2.3 TransportLayerInterface() [2/2]

```
TransportLayerInterface ( ) [protected]
```

15.178.3 Friends And Related Function Documentation

15.178.3.1 IInterface

```
friend class IInterface [friend]
```

15.178.3.2 Interface

```
friend class Interface [friend]
```

15.178.3.3 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

15.178.4 Member Data Documentation

15.178.4.1 ActionCommand

`GenApi::ICommand& ActionCommand`

Description: Issues an Action Command to attached GEV devices on interface.

Visibility: Expert

15.178.4.2 DeviceAccessStatus

`GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus`

Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.178.4.3 DeviceCount

`GenApi::IInteger& DeviceCount`

Description: Number of compatible devices detected on current interface.

Visibility: Expert

15.178.4.4 DeviceID

`GenApi::IString& DeviceID`

Description: [Interface](#) wide unique identifier of the selected device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.178.4.5 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.178.4.6 DeviceSelector

`GenApi::IInteger& DeviceSelector`

Description: Selector for the different devices on this interface.

This value only changes on execution of "DeviceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Expert

15.178.4.7 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description: Serial number of the selected remote device.

Visibility: Expert

15.178.4.8 DeviceUnlock

`GenApi::IString& DeviceUnlock`

Description: Unlocks devices for internal use.

Visibility: Expert

15.178.4.9 DeviceUpdateList

`GenApi::ICommand& DeviceUpdateList`

Description: Updates the internal device list.

Visibility: Expert

15.178.4.10 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.178.4.11 FilterDriverStatus

`GenApi::IEnumerationT<FilterDriverStatusEnum>& FilterDriverStatus`

Description: Reports whether FLIR Light Weight Filter Driver is enabled or not.

Visibility: Expert

15.178.4.12 GevActionDeviceKey

`GenApi::IInteger& GevActionDeviceKey`

Description: Key to authorize the action for the device.

Visibility: Expert

15.178.4.13 `GevActionGroupKey`

`GenApi::Integer& GevActionGroupKey`

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

15.178.4.14 `GevActionGroupMask`

`GenApi::Integer& GevActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

15.178.4.15 `GevActionTime`

`GenApi::Integer& GevActionTime`

Description: Provides the time in nanoseconds when the action is to be executed.

Visibility: Expert

15.178.4.16 `GevDeviceAutoForceIP`

`GenApi::Command& GevDeviceAutoForceIP`

Description: Automatically forces the selected remote device to an IP Address on the same subnet as the GVCP interface.

Visibility: Expert

15.178.4.17 `GevDeviceForceGateway`

`GenApi::Integer& GevDeviceForceGateway`

Description: Static gateway IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

15.178.4.18 `GevDeviceForceIP`

`GenApi::Command& GevDeviceForceIP`

Description: Apply the force IP settings (`GevDeviceForceIPAddress`, `GevDeviceForceSubnetMask` and `GevDeviceForceGateway`) in the selected remote device using `ForceIP` command.

Visibility: Expert

15.178.4.19 GevDeviceForceIPAddress

`GenApi::Integer&   GevDeviceForceIPAddress`

Description: Static IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

15.178.4.20 GevDeviceForceSubnetMask

`GenApi::Integer&   GevDeviceForceSubnetMask`

Description: Static subnet mask to set for GVCP interface of the selected remote device.

Visibility: Expert

15.178.4.21 GevDeviceGateway

`GenApi::Integer&   GevDeviceGateway`

Description: Current gateway IP address of the GVCP interface of the selected remote device.

Visibility: Expert

15.178.4.22 GevDeviceIPAddress

`GenApi::Integer&   GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

15.178.4.23 GevDeviceMACAddress

`GenApi::Integer&   GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

15.178.4.24 GevDeviceSubnetMask

`GenApi::Integer&   GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

15.178.4.25 GevInterfaceGateway

`GenApi::Integer&   GevInterfaceGateway`

Description: IP address of the selected gateway entry of this interface.

Visibility: Expert

15.178.4.26 GevInterfaceGatewaySelector

`GenApi::Integer&   GevInterfaceGatewaySelector`

Description: Selector for the different gateway entries for this interface.

Visibility: Expert

15.178.4.27 GevInterfaceMACAddress

`GenApi::Integer&   GevInterfaceMACAddress`

Description: 48-bit MAC address of this interface.

Visibility: Expert

15.178.4.28 GevInterfaceMTU

`GenApi::Integer&   GevInterfaceMTU`

Description: Maximum transmission unit of this interface.

Visibility: Expert

15.178.4.29 GevInterfaceReceiveLinkSpeed

`GenApi::Integer&   GevInterfaceReceiveLinkSpeed`

Description: Receive link speed of this interface in bits per second.

Visibility: Expert

15.178.4.30 GevInterfaceSubnetIPAddress

`GenApi::Integer&   GevInterfaceSubnetIPAddress`

Description: IP address of the selected subnet of this interface.

Visibility: Expert

15.178.4.31 GevInterfaceSubnetMask

`GenApi::Integer&   GevInterfaceSubnetMask`

Description: Subnet mask of the selected subnet of this interface.

Visibility: Expert

15.178.4.32 GevInterfaceSubnetSelector

`GenApi::Integer&   GevInterfaceSubnetSelector`

Description: Selector for the subnet of this interface.

Visibility: Expert

15.178.4.33 GevInterfaceTransmitLinkSpeed

`GenApi::Integer&   GevInterfaceTransmitLinkSpeed`

Description: Transmit link speed of this interface in bits per second.

Visibility: Expert

15.178.4.34 HostAdapterDriverVersion

`GenApi::String&   HostAdapterDriverVersion`

Description: Driver version of the interface's host adapter.

Visibility: Expert

15.178.4.35 HostAdapterName

`GenApi::String&   HostAdapterName`

Description: User readable name of the interface's host adapter.

Visibility: Expert

15.178.4.36 HostAdapterVendor

`GenApi::String&   HostAdapterVendor`

Description: User readable name of the host adapter's vendor.

Visibility: Expert

15.178.4.37 IncompatibleDeviceCount

`GenApi::Integer& IncompatibleDeviceCount`

Description: Number of incompatible devices detected on current interface.

Visibility: Expert

15.178.4.38 IncompatibleDeviceID

`GenApi::IString& IncompatibleDeviceID`

Description: [Interface](#) wide unique identifier of the selected incompatible device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.178.4.39 IncompatibleDeviceModelName

`GenApi::IString& IncompatibleDeviceModelName`

Description: Name of the incompatible device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.178.4.40 IncompatibleDeviceSelector

`GenApi::Integer& IncompatibleDeviceSelector`

Description: Selector for the devices that are not compatible with [Spinnaker](#) on this interface.

This value only changes on execution of "DeviceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Expert

15.178.4.41 IncompatibleDeviceVendorName

`GenApi::IString& IncompatibleDeviceVendorName`

Description: Name of the incompatible device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.178.4.42 IncompatibleGevDeviceIPAddress

`GenApi::Integer& IncompatibleGevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

15.178.4.43 IncompatibleGevDeviceMACAddress

`GenApi::Integer& IncompatibleGevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

15.178.4.44 IncompatibleGevDeviceSubnetMask

`GenApi::Integer& IncompatibleGevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

15.178.4.45 InterfaceDisplayName

`GenApi::IString& InterfaceDisplayName`

Description: User readable name of the selected interface.

Visibility: Expert

15.178.4.46 InterfaceID

`GenApi::IString& InterfaceID`

Description: Transport layer Producer wide unique identifier of the selected interface.

Visibility: Expert

15.178.4.47 InterfaceType

`GenApi::IEnumerationT<InterfaceTypeEnum>& InterfaceType`

Description: Transport layer type of the interface.

Visibility: Expert

15.178.4.48 POEStatus

`GenApi::IEnumerationT<POEStatusEnum>& POEStatus`

Description: Reports and controls the interface's power over Ethernet status.

Visibility: Expert

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

- include/[TransportLayerInterface.h](#)

15.179 TransportLayerStream Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerStream](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerStream](#) ()

Public Attributes

- [GenApi::IString](#) & [StreamID](#)
Description: Device unique ID for the data stream, e.g.
- [GenApi::IEnumerationT](#)< [StreamTypeEnum](#) > & [StreamType](#)
Description: Stream type of the device.
- [GenApi::IInteger](#) & [StreamBufferCountManual](#)
Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.
- [GenApi::IInteger](#) & [StreamBufferCountResult](#)
Description: Displays the number of buffers to be used on this stream upon acquisition start.
- [GenApi::IInteger](#) & [StreamBufferCountMax](#)
Description: Controls the maximum number of buffers that should be used on this stream.
- [GenApi::IEnumerationT](#)< [StreamBufferCountModeEnum](#) > & [StreamBufferCountMode](#)
Description: Controls access to setting the number of buffers used for the stream.
- [GenApi::IEnumerationT](#)< [StreamBufferHandlingModeEnum](#) > & [StreamBufferHandlingMode](#)
Description: Available buffer handling modes of this data stream: Visibility: Beginner.
- [GenApi::IInteger](#) & [StreamAnnounceBufferMinimum](#)
Description: Minimal number of buffers to announce to enable selected buffer handling mode.
- [GenApi::IInteger](#) & [StreamAnnouncedBufferCount](#)
Description: Number of announced (known) buffers on this stream.
- [GenApi::IInteger](#) & [StreamStartedFrameCount](#)
Description: Number of frames started in the acquisition engine.
- [GenApi::IInteger](#) & [StreamDeliveredFrameCount](#)
Description: Number of delivered frames since last acquisition start.
- [GenApi::IInteger](#) & [StreamLostFrameCount](#)
Description: Number of lost frames due to queue underrun.
- [GenApi::IInteger](#) & [StreamInputBufferCount](#)
Description: Number of buffers in the input buffer pool plus the buffers(s) currently being filled.
- [GenApi::IInteger](#) & [StreamOutputBufferCount](#)
Description: Number of buffers in the output buffer queue.
- [GenApi::IBoolean](#) & [StreamCRCCheckEnable](#)
Description: Enables or disables CRC checks on received images.
- [GenApi::IBoolean](#) & [GevPacketResendMode](#)
Description: Enables or disables the packet resend mechanism.
- [GenApi::IInteger](#) & [GevMaximumNumberResendRequests](#)
Description: Maximum number of resend requests per image.
- [GenApi::IInteger](#) & [GevPacketResendTimeout](#)
Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.
- [GenApi::IBoolean](#) & [StreamIsGrabbing](#)

Description: Flag indicating whether the acquisition engine is started or not.

- [GenApi::Integer](#) & [StreamChunkCountMaximum](#)

Description: Maximum number of chunks to be expected in a buffer.

- [GenApi::Integer](#) & [StreamBufferAlignment](#)

Description: Alignment size in bytes of the buffer passed to DSAnnounceBuffer.

- [GenApi::Integer](#) & [StreamDroppedFrameCount](#)

Description: Number of dropped frames due to queue overrun.

- [GenApi::Integer](#) & [GevTotalPacketCount](#)

Description: Displays number of packets received on this stream.

- [GenApi::Integer](#) & [GevFailedPacketCount](#)

Description: Displays number of packets missed on this stream.

- [GenApi::Integer](#) & [GevResendPacketCount](#)

Description: Displays number of packets received after retransmit request on this stream.

- [GenApi::Integer](#) & [StreamFailedBufferCount](#)

Description: Displays number of images with missing packet.

- [GenApi::Integer](#) & [GevResendRequestCount](#)

Description: Displays number of packets requested to be retransmitted on this stream.

- [GenApi::Integer](#) & [StreamBlockTransferSize](#)

Description: Controls the image breakup size that should be used on this stream.

Protected Member Functions

- [TransportLayerStream](#) ()

Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameraInternal](#)

15.179.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.179.2 Constructor & Destructor Documentation

15.179.2.1 TransportLayerStream() [1/2]

```
TransportLayerStream (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.179.2.2 ~TransportLayerStream()

`~TransportLayerStream ( )`

15.179.2.3 TransportLayerStream() [2/2]

`TransportLayerStream ( )` [protected]

15.179.3 Friends And Related Function Documentation

15.179.3.1 CameraBase

`friend class CameraBase` [friend]

15.179.3.2 CameraInternal

`friend class CameraInternal` [friend]

15.179.3.3 ICameraBase

`friend class ICameraBase` [friend]

15.179.4 Member Data Documentation

15.179.4.1 GevFailedPacketCount

`GenApi::Integer& GevFailedPacketCount`

Description: Displays number of packets missed on this stream.

Successful resent packets are not counted as a missed packet. Visibility: Expert

15.179.4.2 `GevMaximumNumberResendRequests`

`GenApi::Integer& GevMaximumNumberResendRequests`

Description: Maximum number of resend requests per image.

Each resend request consists of a span of consecutive packet IDs. Visibility: Expert

15.179.4.3 `GevPacketResendMode`

`GenApi::Boolean& GevPacketResendMode`

Description: Enables or disables the packet resend mechanism.

Visibility: Expert

15.179.4.4 `GevPacketResendTimeout`

`GenApi::Integer& GevPacketResendTimeout`

Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.

Visibility: Expert

15.179.4.5 `GevResendPacketCount`

`GenApi::Integer& GevResendPacketCount`

Description: Displays number of packets received after retransmit request on this stream.

Visibility: Expert

15.179.4.6 `GevResendRequestCount`

`GenApi::Integer& GevResendRequestCount`

Description: Displays number of packets requested to be retransmitted on this stream.

Visibility: Expert

15.179.4.7 `GevTotalPacketCount`

`GenApi::Integer& GevTotalPacketCount`

Description: Displays number of packets received on this stream.

Visibility: Expert

15.179.4.8 StreamAnnounceBufferMinimum

`GenApi::Integer& StreamAnnounceBufferMinimum`

Description: Minimal number of buffers to announce to enable selected buffer handling mode.

Visibility: Expert

15.179.4.9 StreamAnnouncedBufferCount

`GenApi::Integer& StreamAnnouncedBufferCount`

Description: Number of announced (known) buffers on this stream.

This value is volatile. It may change if additional buffers are announced and/or buffers are revoked by the GenTL Consumer. Visibility: Expert

15.179.4.10 StreamBlockTransferSize

`GenApi::Integer& StreamBlockTransferSize`

Description: Controls the image breakup size that should be used on this stream.

Visibility: Expert

15.179.4.11 StreamBufferAlignment

`GenApi::Integer& StreamBufferAlignment`

Description: Alignment size in bytes of the buffer passed to DSAnnounceBuffer.

Visibility: Expert

15.179.4.12 StreamBufferCountManual

`GenApi::Integer& StreamBufferCountManual`

Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.

Visibility: Expert

15.179.4.13 StreamBufferCountMax

`GenApi::Integer& StreamBufferCountMax`

Description: Controls the maximum number of buffers that should be used on this stream.

This value is calculated based on the available system memory. Visibility: Expert

15.179.4.14 StreamBufferCountMode

`GenApi::IEnumerationT<StreamBufferCountModeEnum>& StreamBufferCountMode`

Description: Controls access to setting the number of buffers used for the stream.

Visibility: Expert

15.179.4.15 StreamBufferCountResult

`GenApi::IInteger& StreamBufferCountResult`

Description: Displays the number of buffers to be used on this stream upon acquisition start.

Recalculated on acquisition start if in auto mode. Visibility: Expert

15.179.4.16 StreamBufferHandlingMode

`GenApi::IEnumerationT<StreamBufferHandlingModeEnum>& StreamBufferHandlingMode`

Description: Available buffer handling modes of this data stream: Visibility: Beginner.

15.179.4.17 StreamChunkCountMaximum

`GenApi::IInteger& StreamChunkCountMaximum`

Description: Maximum number of chunks to be expected in a buffer.

Visibility: Expert

15.179.4.18 StreamCRCCheckEnable

`GenApi::IBoolean& StreamCRCCheckEnable`

Description: Enables or disables CRC checks on received images.

Visibility: Expert

15.179.4.19 StreamDeliveredFrameCount

`GenApi::IInteger& StreamDeliveredFrameCount`

Description: Number of delivered frames since last acquisition start.

It is not reset until the stream is closed. Visibility: Expert

15.179.4.20 StreamDroppedFrameCount

`GenApi::Integer& StreamDroppedFrameCount`

Description: Number of dropped frames due to queue overrun.

This number is initialized with zero at the time the stream is opened and incremented every time old data is dropped from the output list for new data. It is not reset until the stream is closed. Visibility: Expert

15.179.4.21 StreamFailedBufferCount

`GenApi::Integer& StreamFailedBufferCount`

Description: Displays number of images with missing packet.

Visibility: Expert

15.179.4.22 StreamID

`GenApi::IString& StreamID`

Description: Device unique ID for the data stream, e.g.

a GUID. Visibility: Expert

15.179.4.23 StreamInputBufferCount

`GenApi::Integer& StreamInputBufferCount`

Description: Number of buffers in the input buffer pool plus the buffers(s) currently being filled.

Visibility: Expert

15.179.4.24 StreamIsGrabbing

`GenApi::IBoolean& StreamIsGrabbing`

Description: Flag indicating whether the acquisition engine is started or not.

Visibility: Expert

15.179.4.25 StreamLostFrameCount

`GenApi::Integer& StreamLostFrameCount`

Description: Number of lost frames due to queue underrun.

This number is initialized with zero at the time the stream is opened and incremented every time the data could not be acquired because there was no buffer in the input buffer pool. It is not reset until the stream is closed. Visibility: Expert

15.179.4.26 StreamOutputBufferCount

`GenApi::Integer& StreamOutputBufferCount`

Description: Number of buffers in the output buffer queue.

Visibility: Expert

15.179.4.27 StreamStartedFrameCount

`GenApi::Integer& StreamStartedFrameCount`

Description: Number of frames started in the acquisition engine.

This number is incremented every time in case of a new buffer is started and then to be filled (data written to) regardless even if the buffer is later delivered to the user or discarded for any reason. This number is initialized with 0 at the time of the stream is opened. It is not reset until the stream is closed. Visibility: Expert

15.179.4.28 StreamType

`GenApi::EnumerationT<StreamTypeEnum>& StreamType`

Description: Stream type of the device.

Visibility: Expert

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

- [include/TransportLayerStream.h](#)

15.180 TransportLayerSystem Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerSystem](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerSystem](#) ()

Public Attributes

- [GenApi::IString](#) & [TLID](#)
Description: Unique identifier of the GenTL Producer like a GUID.
- [GenApi::IString](#) & [TLVendorName](#)
Description: Name of the GenTL Producer vendor.
- [GenApi::IString](#) & [TLModelName](#)
Description: Name of the GenTL Producer to distinguish different kinds of GenTL Producer implementations from one vendor.
- [GenApi::IString](#) & [TLVersion](#)
Description: Vendor specific version string.
- [GenApi::IString](#) & [TLFileName](#)
Description: Filename including extension of the GenTL Producer.
- [GenApi::IString](#) & [TLDisplayName](#)
Description: User readable name of the GenTL Producer.
- [GenApi::IString](#) & [TLPath](#)
Description: Full path to the GenTL Producer including filename and extension.
- [GenApi::IEnumerationT< TLTypeEnum >](#) & [TLType](#)
Description: Transport layer type of the GenTL Producer implementation.
- [GenApi::Integer](#) & [GenTLVersionMajor](#)
Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.
- [GenApi::Integer](#) & [GenTLVersionMinor](#)
Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.
- [GenApi::Integer](#) & [GenTLSFNCVersionMajor](#)
Description: Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.
- [GenApi::Integer](#) & [GenTLSFNCVersionMinor](#)
Description: Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.
- [GenApi::Integer](#) & [GenTLSFNCVersionSubMinor](#)
Description: Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.
- [GenApi::Integer](#) & [GevVersionMajor](#)
Description: Major version number of the GigE Vision specification the GenTL Producer implementation complies to.
- [GenApi::Integer](#) & [GevVersionMinor](#)
Description: Minor version number of the GigE Vision specification the GenTL Producer implementation complies to.
- [GenApi::ICommand](#) & [InterfaceUpdateList](#)
Description: Updates the internal list of the interfaces.
- [GenApi::Integer](#) & [InterfaceSelector](#)
Description: Selector for the different GenTL Producer interfaces.
- [GenApi::IString](#) & [InterfaceID](#)
Description: GenTL Producer wide unique identifier of the selected interface.
- [GenApi::IString](#) & [InterfaceDisplayName](#)
Description: A user-friendly name of the selected [Interface](#).
- [GenApi::Integer](#) & [GevInterfaceMACAddress](#)
Description: 48-bit MAC address of the selected interface.
- [GenApi::Integer](#) & [GevInterfaceDefaultIPAddress](#)
Description: IP address of the first subnet of the selected interface.
- [GenApi::Integer](#) & [GevInterfaceDefaultSubnetMask](#)
Description: Subnet mask of the first subnet of the selected interface.
- [GenApi::Integer](#) & [GevInterfaceDefaultGateway](#)
Description: Gateway of the selected interface.

- [GenApi::IBoolean](#) & [EnumerateGEVInterfaces](#)

Description: Enables or disables enumeration of GEV Interfaces.

- [GenApi::IBoolean](#) & [EnumerateUSBInterfaces](#)

Description: Enables or disables enumeration of USB Interfaces.

Protected Member Functions

- [TransportLayerSystem](#) ()

Friends

- class [System](#)
- class [ISystem](#)
- class [SystemPtrInternal](#)

15.180.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.180.2 Constructor & Destructor Documentation

15.180.2.1 [TransportLayerSystem](#)() [1/2]

```
TransportLayerSystem (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.180.2.2 [~TransportLayerSystem](#)()

```
~TransportLayerSystem ( )
```

15.180.2.3 [TransportLayerSystem](#)() [2/2]

```
TransportLayerSystem ( ) [protected]
```

15.180.3 Friends And Related Function Documentation

15.180.3.1 ISystem

```
friend class ISystem [friend]
```

15.180.3.2 System

```
friend class System [friend]
```

15.180.3.3 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

15.180.4 Member Data Documentation

15.180.4.1 EnumerateGEVInterfaces

```
GenApi::IBoolean& EnumerateGEVInterfaces
```

Description: Enables or disables enumeration of GEV Interfaces.

Visibility: Expert

15.180.4.2 EnumerateUSBInterfaces

```
GenApi::IBoolean& EnumerateUSBInterfaces
```

Description: Enables or disables enumeration of USB Interfaces.

Visibility: Expert

15.180.4.3 GenTL_SFNCVersionMajor

```
GenApi::Integer& GenTL_SFNCVersionMajor
```

Description: Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

15.180.4.4 GenTlsFncVersionMinor

`GenApi::Integer& GenTlsFncVersionMinor`

Description: Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

15.180.4.5 GenTlsFncVersionSubMinor

`GenApi::Integer& GenTlsFncVersionSubMinor`

Description: Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

15.180.4.6 GenTlVersionMajor

`GenApi::Integer& GenTlVersionMajor`

Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

15.180.4.7 GenTlVersionMinor

`GenApi::Integer& GenTlVersionMinor`

Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

15.180.4.8 GevInterfaceDefaultGateway

`GenApi::Integer& GevInterfaceDefaultGateway`

Description: Gateway of the selected interface.

Visibility: Expert

15.180.4.9 GevInterfaceDefaultIpAddress

`GenApi::Integer& GevInterfaceDefaultIpAddress`

Description: IP address of the first subnet of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

15.180.4.10 `GevInterfaceDefaultSubnetMask`

`GenApi::Integer& GevInterfaceDefaultSubnetMask`

Description: Subnet mask of the first subnet of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

15.180.4.11 `GevInterfaceMACAddress`

`GenApi::Integer& GevInterfaceMACAddress`

Description: 48-bit MAC address of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

15.180.4.12 `GevVersionMajor`

`GenApi::Integer& GevVersionMajor`

Description: Major version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

15.180.4.13 `GevVersionMinor`

`GenApi::Integer& GevVersionMinor`

Description: Minor version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

15.180.4.14 `InterfaceDisplayName`

`GenApi::String& InterfaceDisplayName`

Description: A user-friendly name of the selected [Interface](#).

Visibility: Beginner

15.180.4.15 `InterfaceID`

`GenApi::String& InterfaceID`

Description: GenTL Producer wide unique identifier of the selected interface.

Visibility: Beginner

15.180.4.16 InterfaceSelector

`GenApi::Integer& InterfaceSelector`

Description: Selector for the different GenTL Producer interfaces.

This interface list only changes on execution of "InterfaceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Beginner

15.180.4.17 InterfaceUpdateList

`GenApi::ICommand& InterfaceUpdateList`

Description: Updates the internal list of the interfaces.

This feature is readable even if the execution cannot be performed immediately. The command then returns and the status can be polled. This function interacts with the TLUpdateInterfaceList function of the GenTL producer. It is up to the GenTL consumer to handle access in case both methods are used. Visibility: Beginner

15.180.4.18 TLDisplayName

`GenApi::IString& TLDisplayName`

Description: User readable name of the GenTL Producer.

Visibility: Expert

15.180.4.19 TLFileName

`GenApi::IString& TLFileName`

Description: Filename including extension of the GenTL Producer.

Visibility: Expert

15.180.4.20 TLID

`GenApi::IString& TLID`

Description: Unique identifier of the GenTL Producer like a GUID.

Visibility: Expert

15.180.4.21 TLModelName

`GenApi::IString& TLModelName`

Description: Name of the GenTL Producer to distinguish different kinds of GenTL Producer implementations from one vendor.

Visibility: Beginner

15.180.4.22 TLPath

`GenApi::IString& TLPath`

Description: Full path to the GenTL Producer including filename and extension.

Visibility: Expert

15.180.4.23 TLType

`GenApi::IEnumerationT<TLTypeEnum>& TLType`

Description: Transport layer type of the GenTL Producer implementation.

Visibility: Expert

15.180.4.24 TLVendorName

`GenApi::IString& TLVendorName`

Description: Name of the GenTL Producer vendor.

Visibility: Beginner

15.180.4.25 TLVersion

`GenApi::IString& TLVersion`

Description: Vendor specific version string.

Visibility: Beginner

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

- include/[TransportLayerSystem.h](#)

15.181 U3V_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t [ChunkID](#)
- uint32_t [ChunkLength](#)

15.181.1 Detailed Description

header of a GVCP request packet

15.181.2 Member Data Documentation

15.181.2.1 ChunkID

`uint32_t` ChunkID

15.181.2.2 ChunkLength

`uint32_t` ChunkLength

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

- `include/SpinGenApi/ChunkAdapterU3V.h`

15.182 U3V_COMMAND_HEADER Struct Reference

U3V/GenCP command header.

Public Attributes

- `uint32_t` [Prefix](#)
- `uint16_t` [Flags](#)
- `uint16_t` [CommandId](#)
- `uint16_t` [Length](#)
- `uint16_t` [ReqId](#)

15.182.1 Detailed Description

U3V/GenCP command header.

15.182.2 Member Data Documentation

15.182.2.1 CommandId

uint16_t CommandId

15.182.2.2 Flags

uint16_t Flags

15.182.2.3 Length

uint16_t Length

15.182.2.4 Prefix

uint32_t Prefix

15.182.2.5 ReqId

uint16_t ReqId

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

- include/SpinGenApi/[EventAdapterU3V.h](#)

15.183 U3V_EVENT_DATA Struct Reference

U3V/GenCP EVENT_CMD specific command data.

Public Attributes

- uint16_t [Reserved](#)
- uint16_t [EventId](#)
- uint64_t [Timestamp](#)

15.183.1 Detailed Description

U3V/GenCP EVENT_CMD specific command data.

15.183.2 Member Data Documentation

15.183.2.1 EventId

`uint16_t EventId`

15.183.2.2 Reserved

`uint16_t Reserved`

15.183.2.3 Timestamp

`uint64_t Timestamp`

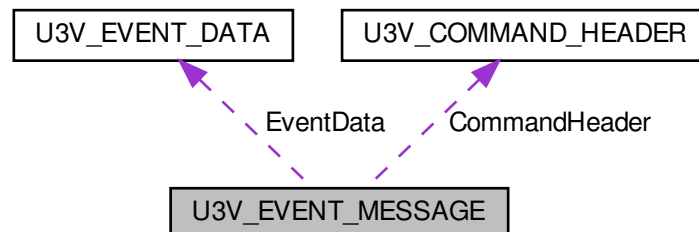
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterU3V.h`

15.184 U3V_EVENT_MESSAGE Struct Reference

Entire event data message (without the variable-sized data field)

Collaboration diagram for U3V_EVENT_MESSAGE:



Public Attributes

- [U3V_COMMAND_HEADER](#) `CommandHeader`
- [U3V_EVENT_DATA](#) `EventData`

15.184.1 Detailed Description

Entire event data message (without the variable-sized data field)

15.184.2 Member Data Documentation

15.184.2.1 CommandHeader

[U3V_COMMAND_HEADER](#) `CommandHeader`

15.184.2.2 EventData

[U3V_EVENT_DATA](#) `EventData`

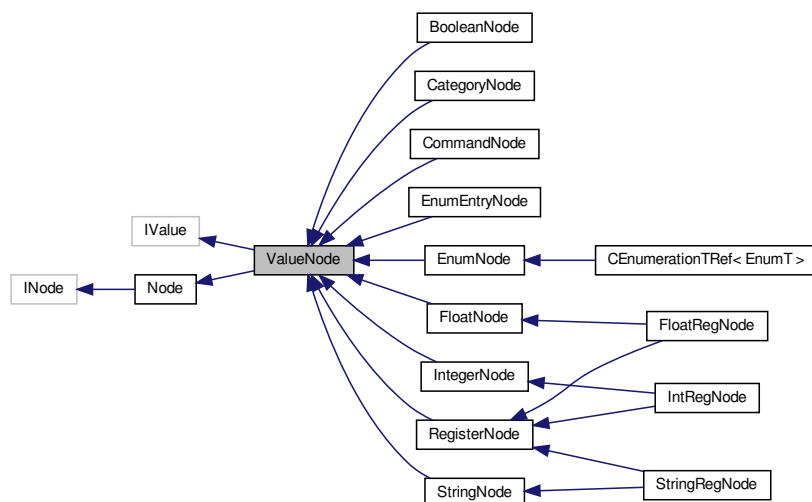
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterU3V.h`

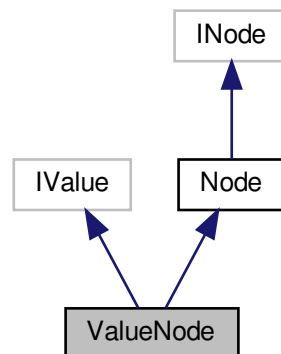
15.185 ValueNode Class Reference

[Interface](#) for value properties.

Inheritance diagram for ValueNode:



Collaboration diagram for ValueNode:



Public Member Functions

- [ValueNode](#) ()
Constructor.
- [ValueNode](#) (std::shared_ptr< Node::NodeImpl > pValue)
constructor with [GenICam](#) IValue
- [~ValueNode](#) ()
Destructor.
- virtual [INode](#) * [GetNode](#) ()
- virtual [GenICam::gcstring](#) [ToString](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get content of the node as string.
- virtual void [FromString](#) (const [GenICam::gcstring](#) &ValueStr, bool [Verify](#)=true)
Set content of the node as string.
- virtual bool [IsValueCacheValid](#) () const
Checks if the value comes from cache or is requested from another node.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Value

Additional Inherited Members

15.185.1 Detailed Description

[Interface](#) for value properties.

15.185.2 Constructor & Destructor Documentation

15.185.2.1 ValueNode() [1/2]

```
ValueNode ( )
```

Constructor.

15.185.2.2 ValueNode() [2/2]

```
ValueNode (
    std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenICam IValue](#)

15.185.2.3 ~ValueNode()

```
~ValueNode ( )
```

Destructor.

15.185.3 Member Function Documentation

15.185.3.1 FromString()

```
virtual void FromString (
    const GenICam::gcstring & ValueStr,
    bool Verify = true ) [virtual]
```

Set content of the node as string.

Parameters

<i>ValueStr</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.185.3.2 GetNode()

```
virtual INode* GetNode ( ) [virtual]
```

15.185.3.3 IsValueCacheValid()

```
virtual bool IsValueCacheValid ( ) const [virtual]
```

Checks if the value comes from cache or is requested from another node.

15.185.3.4 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [Node](#).

Reimplemented in [FloatNode](#), [IntegerNode](#), [EnumNode](#), [CEnumerationTRef< EnumT >](#), [StringNode](#), [RegisterNode](#), [BooleanNode](#), [CommandNode](#), [EnumEntryNode](#), [CategoryNode](#), [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

15.185.3.5 ToString()

```
virtual GenICam::gcstring ToString (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get content of the node as string.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

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

- include/SpinGenApi/[ValueNode.h](#)

15.186 Version_t Struct Reference

Version.

Public Attributes

- uint16_t [Major](#)
- uint16_t [Minor](#)
a is incompatible with b if $a \neq b$
- uint16_t [SubMinor](#)
a is incompatible b $a > b$

15.186.1 Detailed Description

Version.

15.186.2 Member Data Documentation

15.186.2.1 Major

uint16_t Major

15.186.2.2 Minor

uint16_t Minor

a is incompatible with b if $a \neq b$

15.186.2.3 SubMinor

uint16_t SubMinor

a is incompatible b $a > b$

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

- include/SpinGenApi/[GCTypes.h](#)

Chapter 16

File Documentation

16.1 [doc/spindocs/C++/GettingStarted.dox](#) File Reference

16.2 [doc/spindocs/C++/ProgrammerGuide.dox](#) File Reference

16.3 [doc/spindocs/shared/Benefits.dox](#) File Reference

16.4 [doc/spindocs/shared/FlyCapture2Comparison.dox](#) File Reference

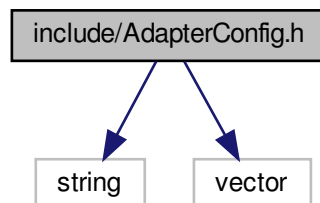
16.5 [doc/spindocs/shared/GenICamGenTL.dox](#) File Reference

16.6 [doc/spindocs/shared/Licensing.dox](#) File Reference

16.7 [doc/spindocs/shared/Maintenance.dox](#) File Reference

16.8 [include/AdapterConfig.h](#) File Reference

Include dependency graph for AdapterConfig.h:



Classes

- struct [IpInfo](#)
- struct [AdapterInfo](#)
- class [AdapterConfigException](#)

Namespaces

- [AdapterConfig](#)

Macros

- `#define ADAPTERCONFIG\_API __declspec(dllimport)`

Enumerations

- enum [AdapterConfigErr](#) {
[IP_ADDRESS_INVALID](#),
[IP_ADDRESS_IS_NOT_V4](#),
[IP_ADDRESS_TOO_LARGE](#),
[IP_ADDRESS_TOO_SMALL](#),
[HOST_ADDRESS_ZERO](#),
[SUBNET_MASK_INVALID](#),
[VALID_SUBNET_NOT_FOUND](#) }

Functions

- [ADAPTERCONFIG_API](#) `std::vector< AdapterInfo > RetrieveAllAdapters ()`
- [ADAPTERCONFIG_API](#) `void AutoPopulateAdapterInfo (std::vector< AdapterInfo > &adaptersToConfigure, const std::vector< AdapterInfo > &allAdapters)`
- [ADAPTERCONFIG_API](#) `void AutoPopulateAdvancedProperties (std::vector< AdapterInfo > &adaptersToConfigure)`
- [ADAPTERCONFIG_API](#) `void PopulateAdapterIpInfo (IpInfo startingIpInfo, std::vector< AdapterInfo > &adaptersToConfigure, const std::vector< AdapterInfo > &allAdapters, bool skipIfIPValid=false)`
- [ADAPTERCONFIG_API](#) `void ValidateIpAddress (const std::string &ipAddr, unsigned int subnetMaskLength)`
- [ADAPTERCONFIG_API](#) `bool IsValidIpAddress (const std::string &ipAddr)`
- [ADAPTERCONFIG_API](#) `bool IsValidSubnetMask (const std::string &subnetMask)`
- [ADAPTERCONFIG_API](#) `bool IsOnSameSubnet (const std::string &ipAddrStr1, const std::string &ipAddrStr2, const unsigned int subnetMaskLength)`
- [ADAPTERCONFIG_API](#) `unsigned int GetSubnetMaskLength (const std::string &subnetMask)`
- [ADAPTERCONFIG_API](#) `std::string GetEnumerationLogFileName ()`
- [ADAPTERCONFIG_API](#) `std::string GetConfigLogFileName ()`
- [ADAPTERCONFIG_API](#) `void ConfigureAdapter (AdapterInfo &adapter, bool configureIP, bool configureAdvancedProperties)`
- [ADAPTERCONFIG_API](#) `unsigned int GetAutoSubnetMaskLength ()`
- [ADAPTERCONFIG_API](#) `std::string GetAutoSubnetMask ()`
- [ADAPTERCONFIG_API](#) `std::string GetMaxIpAddress ()`
- [ADAPTERCONFIG_API](#) `std::string GetMinIpAddress ()`
- [ADAPTERCONFIG_API](#) `std::string GetAutoGigabitDesc ()`
- [ADAPTERCONFIG_API](#) `std::string GetAuto10GDesc ()`
- [ADAPTERCONFIG_API](#) `std::string GetAutoStartIp ()`

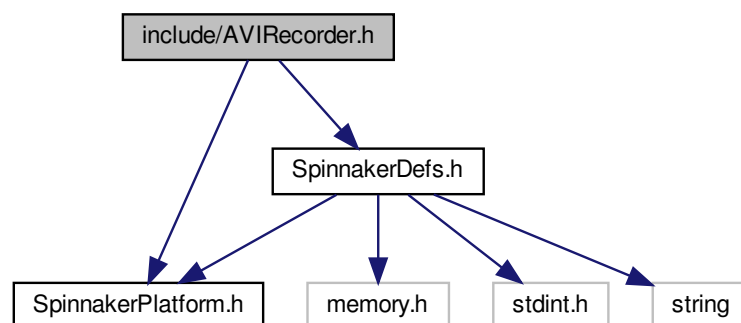
16.8.1 Macro Definition Documentation

16.8.1.1 ADAPTERCONFIG_API

```
#define ADAPTERCONFIG_API __declspec(dllimport)
```

16.9 include/AVIRecorder.h File Reference

Include dependency graph for AVIRecorder.h:



Namespaces

- [Spinnaker](#)

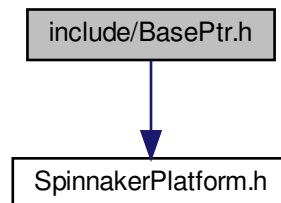
Functions

- class [DEPRECATED_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API [AVIRecorder](#)

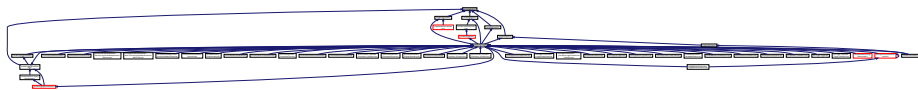
Provides the functionality for the user to record images to an AVI file.

16.10 include/BasePtr.h File Reference

Include dependency graph for BasePtr.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [BasePtr< T, B >](#)

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

Namespaces

- [Spinnaker](#)

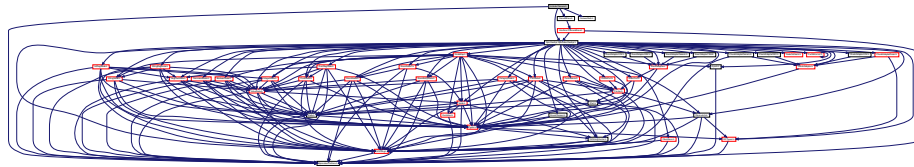
Functions

- `template<class T , class B >`
`bool operator== (const std::nullptr_t, const BasePtr< T, B > &rhs)`

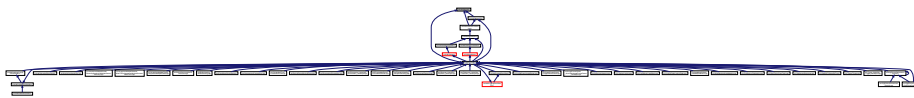
Pointer equal.

16.11 include/Camera.h File Reference

Include dependency graph for Camera.h:



This graph shows which files directly or indirectly include this file:



Classes

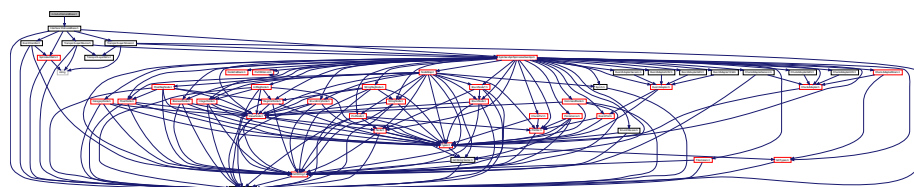
- class [Camera](#)
The camera object class.

Namespaces

- [Spinnaker](#)

16.12 include/CameraBase.h File Reference

Include dependency graph for CameraBase.h:



This graph shows which files directly or indirectly include this file:



Classes

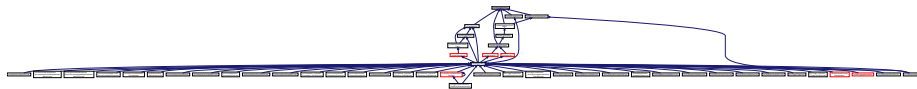
- class [CameraBase](#)
The base class for the camera object.

Namespaces

- [Spinnaker](#)

16.13 include/CameraDefs.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)

Enumerations

- enum [LUTSelectorEnums](#) {
 [LUTSelector_LUT1](#),
 [NUM_LUTSELECTOR](#) }
- *The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.*
- enum [ExposureModeEnums](#) {
 [ExposureMode_Timed](#),
 [ExposureMode_TriggerWidth](#),
 [NUM_EXPOSUREMODE](#) }
- enum [AcquisitionModeEnums](#) {
 [AcquisitionMode_Continuous](#),
 [AcquisitionMode_SingleFrame](#),
 [AcquisitionMode_MultiFrame](#),
 [NUM_ACQUISITIONMODE](#) }
- enum [TriggerSourceEnums](#) {
 [TriggerSource_Software](#),
 [TriggerSource_Line0](#),
 [TriggerSource_Line1](#),
 [TriggerSource_Line2](#),
 [TriggerSource_Line3](#),
 [TriggerSource_UserOutput0](#),
 [TriggerSource_UserOutput1](#),
 [TriggerSource_UserOutput2](#),
 [TriggerSource_UserOutput3](#),
 [TriggerSource_Counter0Start](#),
 [TriggerSource_Counter1Start](#),
 [TriggerSource_Counter0End](#),
 [TriggerSource_Counter1End](#),
 [TriggerSource_LogicBlock0](#),
 [TriggerSource_LogicBlock1](#),
 [TriggerSource_Action0](#),
 [NUM_TRIGGERSOURCE](#) }

- enum [TriggerActivationEnums](#) {
 [TriggerActivation_LevelLow](#),
 [TriggerActivation_LevelHigh](#),
 [TriggerActivation_FallingEdge](#),
 [TriggerActivation_RisingEdge](#),
 [TriggerActivation_AnyEdge](#),
 [NUM_TRIGGERACTIVATION](#) }
- enum [SensorShutterModeEnums](#) {
 [SensorShutterMode_Global](#),
 [SensorShutterMode_Rolling](#),
 [SensorShutterMode_GlobalReset](#),
 [NUM_SENSORSHUTTERMODE](#) }
- enum [TriggerModeEnums](#) {
 [TriggerMode_Off](#),
 [TriggerMode_On](#),
 [NUM_TRIGGERMODE](#) }
- enum [TriggerOverlapEnums](#) {
 [TriggerOverlap_Off](#),
 [TriggerOverlap_ReadOut](#),
 [TriggerOverlap_PreviousFrame](#),
 [NUM_TRIGGEROVERLAP](#) }
- enum [TriggerSelectorEnums](#) {
 [TriggerSelector_AcquisitionStart](#),
 [TriggerSelector_FrameStart](#),
 [TriggerSelector_FrameBurstStart](#),
 [NUM_TRIGGERSELECTOR](#) }
- enum [ExposureAutoEnums](#) {
 [ExposureAuto_Off](#),
 [ExposureAuto_Once](#),
 [ExposureAuto_Continuous](#),
 [NUM_EXPOSUREAUTO](#) }
- enum [EventSelectorEnums](#) {
 [EventSelector_Error](#),
 [EventSelector_ExposureEnd](#),
 [EventSelector_SerialPortReceive](#),
 [NUM_EVENTSELECTOR](#) }
- enum [EventNotificationEnums](#) {
 [EventNotification_On](#),
 [EventNotification_Off](#),
 [NUM_EVENTNOTIFICATION](#) }
- enum [LogicBlockSelectorEnums](#) {
 [LogicBlockSelector_LogicBlock0](#),
 [LogicBlockSelector_LogicBlock1](#),
 [NUM_LOGICBLOCKSELECTOR](#) }
- enum [LogicBlockLUTInputActivationEnums](#) {
 [LogicBlockLUTInputActivation_LevelLow](#),
 [LogicBlockLUTInputActivation_LevelHigh](#),
 [LogicBlockLUTInputActivation_FallingEdge](#),
 [LogicBlockLUTInputActivation_RisingEdge](#),
 [LogicBlockLUTInputActivation_AnyEdge](#),
 [NUM_LOGICBLOCKLUTINPUTACTIVATION](#) }
- enum [LogicBlockLUTInputSelectorEnums](#) {
 [LogicBlockLUTInputSelector_Input0](#),
 [LogicBlockLUTInputSelector_Input1](#),
 [LogicBlockLUTInputSelector_Input2](#),
 [LogicBlockLUTInputSelector_Input3](#),
 [NUM_LOGICBLOCKLUTINPUTSELECTOR](#) }

- enum LogicBlockLUTInputSourceEnums {
 LogicBlockLUTInputSource_Zero,
 LogicBlockLUTInputSource_Line0,
 LogicBlockLUTInputSource_Line1,
 LogicBlockLUTInputSource_Line2,
 LogicBlockLUTInputSource_Line3,
 LogicBlockLUTInputSource_UserOutput0,
 LogicBlockLUTInputSource_UserOutput1,
 LogicBlockLUTInputSource_UserOutput2,
 LogicBlockLUTInputSource_UserOutput3,
 LogicBlockLUTInputSource_Counter0Start,
 LogicBlockLUTInputSource_Counter1Start,
 LogicBlockLUTInputSource_Counter0End,
 LogicBlockLUTInputSource_Counter1End,
 LogicBlockLUTInputSource_LogicBlock0,
 LogicBlockLUTInputSource_LogicBlock1,
 LogicBlockLUTInputSource_ExposureStart,
 LogicBlockLUTInputSource_ExposureEnd,
 LogicBlockLUTInputSource_FrameTriggerWait,
 LogicBlockLUTInputSource_AcquisitionActive,
 NUM_LOGICBLOCKLUTINPUTSOURCE }
- enum LogicBlockLUTSelectorEnums {
 LogicBlockLUTSelector_Value,
 LogicBlockLUTSelector_Enable,
 NUM_LOGICBLOCKLUTSELECTOR }
- enum ColorTransformationSelectorEnums {
 ColorTransformationSelector_RGBtoRGB,
 ColorTransformationSelector_RGBtoYUV,
 NUM_COLORTRANSFORMATIONSELECTOR }
- enum RgbTransformLightSourceEnums {
 RgbTransformLightSource_General,
 RgbTransformLightSource_Tungsten2800K,
 RgbTransformLightSource_WarmFluorescent3000K,
 RgbTransformLightSource_CoolFluorescent4000K,
 RgbTransformLightSource_Daylight5000K,
 RgbTransformLightSource_Cloudy6500K,
 RgbTransformLightSource_Shade8000K,
 RgbTransformLightSource_Custom,
 NUM_RGBTRANSFORMLIGHTSOURCE }
- enum ColorTransformationValueSelectorEnums {
 ColorTransformationValueSelector_Gain00,
 ColorTransformationValueSelector_Gain01,
 ColorTransformationValueSelector_Gain02,
 ColorTransformationValueSelector_Gain10,
 ColorTransformationValueSelector_Gain11,
 ColorTransformationValueSelector_Gain12,
 ColorTransformationValueSelector_Gain20,
 ColorTransformationValueSelector_Gain21,
 ColorTransformationValueSelector_Gain22,
 ColorTransformationValueSelector_Offset0,
 ColorTransformationValueSelector_Offset1,
 ColorTransformationValueSelector_Offset2,
 NUM_COLORTRANSFORMATIONVALUESELECTOR }
- enum DeviceRegistersEndiannessEnums {
 DeviceRegistersEndianness_Little,
 DeviceRegistersEndianness_Big,
 NUM_DEVICEREGISTERSENDIANNES }
- enum DeviceScanTypeEnums {

- DeviceScanType_Areascan,
NUM_DEVICESCANTYPE }
- enum DeviceCharacterSetEnums {
DeviceCharacterSet_UTF8,
DeviceCharacterSet_ASCII,
NUM_DEVICECHARACTERSET }
- enum DeviceTLTypeEnums {
DeviceTLType_GigEVision,
DeviceTLType_CameraLink,
DeviceTLType_CameraLinkHS,
DeviceTLType_CoaXPress,
DeviceTLType_USB3Vision,
DeviceTLType_Custom,
NUM_DEVICETLTYPE }
- enum DevicePowerSupplySelectorEnums {
DevicePowerSupplySelector_External,
NUM_DEVICEPOWERSUPPLYSELECTOR }
- enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICETEMPERATURESELECTOR }
- enum DeviceIndicatorModeEnums {
DeviceIndicatorMode_Inactive,
DeviceIndicatorMode_Active,
DeviceIndicatorMode_ErrorStatus,
NUM_DEVICEINDICATORMODE }
- enum AutoExposureControlPriorityEnums {
AutoExposureControlPriority_Gain,
AutoExposureControlPriority_ExposureTime,
NUM_AUTOEXPOSURECONTROLPRIORITY }
- enum AutoExposureMeteringModeEnums {
AutoExposureMeteringMode_Average,
AutoExposureMeteringMode_Spot,
AutoExposureMeteringMode_Partial,
AutoExposureMeteringMode_CenterWeighted,
AutoExposureMeteringMode_HistogramPeak,
NUM_AUTOEXPOSUREMETERINGMODE }
- enum BalanceWhiteAutoProfileEnums {
BalanceWhiteAutoProfile_Indoor,
BalanceWhiteAutoProfile_Outdoor,
NUM_BALANCEWHITEAUTOPROFILE }
- enum AutoAlgorithmSelectorEnums {
AutoAlgorithmSelector_Awb,
AutoAlgorithmSelector_Ae,
NUM_AUTOALGORITHMSELECTOR }
- enum AutoExposureTargetGreyValueAutoEnums {
AutoExposureTargetGreyValueAuto_Off,
AutoExposureTargetGreyValueAuto_Continuous,
NUM_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum AutoExposureLightingModeEnums {
AutoExposureLightingMode_AutoDetect,
AutoExposureLightingMode_Backlight,
AutoExposureLightingMode_Frontlight,
AutoExposureLightingMode_Normal,
NUM_AUTOEXPOSURELIGHTINGMODE }
- enum GevIEEE1588StatusEnums {
GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,

```

    GevIEEE1588Status_Listening,
    GevIEEE1588Status_PreMaster,
    GevIEEE1588Status_Master,
    GevIEEE1588Status_Passive,
    GevIEEE1588Status_Uncalibrated,
    GevIEEE1588Status_Slave,
    NUM_GEVIEEE1588STATUS }
• enum GevIEEE1588ModeEnums {
    GevIEEE1588Mode_Auto,
    GevIEEE1588Mode_SlaveOnly,
    NUM_GEVIEEE1588MODE }
• enum GevIEEE1588ClockAccuracyEnums {
    GevIEEE1588ClockAccuracy_Unknown,
    NUM_GEVIEEE1588CLOCKACCURACY }
• enum GevCCPEnums {
    GevCCP_OpenAccess,
    GevCCP_ExclusiveAccess,
    GevCCP_ControlAccess,
    NUM_GEVCCP }
• enum GevSupportedOptionSelectorEnums {
    GevSupportedOptionSelector_UserDefinedName,
    GevSupportedOptionSelector_SerialNumber,
    GevSupportedOptionSelector_HeartbeatDisable,
    GevSupportedOptionSelector_LinkSpeed,
    GevSupportedOptionSelector_CCPApplicationSocket,
    GevSupportedOptionSelector_ManifestTable,
    GevSupportedOptionSelector_TestData,
    GevSupportedOptionSelector_DiscoveryAckDelay,
    GevSupportedOptionSelector_DiscoveryAckDelayWritable,
    GevSupportedOptionSelector_ExtendedStatusCodes,
    GevSupportedOptionSelector_Action,
    GevSupportedOptionSelector_PendingAck,
    GevSupportedOptionSelector_EventData,
    GevSupportedOptionSelector_Event,
    GevSupportedOptionSelector_PacketResend,
    GevSupportedOptionSelector_WriteMem,
    GevSupportedOptionSelector_CommandsConcatenation,
    GevSupportedOptionSelector_IPConfigurationLLA,
    GevSupportedOptionSelector_IPConfigurationDHCP,
    GevSupportedOptionSelector_IPConfigurationPersistentIP,
    GevSupportedOptionSelector_StreamChannelSourceSocket,
    GevSupportedOptionSelector_MessageChannelSourceSocket,
    NUM_GEVSUPPORTEDOPTIONSELECTOR }
• enum BlackLevelSelectorEnums {
    BlackLevelSelector_All,
    BlackLevelSelector_Analog,
    BlackLevelSelector_Digital,
    NUM_BLACKLEVELSELECTOR }
• enum BalanceWhiteAutoEnums {
    BalanceWhiteAuto_Off,
    BalanceWhiteAuto_Once,
    BalanceWhiteAuto_Continuous,
    NUM_BALANCEWHITEAUTO }
• enum GainAutoEnums {
    GainAuto_Off,
    GainAuto_Once,
    GainAuto_Continuous,
    NUM_GAINAUTO }

```

- enum [BalanceRatioSelectorEnums](#) {
 [BalanceRatioSelector_Red](#),
 [BalanceRatioSelector_Blue](#),
 [NUM_BALANCERATIOSELECTOR](#) }
- enum [GainSelectorEnums](#) {
 [GainSelector_All](#),
 [NUM_GAINSELECTOR](#) }
- enum [DefectCorrectionModeEnums](#) {
 [DefectCorrectionMode_Average](#),
 [DefectCorrectionMode_Highlight](#),
 [DefectCorrectionMode_Zero](#),
 [NUM_DEFECTCORRECTIONMODE](#) }
- enum [UserSetSelectorEnums](#) {
 [UserSetSelector_Default](#),
 [UserSetSelector_UserSet0](#),
 [UserSetSelector_UserSet1](#),
 [NUM_USERSETSELECTOR](#) }
- enum [UserSetDefaultEnums](#) {
 [UserSetDefault_Default](#),
 [UserSetDefault_UserSet0](#),
 [UserSetDefault_UserSet1](#),
 [NUM_USERSETDEFAULT](#) }
- enum [SerialPortBaudRateEnums](#) {
 [SerialPortBaudRate_Baud300](#),
 [SerialPortBaudRate_Baud600](#),
 [SerialPortBaudRate_Baud1200](#),
 [SerialPortBaudRate_Baud2400](#),
 [SerialPortBaudRate_Baud4800](#),
 [SerialPortBaudRate_Baud9600](#),
 [SerialPortBaudRate_Baud14400](#),
 [SerialPortBaudRate_Baud19200](#),
 [SerialPortBaudRate_Baud38400](#),
 [SerialPortBaudRate_Baud57600](#),
 [SerialPortBaudRate_Baud115200](#),
 [SerialPortBaudRate_Baud230400](#),
 [SerialPortBaudRate_Baud460800](#),
 [SerialPortBaudRate_Baud921600](#),
 [NUM_SERIALPORTBAUDRATE](#) }
- enum [SerialPortParityEnums](#) {
 [SerialPortParity_None](#),
 [SerialPortParity_Odd](#),
 [SerialPortParity_Even](#),
 [SerialPortParity_Mark](#),
 [SerialPortParity_Space](#),
 [NUM_SERIALPORTPARITY](#) }
- enum [SerialPortSelectorEnums](#) {
 [SerialPortSelector_SerialPort0](#),
 [NUM_SERIALPORTSELECTOR](#) }
- enum [SerialPortStopBitsEnums](#) {
 [SerialPortStopBits_Bits1](#),
 [SerialPortStopBits_Bits1AndAHalf](#),
 [SerialPortStopBits_Bits2](#),
 [NUM_SERIALPORTSTOPBITS](#) }
- enum [SerialPortSourceEnums](#) {
 [SerialPortSource_Line0](#),
 [SerialPortSource_Line1](#),
 [SerialPortSource_Line2](#),
 [SerialPortSource_Line3](#),

```
SerialPortSource_Off,  
NUM_SERIALPORTSOURCE }  
• enum SequencerModeEnums {  
    SequencerMode_Off,  
    SequencerMode_On,  
    NUM_SEQUENCERMODE }  
• enum SequencerConfigurationValidEnums {  
    SequencerConfigurationValid_No,  
    SequencerConfigurationValid_Yes,  
    NUM_SEQUENCERCONFIGURATIONVALID }  
• enum SequencerSetValidEnums {  
    SequencerSetValid_No,  
    SequencerSetValid_Yes,  
    NUM_SEQUENCERSETVALID }  
• enum SequencerTriggerActivationEnums {  
    SequencerTriggerActivation_RisingEdge,  
    SequencerTriggerActivation_FallingEdge,  
    SequencerTriggerActivation_AnyEdge,  
    SequencerTriggerActivation_LevelHigh,  
    SequencerTriggerActivation_LevelLow,  
    NUM_SEQUENCERTRIGGERACTIVATION }  
• enum SequencerConfigurationModeEnums {  
    SequencerConfigurationMode_Off,  
    SequencerConfigurationMode_On,  
    NUM_SEQUENCERCONFIGURATIONMODE }  
• enum SequencerTriggerSourceEnums {  
    SequencerTriggerSource_Off,  
    SequencerTriggerSource_FrameStart,  
    NUM_SEQUENCERTRIGGERSOURCE }  
• enum TransferQueueModeEnums {  
    TransferQueueMode_FirstInFirstOut,  
    NUM_TRANSFERQUEUEMODE }  
• enum TransferOperationModeEnums {  
    TransferOperationMode_Continuous,  
    TransferOperationMode_MultiBlock,  
    NUM_TRANSFEROPERATIONMODE }  
• enum TransferControlModeEnums {  
    TransferControlMode_Basic,  
    TransferControlMode_Automatic,  
    TransferControlMode_UserControlled,  
    NUM_TRANSFERCONTROLMODE }  
• enum ChunkGainSelectorEnums {  
    ChunkGainSelector_All,  
    ChunkGainSelector_Red,  
    ChunkGainSelector_Green,  
    ChunkGainSelector_Blue,  
    NUM_CHUNKGAINSELECTOR }  
• enum ChunkSelectorEnums {  
    ChunkSelector_Image,  
    ChunkSelector_CRC,  
    ChunkSelector_FrameID,  
    ChunkSelector_OffsetX,  
    ChunkSelector_OffsetY,  
    ChunkSelector_Width,  
    ChunkSelector_Height,  
    ChunkSelector_ExposureTime,  
    ChunkSelector_Gain,  
    ChunkSelector_BlackLevel,
```

```
ChunkSelector_PixelFormat,  
ChunkSelector_Timestamp,  
ChunkSelector_SequencerSetActive,  
ChunkSelector_SerialData,  
ChunkSelector_ExposureEndLineStatusAll,  
NUM_CHUNKSELECTOR }  
• enum ChunkBlackLevelSelectorEnums {  
    ChunkBlackLevelSelector_All,  
    NUM_CHUNKBLACKLEVELSELECTOR }  
• enum ChunkPixelFormatEnums {  
    ChunkPixelFormat_Mono8,  
    ChunkPixelFormat_Mono12Packed,  
    ChunkPixelFormat_Mono16,  
    ChunkPixelFormat_RGB8Packed,  
    ChunkPixelFormat_YUV422Packed,  
    ChunkPixelFormat_BayerGR8,  
    ChunkPixelFormat_BayerRG8,  
    ChunkPixelFormat_BayerGB8,  
    ChunkPixelFormat_BayerBG8,  
    ChunkPixelFormat_YCbCr601_422_8_CbYCrY,  
    NUM_CHUNKPIXELFORMAT }  
• enum FileOperationStatusEnums {  
    FileOperationStatus_Success,  
    FileOperationStatus_Failure,  
    FileOperationStatus_Overflow,  
    NUM_FILEOPERATIONSTATUS }  
• enum FileOpenModeEnums {  
    FileOpenMode_Read,  
    FileOpenMode_Write,  
    FileOpenMode_ReadWrite,  
    NUM_FILEOPENMODE }  
• enum FileOperationSelectorEnums {  
    FileOperationSelector_Open,  
    FileOperationSelector_Close,  
    FileOperationSelector_Read,  
    FileOperationSelector_Write,  
    FileOperationSelector_Delete,  
    NUM_FILEOPERATIONSELECTOR }  
• enum FileSelectorEnums {  
    FileSelector_UserSetDefault,  
    FileSelector_UserSet0,  
    FileSelector_UserSet1,  
    FileSelector_UserFile1,  
    FileSelector_SerialPort0,  
    NUM_FILESELECTOR }  
• enum BinningSelectorEnums {  
    BinningSelector_All,  
    BinningSelector_Sensor,  
    BinningSelector_ISP,  
    NUM_BINNINGSELECTOR }  
• enum TestPatternGeneratorSelectorEnums {  
    TestPatternGeneratorSelector_Sensor,  
    TestPatternGeneratorSelector_PipelineStart,  
    NUM_TESTPATTERNGENERATORSELECTOR }  
• enum CompressionSaturationPriorityEnums {  
    CompressionSaturationPriority_DropFrame,  
    CompressionSaturationPriority_ReduceFrameRate,  
    NUM_COMPRESSIONSATURATIONPRIORITY }
```

- enum [TestPatternEnums](#) {
 [TestPattern_Off](#),
 [TestPattern_Increment](#),
 [TestPattern_SensorTestPattern](#),
 [NUM_TESTPATTERN](#) }
- enum [PixelColorFilterEnums](#) {
 [PixelColorFilter_None](#),
 [PixelColorFilter_BayerRG](#),
 [PixelColorFilter_BayerGB](#),
 [PixelColorFilter_BayerGR](#),
 [PixelColorFilter_BayerBG](#),
 [NUM_PIXELCOLORFILTER](#) }
- enum [AdcBitDepthEnums](#) {
 [AdcBitDepth_Bit8](#),
 [AdcBitDepth_Bit10](#),
 [AdcBitDepth_Bit12](#),
 [AdcBitDepth_Bit14](#),
 [NUM_ADCBITDEPTH](#) }
- enum [DecimationHorizontalModeEnums](#) {
 [DecimationHorizontalMode_Discard](#),
 [NUM_DECIMATIONHORIZONTALMODE](#) }
- enum [BinningVerticalModeEnums](#) {
 [BinningVerticalMode_Sum](#),
 [BinningVerticalMode_Average](#),
 [NUM_BINNINGVERTICALMODE](#) }
- enum [PixelSizeEnums](#) {
 [PixelSize_Bpp1](#),
 [PixelSize_Bpp2](#),
 [PixelSize_Bpp4](#),
 [PixelSize_Bpp8](#),
 [PixelSize_Bpp10](#),
 [PixelSize_Bpp12](#),
 [PixelSize_Bpp14](#),
 [PixelSize_Bpp16](#),
 [PixelSize_Bpp20](#),
 [PixelSize_Bpp24](#),
 [PixelSize_Bpp30](#),
 [PixelSize_Bpp32](#),
 [PixelSize_Bpp36](#),
 [PixelSize_Bpp48](#),
 [PixelSize_Bpp64](#),
 [PixelSize_Bpp96](#),
 [NUM_PIXELSIZE](#) }
- enum [DecimationSelectorEnums](#) {
 [DecimationSelector_All](#),
 [DecimationSelector_Sensor](#),
 [NUM_DECIMATIONSELECTOR](#) }
- enum [ImageCompressionModeEnums](#) {
 [ImageCompressionMode_Off](#),
 [ImageCompressionMode_Lossless](#),
 [NUM_IMAGECOMPRESSIONMODE](#) }
- enum [BinningHorizontalModeEnums](#) {
 [BinningHorizontalMode_Sum](#),
 [BinningHorizontalMode_Average](#),
 [NUM_BINNINGHORIZONTALMODE](#) }
- enum [PixelFormatEnums](#) {
 [PixelFormat_Mono8](#),
 [PixelFormat_Mono16](#),

PixelFormat_RGB8Packed,
PixelFormat_BayerGR8,
PixelFormat_BayerRG8,
PixelFormat_BayerGB8,
PixelFormat_BayerBG8,
PixelFormat_BayerGR16,
PixelFormat_BayerRG16,
PixelFormat_BayerGB16,
PixelFormat_BayerBG16,
PixelFormat_Mono12Packed,
PixelFormat_BayerGR12Packed,
PixelFormat_BayerRG12Packed,
PixelFormat_BayerGB12Packed,
PixelFormat_BayerBG12Packed,
PixelFormat_YUV411Packed,
PixelFormat_YUV422Packed,
PixelFormat_YUV444Packed,
PixelFormat_Mono12p,
PixelFormat_BayerGR12p,
PixelFormat_BayerRG12p,
PixelFormat_BayerGB12p,
PixelFormat_BayerBG12p,
PixelFormat_YCbCr8,
PixelFormat_YCbCr422_8,
PixelFormat_YCbCr411_8,
PixelFormat_BGR8,
PixelFormat_BGRa8,
PixelFormat_Mono10Packed,
PixelFormat_BayerGR10Packed,
PixelFormat_BayerRG10Packed,
PixelFormat_BayerGB10Packed,
PixelFormat_BayerBG10Packed,
PixelFormat_Mono10p,
PixelFormat_BayerGR10p,
PixelFormat_BayerRG10p,
PixelFormat_BayerGB10p,
PixelFormat_BayerBG10p,
PixelFormat_Mono1p,
PixelFormat_Mono2p,
PixelFormat_Mono4p,
PixelFormat_Mono8s,
PixelFormat_Mono10,
PixelFormat_Mono12,
PixelFormat_Mono14,
PixelFormat_Mono16s,
PixelFormat_Mono32f,
PixelFormat_BayerBG10,
PixelFormat_BayerBG12,
PixelFormat_BayerGB10,
PixelFormat_BayerGB12,
PixelFormat_BayerGR10,
PixelFormat_BayerGR12,
PixelFormat_BayerRG10,
PixelFormat_BayerRG12,
PixelFormat_RGBa8,
PixelFormat_RGBa10,
PixelFormat_RGBa10p,
PixelFormat_RGBa12,

PixelFormat_RGBA12p,
PixelFormat_RGBA14,
PixelFormat_RGBA16,
PixelFormat_RGB8,
PixelFormat_RGB8_Planar,
PixelFormat_RGB10,
PixelFormat_RGB10_Planar,
PixelFormat_RGB10p,
PixelFormat_RGB10p32,
PixelFormat_RGB12,
PixelFormat_RGB12_Planar,
PixelFormat_RGB12p,
PixelFormat_RGB14,
PixelFormat_RGB16,
PixelFormat_RGB16s,
PixelFormat_RGB32f,
PixelFormat_RGB16_Planar,
PixelFormat_RGB565p,
PixelFormat_BGRA10,
PixelFormat_BGRA10p,
PixelFormat_BGRA12,
PixelFormat_BGRA12p,
PixelFormat_BGRA14,
PixelFormat_BGRA16,
PixelFormat_RGBA32f,
PixelFormat_BGR10,
PixelFormat_BGR10p,
PixelFormat_BGR12,
PixelFormat_BGR12p,
PixelFormat_BGR14,
PixelFormat_BGR16,
PixelFormat_BGR565p,
PixelFormat_R8,
PixelFormat_R10,
PixelFormat_R12,
PixelFormat_R16,
PixelFormat_G8,
PixelFormat_G10,
PixelFormat_G12,
PixelFormat_G16,
PixelFormat_B8,
PixelFormat_B10,
PixelFormat_B12,
PixelFormat_B16,
PixelFormat_Coord3D_ABC8,
PixelFormat_Coord3D_ABC8_Planar,
PixelFormat_Coord3D_ABC10p,
PixelFormat_Coord3D_ABC10p_Planar,
PixelFormat_Coord3D_ABC12p,
PixelFormat_Coord3D_ABC12p_Planar,
PixelFormat_Coord3D_ABC16,
PixelFormat_Coord3D_ABC16_Planar,
PixelFormat_Coord3D_ABC32f,
PixelFormat_Coord3D_ABC32f_Planar,
PixelFormat_Coord3D_AC8,
PixelFormat_Coord3D_AC8_Planar,
PixelFormat_Coord3D_AC10p,
PixelFormat_Coord3D_AC10p_Planar,

PixelFormat_Coord3D_AC12p,
PixelFormat_Coord3D_AC12p_Planar,
PixelFormat_Coord3D_AC16,
PixelFormat_Coord3D_AC16_Planar,
PixelFormat_Coord3D_AC32f,
PixelFormat_Coord3D_AC32f_Planar,
PixelFormat_Coord3D_A8,
PixelFormat_Coord3D_A10p,
PixelFormat_Coord3D_A12p,
PixelFormat_Coord3D_A16,
PixelFormat_Coord3D_A32f,
PixelFormat_Coord3D_B8,
PixelFormat_Coord3D_B10p,
PixelFormat_Coord3D_B12p,
PixelFormat_Coord3D_B16,
PixelFormat_Coord3D_B32f,
PixelFormat_Coord3D_C8,
PixelFormat_Coord3D_C10p,
PixelFormat_Coord3D_C12p,
PixelFormat_Coord3D_C16,
PixelFormat_Coord3D_C32f,
PixelFormat_Confidence1,
PixelFormat_Confidence1p,
PixelFormat_Confidence8,
PixelFormat_Confidence16,
PixelFormat_Confidence32f,
PixelFormat_BiColorBGRG8,
PixelFormat_BiColorBGRG10,
PixelFormat_BiColorBGRG10p,
PixelFormat_BiColorBGRG12,
PixelFormat_BiColorBGRG12p,
PixelFormat_BiColorRGBG8,
PixelFormat_BiColorRGBG10,
PixelFormat_BiColorRGBG10p,
PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,
PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,

PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,
PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,
PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,
PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,
PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,
PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,

- PixelFormat_Polarized10p,
- PixelFormat_Polarized12p,
- PixelFormat_Polarized16,
- PixelFormat_BayerRGPolarized8,
- PixelFormat_BayerRGPolarized10p,
- PixelFormat_BayerRGPolarized12p,
- PixelFormat_BayerRGPolarized16,
- PixelFormat_LLCMono8,
- PixelFormat_LLCBayerRG8,
- PixelFormat_JPEGMono8,
- PixelFormat_JPEGColor8,
- PixelFormat_Raw16,
- PixelFormat_Raw8,
- PixelFormat_R12_Jpeg,
- PixelFormat_GR12_Jpeg,
- PixelFormat_GB12_Jpeg,
- PixelFormat_B12_Jpeg,
- UNKNOWN_PIXELFORMAT,
- NUM_PIXELFORMAT }
- enum DecimationVerticalModeEnums {
DecimationVerticalMode_Discard,
NUM_DECIMATIONVERTICALMODE }
- enum LineModeEnums {
LineMode_Input,
LineMode_Output,
NUM_LINEMODE }
- enum LineSourceEnums {
LineSource_Off,
LineSource_Line0,
LineSource_Line1,
LineSource_Line2,
LineSource_Line3,
LineSource_UserOutput0,
LineSource_UserOutput1,
LineSource_UserOutput2,
LineSource_UserOutput3,
LineSource_Counter0Active,
LineSource_Counter1Active,
LineSource_LogicBlock0,
LineSource_LogicBlock1,
LineSource_ExposureActive,
LineSource_FrameTriggerWait,
LineSource_SerialPort0,
LineSource_PPSSignal,
LineSource_AllPixel,
LineSource_AnyPixel,
NUM_LINESOURCE }
- enum LineInputFilterSelectorEnums {
LineInputFilterSelector_Deglitch,
LineInputFilterSelector_Debounce,
NUM_LINEINPUTFILTERSELECTOR }
- enum UserOutputSelectorEnums {
UserOutputSelector_UserOutput0,
UserOutputSelector_UserOutput1,
UserOutputSelector_UserOutput2,
UserOutputSelector_UserOutput3,
NUM_USEROUTPUTSELECTOR }
- enum LineFormatEnums {

```
LineFormat_NoConnect,
LineFormat_TriState,
LineFormat_TTL,
LineFormat_LVDS,
LineFormat_RS422,
LineFormat_OptoCoupled,
LineFormat_OpenDrain,
NUM_LINEFORMAT }

• enum LineSelectorEnums {
    LineSelector_Line0,
    LineSelector_Line1,
    LineSelector_Line2,
    LineSelector_Line3,
    NUM_LINESELECTOR }

• enum ExposureActiveModeEnums {
    ExposureActiveMode_Line1,
    ExposureActiveMode_AnyPixels,
    ExposureActiveMode_AllPixels,
    NUM_EXPOSUREACTIVEMODE }

• enum CounterTriggerActivationEnums {
    CounterTriggerActivation_LevelLow,
    CounterTriggerActivation_LevelHigh,
    CounterTriggerActivation_FallingEdge,
    CounterTriggerActivation_RisingEdge,
    CounterTriggerActivation_AnyEdge,
    NUM_COUNTERTRIGGERACTIVATION }

• enum CounterSelectorEnums {
    CounterSelector_Counter0,
    CounterSelector_Counter1,
    NUM_COUNTERSELECTOR }

• enum CounterStatusEnums {
    CounterStatus_CounterIdle,
    CounterStatus_CounterTriggerWait,
    CounterStatus_CounterActive,
    CounterStatus_CounterCompleted,
    CounterStatus_CounterOverflow,
    NUM_COUNTERSTATUS }

• enum CounterTriggerSourceEnums {
    CounterTriggerSource_Off,
    CounterTriggerSource_Line0,
    CounterTriggerSource_Line1,
    CounterTriggerSource_Line2,
    CounterTriggerSource_Line3,
    CounterTriggerSource_UserOutput0,
    CounterTriggerSource_UserOutput1,
    CounterTriggerSource_UserOutput2,
    CounterTriggerSource_UserOutput3,
    CounterTriggerSource_Counter0Start,
    CounterTriggerSource_Counter1Start,
    CounterTriggerSource_Counter0End,
    CounterTriggerSource_Counter1End,
    CounterTriggerSource_LogicBlock0,
    CounterTriggerSource_LogicBlock1,
    CounterTriggerSource_ExposureStart,
    CounterTriggerSource_ExposureEnd,
    CounterTriggerSource_FrameTriggerWait,
    NUM_COUNTERTRIGGERSOURCE }
```

- enum CounterResetSourceEnums {
CounterResetSource_Off,
CounterResetSource_Line0,
CounterResetSource_Line1,
CounterResetSource_Line2,
CounterResetSource_Line3,
CounterResetSource_UserOutput0,
CounterResetSource_UserOutput1,
CounterResetSource_UserOutput2,
CounterResetSource_UserOutput3,
CounterResetSource_Counter0Start,
CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }
- enum CounterEventSourceEnums {
CounterEventSource_Off,
CounterEventSource_MHzTick,
CounterEventSource_Line0,
CounterEventSource_Line1,
CounterEventSource_Line2,
CounterEventSource_Line3,
CounterEventSource_UserOutput0,
CounterEventSource_UserOutput1,
CounterEventSource_UserOutput2,
CounterEventSource_UserOutput3,
CounterEventSource_Counter0Start,
CounterEventSource_Counter1Start,
CounterEventSource_Counter0End,
CounterEventSource_Counter1End,
CounterEventSource_LogicBlock0,
CounterEventSource_LogicBlock1,
CounterEventSource_ExposureStart,
CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }
- enum CounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }
- enum CounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }
- enum DeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,

```
DeviceType_Transceiver,  
DeviceType_Peripheral,  
NUM_DEVICETYPE }  
  
• enum DeviceConnectionStatusEnums {  
    DeviceConnectionStatus_Active,  
    DeviceConnectionStatus_Inactive,  
    NUM_DEVICECONNECTIONSTATUS }  
  
• enum DeviceLinkThroughputLimitModeEnums {  
    DeviceLinkThroughputLimitMode_On,  
    DeviceLinkThroughputLimitMode_Off,  
    NUM_DEVICELINKTHROUGHPUTLIMITMODE }  
  
• enum DeviceLinkHeartbeatModeEnums {  
    DeviceLinkHeartbeatMode_On,  
    DeviceLinkHeartbeatMode_Off,  
    NUM_DEVICELINKHEARTBEATMODE }  
  
• enum DeviceStreamChannelTypeEnums {  
    DeviceStreamChannelType_Transmitter,  
    DeviceStreamChannelType_Receiver,  
    NUM_DEVICESTREAMCHANNELTYPE }  
  
• enum DeviceStreamChannelEndiannessEnums {  
    DeviceStreamChannelEndianness_Big,  
    DeviceStreamChannelEndianness_Little,  
    NUM_DEVICESTREAMCHANNELENDIANNESS }  
  
• enum DeviceClockSelectorEnums {  
    DeviceClockSelector_Sensor,  
    DeviceClockSelector_SensorDigitization,  
    DeviceClockSelector_CameraLink,  
    NUM_DEVICECLOCKSELECTOR }  
  
• enum DeviceSerialPortSelectorEnums {  
    DeviceSerialPortSelector_CameraLink,  
    NUM_DEVICESERIALPORTSELECTOR }  
  
• enum DeviceSerialPortBaudRateEnums {  
    DeviceSerialPortBaudRate_Baud9600,  
    DeviceSerialPortBaudRate_Baud19200,  
    DeviceSerialPortBaudRate_Baud38400,  
    DeviceSerialPortBaudRate_Baud57600,  
    DeviceSerialPortBaudRate_Baud115200,  
    DeviceSerialPortBaudRate_Baud230400,  
    DeviceSerialPortBaudRate_Baud460800,  
    DeviceSerialPortBaudRate_Baud921600,  
    NUM_DEVICESERIALPORTBAUDRATE }  
  
• enum SensorTapsEnums {  
    SensorTaps_One,  
    SensorTaps_Two,  
    SensorTaps_Three,  
    SensorTaps_Four,  
    SensorTaps_Eight,  
    SensorTaps_Ten,  
    NUM_SENSORTAPS }  
  
• enum SensorDigitizationTapsEnums {  
    SensorDigitizationTaps_One,  
    SensorDigitizationTaps_Two,  
    SensorDigitizationTaps_Three,  
    SensorDigitizationTaps_Four,  
    SensorDigitizationTaps_Eight,  
    SensorDigitizationTaps_Ten,  
    NUM_SENSORDIGITIZATIONTAPS }
```


- enum [RegionSelectorEnums](#) {
[RegionSelector_Region0](#),
[RegionSelector_Region1](#),
[RegionSelector_Region2](#),
[RegionSelector_All](#),
[NUM_REGIONSELECTOR](#) }
- enum [RegionModeEnums](#) {
[RegionMode_Off](#),
[RegionMode_On](#),
[NUM_REGIONMODE](#) }
- enum [RegionDestinationEnums](#) {
[RegionDestination_Stream0](#),
[RegionDestination_Stream1](#),
[RegionDestination_Stream2](#),
[NUM_REGIONDESTINATION](#) }
- enum [ImageComponentSelectorEnums](#) {
[ImageComponentSelector_Intensity](#),
[ImageComponentSelector_Color](#),
[ImageComponentSelector_Infrared](#),
[ImageComponentSelector_Ultraviolet](#),
[ImageComponentSelector_Range](#),
[ImageComponentSelector_Disparity](#),
[ImageComponentSelector_Confidence](#),
[ImageComponentSelector_Scatter](#),
[NUM_IMAGECOMPONENTSELECTOR](#) }
- enum [PixelFormatInfoSelectorEnums](#) {
[PixelFormatInfoSelector_Mono1p](#),
[PixelFormatInfoSelector_Mono2p](#),
[PixelFormatInfoSelector_Mono4p](#),
[PixelFormatInfoSelector_Mono8](#),
[PixelFormatInfoSelector_Mono8s](#),
[PixelFormatInfoSelector_Mono10](#),
[PixelFormatInfoSelector_Mono10p](#),
[PixelFormatInfoSelector_Mono12](#),
[PixelFormatInfoSelector_Mono12p](#),
[PixelFormatInfoSelector_Mono14](#),
[PixelFormatInfoSelector_Mono16](#),
[PixelFormatInfoSelector_Mono16s](#),
[PixelFormatInfoSelector_Mono32f](#),
[PixelFormatInfoSelector_BayerBG8](#),
[PixelFormatInfoSelector_BayerBG10](#),
[PixelFormatInfoSelector_BayerBG10p](#),
[PixelFormatInfoSelector_BayerBG12](#),
[PixelFormatInfoSelector_BayerBG12p](#),
[PixelFormatInfoSelector_BayerBG16](#),
[PixelFormatInfoSelector_BayerGB8](#),
[PixelFormatInfoSelector_BayerGB10](#),
[PixelFormatInfoSelector_BayerGB10p](#),
[PixelFormatInfoSelector_BayerGB12](#),
[PixelFormatInfoSelector_BayerGB12p](#),
[PixelFormatInfoSelector_BayerGB16](#),
[PixelFormatInfoSelector_BayerGR8](#),
[PixelFormatInfoSelector_BayerGR10](#),
[PixelFormatInfoSelector_BayerGR10p](#),
[PixelFormatInfoSelector_BayerGR12](#),
[PixelFormatInfoSelector_BayerGR12p](#),
[PixelFormatInfoSelector_BayerGR16](#),
[PixelFormatInfoSelector_BayerRG8](#),

[PixelFormatInfoSelector_BayerRG10,](#)
[PixelFormatInfoSelector_BayerRG10p,](#)
[PixelFormatInfoSelector_BayerRG12,](#)
[PixelFormatInfoSelector_BayerRG12p,](#)
[PixelFormatInfoSelector_BayerRG16,](#)
[PixelFormatInfoSelector_RGBa8,](#)
[PixelFormatInfoSelector_RGBa10,](#)
[PixelFormatInfoSelector_RGBa10p,](#)
[PixelFormatInfoSelector_RGBa12,](#)
[PixelFormatInfoSelector_RGBa12p,](#)
[PixelFormatInfoSelector_RGBa14,](#)
[PixelFormatInfoSelector_RGBa16,](#)
[PixelFormatInfoSelector_RGB8,](#)
[PixelFormatInfoSelector_RGB8_Planar,](#)
[PixelFormatInfoSelector_RGB10,](#)
[PixelFormatInfoSelector_RGB10_Planar,](#)
[PixelFormatInfoSelector_RGB10p,](#)
[PixelFormatInfoSelector_RGB10p32,](#)
[PixelFormatInfoSelector_RGB12,](#)
[PixelFormatInfoSelector_RGB12_Planar,](#)
[PixelFormatInfoSelector_RGB12p,](#)
[PixelFormatInfoSelector_RGB14,](#)
[PixelFormatInfoSelector_RGB16,](#)
[PixelFormatInfoSelector_RGB16s,](#)
[PixelFormatInfoSelector_RGB32f,](#)
[PixelFormatInfoSelector_RGB16_Planar,](#)
[PixelFormatInfoSelector_RGB565p,](#)
[PixelFormatInfoSelector_BGRa8,](#)
[PixelFormatInfoSelector_BGRa10,](#)
[PixelFormatInfoSelector_BGRa10p,](#)
[PixelFormatInfoSelector_BGRa12,](#)
[PixelFormatInfoSelector_BGRa12p,](#)
[PixelFormatInfoSelector_BGRa14,](#)
[PixelFormatInfoSelector_BGRa16,](#)
[PixelFormatInfoSelector_RGBa32f,](#)
[PixelFormatInfoSelector_BGR8,](#)
[PixelFormatInfoSelector_BGR10,](#)
[PixelFormatInfoSelector_BGR10p,](#)
[PixelFormatInfoSelector_BGR12,](#)
[PixelFormatInfoSelector_BGR12p,](#)
[PixelFormatInfoSelector_BGR14,](#)
[PixelFormatInfoSelector_BGR16,](#)
[PixelFormatInfoSelector_BGR565p,](#)
[PixelFormatInfoSelector_R8,](#)
[PixelFormatInfoSelector_R10,](#)
[PixelFormatInfoSelector_R12,](#)
[PixelFormatInfoSelector_R16,](#)
[PixelFormatInfoSelector_G8,](#)
[PixelFormatInfoSelector_G10,](#)
[PixelFormatInfoSelector_G12,](#)
[PixelFormatInfoSelector_G16,](#)
[PixelFormatInfoSelector_B8,](#)
[PixelFormatInfoSelector_B10,](#)
[PixelFormatInfoSelector_B12,](#)
[PixelFormatInfoSelector_B16,](#)
[PixelFormatInfoSelector_Coord3D_ABC8,](#)
[PixelFormatInfoSelector_Coord3D_ABC8_Planar,](#)
[PixelFormatInfoSelector_Coord3D_ABC10p,](#)

[PixelFormatInfoSelector_Coord3D_ABC10p_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC12p](#),
[PixelFormatInfoSelector_Coord3D_ABC12p_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC16](#),
[PixelFormatInfoSelector_Coord3D_ABC16_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC32f](#),
[PixelFormatInfoSelector_Coord3D_ABC32f_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC8](#),
[PixelFormatInfoSelector_Coord3D_AC8_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC10p](#),
[PixelFormatInfoSelector_Coord3D_AC10p_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC12p](#),
[PixelFormatInfoSelector_Coord3D_AC12p_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC16](#),
[PixelFormatInfoSelector_Coord3D_AC16_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC32f](#),
[PixelFormatInfoSelector_Coord3D_AC32f_Planar](#),
[PixelFormatInfoSelector_Coord3D_A8](#),
[PixelFormatInfoSelector_Coord3D_A10p](#),
[PixelFormatInfoSelector_Coord3D_A12p](#),
[PixelFormatInfoSelector_Coord3D_A16](#),
[PixelFormatInfoSelector_Coord3D_A32f](#),
[PixelFormatInfoSelector_Coord3D_B8](#),
[PixelFormatInfoSelector_Coord3D_B10p](#),
[PixelFormatInfoSelector_Coord3D_B12p](#),
[PixelFormatInfoSelector_Coord3D_B16](#),
[PixelFormatInfoSelector_Coord3D_B32f](#),
[PixelFormatInfoSelector_Coord3D_C8](#),
[PixelFormatInfoSelector_Coord3D_C10p](#),
[PixelFormatInfoSelector_Coord3D_C12p](#),
[PixelFormatInfoSelector_Coord3D_C16](#),
[PixelFormatInfoSelector_Coord3D_C32f](#),
[PixelFormatInfoSelector_Confidence1](#),
[PixelFormatInfoSelector_Confidence1p](#),
[PixelFormatInfoSelector_Confidence8](#),
[PixelFormatInfoSelector_Confidence16](#),
[PixelFormatInfoSelector_Confidence32f](#),
[PixelFormatInfoSelector_BiColorBGRG8](#),
[PixelFormatInfoSelector_BiColorBGRG10](#),
[PixelFormatInfoSelector_BiColorBGRG10p](#),
[PixelFormatInfoSelector_BiColorBGRG12](#),
[PixelFormatInfoSelector_BiColorBGRG12p](#),
[PixelFormatInfoSelector_BiColorRGBG8](#),
[PixelFormatInfoSelector_BiColorRGBG10](#),
[PixelFormatInfoSelector_BiColorRGBG10p](#),
[PixelFormatInfoSelector_BiColorRGBG12](#),
[PixelFormatInfoSelector_BiColorRGBG12p](#),
[PixelFormatInfoSelector_SCF1WBWG8](#),
[PixelFormatInfoSelector_SCF1WBWG10](#),
[PixelFormatInfoSelector_SCF1WBWG10p](#),
[PixelFormatInfoSelector_SCF1WBWG12](#),
[PixelFormatInfoSelector_SCF1WBWG12p](#),
[PixelFormatInfoSelector_SCF1WBWG14](#),
[PixelFormatInfoSelector_SCF1WBWG16](#),
[PixelFormatInfoSelector_SCF1WGWB8](#),
[PixelFormatInfoSelector_SCF1WGWB10](#),
[PixelFormatInfoSelector_SCF1WGWB10p](#),
[PixelFormatInfoSelector_SCF1WGWB12](#),

PixelFormatInfoSelector_SCF1WGWB12p,
PixelFormatInfoSelector_SCF1WGWB14,
PixelFormatInfoSelector_SCF1WGWB16,
PixelFormatInfoSelector_SCF1WGWR8,
PixelFormatInfoSelector_SCF1WGWR10,
PixelFormatInfoSelector_SCF1WGWR10p,
PixelFormatInfoSelector_SCF1WGWR12,
PixelFormatInfoSelector_SCF1WGWR12p,
PixelFormatInfoSelector_SCF1WGWR14,
PixelFormatInfoSelector_SCF1WGWR16,
PixelFormatInfoSelector_SCF1WRWG8,
PixelFormatInfoSelector_SCF1WRWG10,
PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,
PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,

```

PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

• enum DeinterlacingEnums {
    Deinterlacing_Off,
    Deinterlacing_LineDuplication,
    Deinterlacing_Weave,
    NUM_DEINTERLACING }

• enum ImageCompressionRateOptionEnums {
    ImageCompressionRateOption_FixBitrate,
    ImageCompressionRateOption_FixQuality,
    NUM_IMAGECOMPRESSIONRATEOPTION }

• enum ImageCompressionJPEGFormatOptionEnums {
    ImageCompressionJPEGFormatOption_Lossless,
    ImageCompressionJPEGFormatOption_BaselineStandard,
    ImageCompressionJPEGFormatOption_BaselineOptimized,
    ImageCompressionJPEGFormatOption_Progressive,
    NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }

• enum AcquisitionStatusSelectorEnums {
    AcquisitionStatusSelector_AcquisitionTriggerWait,
    AcquisitionStatusSelector_AcquisitionActive,
    AcquisitionStatusSelector_AcquisitionTransfer,
    AcquisitionStatusSelector_FrameTriggerWait,
    AcquisitionStatusSelector_FrameActive,
    AcquisitionStatusSelector_ExposureActive,
    NUM_ACQUISITIONSTATUSSELECTOR }

• enum ExposureTimeModeEnums {
    ExposureTimeMode_Common,
    ExposureTimeMode_Individual,
    NUM_EXPOSURETIMEMODE }

• enum ExposureTimeSelectorEnums {
    ExposureTimeSelector_Common,
    ExposureTimeSelector_Red,
    ExposureTimeSelector_Green,
    ExposureTimeSelector_Blue,

```

```
ExposureTimeSelector_Cyan,  
ExposureTimeSelector_Magenta,  
ExposureTimeSelector_Yellow,  
ExposureTimeSelector_Infrared,  
ExposureTimeSelector_Ultraviolet,  
ExposureTimeSelector_Stage1,  
ExposureTimeSelector_Stage2,  
NUM_EXPOSURETIMESELECTOR }  
  
• enum GainAutoBalanceEnums {  
    GainAutoBalance_Off,  
    GainAutoBalance_Once,  
    GainAutoBalance_Continuous,  
    NUM_GAINAUTOBALANCE }  
  
• enum BlackLevelAutoEnums {  
    BlackLevelAuto_Off,  
    BlackLevelAuto_Once,  
    BlackLevelAuto_Continuous,  
    NUM_BLACKLEVELAUTO }  
  
• enum BlackLevelAutoBalanceEnums {  
    BlackLevelAutoBalance_Off,  
    BlackLevelAutoBalance_Once,  
    BlackLevelAutoBalance_Continuous,  
    NUM_BLACKLEVELAUTOBALANCE }  
  
• enum WhiteClipSelectorEnums {  
    WhiteClipSelector_All,  
    WhiteClipSelector_Red,  
    WhiteClipSelector_Green,  
    WhiteClipSelector_Blue,  
    WhiteClipSelector_Y,  
    WhiteClipSelector_U,  
    WhiteClipSelector_V,  
    WhiteClipSelector_Tap1,  
    WhiteClipSelector_Tap2,  
    NUM_WHITECLIPSELECTOR }  
  
• enum TimerSelectorEnums {  
    TimerSelector_Timer0,  
    TimerSelector_Timer1,  
    TimerSelector_Timer2,  
    NUM_TIMERSELECTOR }  
  
• enum TimerStatusEnums {  
    TimerStatus_TimerIdle,  
    TimerStatus_TimerTriggerWait,  
    TimerStatus_TimerActive,  
    TimerStatus_TimerCompleted,  
    NUM_TIMERSTATUS }  
  
• enum TimerTriggerSourceEnums {  
    TimerTriggerSource_Off,  
    TimerTriggerSource_AcquisitionTrigger,  
    TimerTriggerSource_AcquisitionStart,  
    TimerTriggerSource_AcquisitionEnd,  
    TimerTriggerSource_FrameTrigger,  
    TimerTriggerSource_FrameStart,  
    TimerTriggerSource_FrameEnd,  
    TimerTriggerSource_FrameBurstStart,  
    TimerTriggerSource_FrameBurstEnd,  
    TimerTriggerSource_LineTrigger,  
    TimerTriggerSource_LineStart,  
    TimerTriggerSource_LineEnd,
```

```

TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

• enum TimerTriggerActivationEnums {
    TimerTriggerActivation_RisingEdge,
    TimerTriggerActivation_FallingEdge,
    TimerTriggerActivation_AnyEdge,
    TimerTriggerActivation_LevelHigh,
    TimerTriggerActivation_LevelLow,
    NUM_TIMERTRIGGERACTIVATION }

• enum EncoderSelectorEnums {
    EncoderSelector_Encoder0,
    EncoderSelector_Encoder1,
    EncoderSelector_Encoder2,
    NUM_ENCODERSELECTOR }

• enum EncoderSourceAEnums {
    EncoderSourceA_Off,
    EncoderSourceA_Line0,
    EncoderSourceA_Line1,
    EncoderSourceA_Line2,
    NUM_ENCODERSOURCEA }

• enum EncoderSourceBEnums {
    EncoderSourceB_Off,
    EncoderSourceB_Line0,
    EncoderSourceB_Line1,
    EncoderSourceB_Line2,
    NUM_ENCODERSOURCEB }

• enum EncoderModeEnums {

```

```
EncoderMode_FourPhase,  
EncoderMode_HighResolution,  
NUM_ENCODERMODE }  
  
• enum EncoderOutputModeEnums {  
EncoderOutputMode_Off,  
EncoderOutputMode_PositionUp,  
EncoderOutputMode_PositionDown,  
EncoderOutputMode_DirectionUp,  
EncoderOutputMode_DirectionDown,  
EncoderOutputMode_Motion,  
NUM_ENCODEROUTPUTMODE }  
  
• enum EncoderStatusEnums {  
EncoderStatus_EncoderUp,  
EncoderStatus_EncoderDown,  
EncoderStatus_EncoderIdle,  
EncoderStatus_EncoderStatic,  
NUM_ENCODERSTATUS }  
  
• enum EncoderResetSourceEnums {  
EncoderResetSource_Off,  
EncoderResetSource_AcquisitionTrigger,  
EncoderResetSource_AcquisitionStart,  
EncoderResetSource_AcquisitionEnd,  
EncoderResetSource_FrameTrigger,  
EncoderResetSource_FrameStart,  
EncoderResetSource_FrameEnd,  
EncoderResetSource_ExposureStart,  
EncoderResetSource_ExposureEnd,  
EncoderResetSource_Line0,  
EncoderResetSource_Line1,  
EncoderResetSource_Line2,  
EncoderResetSource_Counter0Start,  
EncoderResetSource_Counter1Start,  
EncoderResetSource_Counter2Start,  
EncoderResetSource_Counter0End,  
EncoderResetSource_Counter1End,  
EncoderResetSource_Counter2End,  
EncoderResetSource_Timer0Start,  
EncoderResetSource_Timer1Start,  
EncoderResetSource_Timer2Start,  
EncoderResetSource_Timer0End,  
EncoderResetSource_Timer1End,  
EncoderResetSource_Timer2End,  
EncoderResetSource_UserOutput0,  
EncoderResetSource_UserOutput1,  
EncoderResetSource_UserOutput2,  
EncoderResetSource_SoftwareSignal0,  
EncoderResetSource_SoftwareSignal1,  
EncoderResetSource_SoftwareSignal2,  
EncoderResetSource_Action0,  
EncoderResetSource_Action1,  
EncoderResetSource_Action2,  
EncoderResetSource_LinkTrigger0,  
EncoderResetSource_LinkTrigger1,  
EncoderResetSource_LinkTrigger2,  
NUM_ENCODERRESETSOURCE }  
  
• enum EncoderResetActivationEnums {  
EncoderResetActivation_RisingEdge,  
EncoderResetActivation_FallingEdge,
```



```

EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }
• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }
• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }
• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,
NUM_SOURCESELECTOR }
• enum TransferSelectorEnums {
TransferSelector_Stream0,
TransferSelector_Stream1,
TransferSelector_Stream2,
TransferSelector_All,
NUM_TRANSFERSELECTOR }
• enum TransferTriggerSelectorEnums {
TransferTriggerSelector_TransferStart,
TransferTriggerSelector_TransferStop,
TransferTriggerSelector_TransferAbort,
TransferTriggerSelector_TransferPause,
TransferTriggerSelector_TransferResume,
TransferTriggerSelector_TransferActive,
TransferTriggerSelector_TransferBurstStart,
TransferTriggerSelector_TransferBurstStop,
NUM_TRANSFERTRIGGERSELECTOR }
• enum TransferTriggerModeEnums {
TransferTriggerMode_Off,
TransferTriggerMode_On,
NUM_TRANSFERTRIGGERMODE }
• enum TransferTriggerSourceEnums {
TransferTriggerSource_Line0,
TransferTriggerSource_Line1,
TransferTriggerSource_Line2,
TransferTriggerSource_Counter0Start,
TransferTriggerSource_Counter1Start,
TransferTriggerSource_Counter2Start,
TransferTriggerSource_Counter0End,
TransferTriggerSource_Counter1End,
TransferTriggerSource_Counter2End,
TransferTriggerSource_Timer0Start,
TransferTriggerSource_Timer1Start,
TransferTriggerSource_Timer2Start,
TransferTriggerSource_Timer0End,
TransferTriggerSource_Timer1End,
TransferTriggerSource_Timer2End,
TransferTriggerSource_SoftwareSignal0,
TransferTriggerSource_SoftwareSignal1,
TransferTriggerSource_SoftwareSignal2,

```

```

TransferTriggerSource_Action0,
TransferTriggerSource_Action1,
TransferTriggerSource_Action2,
NUM_TRANSFERTRIGGERSOURCE }
• enum TransferTriggerActivationEnums {
TransferTriggerActivation_RisingEdge,
TransferTriggerActivation_FallingEdge,
TransferTriggerActivation_AnyEdge,
TransferTriggerActivation_LevelHigh,
TransferTriggerActivation_LevelLow,
NUM_TRANSFERTRIGGERACTIVATION }
• enum TransferStatusSelectorEnums {
TransferStatusSelector_Streaming,
TransferStatusSelector_Paused,
TransferStatusSelector_Stopping,
TransferStatusSelector_Stopped,
TransferStatusSelector_QueueOverflow,
NUM_TRANSFERSTATUSSELECTOR }
• enum TransferComponentSelectorEnums {
TransferComponentSelector_Red,
TransferComponentSelector_Green,
TransferComponentSelector_Blue,
TransferComponentSelector_All,
NUM_TRANSFERCOMPONENTSELECTOR }
• enum Scan3dDistanceUnitEnums {
Scan3dDistanceUnit_Millimeter,
Scan3dDistanceUnit_Inch,
NUM_SCAN3DDISTANCEUNIT }
• enum Scan3dCoordinateSystemEnums {
Scan3dCoordinateSystem_Cartesian,
Scan3dCoordinateSystem_Spherical,
Scan3dCoordinateSystem_Cylindrical,
NUM_SCAN3DCOORDINATESYSTEM }
• enum Scan3dOutputModeEnums {
Scan3dOutputMode_UncalibratedC,
Scan3dOutputMode_CalibratedABC_Grid,
Scan3dOutputMode_CalibratedABC_PointCloud,
Scan3dOutputMode_CalibratedAC,
Scan3dOutputMode_CalibratedAC_Linescan,
Scan3dOutputMode_CalibratedC,
Scan3dOutputMode_CalibratedC_Linescan,
Scan3dOutputMode_RectifiedC,
Scan3dOutputMode_RectifiedC_Linescan,
Scan3dOutputMode_DisparityC,
Scan3dOutputMode_DisparityC_Linescan,
NUM_SCAN3DOUTPUTMODE }
• enum Scan3dCoordinateSystemReferenceEnums {
Scan3dCoordinateSystemReference_Anchor,
Scan3dCoordinateSystemReference_Transformed,
NUM_SCAN3DCOORDINATESYSTEMREFERENCE }
• enum Scan3dCoordinateSelectorEnums {
Scan3dCoordinateSelector_CoordinateA,
Scan3dCoordinateSelector_CoordinateB,
Scan3dCoordinateSelector_CoordinateC,
NUM_SCAN3DCOORDINATESELECTOR }
• enum Scan3dCoordinateTransformSelectorEnums {
Scan3dCoordinateTransformSelector_RotationX,
Scan3dCoordinateTransformSelector_RotationY,

```

```

Scan3dCoordinateTransformSelector_RotationZ,
Scan3dCoordinateTransformSelector_TranslationX,
Scan3dCoordinateTransformSelector_TranslationY,
Scan3dCoordinateTransformSelector_TranslationZ,
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }

• enum Scan3dCoordinateReferenceSelectorEnums {
    Scan3dCoordinateReferenceSelector_RotationX,
    Scan3dCoordinateReferenceSelector_RotationY,
    Scan3dCoordinateReferenceSelector_RotationZ,
    Scan3dCoordinateReferenceSelector_TranslationX,
    Scan3dCoordinateReferenceSelector_TranslationY,
    Scan3dCoordinateReferenceSelector_TranslationZ,
    NUM_SCAN3DCOORDINATEREFERENCESELECTOR }

• enum ChunkImageComponentEnums {
    ChunkImageComponent_Intensity,
    ChunkImageComponent_Color,
    ChunkImageComponent_Infrared,
    ChunkImageComponent_Ultraviolet,
    ChunkImageComponent_Range,
    ChunkImageComponent_Disparity,
    ChunkImageComponent_Confidence,
    ChunkImageComponent_Scatter,
    NUM_CHUNKIMAGECOMPONENT }

• enum ChunkCounterSelectorEnums {
    ChunkCounterSelector_Counter0,
    ChunkCounterSelector_Counter1,
    ChunkCounterSelector_Counter2,
    NUM_CHUNKCOUNTERSELECTOR }

• enum ChunkTimerSelectorEnums {
    ChunkTimerSelector_Timer0,
    ChunkTimerSelector_Timer1,
    ChunkTimerSelector_Timer2,
    NUM_CHUNKTIMERSELECTOR }

• enum ChunkEncoderSelectorEnums {
    ChunkEncoderSelector_Encoder0,
    ChunkEncoderSelector_Encoder1,
    ChunkEncoderSelector_Encoder2,
    NUM_CHUNKENCODERSELECTOR }

• enum ChunkEncoderStatusEnums {
    ChunkEncoderStatus_EncoderUp,
    ChunkEncoderStatus_EncoderDown,
    ChunkEncoderStatus_EncoderIdle,
    ChunkEncoderStatus_EncoderStatic,
    NUM_CHUNKENCODERSTATUS }

• enum ChunkExposureTimeSelectorEnums {
    ChunkExposureTimeSelector_Common,
    ChunkExposureTimeSelector_Red,
    ChunkExposureTimeSelector_Green,
    ChunkExposureTimeSelector_Blue,
    ChunkExposureTimeSelector_Cyan,
    ChunkExposureTimeSelector_Magenta,
    ChunkExposureTimeSelector_Yellow,
    ChunkExposureTimeSelector_Infrared,
    ChunkExposureTimeSelector_Ultraviolet,
    ChunkExposureTimeSelector_Stage1,
    ChunkExposureTimeSelector_Stage2,
    NUM_CHUNKEXPOSURETIMESELECTOR }

```

- enum ChunkSourceIDEnums {
 ChunkSourceID_Source0,
 ChunkSourceID_Source1,
 ChunkSourceID_Source2,
 NUM_CHUNKSOURCEID }
- enum ChunkRegionIDEnums {
 ChunkRegionID_Region0,
 ChunkRegionID_Region1,
 ChunkRegionID_Region2,
 NUM_CHUNKREGIONID }
- enum ChunkTransferStreamIDEnums {
 ChunkTransferStreamID_Stream0,
 ChunkTransferStreamID_Stream1,
 ChunkTransferStreamID_Stream2,
 ChunkTransferStreamID_Stream3,
 NUM_CHUNKTRANSFERSTREAMID }
- enum ChunkScan3dDistanceUnitEnums {
 ChunkScan3dDistanceUnit_Millimeter,
 ChunkScan3dDistanceUnit_Inch,
 NUM_CHUNKSCAN3DDISTANCEUNIT }
- enum ChunkScan3dOutputModeEnums {
 ChunkScan3dOutputMode_UncalibratedC,
 ChunkScan3dOutputMode_CalibratedABC_Grid,
 ChunkScan3dOutputMode_CalibratedABC_PointCloud,
 ChunkScan3dOutputMode_CalibratedAC,
 ChunkScan3dOutputMode_CalibratedAC_Linescan,
 ChunkScan3dOutputMode_CalibratedC,
 ChunkScan3dOutputMode_CalibratedC_Linescan,
 ChunkScan3dOutputMode_RectifiedC,
 ChunkScan3dOutputMode_RectifiedC_Linescan,
 ChunkScan3dOutputMode_DisparityC,
 ChunkScan3dOutputMode_DisparityC_Linescan,
 NUM_CHUNKSCAN3DOUTPUTMODE }
- enum ChunkScan3dCoordinateSystemEnums {
 ChunkScan3dCoordinateSystem_Cartesian,
 ChunkScan3dCoordinateSystem_Spherical,
 ChunkScan3dCoordinateSystem_Cylindrical,
 NUM_CHUNKSCAN3DCOORDINATESYSTEM }
- enum ChunkScan3dCoordinateSystemReferenceEnums {
 ChunkScan3dCoordinateSystemReference_Anchor,
 ChunkScan3dCoordinateSystemReference_Transformed,
 NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
- enum ChunkScan3dCoordinateSelectorEnums {
 ChunkScan3dCoordinateSelector_CoordinateA,
 ChunkScan3dCoordinateSelector_CoordinateB,
 ChunkScan3dCoordinateSelector_CoordinateC,
 NUM_CHUNKSCAN3DCOORDINATESELECTOR }
- enum ChunkScan3dCoordinateTransformSelectorEnums {
 ChunkScan3dCoordinateTransformSelector_RotationX,
 ChunkScan3dCoordinateTransformSelector_RotationY,
 ChunkScan3dCoordinateTransformSelector_RotationZ,
 ChunkScan3dCoordinateTransformSelector_TranslationX,
 ChunkScan3dCoordinateTransformSelector_TranslationY,
 ChunkScan3dCoordinateTransformSelector_TranslationZ,
 NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }
- enum ChunkScan3dCoordinateReferenceSelectorEnums {
 ChunkScan3dCoordinateReferenceSelector_RotationX,
 ChunkScan3dCoordinateReferenceSelector_RotationY,

```

ChunkScan3dCoordinateReferenceSelector_RotationZ,
ChunkScan3dCoordinateReferenceSelector_TranslationX,
ChunkScan3dCoordinateReferenceSelector_TranslationY,
ChunkScan3dCoordinateReferenceSelector_TranslationZ,
NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }

```

- enum DeviceTapGeometryEnums {


```

DeviceTapGeometry_Geometry_1X_1Y,
DeviceTapGeometry_Geometry_1X2_1Y,
DeviceTapGeometry_Geometry_1X2_1Y2,
DeviceTapGeometry_Geometry_2X_1Y,
DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
DeviceTapGeometry_Geometry_2XE_1Y2,
DeviceTapGeometry_Geometry_2XM_1Y,
DeviceTapGeometry_Geometry_2XM_1Y2,
DeviceTapGeometry_Geometry_1X_1Y2,
DeviceTapGeometry_Geometry_1X_2YE,
DeviceTapGeometry_Geometry_1X3_1Y,
DeviceTapGeometry_Geometry_3X_1Y,
DeviceTapGeometry_Geometry_1X,
DeviceTapGeometry_Geometry_1X2,
DeviceTapGeometry_Geometry_2X,
DeviceTapGeometry_Geometry_2XE,
DeviceTapGeometry_Geometry_2XM,
DeviceTapGeometry_Geometry_1X3,
DeviceTapGeometry_Geometry_3X,
DeviceTapGeometry_Geometry_1X4_1Y,
DeviceTapGeometry_Geometry_4X_1Y,
DeviceTapGeometry_Geometry_2X2_1Y,
DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
DeviceTapGeometry_Geometry_1X2_2YE,
DeviceTapGeometry_Geometry_2X_2YE,
DeviceTapGeometry_Geometry_2XE_2YE,
DeviceTapGeometry_Geometry_2XM_2YE,
DeviceTapGeometry_Geometry_1X4,
DeviceTapGeometry_Geometry_4X,
DeviceTapGeometry_Geometry_2X2,
DeviceTapGeometry_Geometry_2X2E,
DeviceTapGeometry_Geometry_2X2M,
DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

```
- enum GevPhysicalLinkConfigurationEnums {


```

GevPhysicalLinkConfiguration_SingleLink,
GevPhysicalLinkConfiguration_MultiLink,
GevPhysicalLinkConfiguration_StaticLAG,
GevPhysicalLinkConfiguration_DynamicLAG,
NUM_GEVPHYSICALLINKCONFIGURATION }

```

- enum `GevCurrentPhysicalLinkConfigurationEnums` {
`GevCurrentPhysicalLinkConfiguration_SingleLink`,
`GevCurrentPhysicalLinkConfiguration_MultiLink`,
`GevCurrentPhysicalLinkConfiguration_StaticLAG`,
`GevCurrentPhysicalLinkConfiguration_DynamicLAG`,
`NUM_GEVCURRENTPHYSICALLINKCONFIGURATION` }
- enum `GevIPConfigurationStatusEnums` {
`GevIPConfigurationStatus_None`,
`GevIPConfigurationStatus_PersistentIP`,
`GevIPConfigurationStatus_DHCP`,
`GevIPConfigurationStatus_LLA`,
`GevIPConfigurationStatus_ForceIP`,
`NUM_GEVIPCONFIGURATIONSTATUS` }
- enum `GevGVCPExtendedStatusCodesSelectorEnums` {
`GevGVCPExtendedStatusCodesSelector_Version1_1`,
`GevGVCPExtendedStatusCodesSelector_Version2_0`,
`NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR` }
- enum `GevGVSPExtendedIDModeEnums` {
`GevGVSPExtendedIDMode_Off`,
`GevGVSPExtendedIDMode_On`,
`NUM_GEVGVSPEXTENDEDIDMODE` }
- enum `CIConfigurationEnums` {
`CIConfiguration_Base`,
`CIConfiguration_Medium`,
`CIConfiguration_Full`,
`CIConfiguration_DualBase`,
`CIConfiguration_EightyBit`,
`NUM_CLCONFIGURATION` }
- enum `CITimeSlotsCountEnums` {
`CITimeSlotsCount_One`,
`CITimeSlotsCount_Two`,
`CITimeSlotsCount_Three`,
`NUM_CLTIMESLOTSCOUNT` }
- enum `CxpLinkConfigurationStatusEnums` {
`CxpLinkConfigurationStatus_None`,
`CxpLinkConfigurationStatus_Pending`,
`CxpLinkConfigurationStatus_CXP1_X1`,
`CxpLinkConfigurationStatus_CXP2_X1`,
`CxpLinkConfigurationStatus_CXP3_X1`,
`CxpLinkConfigurationStatus_CXP5_X1`,
`CxpLinkConfigurationStatus_CXP6_X1`,
`CxpLinkConfigurationStatus_CXP1_X2`,
`CxpLinkConfigurationStatus_CXP2_X2`,
`CxpLinkConfigurationStatus_CXP3_X2`,
`CxpLinkConfigurationStatus_CXP5_X2`,
`CxpLinkConfigurationStatus_CXP6_X2`,
`CxpLinkConfigurationStatus_CXP1_X3`,
`CxpLinkConfigurationStatus_CXP2_X3`,
`CxpLinkConfigurationStatus_CXP3_X3`,
`CxpLinkConfigurationStatus_CXP5_X3`,
`CxpLinkConfigurationStatus_CXP6_X3`,
`CxpLinkConfigurationStatus_CXP1_X4`,
`CxpLinkConfigurationStatus_CXP2_X4`,
`CxpLinkConfigurationStatus_CXP3_X4`,
`CxpLinkConfigurationStatus_CXP5_X4`,
`CxpLinkConfigurationStatus_CXP6_X4`,
`CxpLinkConfigurationStatus_CXP1_X5`,
`CxpLinkConfigurationStatus_CXP2_X5`,

```

CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,
CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

```

- **enum** CxpLinkConfigurationPreferredEnums {


```

CxpLinkConfigurationPreferred_CXP1_X1,
CxpLinkConfigurationPreferred_CXP2_X1,
CxpLinkConfigurationPreferred_CXP3_X1,
CxpLinkConfigurationPreferred_CXP5_X1,
CxpLinkConfigurationPreferred_CXP6_X1,
CxpLinkConfigurationPreferred_CXP1_X2,
CxpLinkConfigurationPreferred_CXP2_X2,
CxpLinkConfigurationPreferred_CXP3_X2,
CxpLinkConfigurationPreferred_CXP5_X2,
CxpLinkConfigurationPreferred_CXP6_X2,
CxpLinkConfigurationPreferred_CXP1_X3,
CxpLinkConfigurationPreferred_CXP2_X3,
CxpLinkConfigurationPreferred_CXP3_X3,
CxpLinkConfigurationPreferred_CXP5_X3,
CxpLinkConfigurationPreferred_CXP6_X3,
CxpLinkConfigurationPreferred_CXP1_X4,
CxpLinkConfigurationPreferred_CXP2_X4,
CxpLinkConfigurationPreferred_CXP3_X4,
CxpLinkConfigurationPreferred_CXP5_X4,
CxpLinkConfigurationPreferred_CXP6_X4,
CxpLinkConfigurationPreferred_CXP1_X5,
CxpLinkConfigurationPreferred_CXP2_X5,
CxpLinkConfigurationPreferred_CXP3_X5,
CxpLinkConfigurationPreferred_CXP5_X5,
CxpLinkConfigurationPreferred_CXP6_X5,
CxpLinkConfigurationPreferred_CXP1_X6,
CxpLinkConfigurationPreferred_CXP2_X6,
CxpLinkConfigurationPreferred_CXP3_X6,
CxpLinkConfigurationPreferred_CXP5_X6,
CxpLinkConfigurationPreferred_CXP6_X6,
NUM_CXPLINKCONFIGURATIONPREFERRED }

```
- **enum** CxpLinkConfigurationEnums {


```

CxpLinkConfiguration_Auto,
CxpLinkConfiguration_CXP1_X1,
CxpLinkConfiguration_CXP2_X1,
CxpLinkConfiguration_CXP3_X1,
CxpLinkConfiguration_CXP5_X1,
CxpLinkConfiguration_CXP6_X1,
CxpLinkConfiguration_CXP1_X2,
CxpLinkConfiguration_CXP2_X2,
CxpLinkConfiguration_CXP3_X2,
CxpLinkConfiguration_CXP5_X2,
CxpLinkConfiguration_CXP6_X2,
CxpLinkConfiguration_CXP1_X3,
CxpLinkConfiguration_CXP2_X3,
CxpLinkConfiguration_CXP3_X3,
CxpLinkConfiguration_CXP5_X3,
CxpLinkConfiguration_CXP6_X3,

```

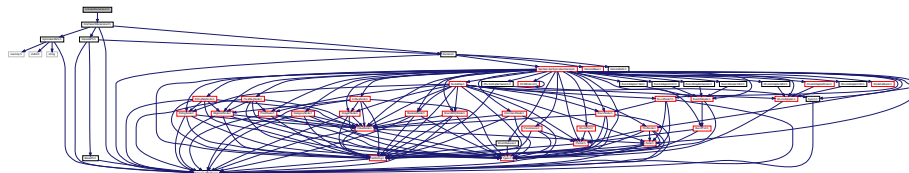
```

CxpLinkConfiguration_CXP1_X4,
CxpLinkConfiguration_CXP2_X4,
CxpLinkConfiguration_CXP3_X4,
CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,
CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }
• enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
• enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

```

16.14 include/CameraList.h File Reference

Include dependency graph for CameraList.h:



This graph shows which files directly or indirectly include this file:



Classes

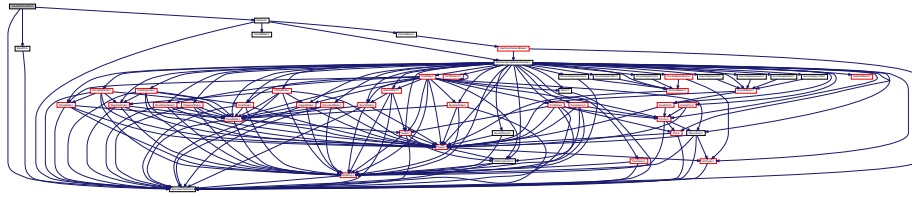
- class [CameraList](#)
Used to hold a list of camera objects.

Namespaces

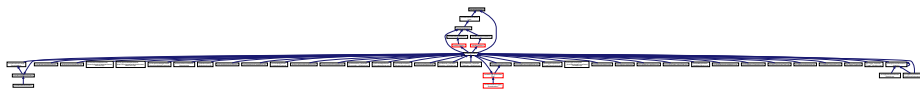
- [Spinnaker](#)

16.15 include/CameraPtr.h File Reference

Include dependency graph for CameraPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

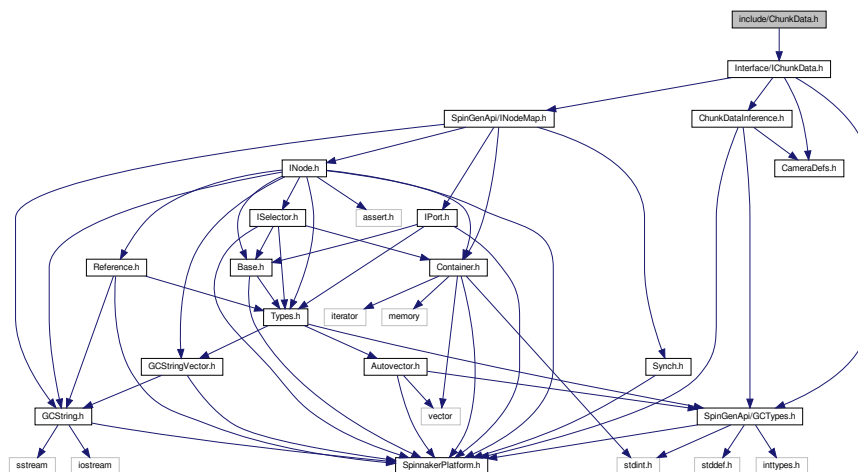
- class [CameraPtr](#)
A reference tracked pointer to a camera object.

Namespaces

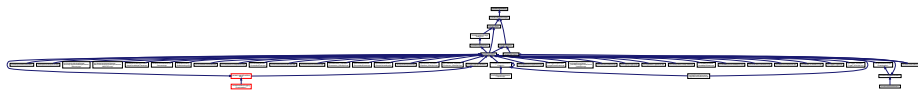
- [Spinnaker](#)

16.16 include/ChunkData.h File Reference

Include dependency graph for ChunkData.h:



This graph shows which files directly or indirectly include this file:



Classes

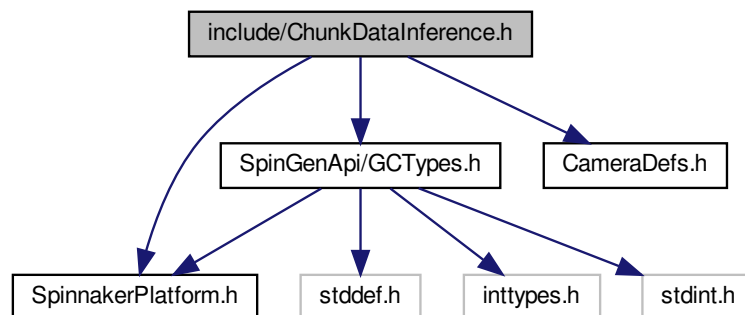
- class [ChunkData](#)
The chunk data which contains additional information about an image.

Namespaces

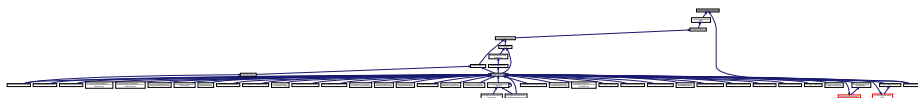
- [Spinnaker](#)

16.17 include/ChunkDataInference.h File Reference

Include dependency graph for ChunkDataInference.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [InferenceBoxRect](#)
Inference Bounding Box Type Data Structures.
- struct [InferenceBoxCircle](#)
- struct [InferenceBoxRotatedRect](#)
- struct [InferenceBoundingBox](#)
Inference Bounding Boxes data structure.
- class [InferenceBoundingBoxResult](#)
An inference bounding boxes object which holds information about the detected bounding boxes.

Namespaces

- [Spinnaker](#)

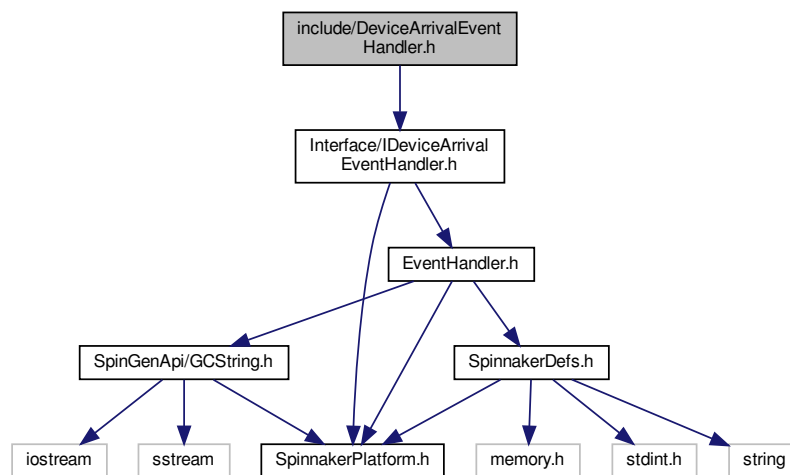
Enumerations

- enum [InferenceBoxType](#) {
[INFERENCE_BOX_TYPE_RECTANGLE](#) = 0,
[INFERENCE_BOX_TYPE_CIRCLE](#) = 1,
[INFERENCE_BOX_TYPE_ROTATED_RECTANGLE](#) = 2 }

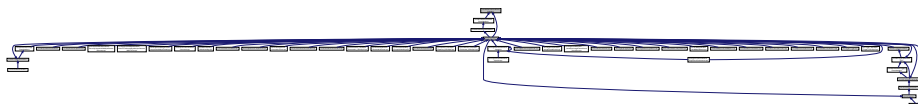
Inference Bounding Box Type.

16.18 include/DeviceArrivalEventHandler.h File Reference

Include dependency graph for DeviceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DeviceArrivalEventHandler](#)

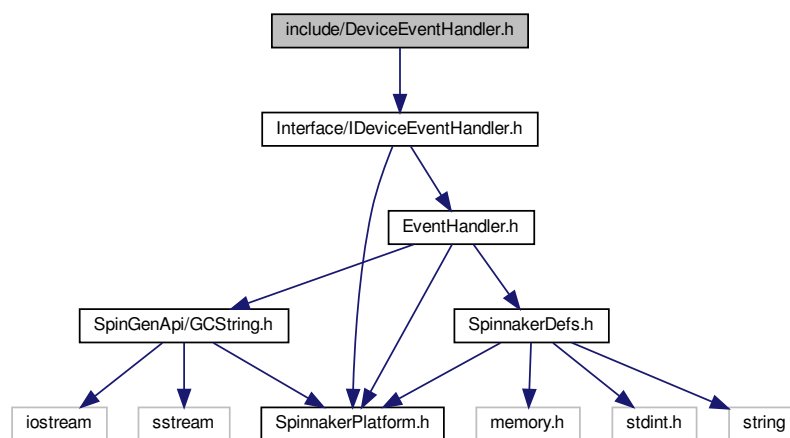
An event handler for capturing the device arrival event.

Namespaces

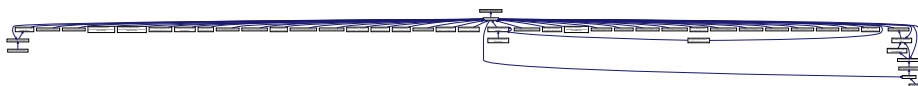
- [Spinnaker](#)

16.19 include/DeviceEventHandler.h File Reference

Include dependency graph for DeviceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

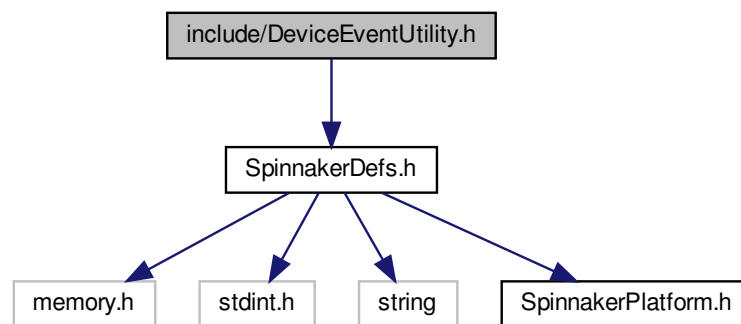
- class [DeviceEventHandler](#)
A handler to device events.

Namespaces

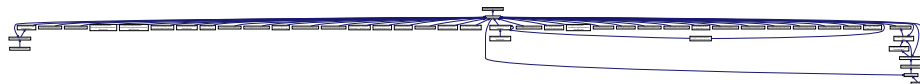
- [Spinnaker](#)

16.20 include/DeviceEventUtility.h File Reference

Include dependency graph for DeviceEventUtility.h:



This graph shows which files directly or indirectly include this file:



Classes

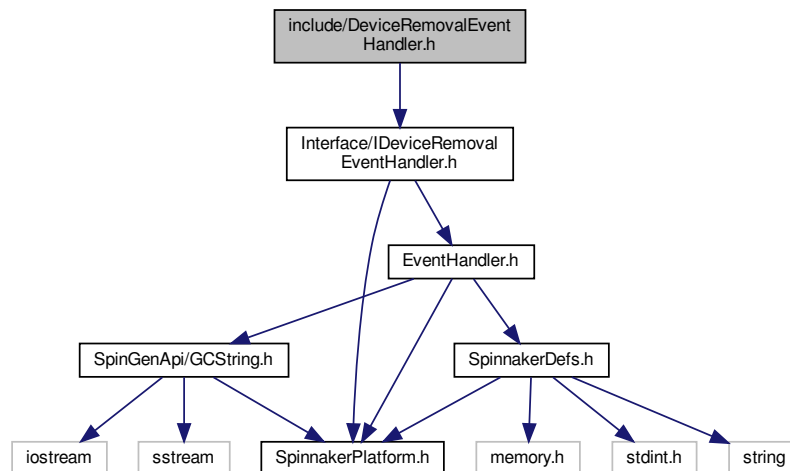
- class [DeviceEventUtility](#)

Namespaces

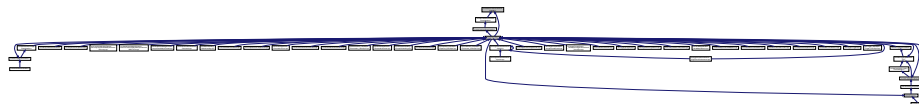
- [Spinnaker](#)

16.21 include/DeviceRemovalEventHandler.h File Reference

Include dependency graph for DeviceRemovalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DeviceRemovalEventHandler](#)

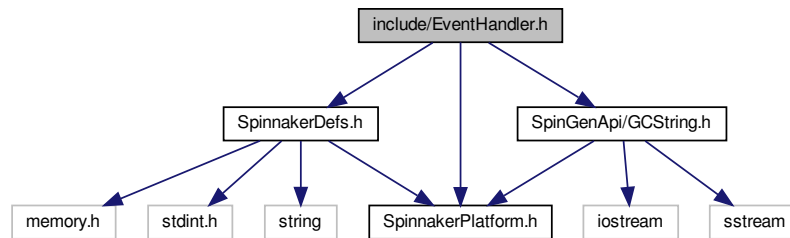
An event handler for capturing the device removal event.

Namespaces

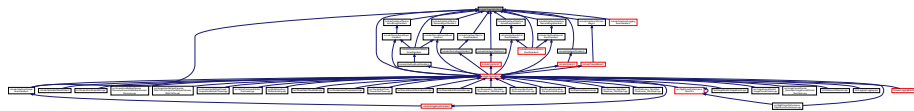
- [Spinnaker](#)

16.22 include/EventHandler.h File Reference

Include dependency graph for EventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

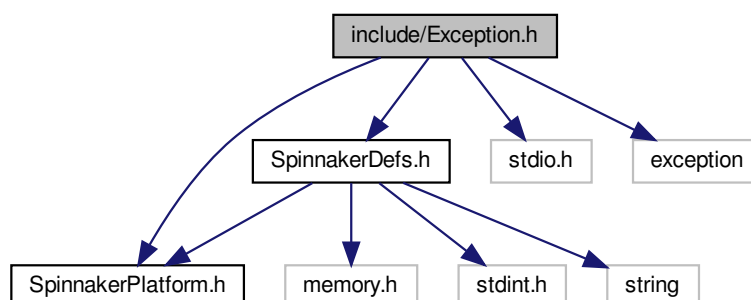
- class [EventHandler](#)
The base class for all event handler types.

Namespaces

- [Spinnaker](#)

16.23 include/Exception.h File Reference

Include dependency graph for Exception.h:

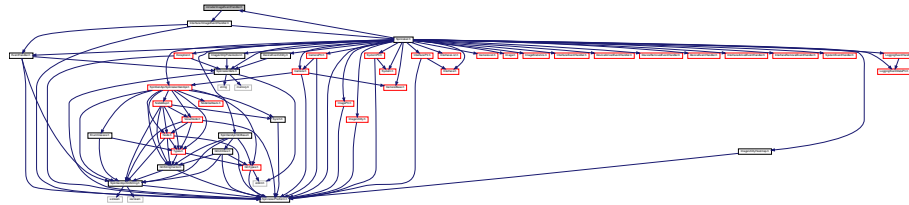


Namespaces

- [Spinnaker](#)

16.25 include/ImageEventHandler.h File Reference

Include dependency graph for ImageEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

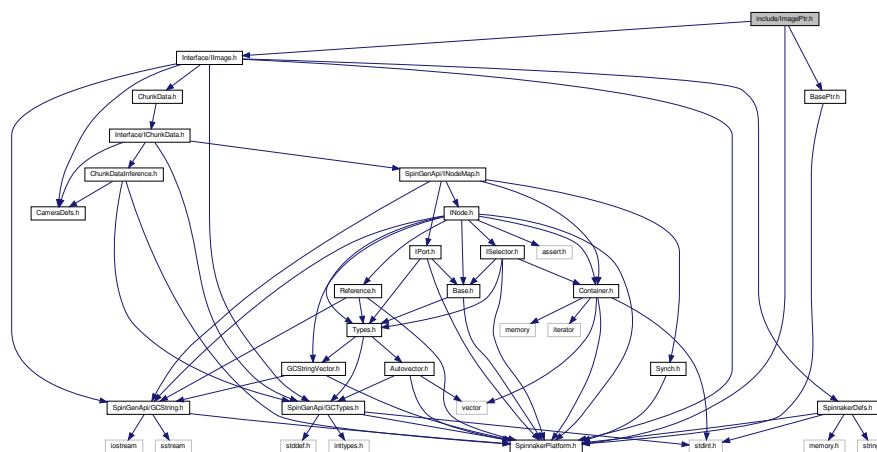
- class [ImageEventHandler](#)
A handler for capturing image arrival events.

Namespaces

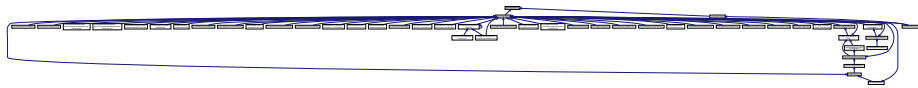
- [Spinnaker](#)

16.26 include/ImagePtr.h File Reference

Include dependency graph for ImagePtr.h:



This graph shows which files directly or indirectly include this file:



Classes

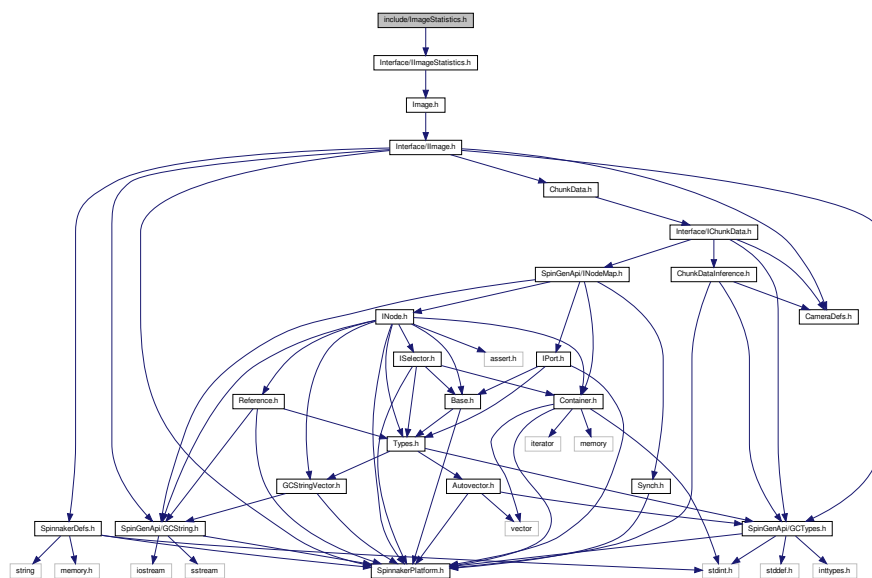
- class [ImagePtr](#)
A reference tracked pointer to an image object.

Namespaces

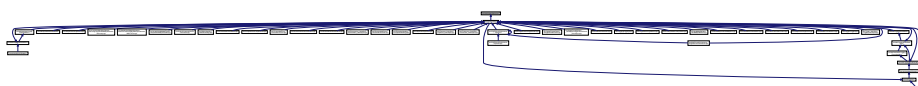
- [Spinnaker](#)

16.27 include/ImageStatistics.h File Reference

Include dependency graph for ImageStatistics.h:



This graph shows which files directly or indirectly include this file:



Classes

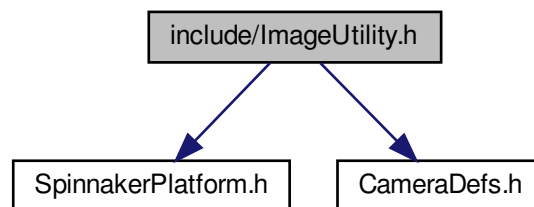
- class [ImageStatistics](#)
Represents image statistics for an image.

Namespaces

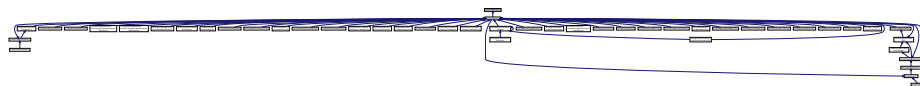
- [Spinnaker](#)

16.28 include/ImageUtility.h File Reference

Include dependency graph for ImageUtility.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageUtility](#)

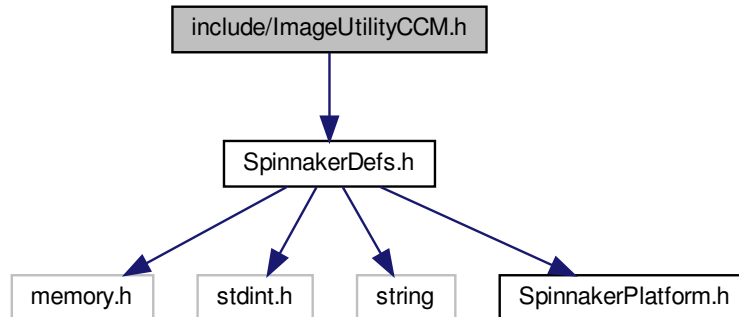
Static helper functions for the image object class.

Namespaces

- [Spinnaker](#)

16.29 include/ImageUtilityCCM.h File Reference

Include dependency graph for ImageUtilityCCM.h:



Classes

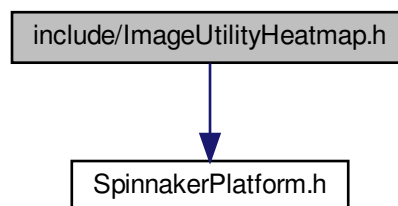
- class [ImageUtilityCCM](#)
Static function to create color corrected images from an image object.

Namespaces

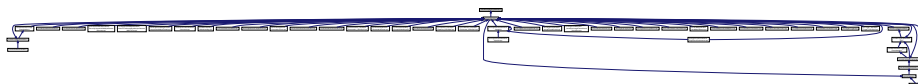
- [Spinnaker](#)

16.30 include/ImageUtilityHeatmap.h File Reference

Include dependency graph for ImageUtilityHeatmap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageUtilityHeatmap](#)

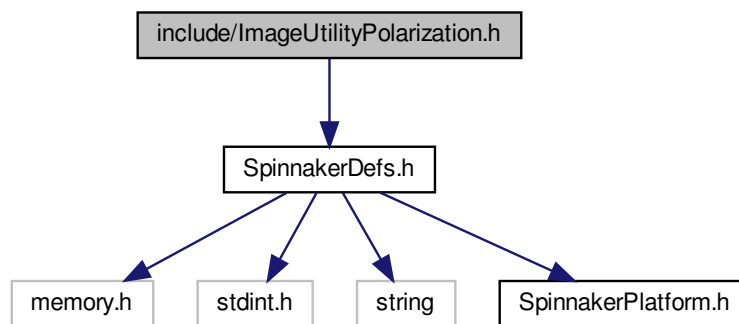
Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

Namespaces

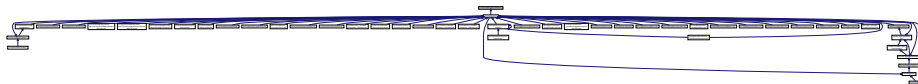
- [Spinnaker](#)

16.31 include/ImageUtilityPolarization.h File Reference

Include dependency graph for ImageUtilityPolarization.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageUtilityPolarization](#)

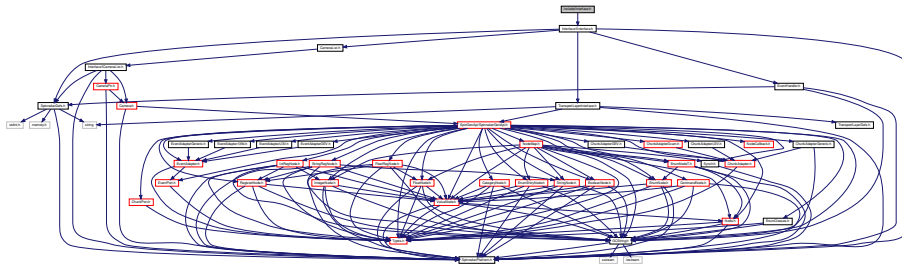
Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

Namespaces

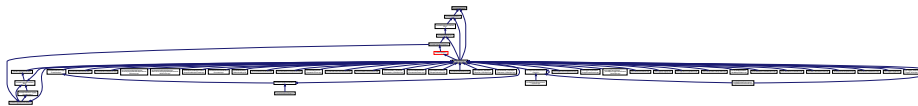
- [Spinnaker](#)

16.32 include/Interface.h File Reference

Include dependency graph for Interface.h:



This graph shows which files directly or indirectly include this file:



Classes

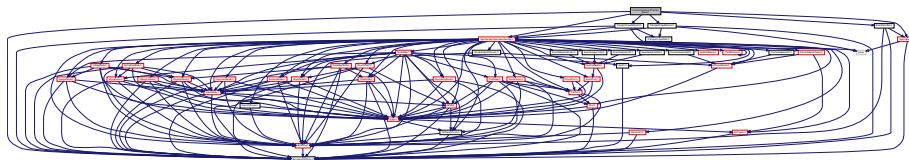
- class [Interface](#)
An interface object which holds a list of cameras.

Namespaces

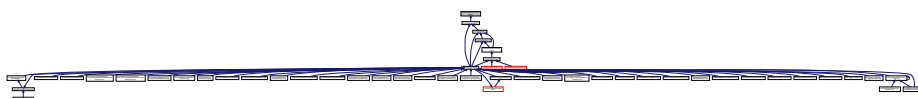
- [Spinnaker](#)

16.33 include/Interface/ICameraBase.h File Reference

Include dependency graph for ICameraBase.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ICameraBase](#)

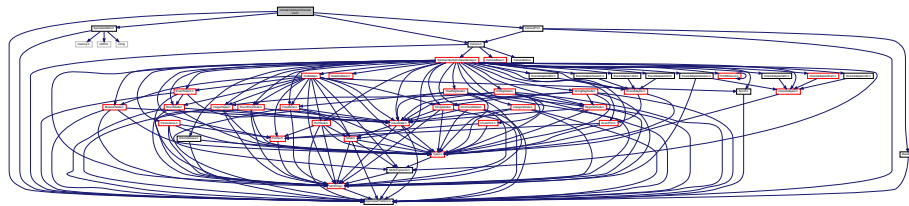
The interface file for base class for the camera object.

Namespaces

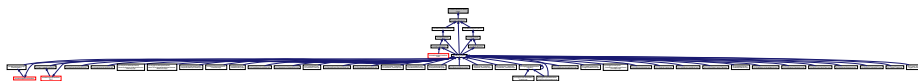
- [Spinnaker](#)

16.34 include/Interface/ICameraList.h File Reference

Include dependency graph for ICameraList.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ICameraList](#)

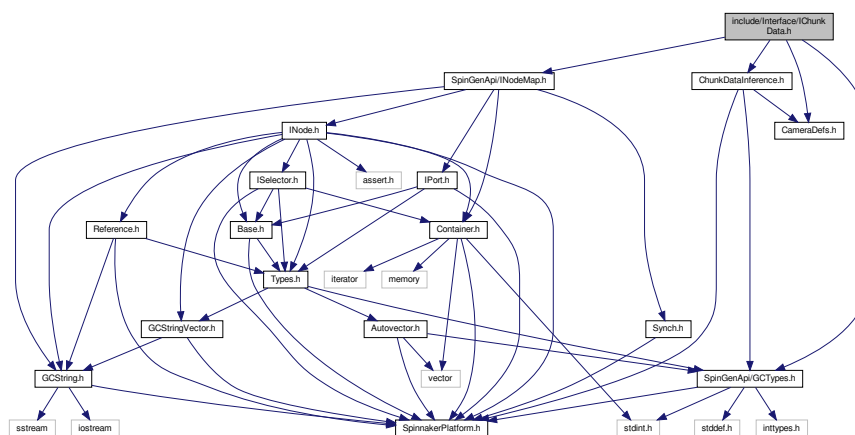
Used to hold a list of camera objects.

Namespaces

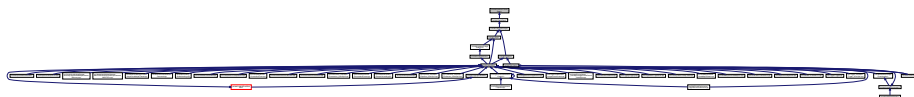
- [Spinnaker](#)

16.35 include/Interface/IChunkData.h File Reference

Include dependency graph for IChunkData.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IChunkData](#)

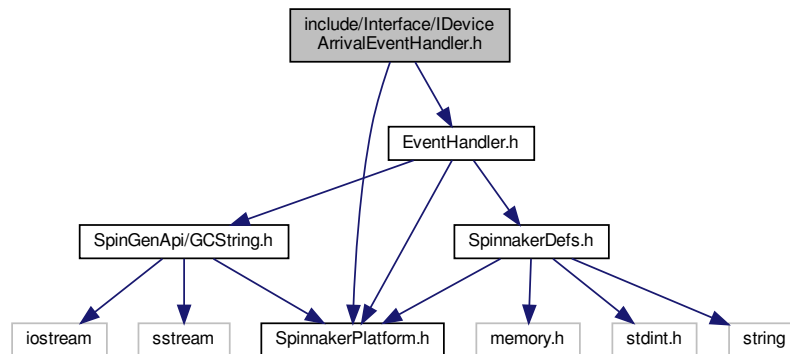
The *Interface* file for *ChunkData*.

Namespaces

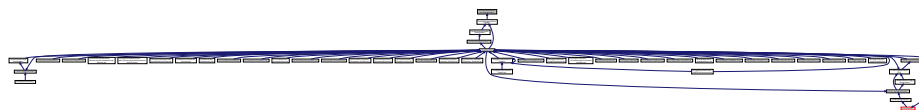
- [Spinaker](#)

16.36 include/Interface/IDeviceArrivalEventHandler.h File Reference

Include dependency graph for IDeviceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

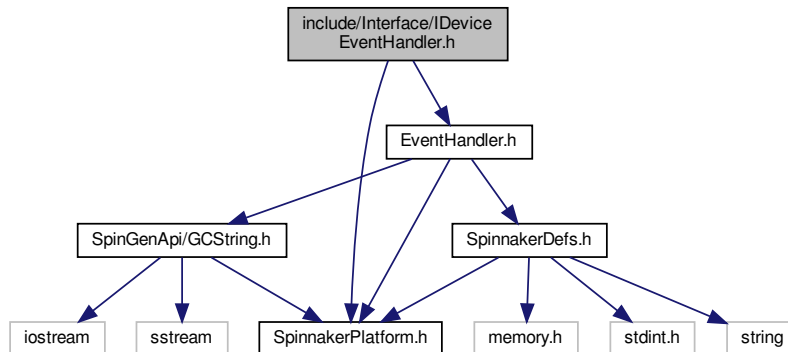
- class [IDeviceArrivalEventHandler](#)

Namespaces

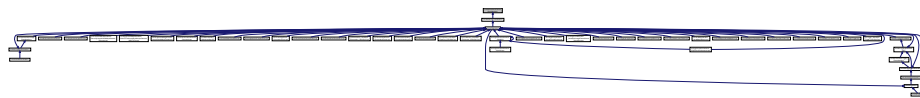
- [Spinnaker](#)

16.37 include/Interface/IDeviceEventHandler.h File Reference

Include dependency graph for IDeviceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

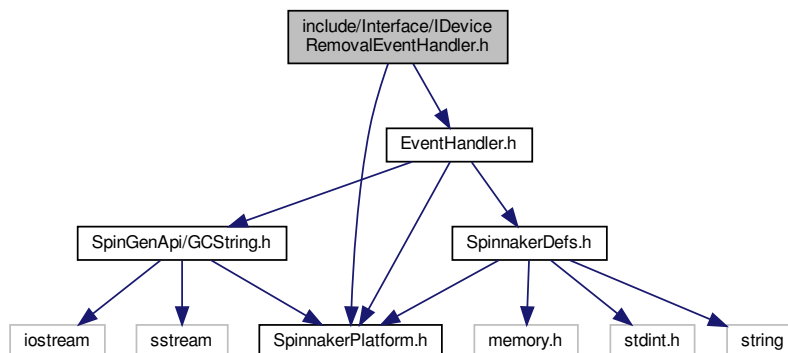
- class [IDeviceEventHandler](#)

Namespaces

- [Spinnaker](#)

16.38 include/Interface/IDeviceRemovalEventHandler.h File Reference

Include dependency graph for IDeviceRemovalEventHandler.h:

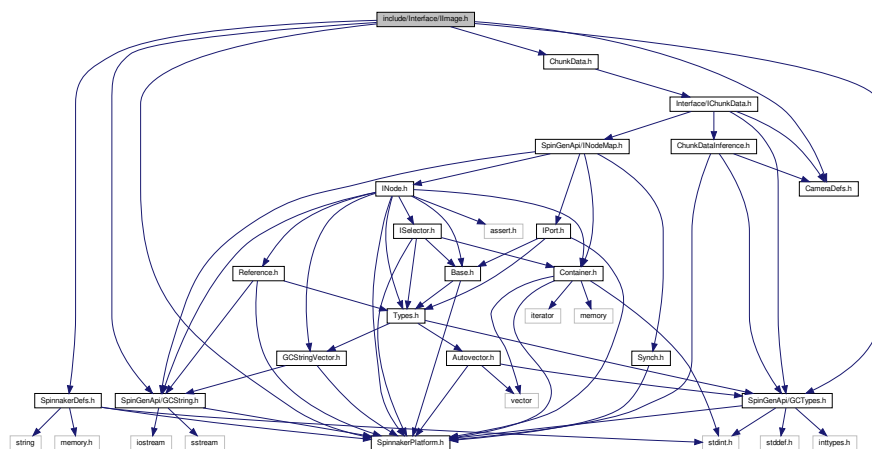


The diagram illustrates a network structure. A central vertical axis is flanked by two horizontal rows of nodes. The nodes are represented by small rectangles, and the edges are represented by lines connecting them. A legend at the bottom indicates that the nodes are labeled 'Node' and the edges are labeled 'Edge'.

- class `IDeviceRemovalEventHandler`

- Spinnaker

Include dependency graph for Image.h:



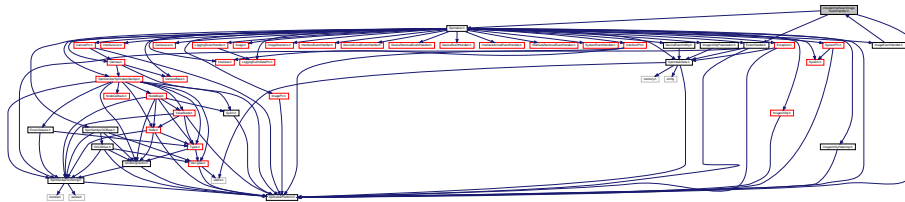
- class `Image`
The interface file for `Image`.

Namespaces

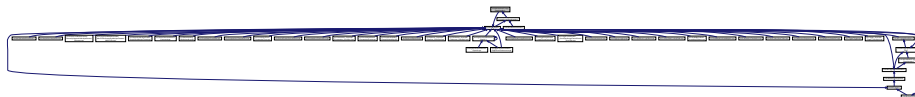
- [Spinnaker](#)

16.40 include/Interface/IImageEventHandler.h File Reference

Include dependency graph for IImageEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

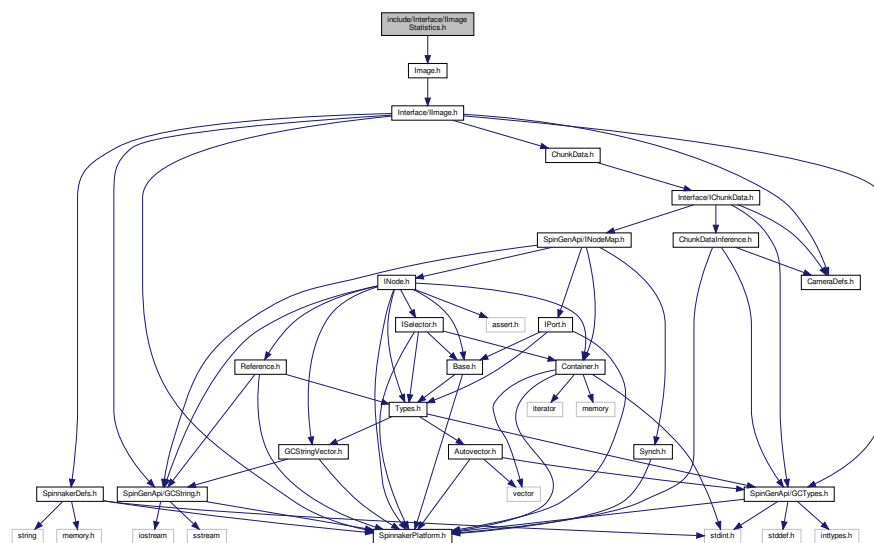
- class [IImageEventHandler](#)

Namespaces

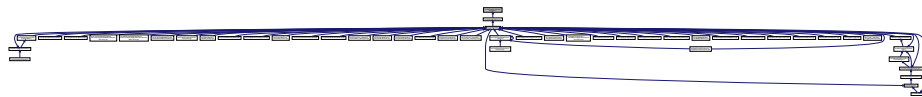
- [Spinnaker](#)

16.41 include/Interface/IImageStatistics.h File Reference

Include dependency graph for IImageStatistics.h:



This graph shows which files directly or indirectly include this file:



Classes

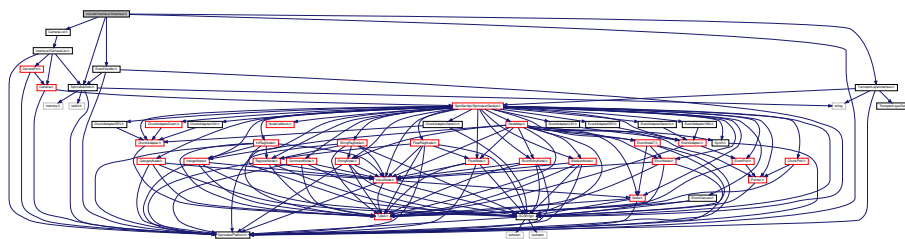
- class [IImageStatistics](#)
The interface file for image statistics.

Namespaces

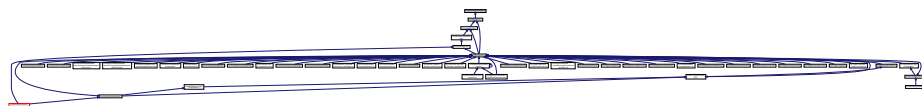
- [Spinnaker](#)

16.42 include/Interface/IInterface.h File Reference

Include dependency graph for IInterface.h:



This graph shows which files directly or indirectly include this file:



Classes

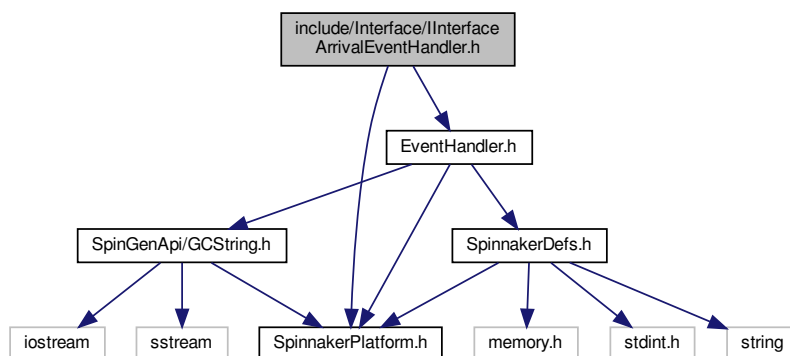
- class [IInterface](#)
The interface file for [IInterface](#).

Namespaces

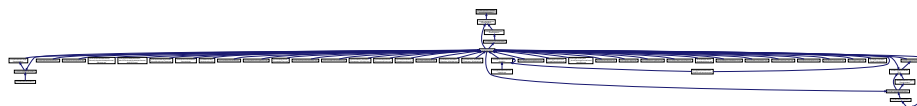
- [Spinnaker](#)

16.43 include/Interface/InterfaceArrivalEventHandler.h File Reference

Include dependency graph for InterfaceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

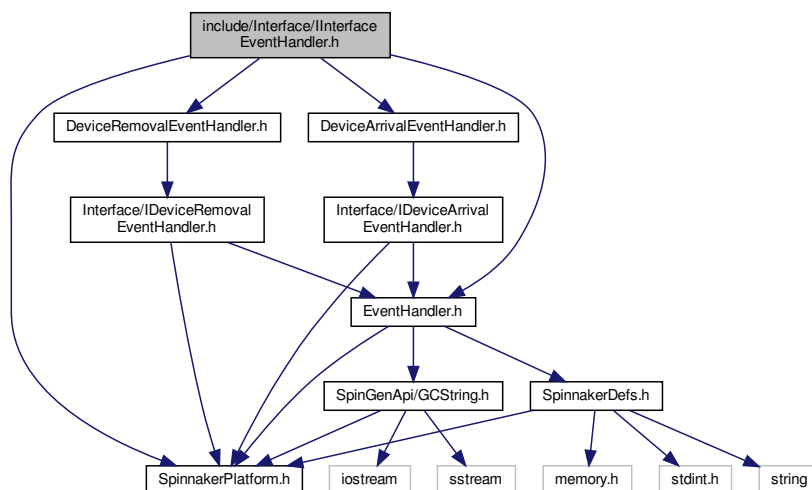
- class [InterfaceArrivalEventHandler](#)

Namespaces

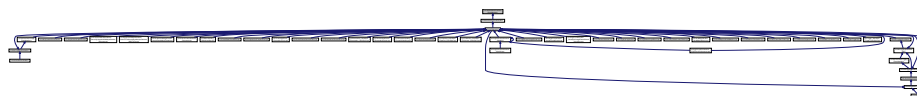
- [Spinnaker](#)

16.44 include/Interface/IInterfaceEventHandler.h File Reference

Include dependency graph for IInterfaceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

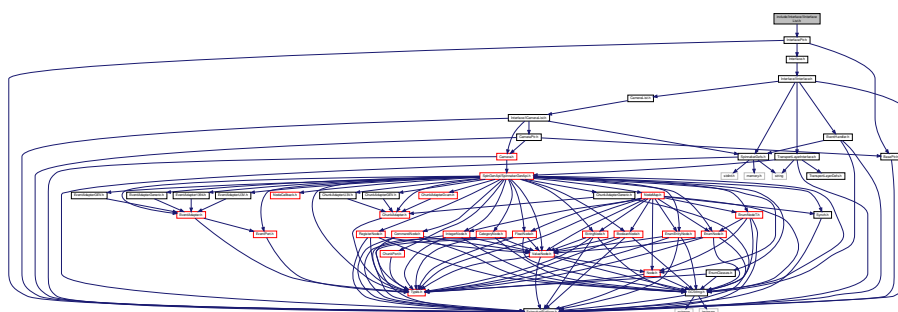
- class [IInterfaceEventHandler](#)

Namespaces

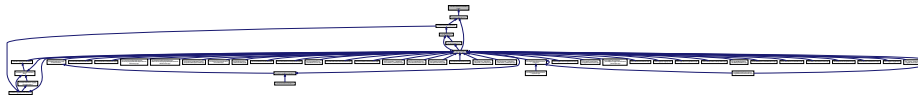
- [Spinnaker](#)

16.45 include/Interface/IInterfaceList.h File Reference

Include dependency graph for IInterfaceList.h:



This graph shows which files directly or indirectly include this file:



Classes

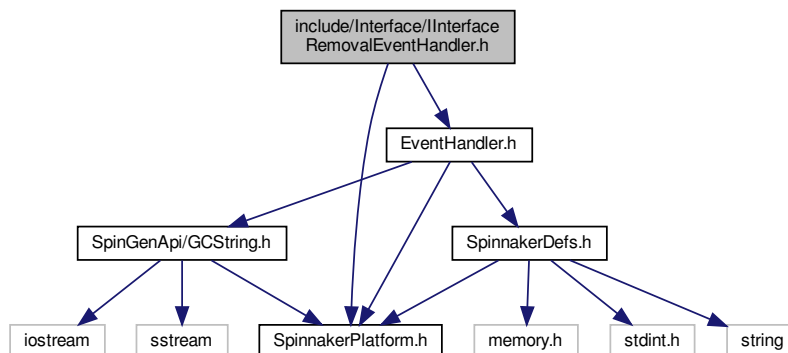
- class [InterfaceList](#)
The interface file for [InterfaceList](#) class.

Namespaces

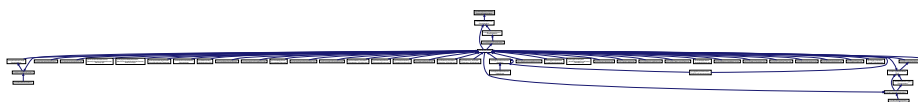
- [Spinnaker](#)

16.46 include/Interface/InterfaceRemovalEventHandler.h File Reference

Include dependency graph for InterfaceRemovalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

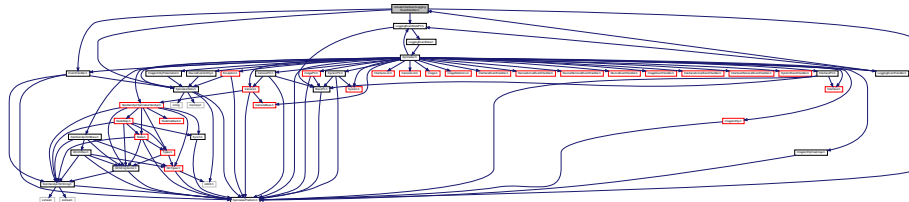
- class [InterfaceRemovalEventHandler](#)

Namespaces

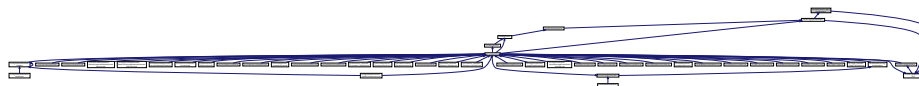
- [Spinnaker](#)

16.47 include/Interface/ILoggingEventHandler.h File Reference

Include dependency graph for ILoggingEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

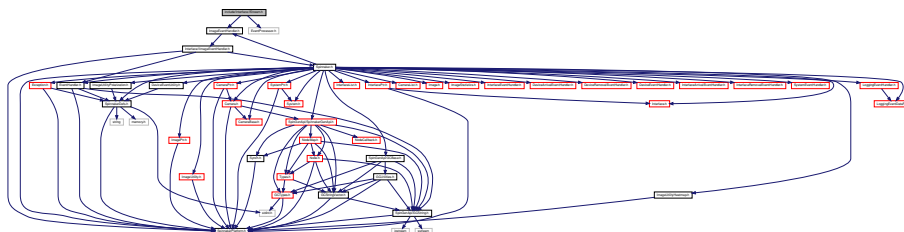
- class [ILoggingEventHandler](#)

Namespaces

- [Spinnaker](#)

16.48 include/Interface/IStream.h File Reference

Include dependency graph for IStream.h:



Classes

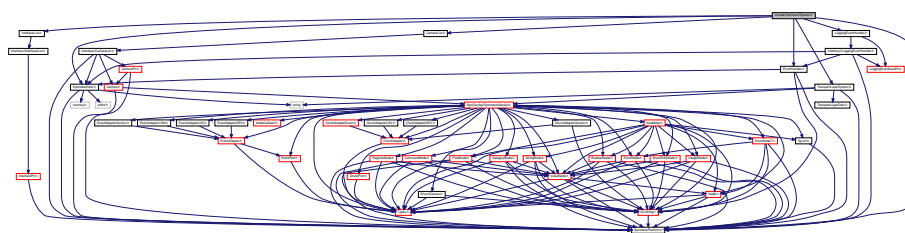
- class [IDataStream](#)

Namespaces

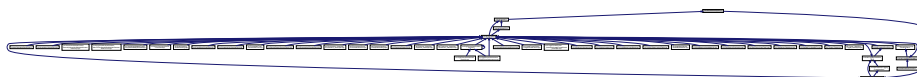
- [Spinnaker](#)

16.49 include/Interface/ISystem.h File Reference

Include dependency graph for ISystem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ISystem](#)

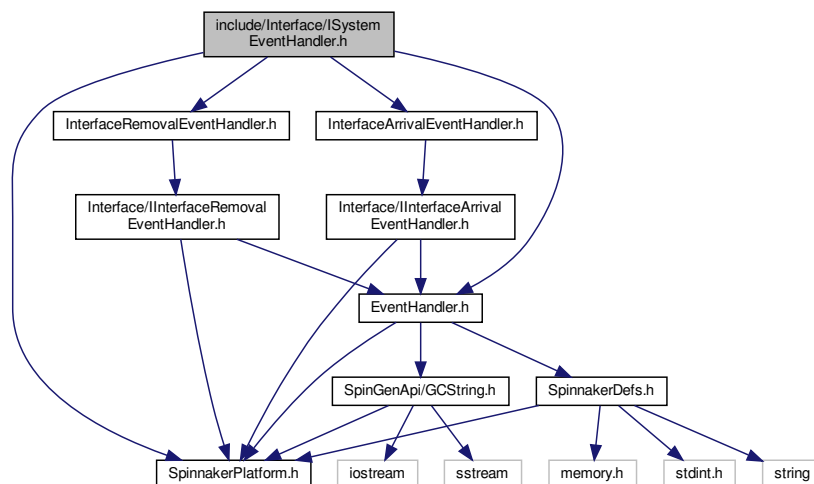
The interface file for [System](#).

Namespaces

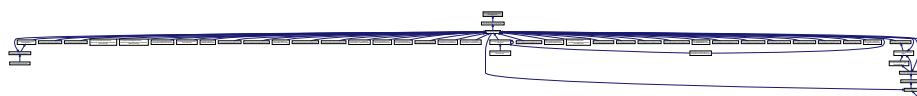
- [Spinnaker](#)

16.50 include/Interface/ISystemEventHandler.h File Reference

Include dependency graph for ISystemEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

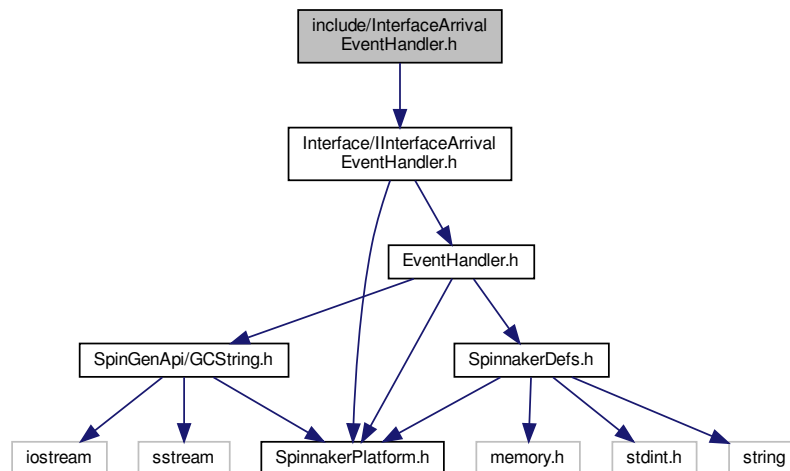
- class [ISystemEventHandler](#)

Namespaces

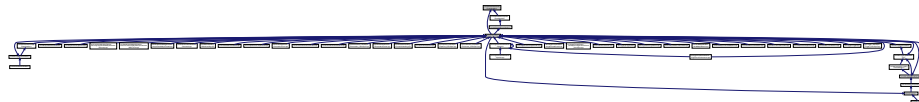
- [Spinnaker](#)

16.51 include/InterfaceArrivalEventHandler.h File Reference

Include dependency graph for InterfaceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [InterfaceArrivalEventHandler](#)

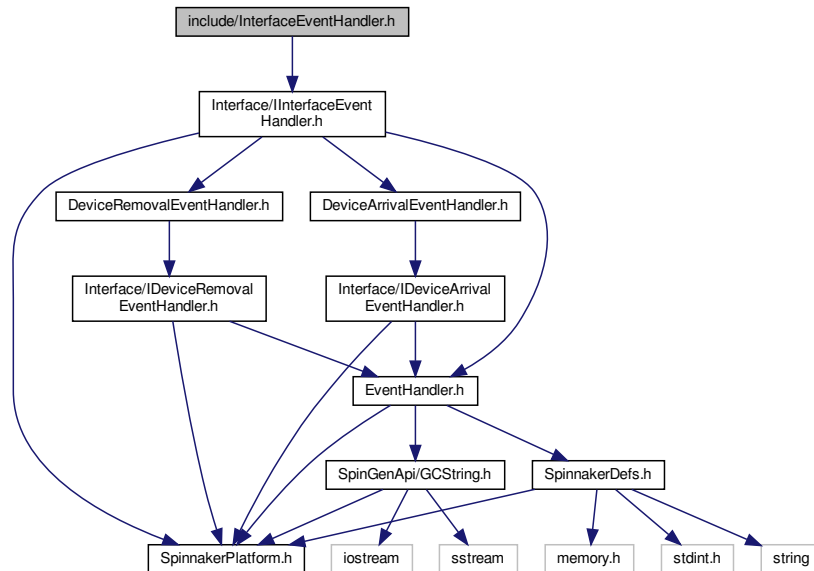
An event handler for capturing the interface arrival event.

Namespaces

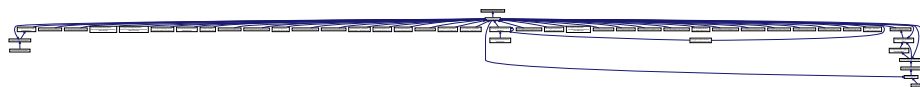
- [Spinnaker](#)

16.52 include/InterfaceEventHandler.h File Reference

Include dependency graph for InterfaceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [InterfaceEventHandler](#)

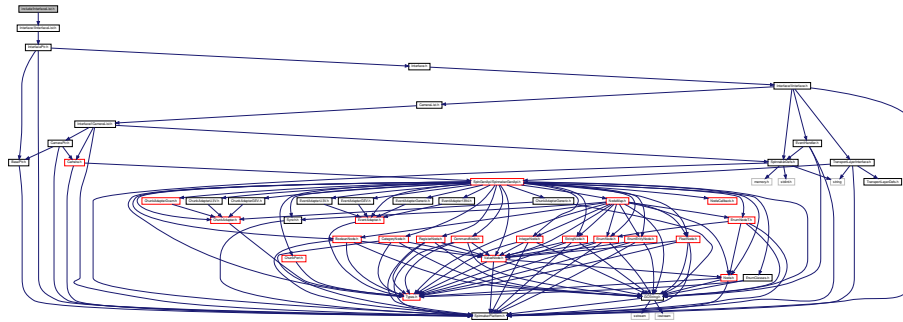
A handler to device arrival and removal events on all interfaces.

Namespaces

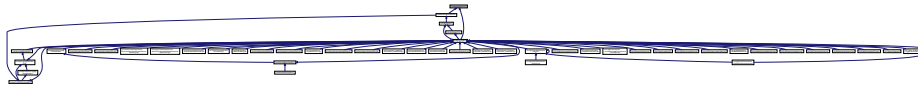
- [Spinnaker](#)

16.53 `include/InterfaceList.h` File Reference

Include dependency graph for `InterfaceList.h`:



This graph shows which files directly or indirectly include this file:



Classes

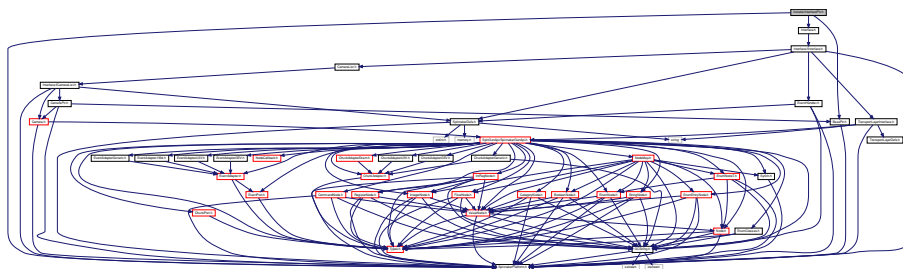
- class [InterfaceList](#)
A list of the available interfaces on the system.

Namespaces

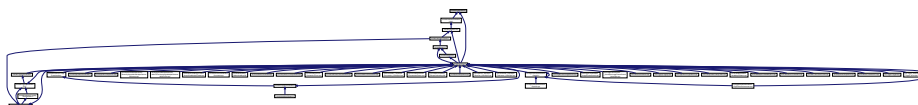
- [Spinnaker](#)

16.54 `include/InterfacePtr.h` File Reference

Include dependency graph for `InterfacePtr.h`:



This graph shows which files directly or indirectly include this file:



Classes

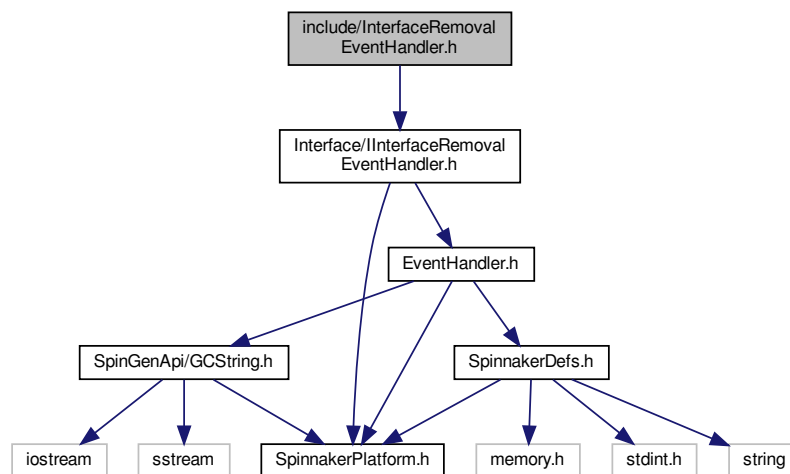
- class [InterfacePtr](#)
A reference tracked pointer to the interface object.

Namespaces

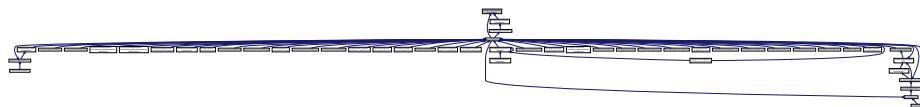
- [Spinnaker](#)

16.55 include/InterfaceRemovalEventHandler.h File Reference

Include dependency graph for InterfaceRemovalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

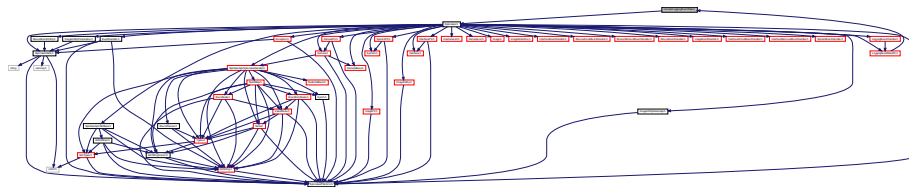
- class [InterfaceRemovalEventHandler](#)
An event handler for capturing the interface removal event.

Namespaces

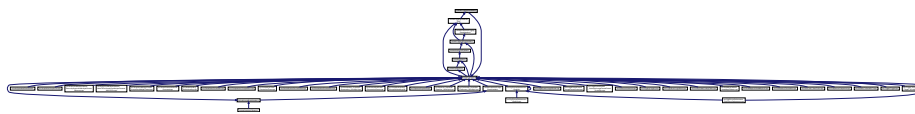
- [Spinnaker](#)

16.56 include/LoggingEventData.h File Reference

Include dependency graph for LoggingEventData.h:



This graph shows which files directly or indirectly include this file:



Classes

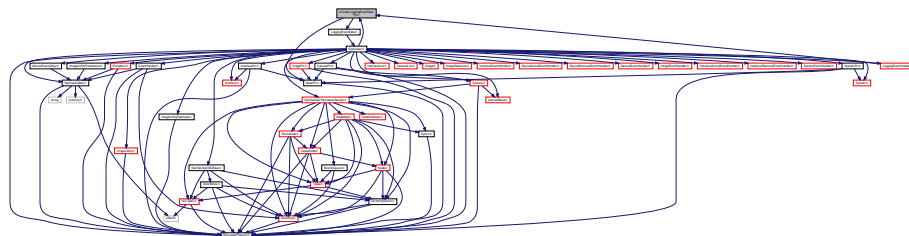
- class [LoggingEventData](#)
The *LoggingEventData* object.

Namespaces

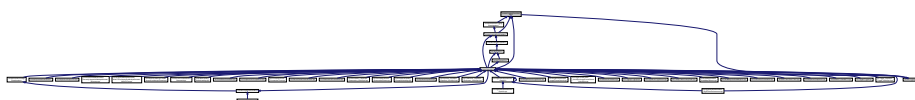
- [Spinnaker](#)

16.57 include/LoggingEventDataPtr.h File Reference

Include dependency graph for LoggingEventDataPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [LoggingEventDataPtr](#)

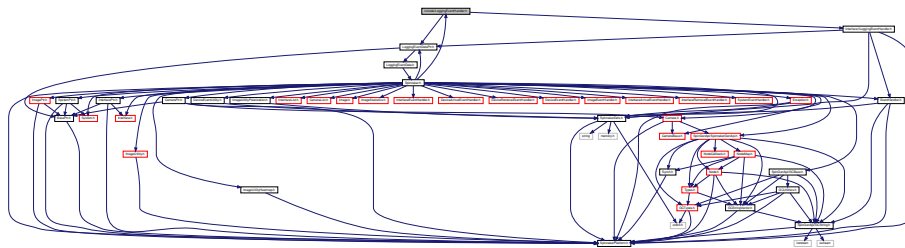
A reference tracked pointer to the LoggingEvent object.

Namespaces

- [Spinnaker](#)

16.58 include/LoggingEventHandler.h File Reference

Include dependency graph for LoggingEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [LoggingEventHandler](#)

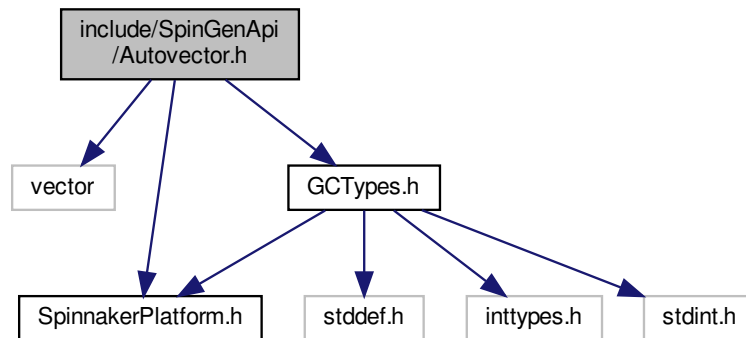
An event handler for capturing the device logging event.

Namespaces

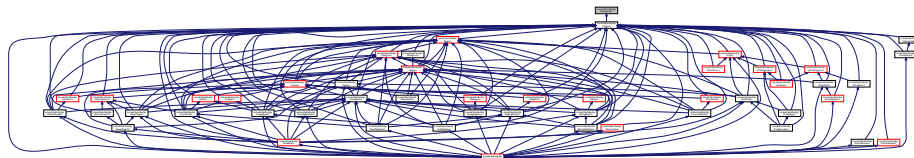
- [Spinnaker](#)

16.59 include/SpinGenApi/Autovector.h File Reference

Include dependency graph for Autovector.h:



This graph shows which files directly or indirectly include this file:



Classes

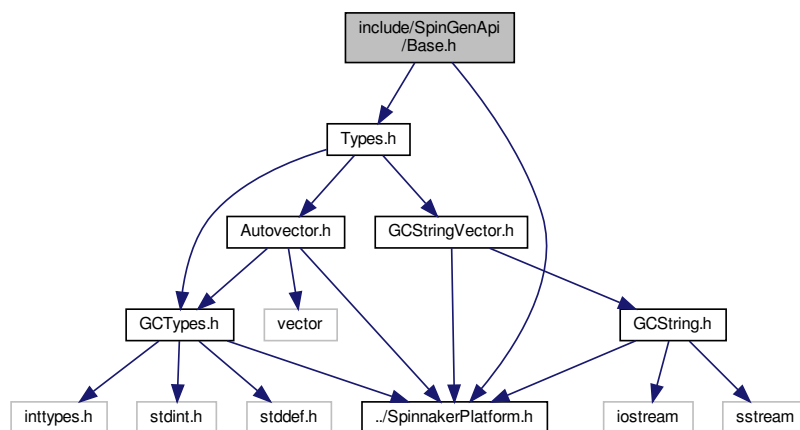
- class `int64_autovector_t`
Vector of integers with reference counting.
- class `double_autovector_t`
Vector of doubles with reference counting.

Namespaces

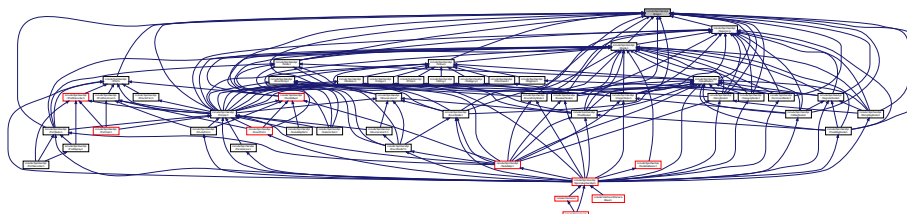
- `Spinnaker`
- `Spinnaker::GenApi`

16.60 include/SpinGenApi/Base.h File Reference

Include dependency graph for Base.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

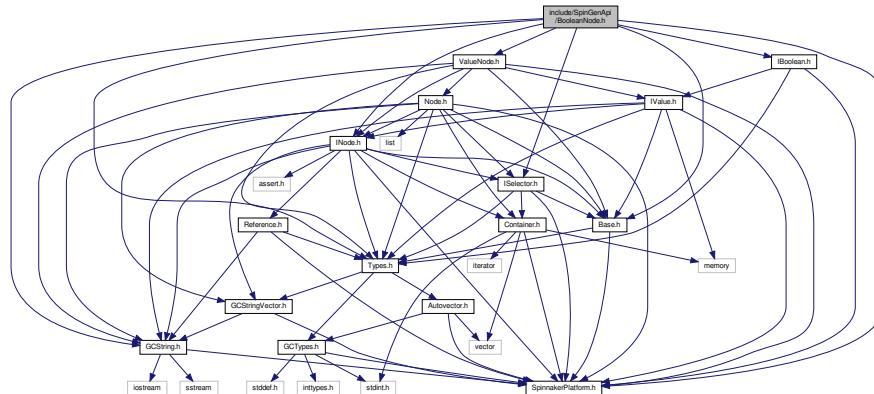
Variables

- [interface SPINNAKER_API_ABSTRACT IBase](#)

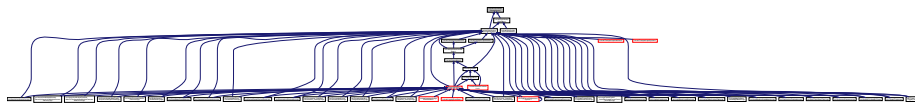
Base interface common to all nodes.

16.61 include/SpinGenApi/BooleanNode.h File Reference

Include dependency graph for BooleanNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `BooleanNode`
Interface for string properties.

Namespaces

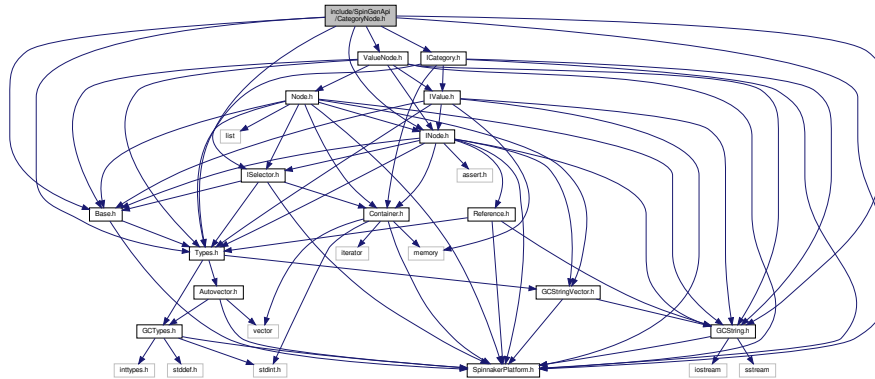
- Spinnaker
- Spinnaker::GenApi

Typedefs

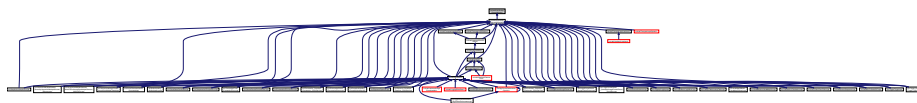
- typedef BooleanNode CBooleanRef

16.62 include/SpinGenApi/CategoryNode.h File Reference

Include dependency graph for CategoryNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CategoryNode](#)
Interface for string properties.

Namespaces

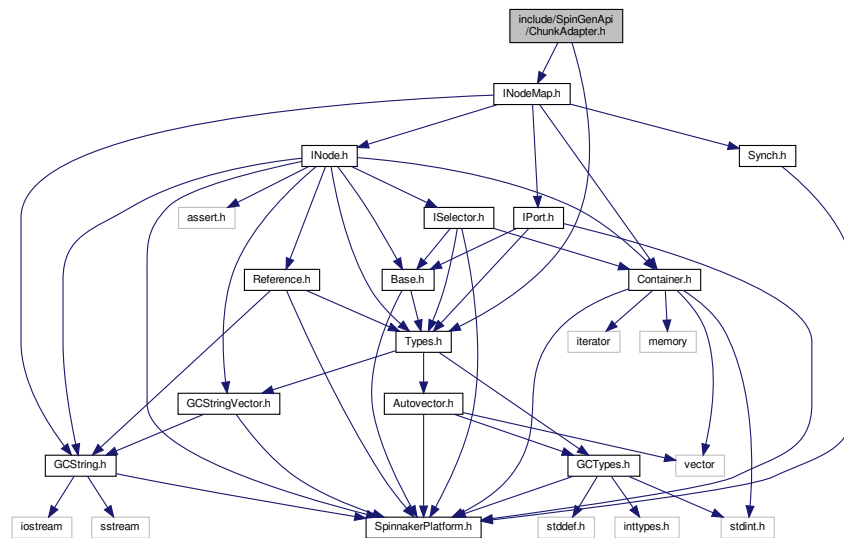
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

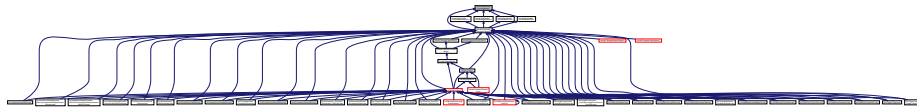
- typedef [CategoryNode](#) [CCategoryRef](#)

16.63 include/SpinGenApi/ChunkAdapter.h File Reference

Include dependency graph for ChunkAdapter.h:



This graph shows which files directly or indirectly include this file:



Classes

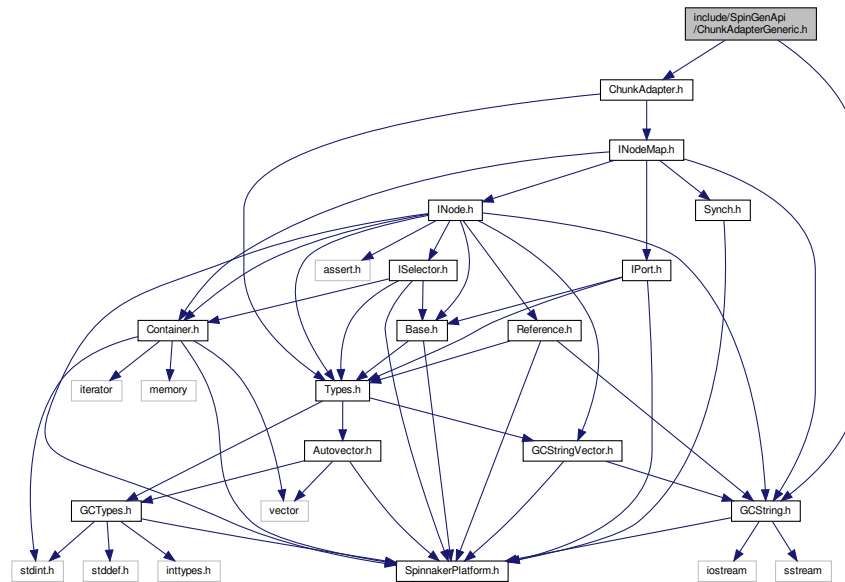
- struct [AttachStatistics_t](#)
Delivers information about the attached chunks and nodes.
- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.

Namespaces

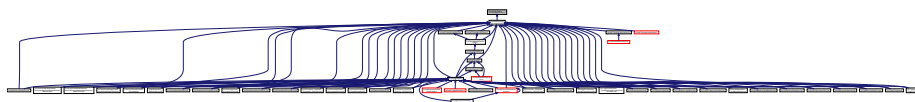
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.65 include/SpinGenApi/ChunkAdapterGeneric.h File Reference

Include dependency graph for ChunkAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



Classes

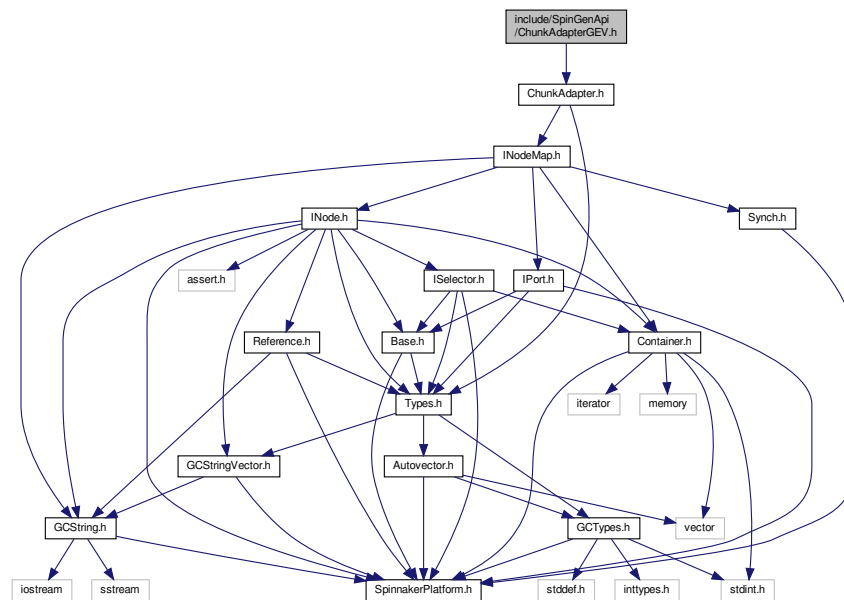
- struct [SingleChunkData_t](#)
- struct [SingleChunkDataStr_t](#)
- class [CChunkAdapterGeneric](#)

Namespaces

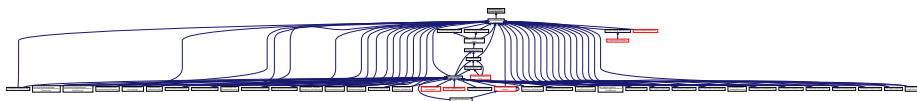
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.66 include/SpinGenApi/ChunkAdapterGEV.h File Reference

Include dependency graph for ChunkAdapterGEV.h:



This graph shows which files directly or indirectly include this file:



Classes

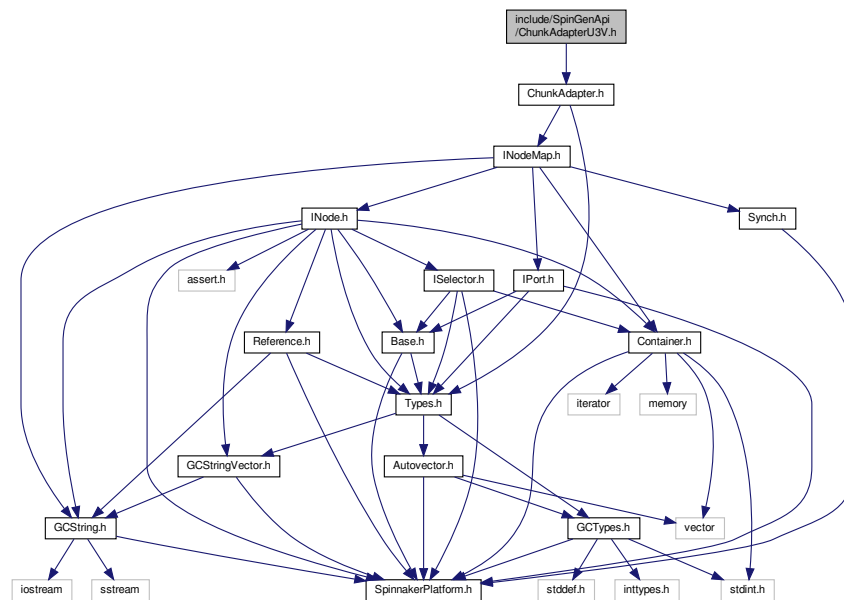
- struct [GVCP_CHUNK_TRAILER](#)
header of a GVCP request packet
- class [CChunkAdapterGEV](#)
Connects a chunked DCAM buffer to a node map.

Namespaces

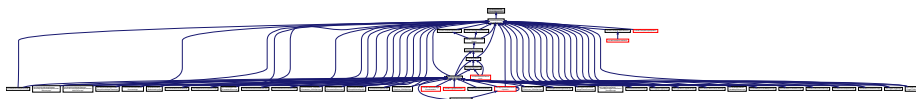
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.67 include/SpinGenApi/ChunkAdapterU3V.h File Reference

Include dependency graph for ChunkAdapterU3V.h:



This graph shows which files directly or indirectly include this file:



Classes

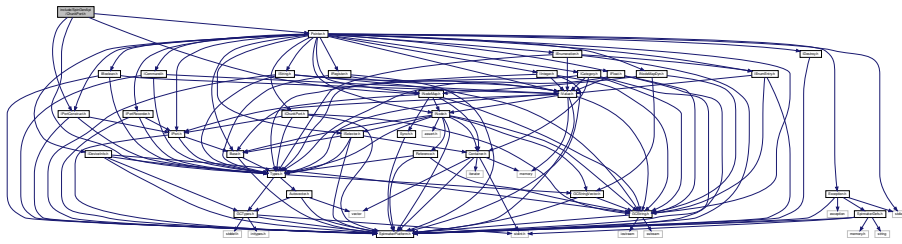
- struct [U3V_CHUNK_TRAILER](#)
header of a GVCP request packet
- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.

Namespaces

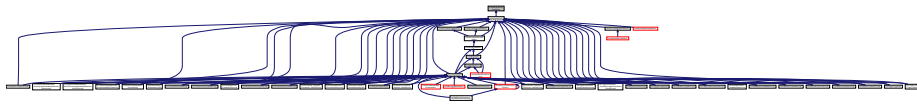
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.68 include/SpinGenApi/ChunkPort.h File Reference

Include dependency graph for ChunkPort.h:



This graph shows which files directly or indirectly include this file:



Classes

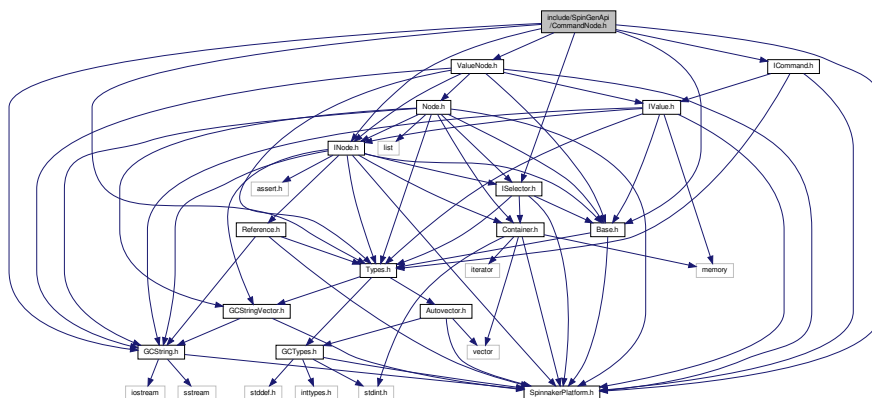
- class [CChunkPort](#)
Port attachable to a chunk in a buffer.

Namespaces

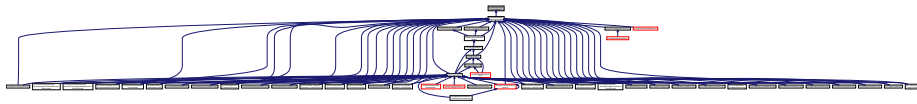
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.69 include/SpinGenApi/CommandNode.h File Reference

Include dependency graph for CommandNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CommandNode](#)
Interface for string properties.

Namespaces

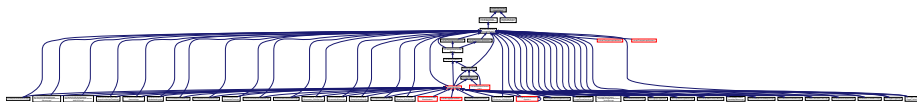
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef [CommandNode](#) [CCommandRef](#)

16.70 include/SpinGenApi/Compatibility.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [FMT_I64](#) "l"

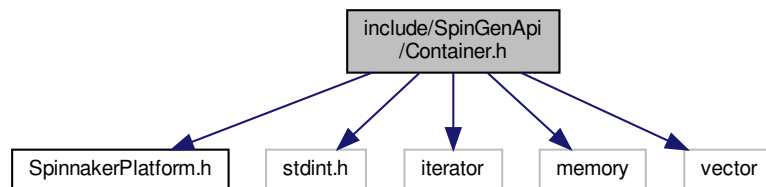
16.70.1 Macro Definition Documentation

16.70.1.1 FMT_I64

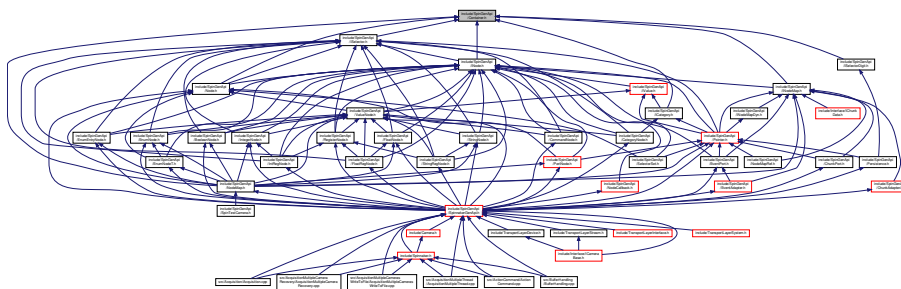
```
#define FMT_I64 "l"
```

16.71 include/SpinGenApi/Container.h File Reference

Include dependency graph for Container.h:



This graph shows which files directly or indirectly include this file:



16.72 include/SpinGenApi/Counter.h File Reference

Classes

- class [Counter](#)

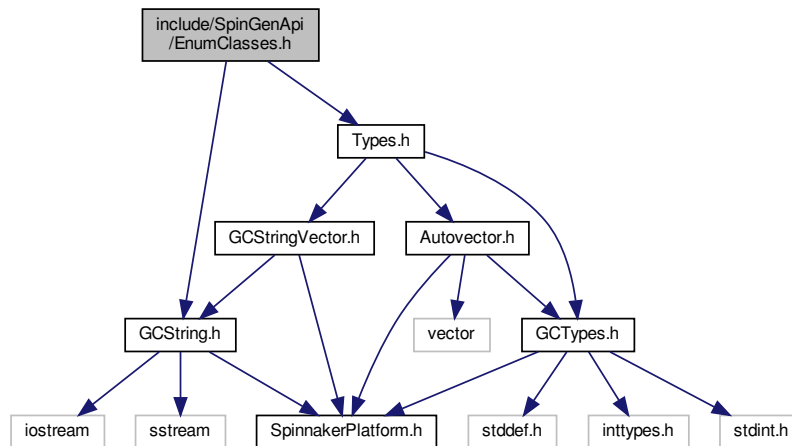
Definition of a simple [Counter](#) class.

Namespaces

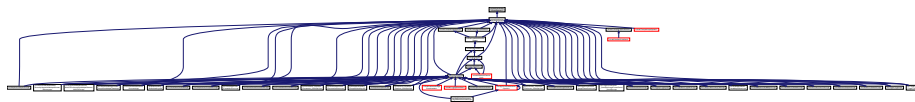
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.73 include/SpinGenApi/EnumClasses.h File Reference

Include dependency graph for EnumClasses.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [EEndiannessClass](#)
Holds conversion methods for the endianness enumeration.
- class [ERepresentationClass](#)
Holds conversion methods for the representation enumeration.
- class [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- class [EAccessModeClass](#)
Holds conversion methods for the access mode enumeration.
- class [ECachingModeClass](#)
Holds conversion methods for the caching mode enumeration.
- class [ENamespaceClass](#)
Holds conversion methods for the namespace enumeration.
- class [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [ESlopeClass](#)

Holds conversion methods for the converter formulas.

- class EDisplayNotationClass

Holds conversion methods for the notation type of floats.

- class `EInputDirectionClass`

Holds conversion methods for the notation type of floats.

- class EGenApiSchemaVersionClass

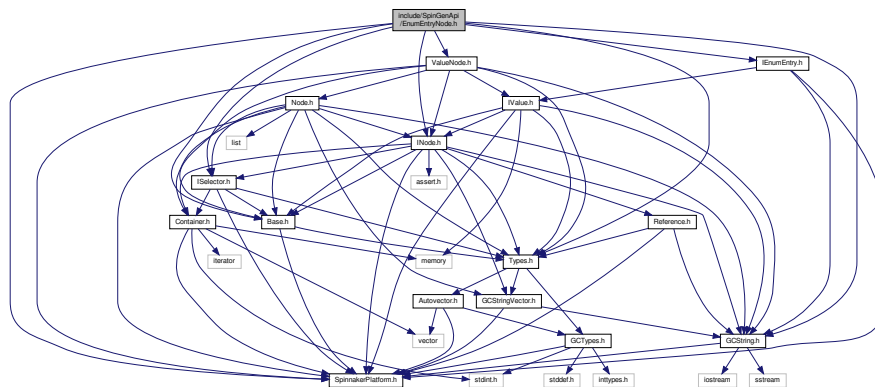
helper class converting EGenApiSchemaVersion from and to string

Namespaces

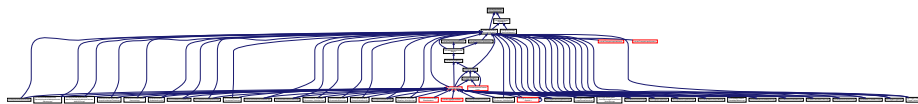
- Spinnaker
- Spinnaker::GenApi

16.74 include/SpinGenApi/EnumEntryNode.h File Reference

Include dependency graph for EnumEntryNode.h:



This graph shows which files directly or indirectly include this file:



Classes

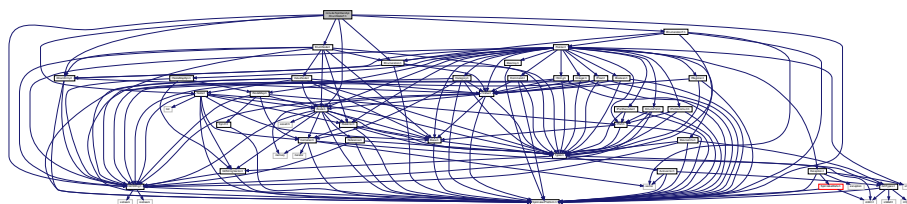
- class EnumEntryNode
Interface for string properties.

Namespaces

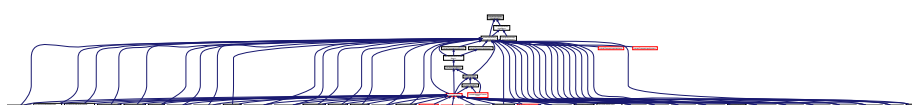
- Spinnaker
- Spinnaker::GenApi

16.76 include/SpinGenApi/EnumNodeT.h File Reference

Include dependency graph for EnumNodeT.h:



This graph shows which files directly or indirectly include this file:



Classes

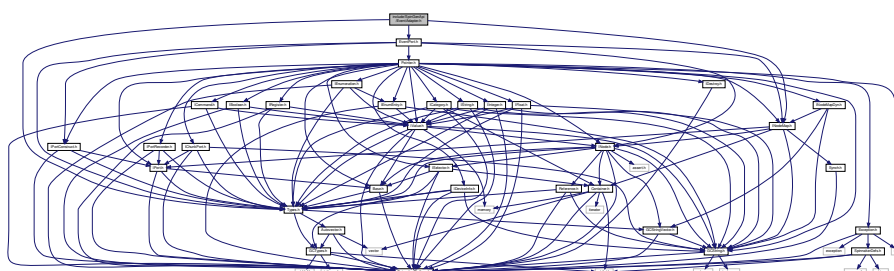
- class [CEnumerationTRef< EnumT >](#)
Interface for string properties.

Namespaces

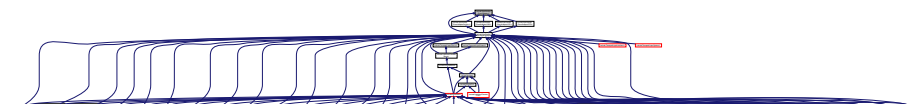
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.77 include/SpinGenApi/EventAdapter.h File Reference

Include dependency graph for EventAdapter.h:



This graph shows which files directly or indirectly include this file:



Classes

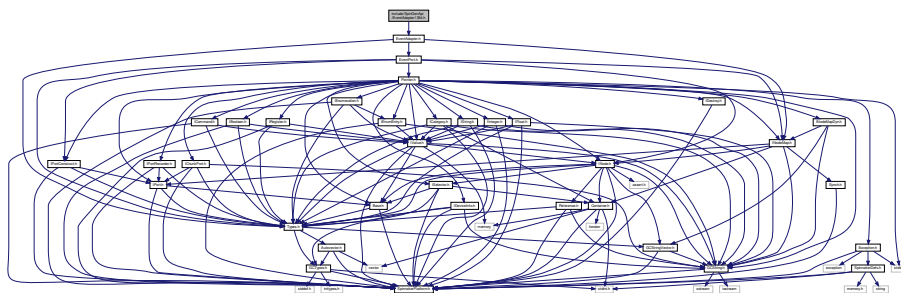
- class [CEventAdapter](#)
Delivers Events to ports.

Namespaces

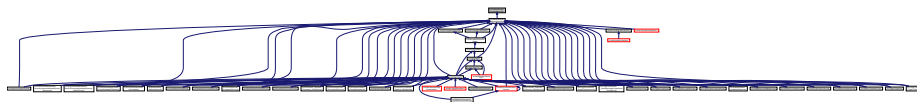
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.78 include/SpinGenApi/EventAdapter1394.h File Reference

Include dependency graph for EventAdapter1394.h:



This graph shows which files directly or indirectly include this file:



Classes

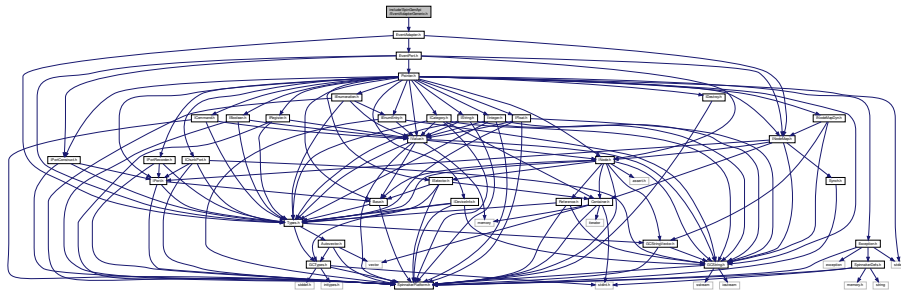
- class [CEventAdapter1394](#)
Distribute the events to the node map.

Namespaces

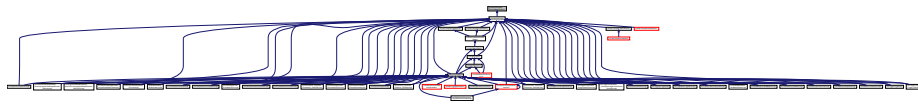
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.79 include/SpinGenApi/EventAdapterGeneric.h File Reference

Include dependency graph for EventAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



Classes

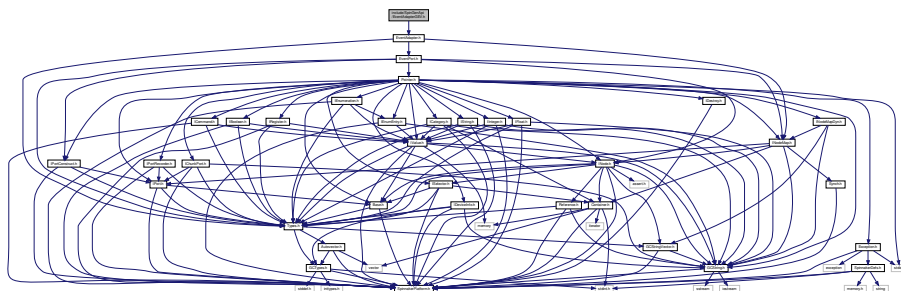
- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.

Namespaces

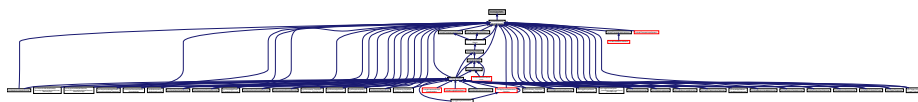
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.80 include/SpinGenApi/EventAdapterGEV.h File Reference

Include dependency graph for EventAdapterGEV.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [GVCP_REQUEST_HEADER](#)
header of a GVCP request packet
- struct [GVCP_EVENT_ITEM_BASIC](#)
layout of a GVCP event item (common to all types)
- struct [GVCP_EVENT_ITEM](#)
layout of a GVCP event item (Extended ID flag not set)
- struct [GVCP_EVENT_REQUEST](#)
Layout of a GVCP event request packet (Extended ID flag not set)
- struct [GVCP_EVENTDATA_REQUEST](#)
Layout of a GVCP event data request packet (Extended ID flag not set)
- struct [GVCP_EVENT_ITEM_EXTENDED_ID](#)
layout of a GVCP event item (Extended ID flag set)
- struct [GVCP_EVENT_REQUEST_EXTENDED_ID](#)
Layout of a GVCP event request packet (Extended ID flag set)
- struct [GVCP_EVENTDATA_REQUEST_EXTENDED_ID](#)
Layout of a GVCP event data request packet (Extended ID flag set)
- class [CEventAdapterGEV](#)
Connects a GigE Event to a node map.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Enumerations

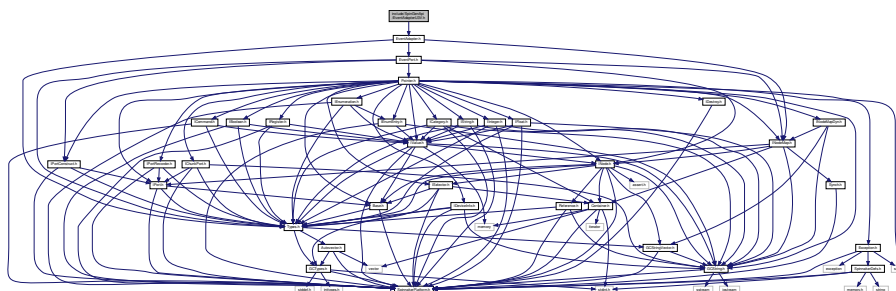
- enum [GVCP_MESSAGE_TAGS](#) {
 [TAG_EVENT_CMD](#) = 0xc0,
 [TAG_EVENTDATA_CMD](#) = 0xc2 }

Variables

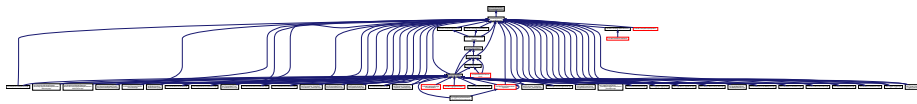
- const uint8_t [COMMAND_MAGIC](#) = 0x42

16.81 include/SpinGenApi/EventAdapterU3V.h File Reference

Include dependency graph for EventAdapterU3V.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [U3V_COMMAND_HEADER](#)
U3V/GenCP command header.
- struct [U3V_EVENT_DATA](#)
U3V/GenCP EVENT_CMD specific command data.
- struct [U3V_EVENT_MESSAGE](#)
Entire event data message (without the variable-sized data field)
- class [CEventAdapterU3V](#)
Connects a U3V Event to a node map.

Namespaces

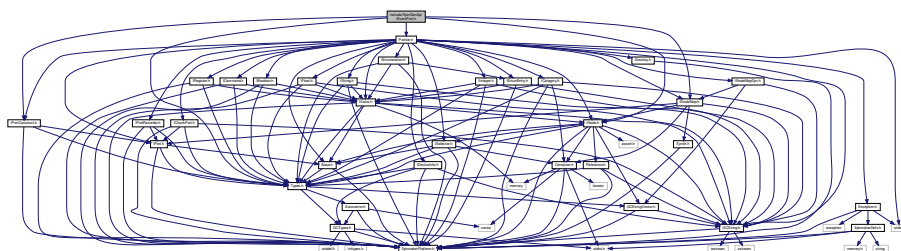
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Variables

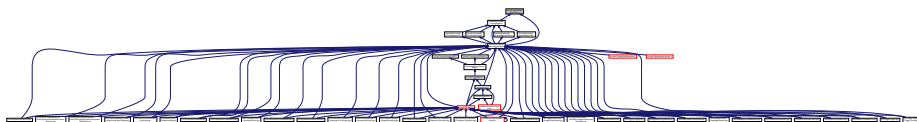
- const uint32_t [U3V_EVENT_PREFIX](#) = 0x45563355
- const uint16_t [GENCP_EVENT_CMD_ID](#) = 0x0C00
- const size_t [GENCP_COMMAND_HEADER_SIZE](#) = sizeof([U3V_COMMAND_HEADER](#))
- const size_t [GENCP_EVENT_BASIC_SIZE](#) = sizeof([U3V_EVENT_MESSAGE](#))

16.82 include/SpinGenApi/EventPort.h File Reference

Include dependency graph for EventPort.h:



This graph shows which files directly or indirectly include this file:



Classes

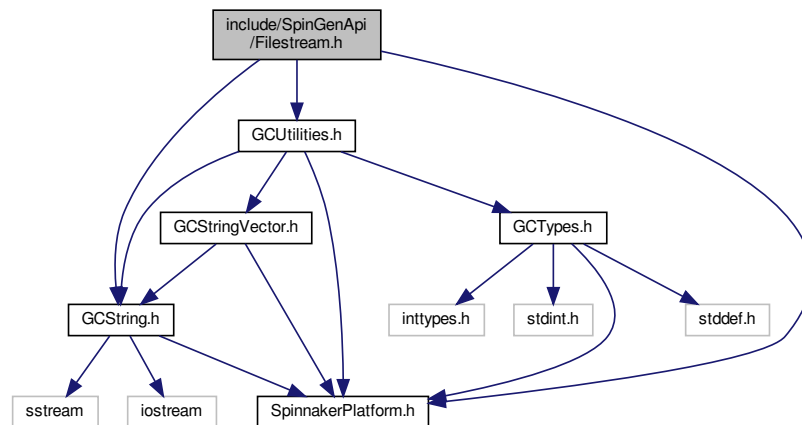
- class [CEventPort](#)
Port attachable to an event.

Namespaces

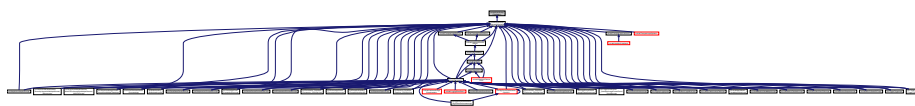
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.83 include/SpinGenApi/Filestream.h File Reference

Include dependency graph for FileStream.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [FileProtocolAdapter](#)
Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.
- class [IDevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBase< CharType, Traits >](#)
- class [IDevFileStreamBase< CharType, Traits >](#)

Namespaces

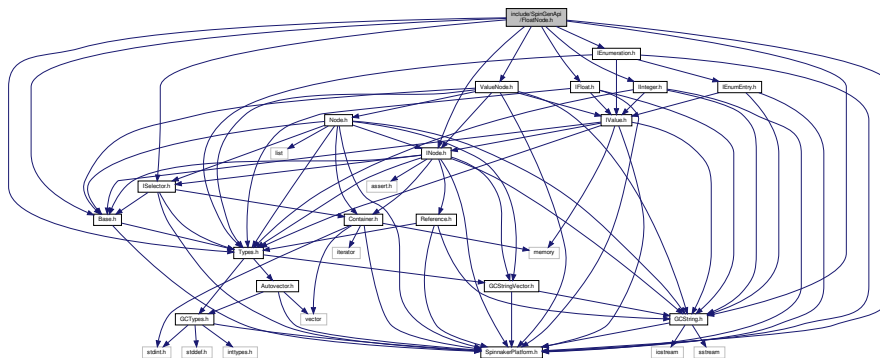
- Spinnaker
- Spinnaker::GenApi

Typedefs

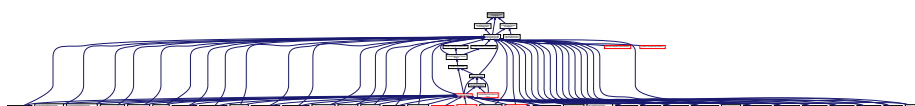
- typedef ODevFileStreamBase< char, std::char_traits< char > > ODevFileStream
- typedef IDevFileStreamBase< char, std::char_traits< char > > IDevFileStream

16.84 include/SpinGenApi/FloatNode.h File Reference

Include dependency graph for FloatNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `FloatNode`
Interface for string properties.

Namespaces

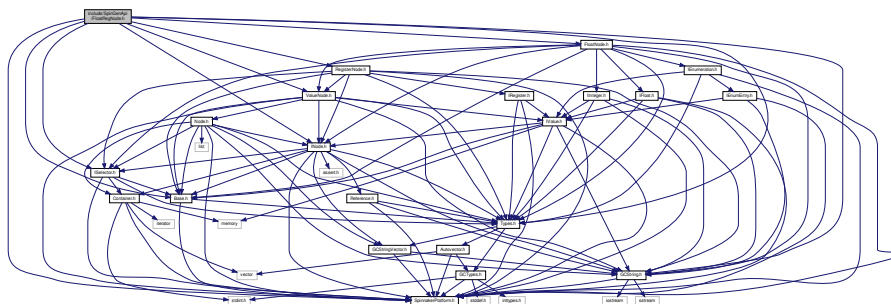
- Spinnaker
- Spinnaker::GenApi

Typedefs

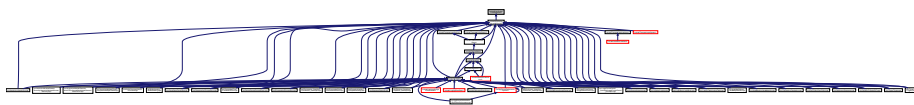
- typedef FloatNode CFloatRef

16.85 include/SpinGenApi/FloatRegNode.h File Reference

Include dependency graph for FloatRegNode.h:



This graph shows which files directly or indirectly include this file:



Classes

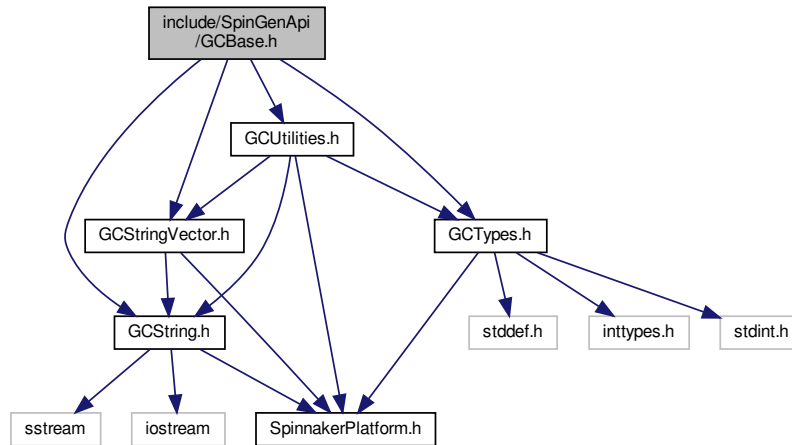
- class `FloatRegNode`
Interface for string properties.

Namespaces

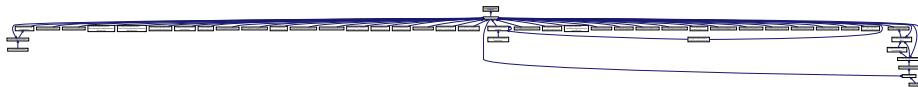
- Spinnaker
- Spinnaker::GenApi

16.86 include/SpinGenApi/GCBase.h File Reference

Include dependency graph for GCBase.h:

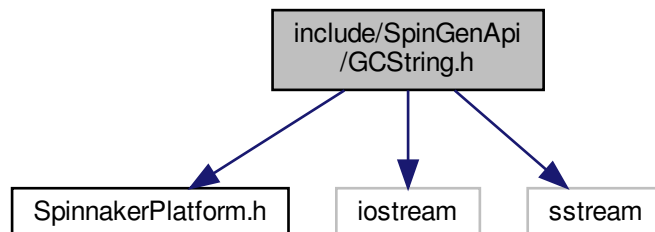


This graph shows which files directly or indirectly include this file:

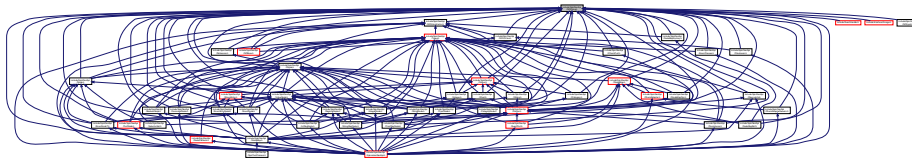


16.87 include/SpinGenApi/GCString.h File Reference

Include dependency graph for GCString.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gcstring](#)

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenlCam](#)

Macros

- `#define GCSTRING\_NPOS size_t(-1)`

Functions

- [SPINNAKER_API](#) void [ThrowBadAlloc](#) ()
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenlCam::gcstring](#) &str)
STL getline.
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenlCam::gcstring](#) &str, char delim)
STL getline.
- std::ostream & [operator<<](#) (std::ostream &ostr, const [Spinnaker::GenlCam::gcstring](#) &str)
STL operator out.
- std::istream & [operator>>](#) (std::istream &istr, [Spinnaker::GenlCam::gcstring](#) &str)
STL operator in.

16.87.1 Macro Definition Documentation

16.87.1.1 GCSTRING_NPOS

```
#define GCSTRING_NPOS size_t(-1)
```

16.87.2 Function Documentation

16.87.2.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & ostr,
    const Spinnaker::GenICam::gcstring & str ) [inline]
```

STL operator out.

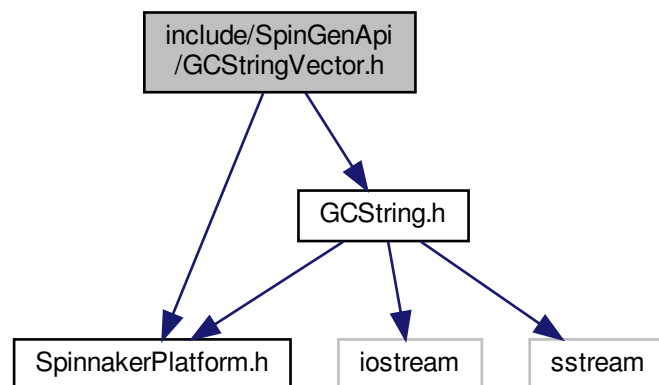
16.87.2.2 operator>>()

```
std::istream& operator>> (
    std::istream & istr,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

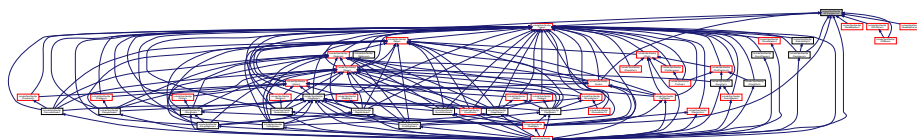
STL operator in.

16.88 include/SpinGenApi/GCStringVector.h File Reference

Include dependency graph for GCStringVector.h:

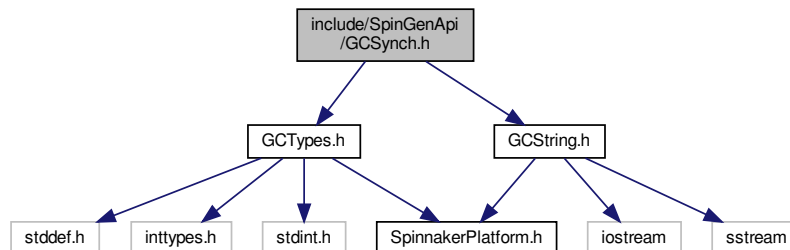


This graph shows which files directly or indirectly include this file:



16.89 include/SpinGenApi/GCSynch.h File Reference

Include dependency graph for GCSynch.h:



Classes

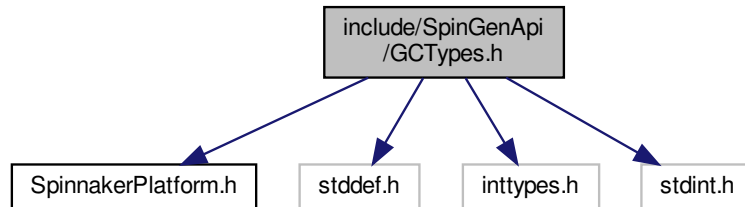
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)
- class [LockableObject< Object >](#)
Instance-Lock for an object.
- class [LockableObject< Object >::Lock](#)
A scopelevel [Lock](#) class.
- class [CGlobalLock](#)
Named global lock which can be used over process boundaries.
- class [CGlobalLockUnlocker](#)
Unlocks the global lock object on destruction.

Namespaces

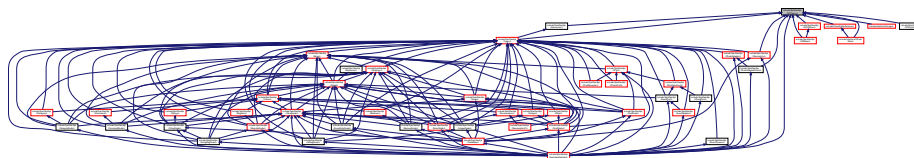
- [Spinnaker](#)
- [Spinnaker::GenICam](#)

16.90 include/SpinGenApi/GCTypes.h File Reference

Include dependency graph for GCTypes.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Version_t](#)
Version.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

Macros

- `#define __STDC_LIMIT_MACROS`
- `#define __STDC_CONSTANT_MACROS`
- `#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */`
- `#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */`
- `#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */`
- `#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */`
- `#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */`
- `#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */`
- `#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */`
- `#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /* minimum signed int8 value */`
- `#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */`

Typedefs

- typedef float [float32_t](#)
32 bit floating point
- typedef double [float64_t](#)
64 bit floating point

16.90.1 Macro Definition Documentation

16.90.1.1 __STDC_CONSTANT_MACROS

```
#define __STDC_CONSTANT_MACROS
```

16.90.1.2 __STDC_LIMIT_MACROS

```
#define __STDC_LIMIT_MACROS
```

16.90.1.3 GC_INT32_MAX

```
#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */
```

16.90.1.4 GC_INT32_MIN

```
#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */
```

16.90.1.5 GC_INT64_MAX

```
#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */
```

16.90.1.6 GC_INT64_MIN

```
#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */
```

16.90.1.7 GC_INT8_MAX

```
#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */
```

16.90.1.8 GC_INT8_MIN

```
#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /* minimum signed int8 value */
```

16.90.1.9 GC_UINT32_MAX

```
#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffUULL) /* maximum unsigned int32 value */
```

16.90.1.10 GC_UINT64_MAX

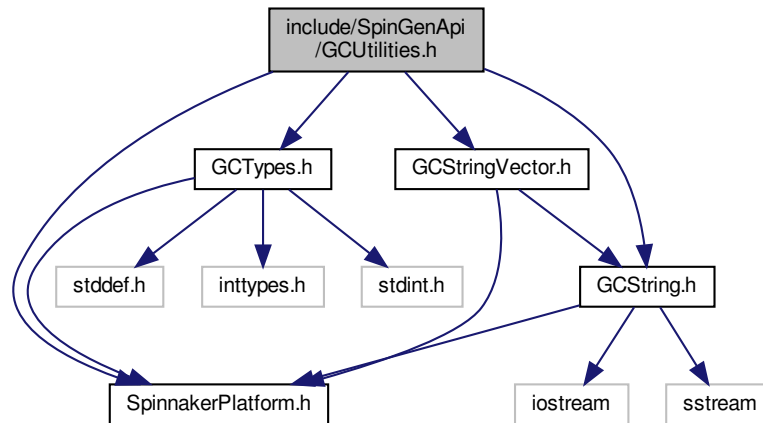
```
#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffUULL) /* maximum unsigned int64 value */
```

16.90.1.11 GC_UINT8_MAX

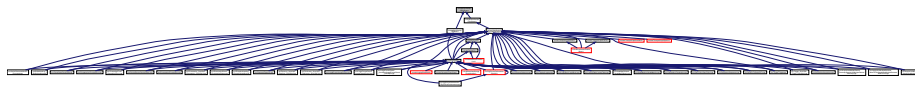
```
#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */
```

16.91 include/SpinGenApi/GCUtilities.h File Reference

Include dependency graph for GCUtilities.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

Macros

- `#define USE_TEMP_CACHE_FILE 1`
- `#define USE_TEMP_CACHE_FILE 1`
- `#define GC_COUNTOF(arr) (sizeof(arr) / sizeof(arr)[0])`
- `#define GENICAM_UNUSED(unused_var) ((void)(unused_var))`
- `#define GENICAM_DEPRECATED(FUNCTION) FUNCTION`
- `#define _TO_STRING(__stN) # __stN`
- `#define EXPAND_TO_STRINGISE(__stN) _TO_STRING(__stN)`
- `#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)`
- `#define __LOCATION__ __FILE__ "(" __LINE_STR__ ")"`
- `#define __OUTPUT_FORMATER__(_type) __LOCATION__ ": " _type " : "`
- `#define __WARN__ __OUTPUT_FORMATER__("WARNING")`
- `#define __ERR__ __OUTPUT_FORMATER__("ERROR")`
- `#define __TODO__ __OUTPUT_FORMATER__("TBD")`

Functions

- `template<typename Td , typename Ts >`
`Td INTEGRAL_CAST2 (Ts s)`
This verifies at runtime if there was no loss of data if an type Ts (e.g.
- `template<typename T >`
`T INTEGRAL_CAST (int64_t ll)`
This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.
- `SPINNAKER_API bool DoesEnvironmentVariableExist (const Spinnaker::GenICam::gcstring &VariableName)`
Returns true if an environment variable exists.
- `SPINNAKER_API gcstring GetValueOfEnvironmentVariable (const gcstring &VariableName)`
Retrieve the value of an environment variable.
- `SPINNAKER_API bool GetValueOfEnvironmentVariable (const gcstring &VariableName, gcstring &VariableContent)`
Retrieve the value of an environment variable.
- `SPINNAKER_API gcstring UriEncode (const gcstring &Input)`
Converts \ to / and replaces all unsafe characters by their xx equivalent.
- `SPINNAKER_API gcstring UriDecode (const gcstring &Input)`
Replaces xx escapes by their char equivalent.
- `SPINNAKER_API void ReplaceEnvironmentVariables (gcstring &Buffer, bool ReplaceBlankBy20=false)`
Replaces in a string and replace ' ' with %20.
- `SPINNAKER_API gcstring GetGenICamCacheFolder (void)`
Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling SetGenICamCacheFolder().
- `SPINNAKER_API gcstring GetGenICamLogConfig (void)`
Retrieve the path of the GenICam logging properties file.
- `SPINNAKER_API gcstring GetGenICamCLProtocolFolder (void)`
Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling SetGenICamCLProtocolFolder().
- `SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)`
Stores the path of the GenICam cache folder.
- `SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)`
Stores the path of the GenICam logging properties file.
- `SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)`
Stores the path of the CLProtocol folder.
- `SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")`
splits str input string into a list of tokens using the delimiter
- `SPINNAKER_API void GetFiles (const gcstring &FileTemplate, gcstring_vector &FileNames, const bool DirectoriesOnly=false)`
Gets a list of files or directories matching a given FileTemplate.
- `SPINNAKER_API gcstring GetModulePathFromFunction (void *pFunction)`
Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

16.91.1 Macro Definition Documentation

16.91.1.1 __ERR__

```
#define __ERR__  __OUTPUT_FORMATER__("ERROR")
```

16.91.1.2 __LINE_STR__

```
#define __LINE_STR__  EXPAND_TO_STRINGISE(__LINE__)
```

16.91.1.3 __LOCATION__

```
#define __LOCATION__  __FILE__ "(" __LINE_STR__ ")"
```

16.91.1.4 __OUTPUT_FORMATER__

```
#define __OUTPUT_FORMATER__(  
    _type )  __LOCATION__ " : " _type " : "
```

16.91.1.5 __TODO__

```
#define __TODO__  __OUTPUT_FORMATER__("TBD")
```

16.91.1.6 __WARN__

```
#define __WARN__  __OUTPUT_FORMATER__("WARNING")
```

16.91.1.7 _TO_STRING

```
#define _TO_STRING(  
    __stN )  #__stN
```

16.91.1.8 EXPAND_TO_STRINGISE

```
#define EXPAND_TO_STRINGISE(  
    __stN ) _TO_STRING(__stN)
```

16.91.1.9 GC_COUNTOF

```
#define GC_COUNTOF(  
    arr ) (sizeof(arr) / sizeof(arr)[0])
```

16.91.1.10 GENICAM_DEPRECATED

```
#define GENICAM_DEPRECATED(  
    FUNCTION ) FUNCTION
```

16.91.1.11 GENICAM_UNUSED

```
#define GENICAM_UNUSED(  
    unused_var ) ((void)(unused_var))
```

16.91.1.12 USE_TEMP_CACHE_FILE [1/2]

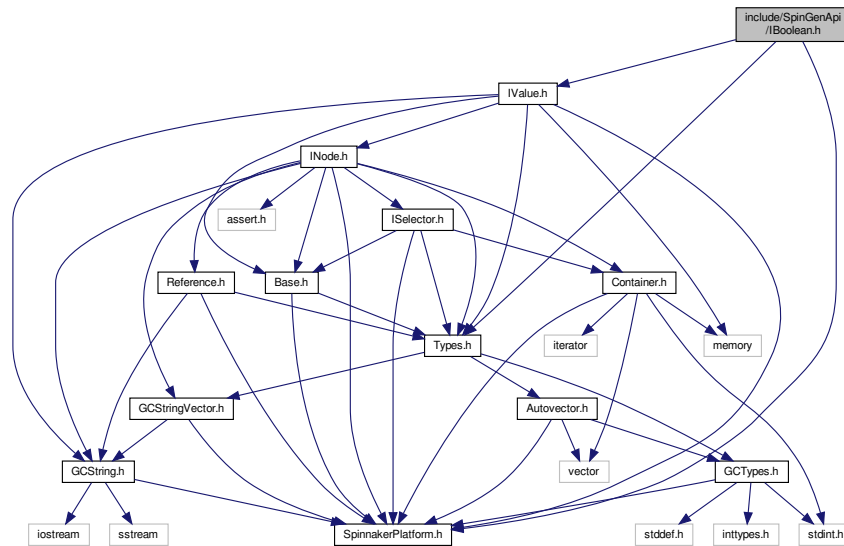
```
#define USE_TEMP_CACHE_FILE 1
```

16.91.1.13 USE_TEMP_CACHE_FILE [2/2]

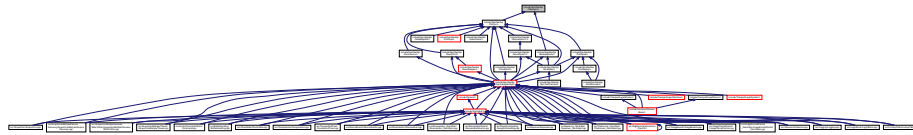
```
#define USE_TEMP_CACHE_FILE 1
```

16.92 include/SpinGenApi/IBoolean.h File Reference

Include dependency graph for IBoolean.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

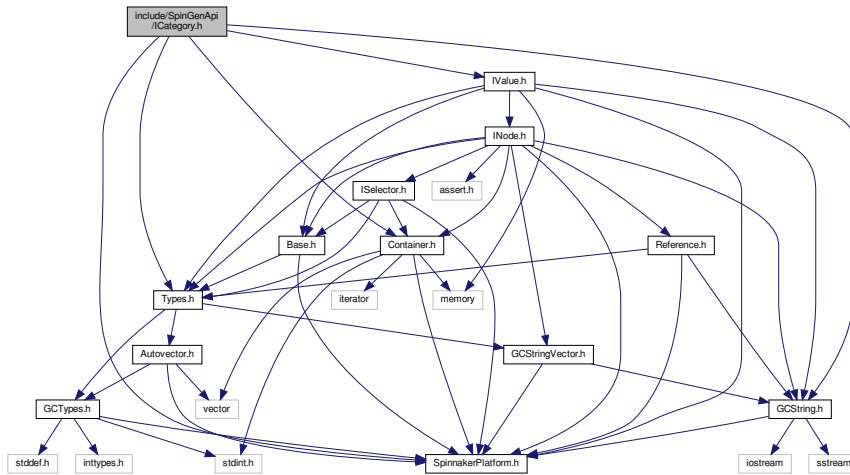
- virtual void [operator=](#) (bool Value)
Set node value.
- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const
Get node value.

Variables

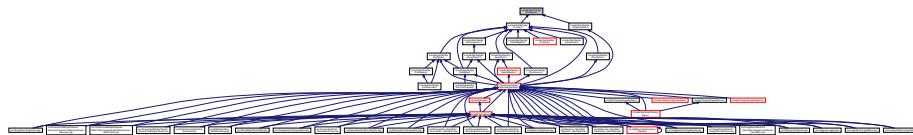
- [interface SPINNAKER_API_ABSTRACT IBoolean](#)
Interface for Boolean properties.
- [interface SPINNAKER_API_ABSTRACT](#) bool [Verify](#) = true) = 0

16.93 include/SpinGenApi/ICategory.h File Reference

Include dependency graph for ICategory.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

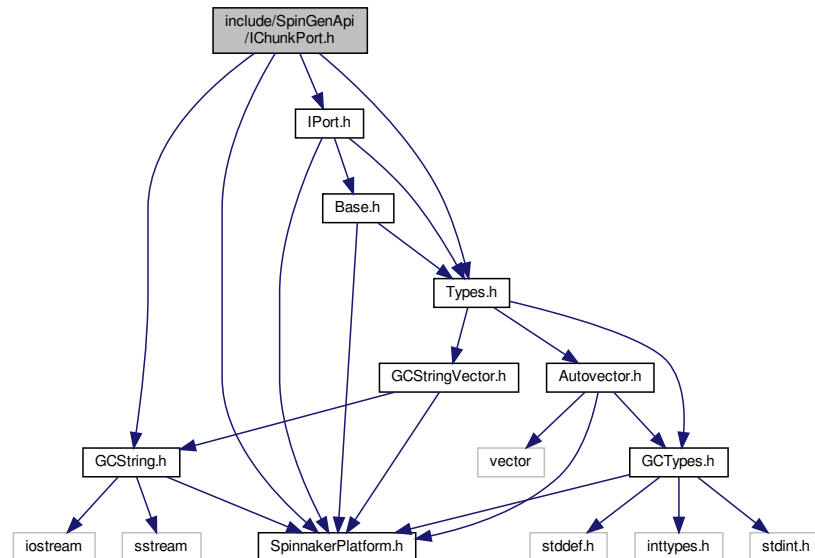
Variables

- [interface SPINNAKER_API_ABSTRACT ICategory](#)

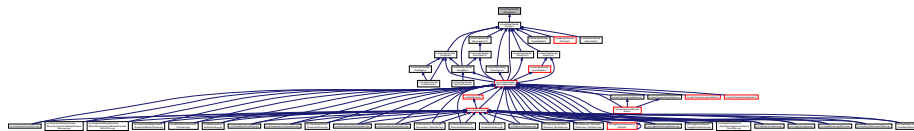
Gives access to a category node.

16.94 include/SpinGenApi/IChunkPort.h File Reference

Include dependency graph for IChunkPort.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Macros

- `#define` [CHUNK_BASE_ADDRESS_REGISTER](#) [GC_INT64_MAX](#)
Address of a `int64_t` pseudo register containing the base address of the chunk (`MAX_INT64`)
- `#define` [CHUNK_BASE_ADDRESS_REGISTER_LEN](#) 8
Length of the `CHUNK_BASE_ADDRESS_REGISTER` pseudo register.
- `#define` [CHUNK_LENGTH_REGISTER](#) ([GC_INT64_MAX](#) - 15)
Address of a `int64_t` pseudo register containing the length of the chunk.
- `#define` [CHUNK_LENGTH_REGISTER_LEN](#) 8
Length of the `CHUNK_LENGTH_REGISTER` pseudo register.

Functions

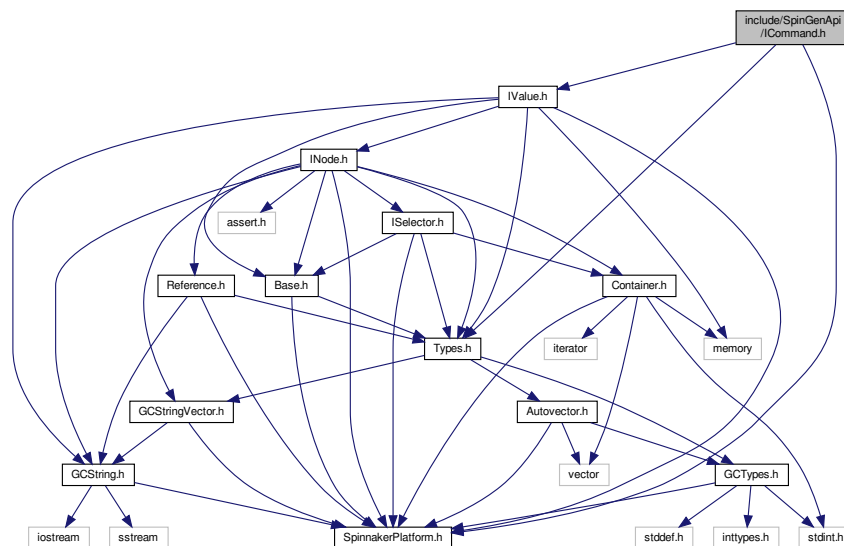
- virtual EYesNo [CacheChunkData](#) () const =0
Indicates if the chunk a adapter must hold a cached version of the chunk data.

Variables

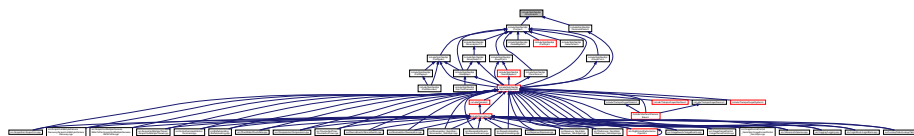
- [interface SPINNAKER_API_ABSTRACT IChunkPort](#)
Interface for ports attached to a chunk.

16.95 include/SpinGenApi/ICommand.h File Reference

Include dependency graph for ICommand.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

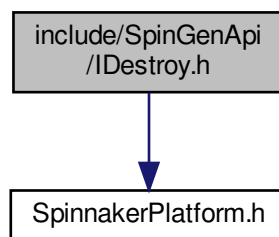
- virtual bool [operator\(\)](#) () const
Get node value.
- virtual bool [IsDone](#) (bool Verify=true)=0
Query whether the command is executed.

Variables

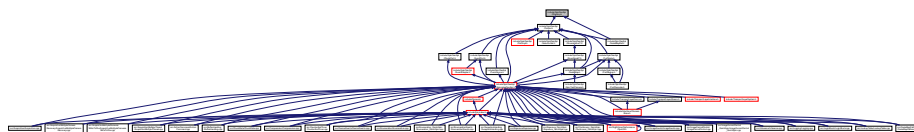
- [interface SPINNAKER_API_ABSTRACT ICommand](#)
Interface for command like properties.

16.96 include/SpinGenApi/IDestroy.h File Reference

Include dependency graph for IDestroy.h:



This graph shows which files directly or indirectly include this file:



Namespaces

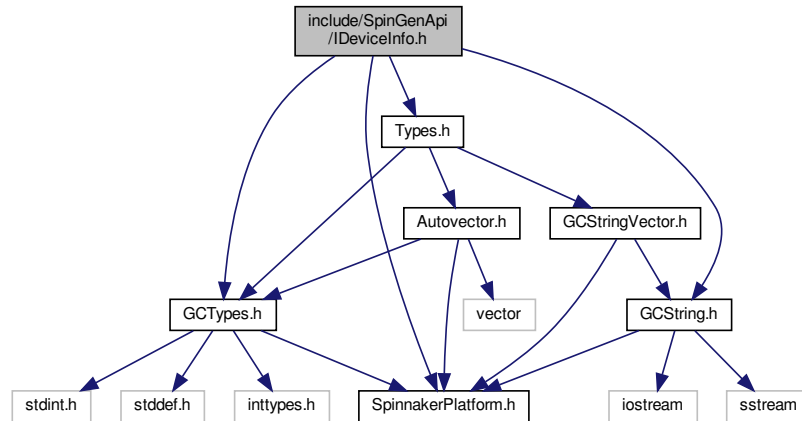
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Variables

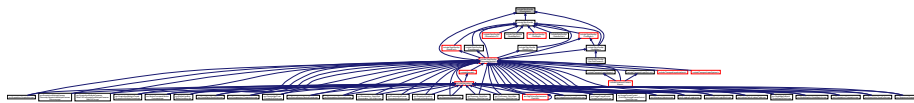
- [interface SPINNAKER_API_ABSTRACT IDestroy](#)
Interface to destroy an object.

16.97 include/SpinGenApi/IDeviceInfo.h File Reference

Include dependency graph for IDeviceInfo.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

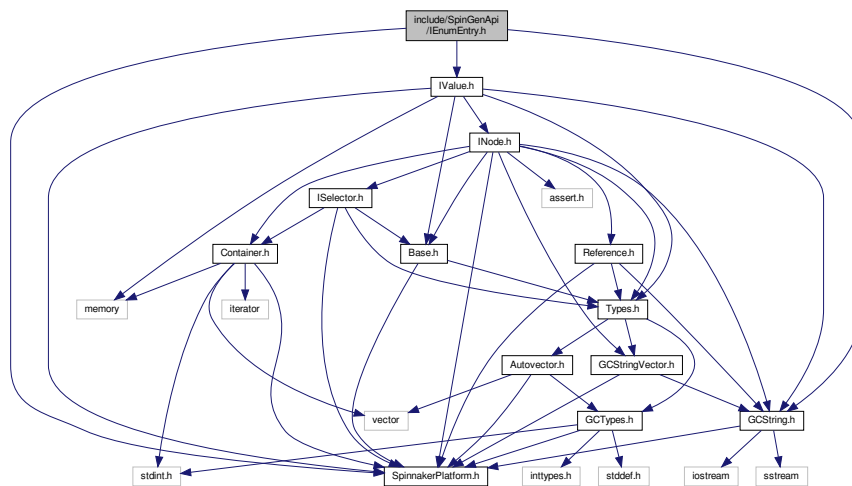
- virtual GenICam::gcstring [GetVendorName](#) ()=0
Get the vendor name.
- virtual GenICam::gcstring [GetToolTip](#) ()=0
Get tool tip.
- virtual GenICam::gcstring [GetStandardNameSpace](#) ()=0
Get the standard name space.
- virtual void [GetGenApiVersion](#) (GenICam::Version_t &Version, uint16_t &Build)=0
Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) (GenICam::Version_t &Version)=0
Get the schema version number.
- virtual void [GetDeviceVersion](#) (GenICam::Version_t &Version)=0
Get the version of the device description file.
- virtual GenICam::gcstring [GetProductGuid](#) ()=0
Get the Guid describing the product.
- virtual GenICam::gcstring [GetVersionGuid](#) ()=0
Get the Guid describing the product version.

Variables

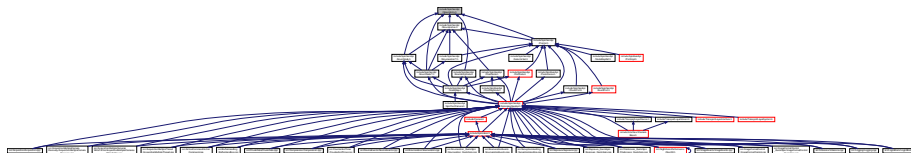
- [interface SPINNAKER_API_ABSTRACT IDeviceInfo](#)
Interface to get information about the device (= nodemap)

16.98 include/SpinGenApi/IEnumEntry.h File Reference

Include dependency graph for IEnumEntry.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

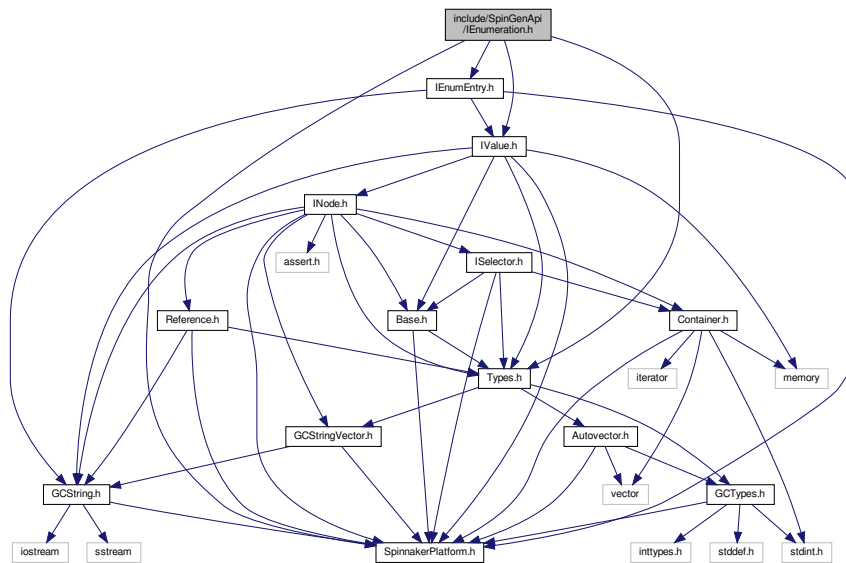
- virtual `GenICam::gcstring` [GetSymbolic](#) () const =0
Get symbolic enum value.
- virtual `double` [GetNumericValue](#) ()=0
Get double number associated with the entry.
- virtual `bool` [IsSelfClearing](#) ()=0
Indicates if the corresponding EnumEntry is self clearing.

Variables

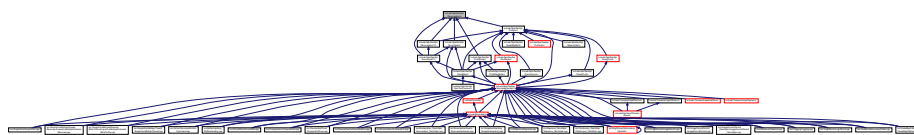
- [interface SPINNAKER_API_ABSTRACT IEnumEntry](#)
Interface of single enum value.

16.99 include/SpinGenApi/IEnumeration.h File Reference

Include dependency graph for IEnumeration.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinaker](#)
- [Spinaker::GenApi](#)

Functions

- virtual void [GetEntries](#) (NodeList_t &Entries)=0
Get list of entry nodes.
- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0
Set string node value.
- virtual void [SetIntValue](#) (int64_t Value, bool Verify=true)=0

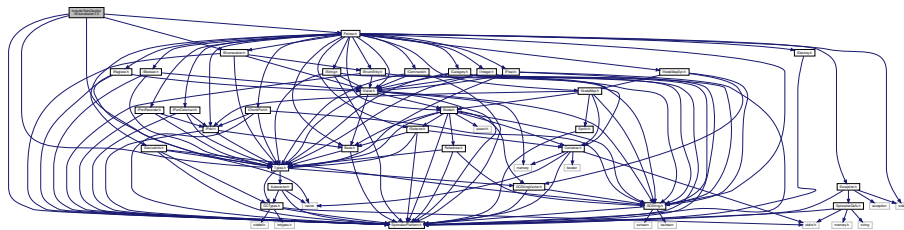
- Set integer node value.*
- virtual GenICam::gcstring [operator*](#) ()=0
- Get string node value.*
- virtual int64_t [GetIntValue](#) (bool Verify=false, bool IgnoreCache=false)=0
- Get integer node value.*
- virtual IEnumEntry * [GetEntryByName](#) (const GenICam::gcstring &Symbolic)=0
- Get an entry node by name.*
- virtual IEnumEntry * [GetEntry](#) (const int64_t IntValue)=0
- Get an entry node by its IntValue.*
- virtual IEnumEntry * [GetCurrentEntry](#) (bool Verify=false, bool IgnoreCache=false)=0
- Get the current entry.*

Variables

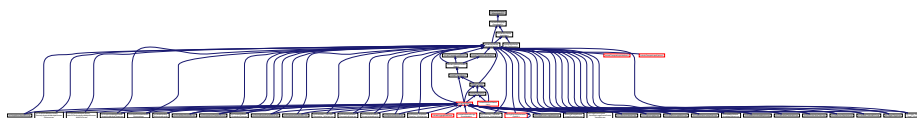
- [interface SPINNAKER_API_ABSTRACT IEnumeration](#)
Interface for enumeration properties.

16.100 include/SpinGenApi/IEnumerationT.h File Reference

Include dependency graph for IEnumerationT.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

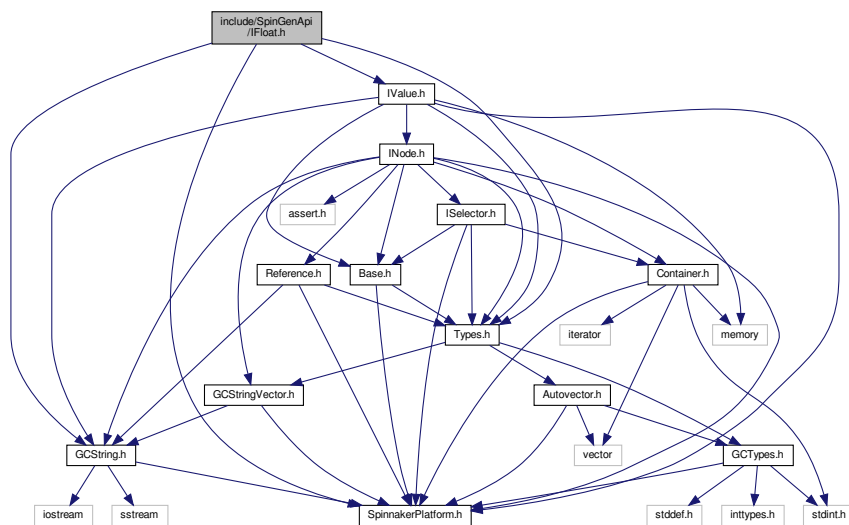
- virtual IEnumeration & **operator=** (EnumT Value)=0
Set node value.
- virtual bool **GetValue** (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool **operator()** () const
Get node value.
- virtual IEnumeration & **operator=** (const GenICam::gcstring &ValueStr)=0
Set string node value.
- virtual IEnumEntry * **GetEntry** (const int64_t IntValue)=0
Get an entry node by its IntValue.
- virtual IEnumEntry * **GetEntry** (const EnumT Value)=0
returns the EnumEntry object belonging to the Value
- virtual IEnumEntry * **GetCurrentEntry** (bool Verify=false, bool IgnoreCache=false)=0
Get the current entry.

Variables

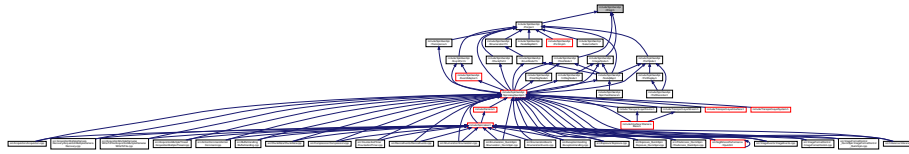
- template<typename EnumT >
interface SPINNAKER_API_ABSTRACT IEnumerationT
Interface for enumeration properties.
- template<typename EnumT >
interface SPINNAKER_API_ABSTRACT virtual public IEnumReference
Interface to construct an enum reference.

16.101 include/SpinGenApi/IFloat.h File Reference

Include dependency graph for IFloat.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

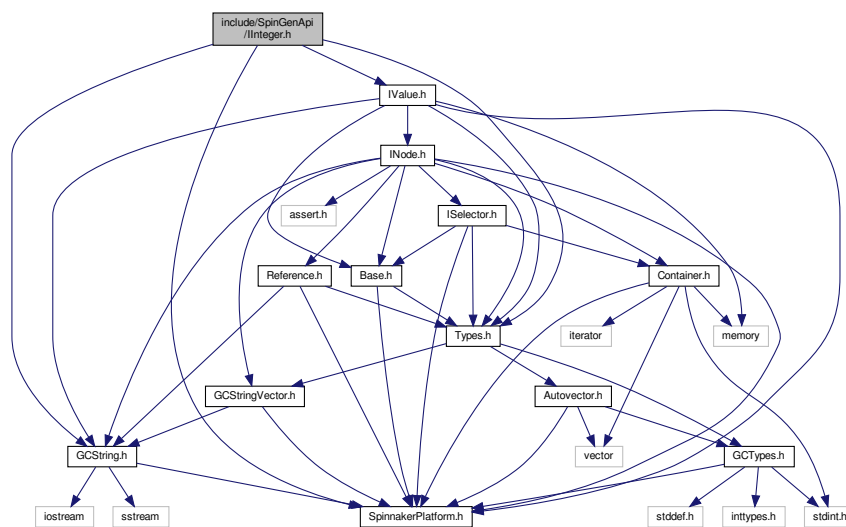
- virtual IFloat & [operator=](#) (double Value)=0
Set node value.
- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const
Get node value.
- virtual GenICam::gcstring [operator*](#) ()=0
Get string node value.
- virtual double [GetMin](#) ()=0
Get minimum value allowed.
- virtual double [GetMax](#) ()=0
Get maximum value allowed.
- virtual bool [HasInc](#) ()=0
True if the float has a constant increment.
- virtual EIncMode [GetIncMode](#) ()=0
Get increment mode.
- virtual double [GetInc](#) ()=0
Get the constant increment if there is any.
- virtual double_autovector_t [GetListOfValidValues](#) (bool bounded=true)=0
Get list of valid value.
- virtual ERepresentation [GetRepresentation](#) ()=0
Get recommended representation.
- virtual GenICam::gcstring [GetUnit](#) () const =0
Get the physical unit name.
- virtual EDisplayNotation [GetDisplayNotation](#) () const =0
Get the way the float should be converted to a string.
- virtual int64_t [GetDisplayPrecision](#) () const =0
Get the precision to be used when converting the float to a string.
- virtual void [ImposeMin](#) (double Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)=0
Restrict maximum value.

Variables

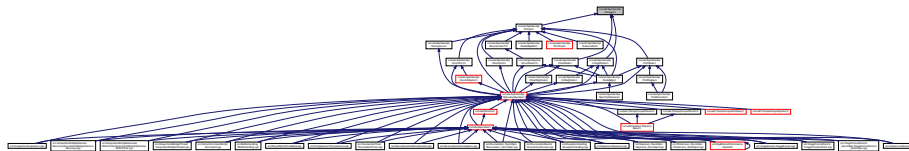
- [interface SPINNAKER_API_ABSTRACT IFloat](#)
Interface for float properties.

16.102 include/SpinGenApi/IInteger.h File Reference

Include dependency graph for IInteger.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

- virtual `IInteger & operator= (int64_t Value)=0`
Set node value.
- virtual `bool GetValue (bool Verify=false, bool IgnoreCache=false) const =0`
Get node value.
- virtual `bool operator() () const`

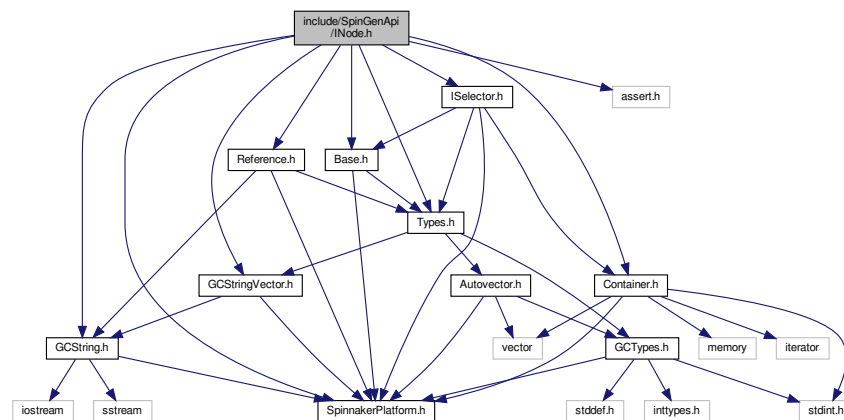
- Get node value.
 • virtual GenICam::gcstring **operator*** ()=0
- Get string node value.
 • virtual double **GetMin** ()=0
- Get minimum value allowed.
 • virtual double **GetMax** ()=0
- Get maximum value allowed.
 • virtual EIncMode **GetIncMode** ()=0
- Get increment mode.
 • virtual double **GetInc** ()=0
- Get the constant increment if there is any.
 • virtual double_autovector_t **GetListOfValidValues** (bool bounded=true)=0
- Get list of valid value.
 • virtual ERepresentation **GetRepresentation** ()=0
- Get recommended representation.
 • virtual GenICam::gcstring **GetUnit** () const =0
- Get the physical unit name.
 • virtual void **ImposeMin** (int64_t Value)=0
- Restrict minimum value.
 • virtual void **ImposeMax** (int64_t Value)=0
- Restrict maximum value.

Variables

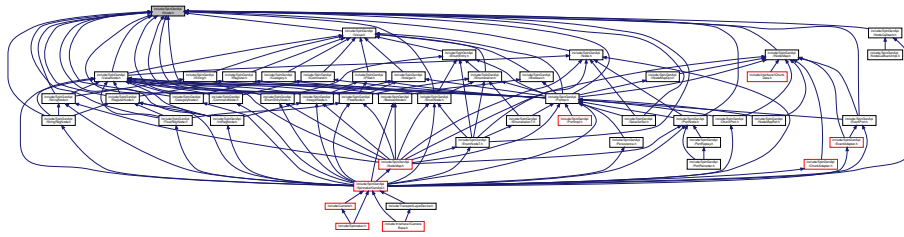
- interface SPINNAKER_API_ABSTRACT Integer
Interface for integer properties.

16.103 include/SpinGenApi/INode.h File Reference

Include dependency graph for INode.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef node_vector [NodeList_t](#)
a list of node references
- typedef intptr_t [CallbackHandleType](#)
the callback handle for nodes

Functions

- virtual GenApi::ENamespace [GetNameSpace](#) () const =0
Get name space.
- virtual EVisibility [GetVisibility](#) () const =0
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()=0
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const =0
Is the node value cacheable.
- virtual EYesNo [IsAccessModeCacheable](#) () const =0
True if the AccessMode can be cached.
- virtual ECachingMode [GetCachingMode](#) () const =0
Get Caching Mode.
- virtual int64_t [GetPollingTime](#) () const =0
recommended polling time (for non-cacheable nodes)
- virtual GenICam::gcstring [GetToolTip](#) ()=0
Get tool tip.
- virtual GenICam::gcstring [GetDescription](#) () const =0
Get a long description of the node.
- virtual GenICam::gcstring [GetDisplayName](#) () const =0
Get a name string for display.
- virtual GenICam::gcstring [GetDeviceName](#) () const =0
Get a name of the device.
- virtual void [GetChildren](#) (GenApi::NodeList_t &Children, ELinkType LinkType=ctReadingChildren) const =0
Get all nodes this node directly depends on.

- virtual void [GetParents](#) (GenApi::NodeList_t &Parents) const =0
Gets all nodes this node is directly depending on.
- virtual CallbackHandleType [RegisterCallback](#) (CNodeCallback *pCallback)=0
Register change callback Takes ownership of the [CNodeCallback](#) object.
- virtual bool [DeregisterCallback](#) (CallbackHandleType hCallback)=0
De register change callback Destroys [CNodeCallback](#) object.
- virtual INodeMap * [GetNodeMap](#) () const =0
Retrieves the central node map.
- virtual GenICam::gcstring [GetEventID](#) () const =0
Get the EventId of the node.
- virtual bool [IsStreamable](#) () const =0
True if the node is streamable.
- virtual void [GetPropertyNames](#) (GenICam::gcstring_vector &PropertyNames) const =0
Returns a list of the names all properties set during initialization.
- virtual bool [GetProperty](#) (const GenICam::gcstring &PropertyName, GenICam::gcstring &ValueStr, GenICam::gcstring &AttributeStr)=0
Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.
- virtual void [ImposeAccessMode](#) (EAccessMode ImposedAccessMode)=0
Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) (EVisibility ImposedVisibility)=0
Imposes a visibility to the natural visibility of the node.
- virtual INode * [GetAlias](#) () const =0
Retrieves the a node which describes the same feature in a different way.
- virtual INode * [GetCastAlias](#) () const =0
Retrieves the a node which describes the same feature so that it can be casted.
- virtual GenICam::gcstring [GetDocuURL](#) () const =0
Gets a URL pointing to the documentation of that feature.
- virtual bool [IsDeprecated](#) () const =0
True if the node should not be used any more.
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const =0
Get the type of the main interface of a node.
- virtual bool [IsFeature](#) () const =0
True if the node can be reached via category nodes from a category node named "Root".
- virtual bool [operator==](#) (int nullPtr) const =0
- virtual bool [operator!=](#) (int nullPtr) const =0
- bool [IsReadable](#) (EAccessMode AccessMode)
Tests if readable.
- bool [IsReadable](#) (const IBase *p)
Checks if a node is readable.
- bool [IsReadable](#) (const IBase &r)
Checks if a node is readable.
- bool [IsWritable](#) (EAccessMode AccessMode)
Tests if writable.
- bool [IsWritable](#) (const IBase *p)
Checks if a node is writable.
- bool [IsWritable](#) (const IBase &r)
Checks if a node is writable.
- bool [IsImplemented](#) (EAccessMode AccessMode)
Tests if implemented.
- bool [IsImplemented](#) (const IBase *p)

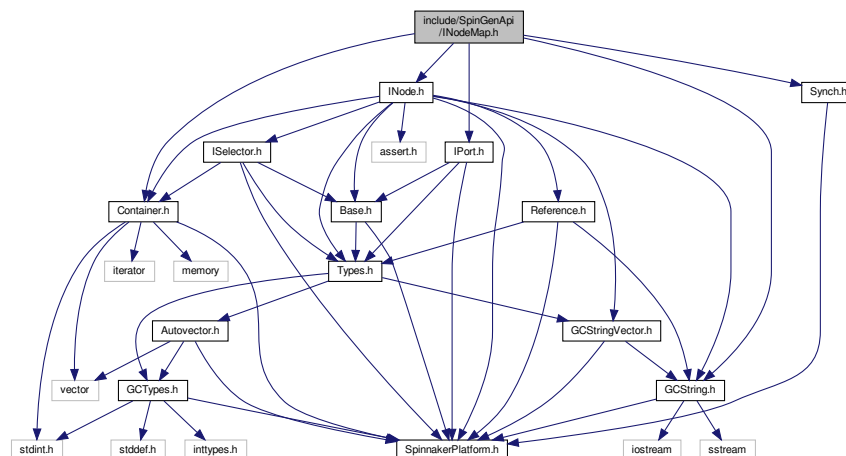
- Checks if a node is implemented.*
- bool [IsImplemented](#) (const IBase &r)
- Checks if a node is implemented.*
- bool [IsAvailable](#) (EAccessMode AccessMode)
- Tests if available.*
- bool [IsAvailable](#) (const IBase *p)
- Checks if a node is available.*
- bool [IsAvailable](#) (const IBase &r)
- Checks if a node is available.*
- EAccessMode [Combine](#) (EAccessMode Peter, EAccessMode Paul)
- Computes which access mode the two guards allow together.*
- bool [IsVisible](#) (EVisibility Visibility, EVisibility MaxVisibility)
- Tests Visibility CAVE : this relies on the EVisibility enum's coding.*
- EVisibility [Combine](#) (EVisibility Peter, EVisibility Paul)
- Computes which visibility the two guards allow together.*
- bool [IsCacheable](#) (ECachingMode CachingMode)
- Tests Cacheability.*
- ECachingMode [Combine](#) (ECachingMode Peter, ECachingMode Paul)
- Computes which CachingMode results from a combination.*

Variables

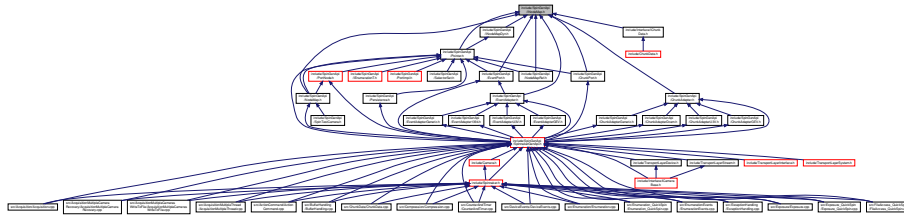
- [interface SPINNAKER_API_ABSTRACT INode](#)
- Interface common to all nodes.*
- [interface SPINNAKER_API_ABSTRACT](#) virtual public [IReference](#)
- Interface to construct a reference.*

16.104 include/SpinGenApi/INodeMap.h File Reference

Include dependency graph for INodeMap.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

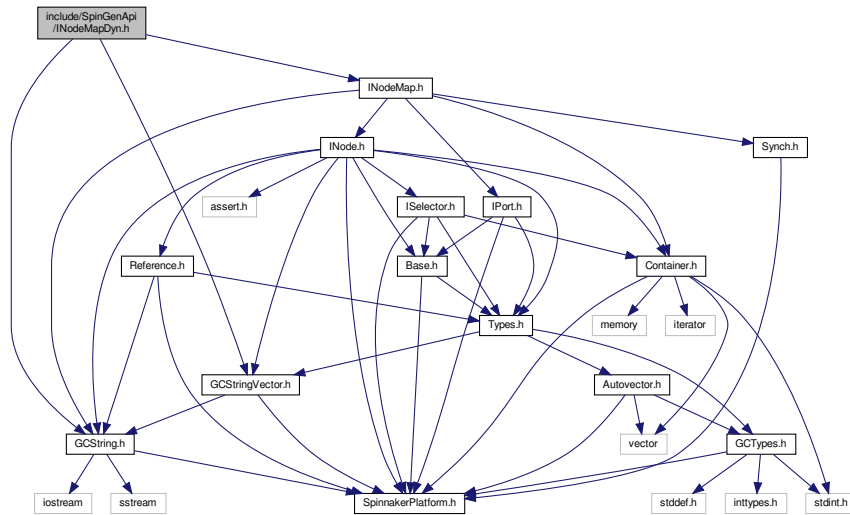
- virtual `INode *` [GetNode](#) (`const GenICam::gcstring &Name`) `const =0`
Retrieves the node from the central map by Name.
- virtual `void` [InvalidateNodes](#) () `const =0`
Invalidates all nodes.
- virtual `bool` [Connect](#) (`IPort *pPort`, `const GenICam::gcstring &PortName`) `const =0`
Connects a port to a port node with given name.
- virtual `bool` [Connect](#) (`IPort *pPort`) `const =0`
Connects a port to the standard port "Device".
- virtual `GenICam::gcstring` [GetDeviceName](#) () `const =0`
Get a name of the device.
- virtual `void` [Poll](#) (`int64_t ElapsedTime`)`=0`
Fires nodes which have a polling time.
- virtual `CLock &` [GetLock](#) () `const =0`
Returns the lock which guards the node map.
- virtual `uint64_t` [GetNumNodes](#) () `const =0`
Get the number of nodes in the map.

Variables

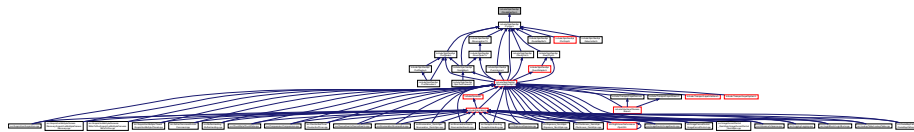
- [interface SPINNAKER_API_ABSTRACT INodeMap](#)
Interface to access the node map.

16.105 include/SpinGenApi/INodeMapDyn.h File Reference

Include dependency graph for INodeMapDyn.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

- virtual void [LoadXMLFromFile](#) (const GenICam::gcstring &FileName)=0
Loads an XML from a file.
- virtual void [LoadXMLFromFileInject](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)=0
Loads an XML from a file with injection.
- virtual void [LoadXMLFromString](#) (const GenICam::gcstring &XMLData)=0
Loads an XML from a string.
- virtual void [LoadXMLFromStringInject](#) (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)=0
Loads an XML from a string with injection.
- virtual void [PreprocessXMLFromFile](#) (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xv↔ Default)=0

Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

- virtual void [MergeXMLFiles](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectedFileName, const GenICam::gcstring &OutputFileName)=0

Injects an XML file into a target file.

- virtual void [ExtractIndependentSubtree](#) (const GenICam::gcstring &XMLData, const GenICam::gcstring &InjectXMLData, const GenICam::gcstring &SubTreeRootNodeName, GenICam::gcstring &ExtractedSubtree)=0

Extract independent subtree.

- virtual void [GetSupportedSchemaVersions](#) (GenICam::gcstring_vector &SchemaVersions)=0

Gets a list of supported schema versions.

- virtual void [LoadXMLFromZIPFile](#) (const GenICam::gcstring &ZipFileName)=0

Loads an XML from a ZIP file.

- virtual void [LoadXMLFromZIPData](#) (const void *zipData, size_t zipSize)=0

Loads an XML from a ZIP data buffer.

- virtual void [PreprocessXMLFromZIPFile](#) (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xvDefault)=0

Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

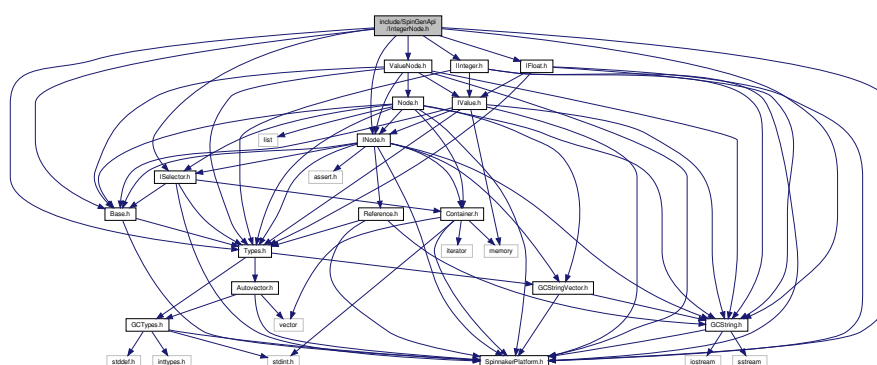
Variables

- [interface SPINNAKER_API_ABSTRACT INodeMapDyn](#)

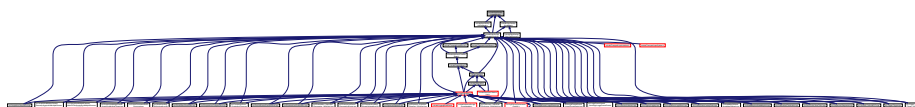
Interface to access the node map.

16.106 include/SpinGenApi/IntegerNode.h File Reference

Include dependency graph for IntegerNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IntegerNode](#)
Interface for string properties.

Namespaces

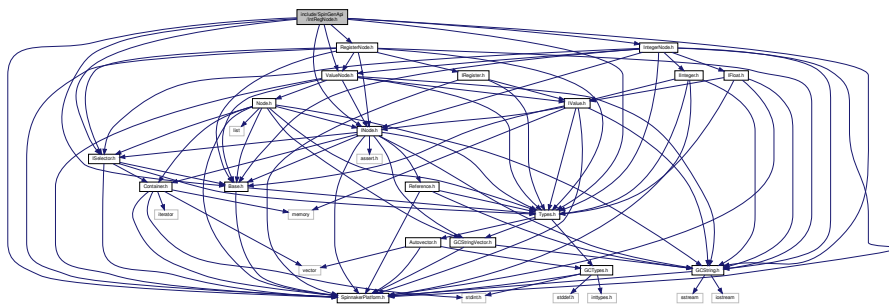
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

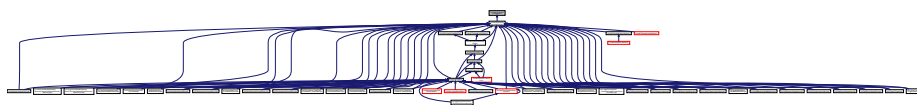
- typedef IntegerNode [CIntegerRef](#)

16.107 include/SpinGenApi/IntRegNode.h File Reference

Include dependency graph for IntRegNode.h:



This graph shows which files directly or indirectly include this file:



Classes

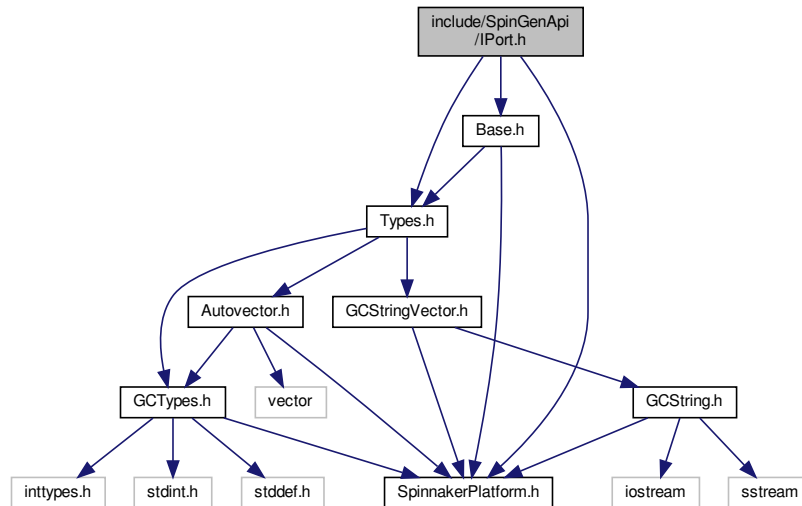
- class [IntRegNode](#)
Interface for string properties.

Namespaces

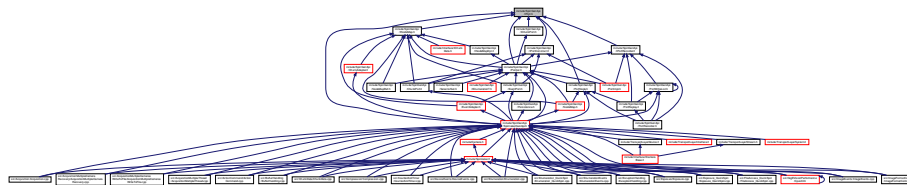
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.108 include/SpinGenApi/IPort.h File Reference

Include dependency graph for IPort.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

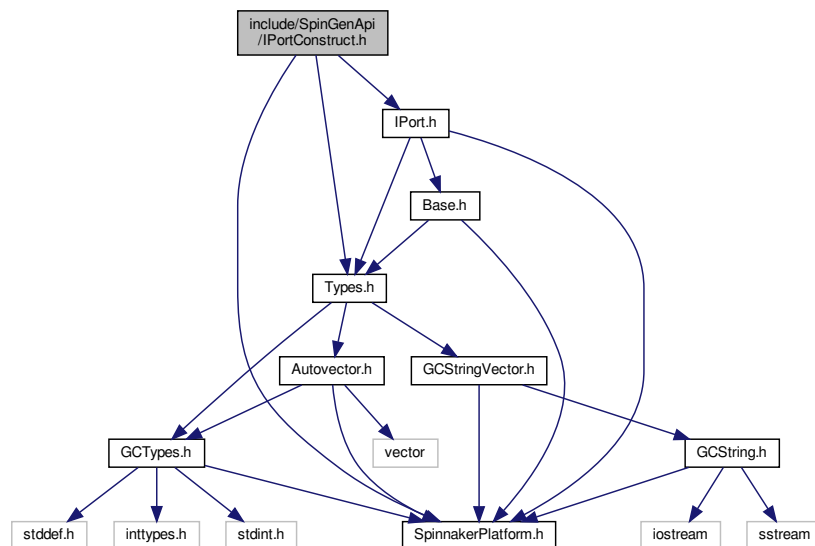
- virtual void [Write](#) (const void *pBuffer, int64_t Address, int64_t Length)=0
Writes a chunk of bytes to the port.

Variables

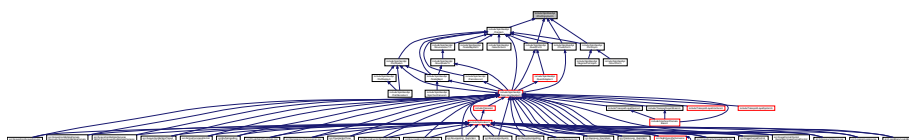
- [interface SPINNAKER_API_ABSTRACT IPort](#)
Interface for ports.
- [interface SPINNAKER_API_ABSTRACT int64_t Address](#)
- [interface SPINNAKER_API_ABSTRACT int64_t int64_t Length = 0](#)

16.109 include/SpinGenApi/IPortConstruct.h File Reference

Include dependency graph for IPortConstruct.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

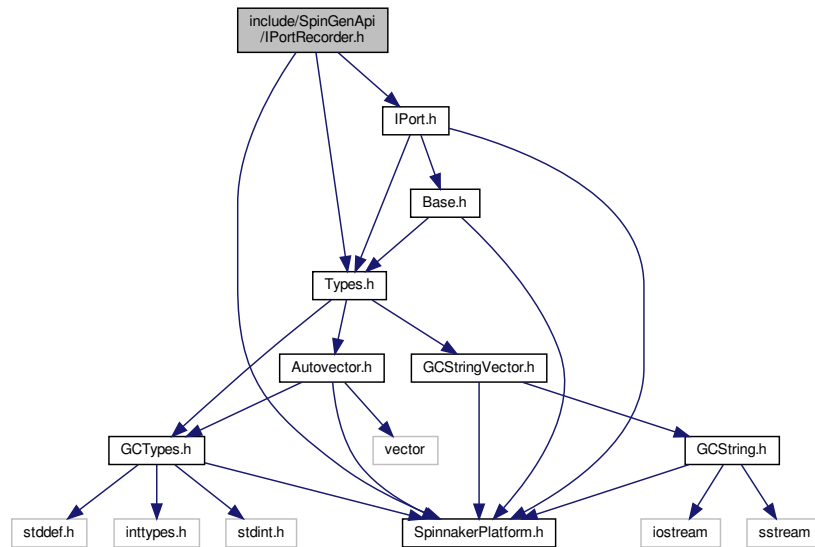
- virtual EYesNo [GetSwapEndianness](#) ()=0
Determines if the port adapter must perform an endianness swap.

Variables

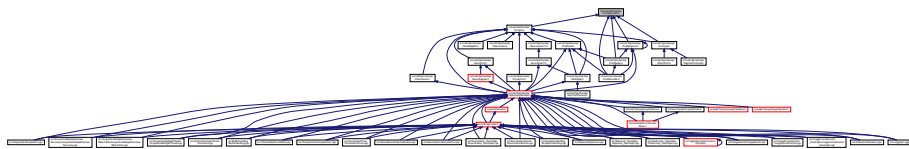
- [interface SPINNAKER_API IPortConstruct](#)
Interface for ports.

16.110 include/SpinGenApi/IPortRecorder.h File Reference

Include dependency graph for IPortRecorder.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

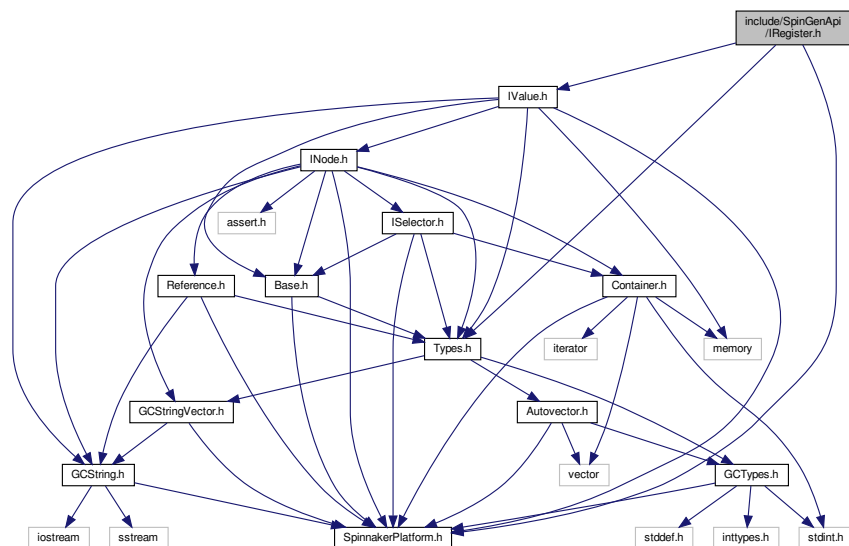
- virtual void [Replay](#) (IPort *pPort)=0
Replays the write command to the given port interface.
- virtual void [SetCookie](#) (const int64_t Value)=0
Sets a cookie in case the port implementation want to cache a command list.
- virtual int64_t [GetCookie](#) ()=0
Gets the cookie a port implementation may have set for caching a command list.
- virtual void [StopRecording](#) ()=0
Stops recording.

Variables

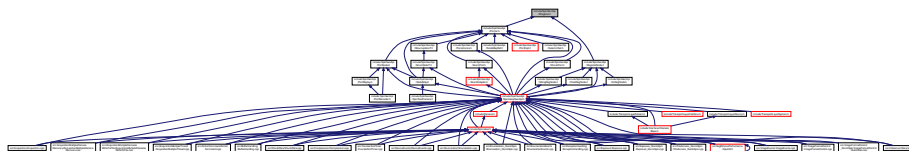
- [interface SPINNAKER_API_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER_API_ABSTRACT IPortReplay](#)
Interface for replaying write commands on a port.
- [interface SPINNAKER_API_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER_API_ABSTRACT IPortRecorder](#)
Interface for recording write commands on a port.

16.111 include/SpinGenApi/IRegister.h File Reference

Include dependency graph for IRegister.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

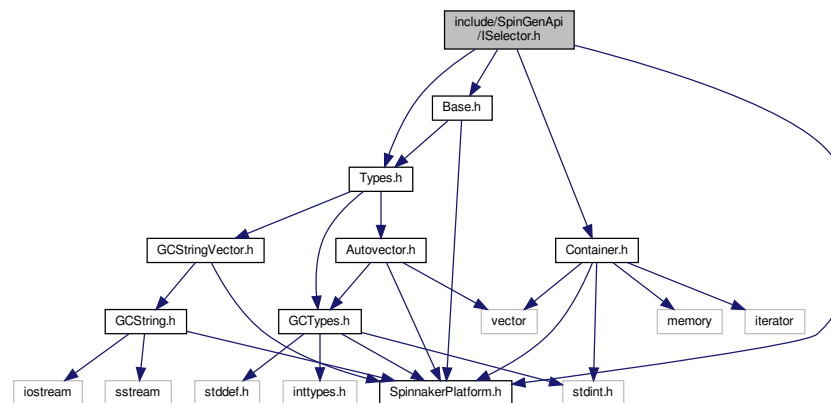
- virtual void [Get](#) (uint8_t *pBuffer, int64_t Length, bool Verify=false, bool IgnoreCache=false)=0
Fills a buffer with the register's contents.
- virtual int64_t [GetLength](#) ()=0
Retrieves the Length of the register [Bytes].
- virtual int64_t [GetAddress](#) ()=0
Retrieves the Address of the register.

Variables

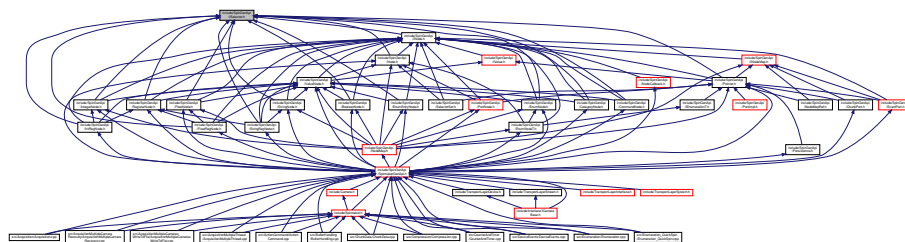
- [interface SPINNAKER_API_ABSTRACT IRegister](#)
Interface for registers.

16.112 include/SpinGenApi/ISelector.h File Reference

Include dependency graph for ISelector.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

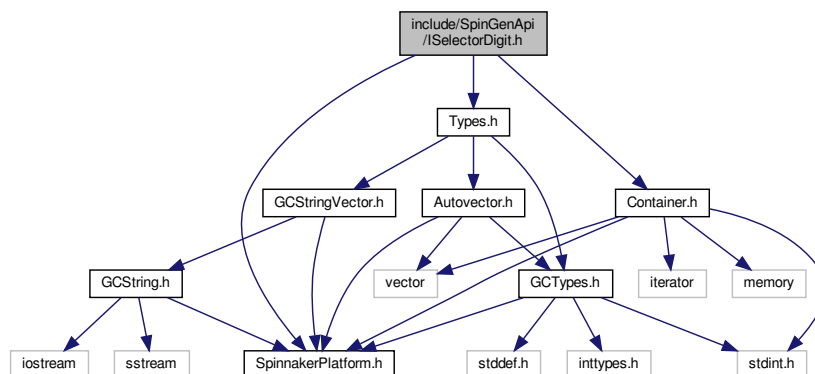
- virtual void [GetSelectedFeatures](#) (FeatureList_t &) const =0
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) (FeatureList_t &) const =0
retrieve the group of features selecting this node

Variables

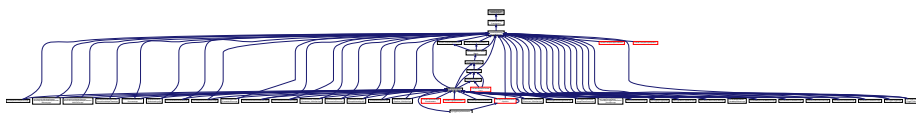
- [interface SPINNAKER_API_ABSTRACT ISelector](#)
Interface for groups of features selected by a single one.

16.113 include/SpinGenApi/ISelectorDigit.h File Reference

Include dependency graph for ISelectorDigit.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

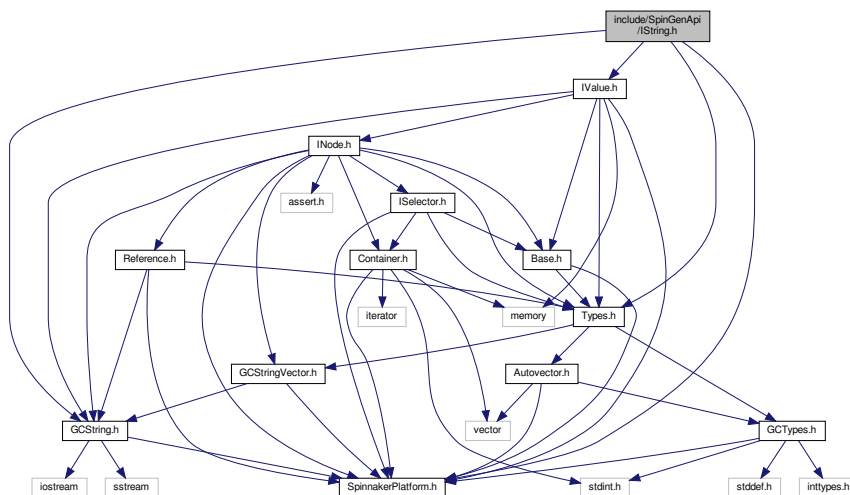
- virtual bool [SetNext](#) (bool Tick=true)=0
Sets digit to next value.
- virtual void [Restore](#) ()=0
Restores the selectors' values found at creation.
- virtual GenICam::gcstring [ToString](#) ()=0
Returns a string representation of the digit.
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)=0
Retrieves an ordered list of selectors.

Variables

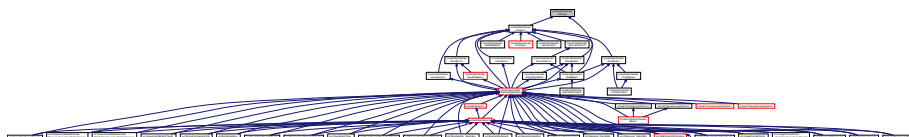
- [interface SPINNAKER_API_ABSTRACT ISelectorDigit](#)
Interface of a "digit" of the "counter" formed by the selector set.

16.114 include/SpinGenApi/IString.h File Reference

Include dependency graph for IString.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

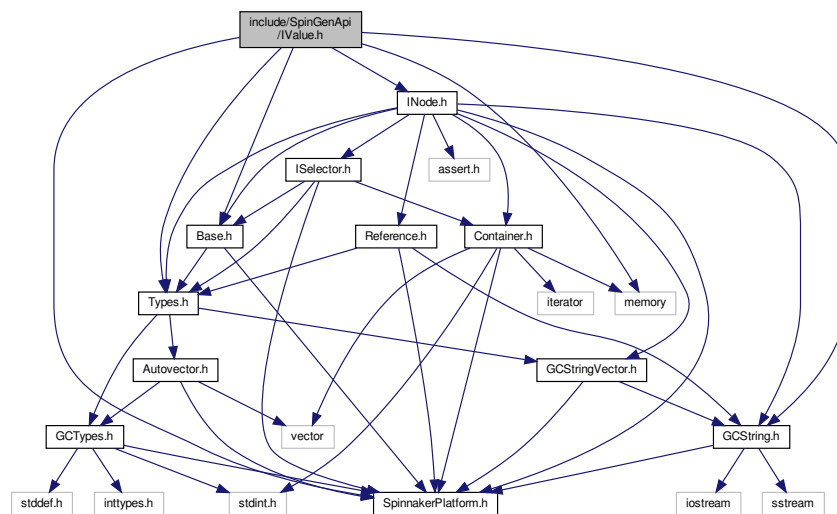
- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0
Set string node value.
- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const
Get node value.
- virtual GenICam::gcstring [operator*](#) ()=0
Get string node value.
- virtual int64_t [GetMaxLength](#) ()=0
Retrieves the maximum length of the string in bytes.

Variables

- [interface SPINNAKER_API_ABSTRACT IString](#)
Interface for string properties.

16.115 include/SpinGenApi/IValue.h File Reference

Include dependency graph for IValue.h:

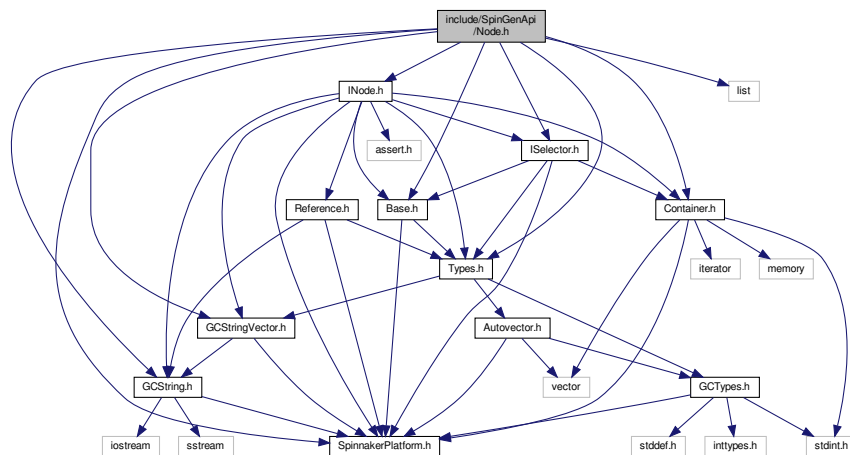


- Spinnaker
- Spinnaker::GenApi

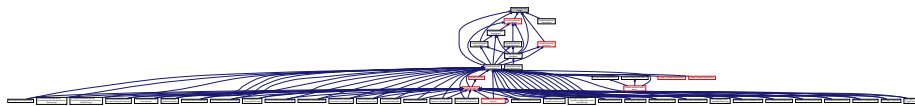
- virtual GenlCam::gcstring **ToString** (bool Verify=false, bool IgnoreCache=false)=0
Get content of the node as string.
- virtual void **FromString** (const GenlCam::gcstring &ValueStr, bool Verify=true)=0
Set content of the node as string.
- virtual bool **IsValueCacheValid** () const =0
Checks if the value comes from cache or is requested from another node.

- interface SPINNAKER_API_ABSTRACT IValue
Interface for value properties.

Include dependency graph for Node.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Node](#)
class common to all nodes

Namespaces

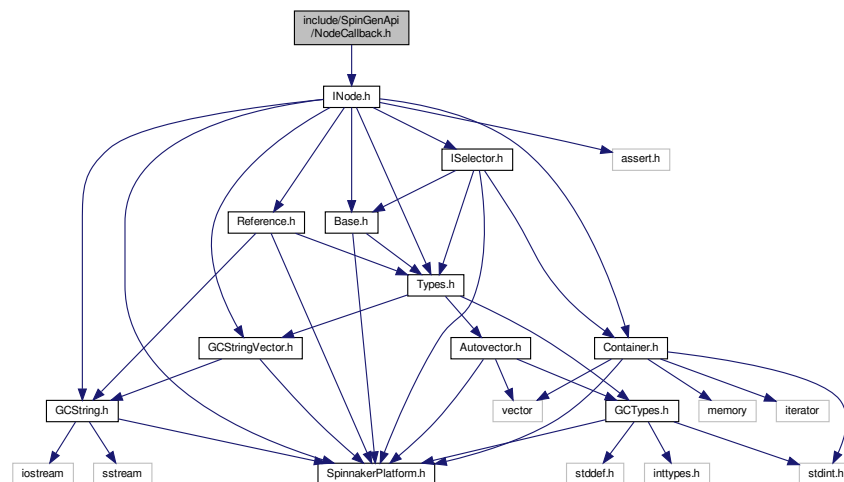
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

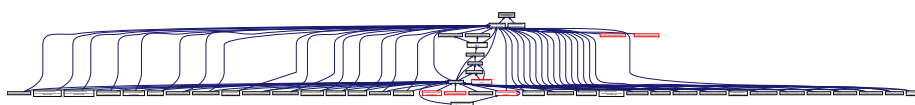
- typedef Node [CNodeRef](#)
- typedef Node [CSelectorRef](#)

16.117 include/SpinGenApi/NodeCallback.h File Reference

Include dependency graph for NodeCallback.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CNodeCallback](#)
callback body instance for INode pointers
- class [Function_NodeCallback](#)< [Function](#) >
Container for a function pointer.
- class [Member_NodeCallback](#)< [Client](#), [Member](#) >
Container for a member function pointer.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Enumerations

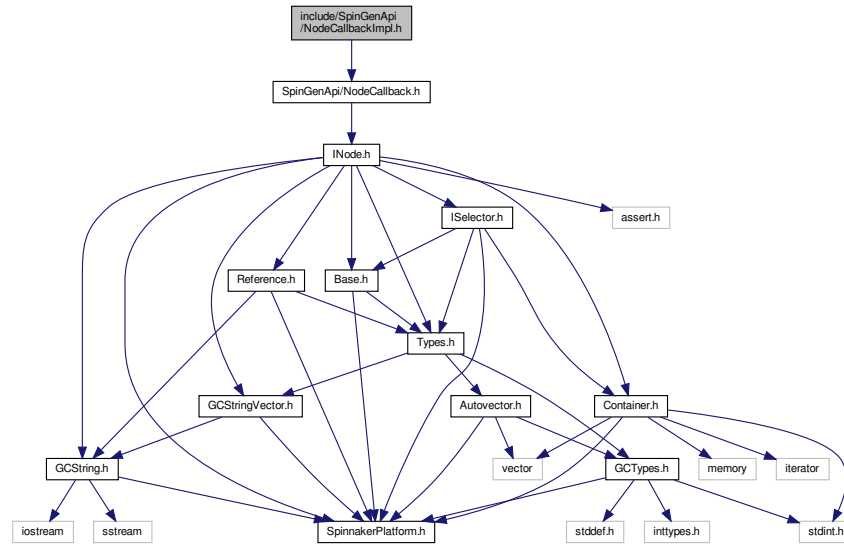
- enum [ECallbackType](#) {
 [cbPostInsideLock](#) = 1,
 [cbPostOutsideLock](#) = 2 }
the type of callback

Functions

- template<class [Function](#) >
 [CNodeCallback](#) * [make_NodeCallback](#) ([INode](#) *pNode, [Function](#) function, [ECallbackType](#) CallbackType)
 make a new callback object for C functions
- template<class [Function](#) >
 intptr_t [Register](#) ([INode](#) *pNode, [Function](#) f, [ECallbackType](#) CallbackType=cbPostInsideLock)
 Register a C-function as a callback.
- template<class [Client](#) , class [Member](#) >
 [CNodeCallback](#) * [make_NodeCallback](#) ([INode](#) *pNode, [Client](#) &client, [Member](#) member, [ECallbackType](#) CallbackType)
 make a new callback object for member functions
- template<class [Client](#) , class [Member](#) >
 intptr_t [Register](#) ([INode](#) *pNode, [Client](#) &c, [Member](#) m, [ECallbackType](#) CallbackType=cbPostInsideLock)
 Register a C++-member function a callback.
- [SPINNAKER_API](#) void [Deregister](#) ([GenApi::CallbackHandleType](#) pCallbackInfo)
 Unregistering callback by handle.

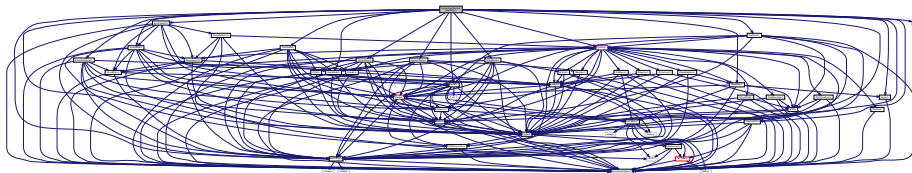
16.118 include/SpinGenApi/NodeCallbackImpl.h File Reference

Include dependency graph for NodeCallbackImpl.h:

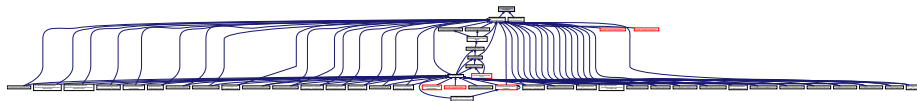


16.119 include/SpinGenApi/NodeMap.h File Reference

Include dependency graph for NodeMap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [NodeMap](#)

Smart pointer template for NodeMaps with create function.

Namespaces

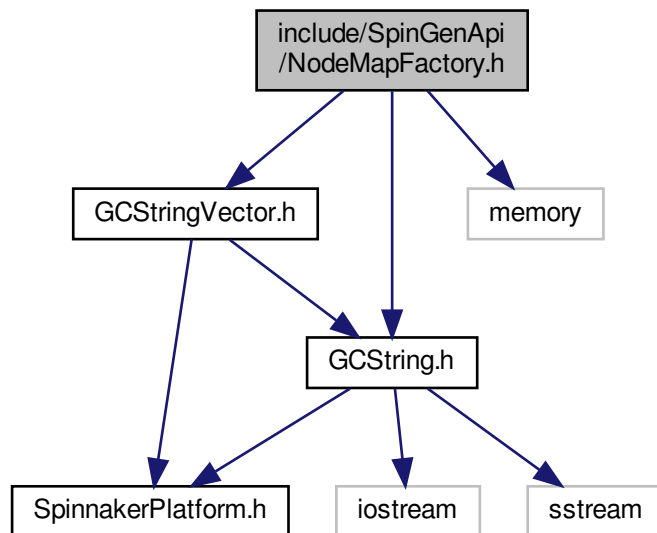
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

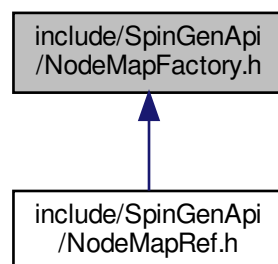
- typedef NodeMap [CNodeMapRef](#)

16.120 include/SpinGenApi/NodeMapFactory.h File Reference

Include dependency graph for NodeMapFactory.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CNodeMapFactory](#)
The node map factory is used for creating node maps from camera description files.
- struct [CNodeMapFactory::NodeStatistics_t](#)

Namespaces

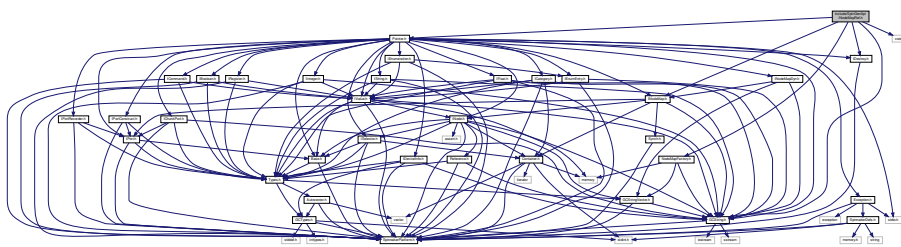
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Enumerations

- enum [ECacheUsage_t](#) {
 [CacheUsage_Automatic](#),
 [CacheUsage_ForceWrite](#),
 [CacheUsage_ForceRead](#),
 [CacheUsage_Ignore](#) }
Lists the cache usage strategies.
- enum [EContentType_t](#) {
 [ContentType_Xml](#),
 [ContentType_ZippedXml](#) }
Lists the processable file types.

16.121 include/SpinGenApi/NodeMapRef.h File Reference

Include dependency graph for NodeMapRef.h:



Classes

- class [CNodeMapRefT< TCameraParams >](#)
Smartpointer template for NodeMaps with create function.
- class [CGeneric_XMLLoaderParams](#)
Empty base class used by class [CNodeMapRef](#) as generic template argument.
- class [CNodeMapRef](#)
Smartpointer for NodeMaps with create function.

Namespaces

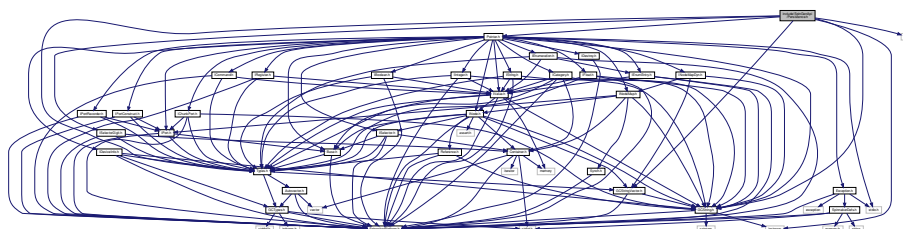
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

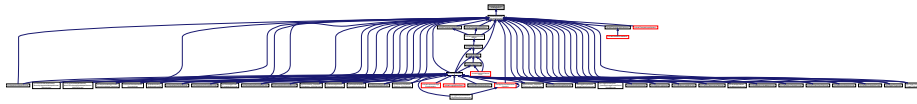
- [SPINNAKER_API](#) IDestroy * [CastToIDestroy](#) (INodeMap *pNodeMap)
makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)
- `template<class TCameraParams >`
`void \_LoadXMLFromFile (const GenICam::gcstring &FileName)`
- `template<class TCameraParams >`
`void \_LoadXMLFromZIPFile (const GenICam::gcstring &ZipFileName)`
- `template<class TCameraParams >`
`void \_LoadXMLFromFileInject (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)`
- `template<class TCameraParams >`
`void \_LoadXMLFromString (const GenICam::gcstring &XMLData)`
- `template<class TCameraParams >`
`void \_LoadXMLFromZIPData (const void *zipData, size_t zipSize)`
- `template<class TCameraParams >`
`void \_LoadXMLFromStringInject (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)`
- `template<class TCameraParams >`
`void \_GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions)`
- `template<class TCameraParams >`
`GenICam::gcstring \_GetDeviceName ()`
- `template<class TCameraParams >`
`void \_Poll (int64_t ElapsedTime)`
- `template<class TCameraParams >`
`void \_GetNodes (NodeList_t &Nodes)`
- `template<class TCameraParams >`
`INode * \_GetNode (const GenICam::gcstring &key)`
- `template<class TCameraParams >`
`void \_InvalidateNodes ()`
- `template<class TCameraParams >`
`bool \_Connect (IPort *pPort, const GenICam::gcstring &PortName)`
- `template<class TCameraParams >`
`bool \_Connect (IPort *pPort)`
- `template<class TCameraParams >`
`bool \_ClearXMLCache ()`

16.122 include/SpinGenApi/Persistence.h File Reference

Include dependency graph for Persistence.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CFeatureBag](#)
Bag holding streamable features of a nodetree.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

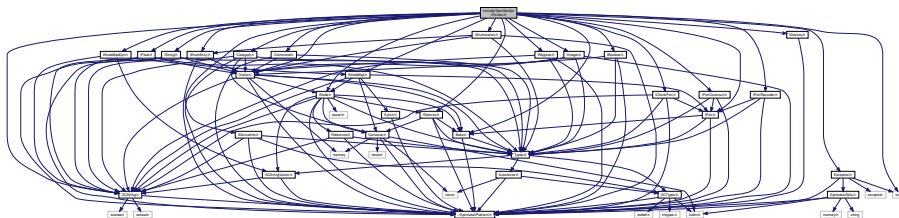
- virtual void [PersistFeature](#) (IValue &item)=0
Stores a feature.
- [SPINNAKER_API](#) std::istream & [EatComments](#) (std::istream &is)
*Helper function ignoring lines starting with comment character '#'.
Reads in persistent data from a stream.*
- [SPINNAKER_API](#) std::istream & [operator>>](#) (std::istream &is, CFeatureBag &FeatureBag)
Reads in persistent data from a stream.
- [SPINNAKER_API](#) std::ostream & [operator<<](#) (std::ostream &os, const CFeatureBag &FeatureBag)
writes out persistent data to a stream

Variables

- [interface SPINNAKER_API_ABSTRACT IPersistScript](#)
Basic interface to persist values to.

16.123 include/SpinGenApi/Pointer.h File Reference

Include dependency graph for Pointer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CPointer< T, B >](#)
Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.
- class [CFloatPtr](#)
SmartPointer for IFloat interface pointer.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef [CPointer< IBase >](#) [CBasePtr](#)
SmartPointer for IBase interface pointer.
- typedef [CPointer< INode, IBase >](#) [CNodePtr](#)
SmartPointer for INode interface pointer.
- typedef [CPointer< IValue >](#) [CValuePtr](#)
SmartPointer for IValue interface pointer.
- typedef [CPointer< ICategory >](#) [CCategoryPtr](#)
SmartPointer for ICategory interface pointer.
- typedef [CPointer< IBoolean >](#) [CBooleanPtr](#)
SmartPointer for IBoolean interface pointer.
- typedef [CPointer< IInteger >](#) [CIntegerPtr](#)
SmartPointer for IInteger interface pointer.
- typedef [CPointer< IString >](#) [CStringPtr](#)
SmartPointer for IString interface pointer.
- typedef [CPointer< IRegister >](#) [CRegisterPtr](#)
SmartPointer for IRegister interface pointer.
- typedef [CPointer< IEnumeration >](#) [CEnumerationPtr](#)
SmartPointer for IEnumeration interface pointer.
- typedef [CPointer< IEnumEntry >](#) [CEnumEntryPtr](#)
SmartPointer for IEnumEntry interface pointer.
- typedef [CPointer< IPort >](#) [CPortPtr](#)
SmartPointer for IPort interface pointer.
- typedef [CPointer< IPortReplay >](#) [CPortReplayPtr](#)
SmartPointer for IPortReplay interface pointer.
- typedef [CPointer< IPortRecorder >](#) [CPortRecorderPtr](#)
SmartPointer for IPortRecorder interface pointer.
- typedef [CPointer< IPortWriteList, IPortWriteList >](#) [CPortWriteListPtr](#)
SmartPointer for IPortWriteList interface pointer.
- typedef [CPointer< IChunkPort >](#) [CChunkPortPtr](#)
SmartPointer for IChunkPort interface pointer.
- typedef [CPointer< INodeMap, INodeMap >](#) [CNodeMapPtr](#)
SmartPointer for INodeMap interface pointer.
- typedef [CPointer< INodeMapDyn, INodeMap >](#) [CNodeMapDynPtr](#)
SmartPointer for INodeMapDyn interface pointer.
- typedef [CPointer< IDeviceInfo, INodeMap >](#) [CDeviceInfoPtr](#)
SmartPointer for IDeviceInfo interface pointer.

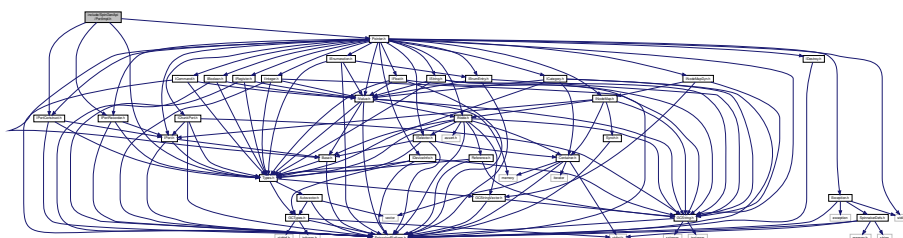
- typedef CPointer< ISelector > [CSelectorPtr](#)
SmartPointer for ISelector interface pointer.
- typedef CPointer< ICommand > [CCommandPtr](#)
SmartPointer for ICommand interface pointer.
- typedef CPointer< IPortConstruct > [CPortConstructPtr](#)
SmartPointer for IPortConstruct interface pointer.

Functions

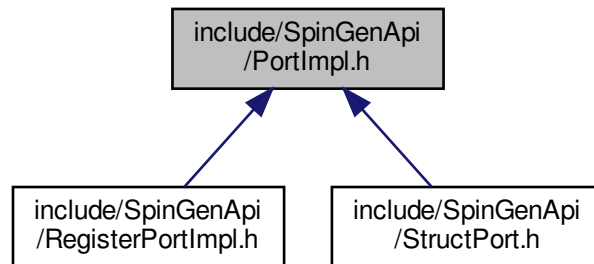
- template<class T , class B >
bool [IsReadable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is readable.
- template<class T , class B >
bool [IsWritable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Writable.
- template<class T , class B >
bool [IsImplemented](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Implemented.
- template<class T , class B >
bool [IsAvailable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Available.
- GenICam::gcstring [GetInterfaceName](#) (IBase *pBase)
Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

16.124 include/SpinGenApi/PortImpl.h File Reference

Include dependency graph for PortImpl.h:



This graph shows which files directly or indirectly include this file:



Classes

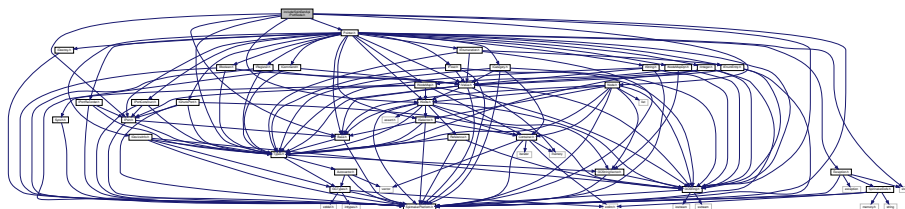
- class [CPortImpl](#)
Standard implementation for a port.

Namespaces

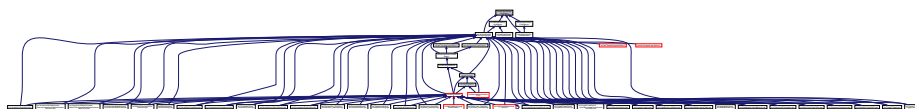
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.125 include/SpinGenApi/PortNode.h File Reference

Include dependency graph for PortNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [PortNode](#)
Interface for value properties.

Namespaces

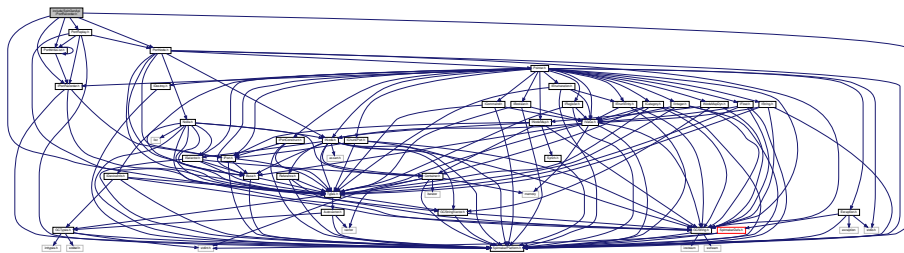
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef PortNode [CPortRef](#)

16.126 include/SpinGenApi/PortRecorder.h File Reference

Include dependency graph for PortRecorder.h:



Classes

- class [PortRecorder](#)
Interface for recording write commands on a port.

Namespaces

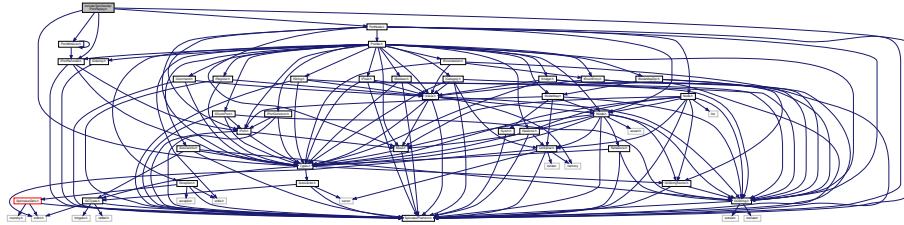
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

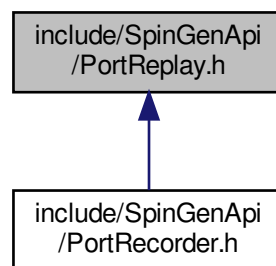
- typedef PortRecorder [CPortRecorderRef](#)
Reference to an IPortRecorder pointer.

16.127 include/SpinGenApi/PortReplay.h File Reference

Include dependency graph for PortReplay.h:



This graph shows which files directly or indirectly include this file:



Classes

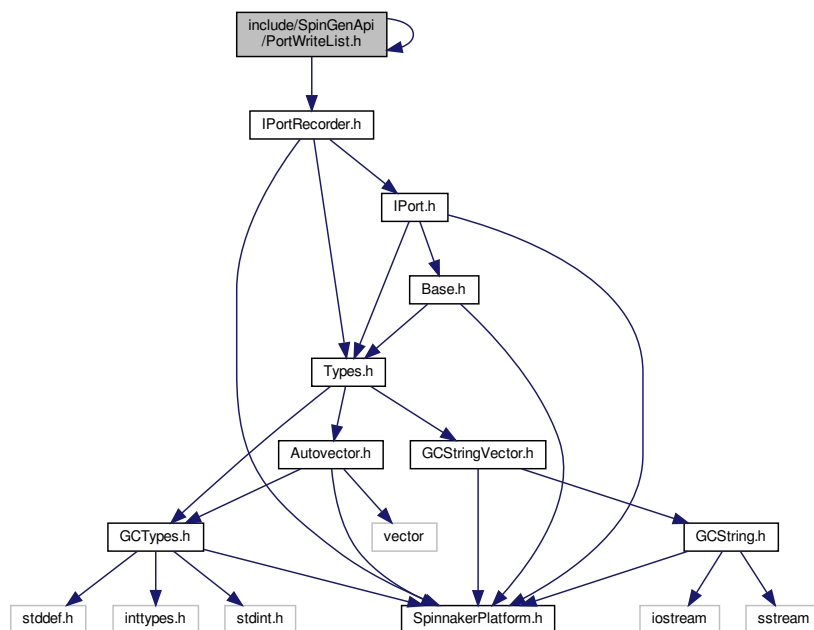
- class [PortReplay](#)
Interface for replaying write commands on a port.

Namespaces

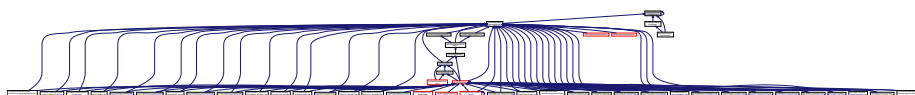
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.128 include/SpinGenApi/PortWriteList.h File Reference

Include dependency graph for PortWriteList.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CPortWriteList](#)

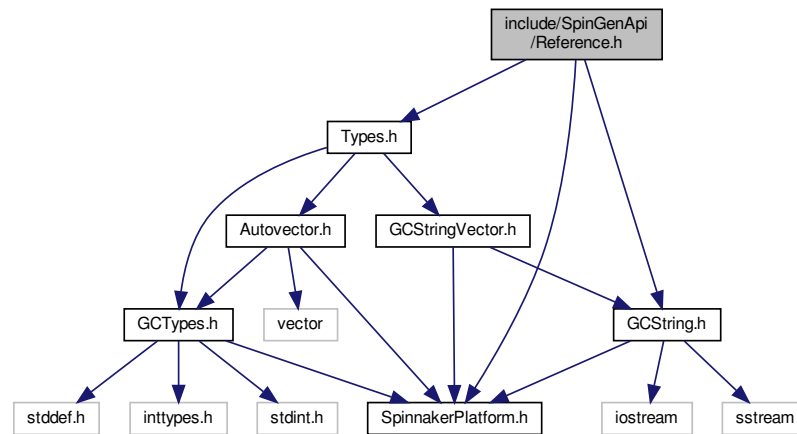
Container holding a list of port write commands.

Namespaces

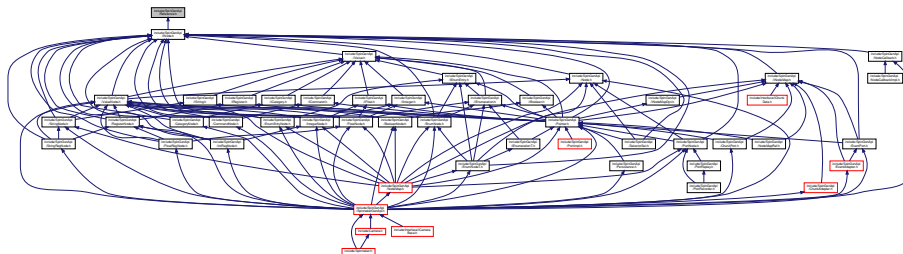
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.129 include/SpinGenApi/Reference.h File Reference

Include dependency graph for Reference.h:



This graph shows which files directly or indirectly include this file:



Namespaces

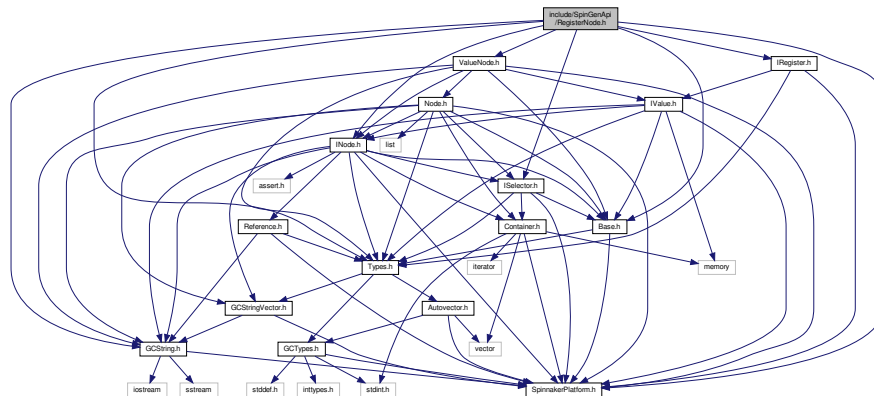
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

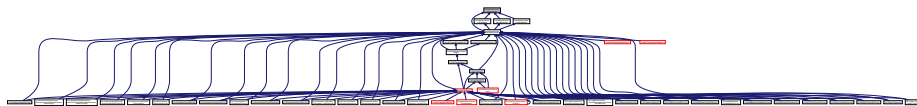
- virtual void [SetNumEnums](#) (int NumEnums)=0
sets the number of enum values

16.130 include/SpinGenApi/RegisterNode.h File Reference

Include dependency graph for RegisterNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [RegisterNode](#)
Interface for string properties.

Namespaces

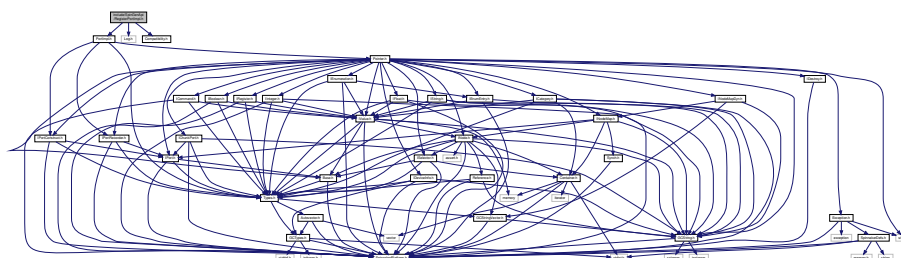
- [Spinaker](#)
- [Spinaker::GenApi](#)

Typedefs

- typedef RegisterNode [CRegisterRef](#)

16.131 include/SpinGenApi/RegisterPortImpl.h File Reference

Include dependency graph for RegisterPortImpl.h:



Classes

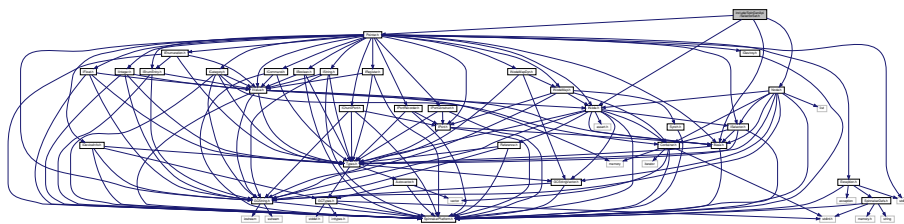
- class [CRegisterPortImpl](#)
Standard implementation for a port using a register based transport layer.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.132 include/SpinGenApi/SelectorSet.h File Reference

Include dependency graph for SelectorSet.h:



Classes

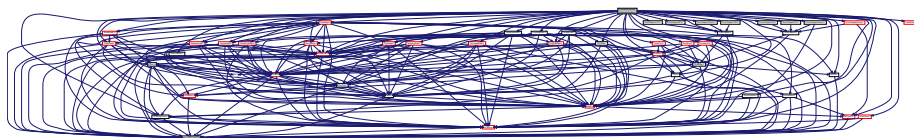
- class [CSelectorSet](#)
The set of selectors selecting a given node.

Namespaces

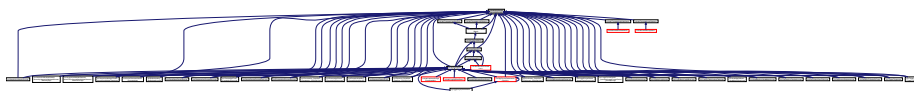
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.133 include/SpinGenApi/SpinnakerGenApi.h File Reference

Include dependency graph for SpinnakerGenApi.h:

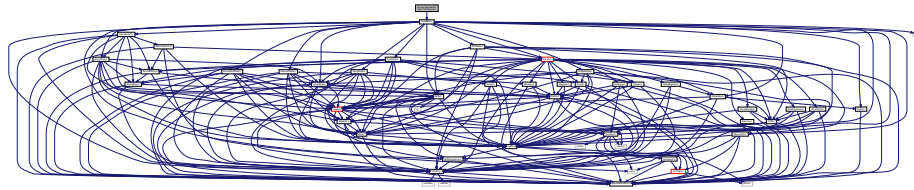


This graph shows which files directly or indirectly include this file:



16.134 include/SpinGenApi/SpinTestCamera.h File Reference

Include dependency graph for SpinTestCamera.h:



Classes

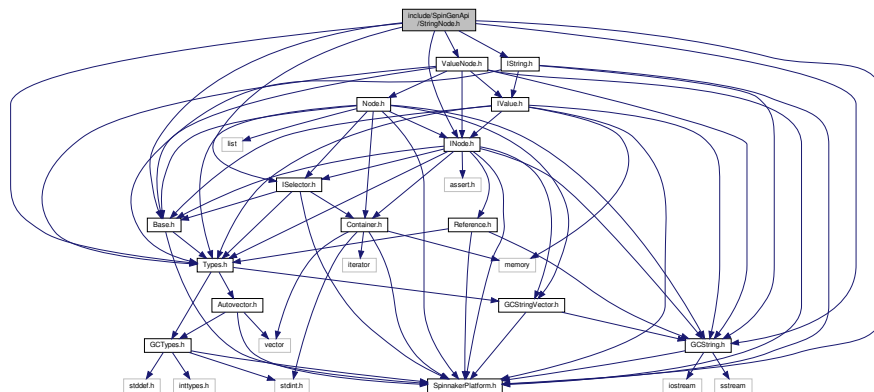
- class [SpinTestCamera](#)

Namespaces

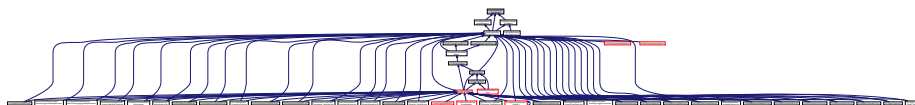
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.135 include/SpinGenApi/StringNode.h File Reference

Include dependency graph for StringNode.h:

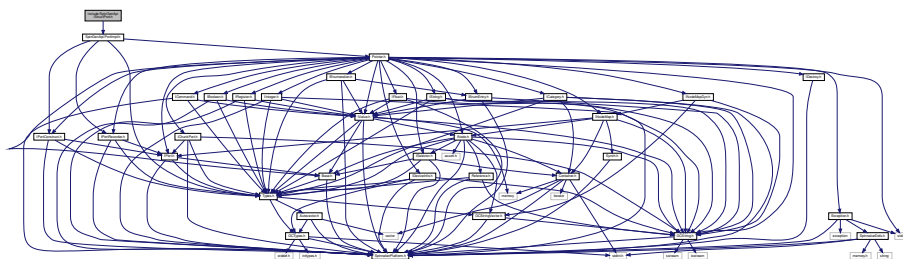


This graph shows which files directly or indirectly include this file:



16.137 include/SpinGenApi/StructPort.h File Reference

Include dependency graph for StructPort.h:



Classes

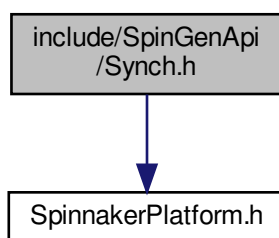
- class `CTestPortStruct< CDataStruct >`
Implements a register spaces based on a C++ struct.

Namespaces

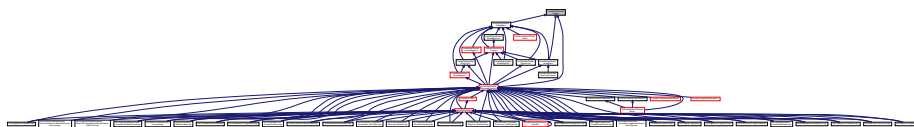
- `Spinnaker`
- `Spinnaker::GenApi`

16.138 include/SpinGenApi/Synch.h File Reference

Include dependency graph for Synch.h:



This graph shows which files directly or indirectly include this file:



Classes

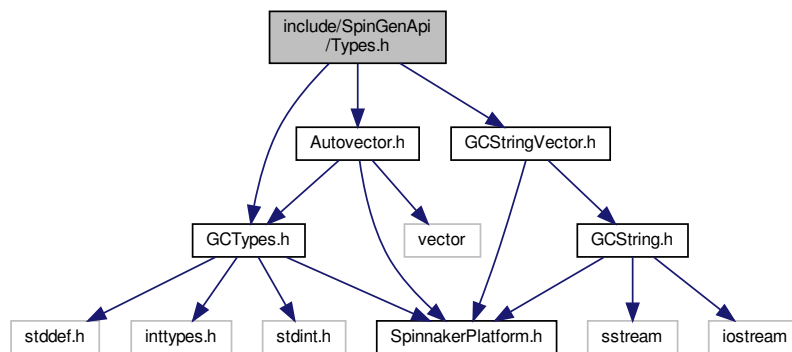
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)

Namespaces

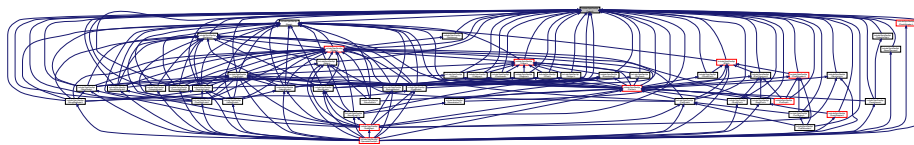
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.139 include/SpinGenApi/Types.h File Reference

Include dependency graph for Types.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Macros

- `#define` [interface](#) struct
- `#define` [_UndefinedRepresentation](#) _UndefinedRepresentation

Typedefs

- typedef GenICam::gcstring_vector [StringList_t](#)
A list of strings.

Enumerations

- enum [ESign](#) {
 [Signed](#),
 [Unsigned](#),
 [_UndefinedSign](#) }
signed or unsigned integers
- enum [EAccessMode](#) {
 [NI](#),
 [NA](#),
 [WO](#),
 [RO](#),
 [RW](#),
 [_UndefinedAccesMode](#),
 [_CycleDetectAccesMode](#) }
access mode of a node
- enum [EVisibility](#) {
 [Beginner](#) = 0,
 [Expert](#) = 1,
 [Guru](#) = 2,
 [Invisible](#) = 3,
 [_UndefinedVisibility](#) = 99 }
recommended visibility of a node
- enum [ECachingMode](#) {
 [NoCache](#),
 [WriteThrough](#),
 [WriteAround](#),
 [_UndefinedCachingMode](#) }
caching mode of a register
- enum [ERepresentation](#) {
 [Linear](#),
 [Logarithmic](#),
 [Boolean](#),
 [PureNumber](#),
 [HexNumber](#),
 [IPV4Address](#),
 [MACAddress](#),
 [_UndefinedRepresentation](#) }
recommended representation of a node value
- enum [EEndianess](#) {
 [BigEndian](#),
 [LittleEndian](#),
 [_UndefinedEndian](#) }
Endianess of a value in a register.
- enum [ENamespace](#) {
 [Custom](#),
 [Standard](#),
 [_UndefinedNameSpace](#) }
Defines if a node name is standard or custom.

- enum `EStandardNameSpace` {
`None`,
`GEV`,
`IIDC`,
`CL`,
`USB`,
`_UndefinedStandardNameSpace` }

Defines from which standard namespace a node name comes from.

- enum `EYesNo` {
`Yes` = 1,
`No` = 0,
`_UndefinedYesNo` = 2 }

Defines the choices of a Yes/No alternative.

- enum `ESlope` {
`Increasing`,
`Decreasing`,
`Varying`,
`Automatic`,
`_UndefinedESlope` }

typedef for formula type

- enum `EXMLValidation` {
`xvLoad` = 0x00000001L,
`xvCycles` = 0x00000002L,
`xvSFNC` = 0x00000004L,
`xvDefault` = 0x00000000L,
`xvAll` = 0xffffffffL,
`_UndefinedEXMLValidation` = 0x80000000L }

typedef describing the different validity checks which can be performed on an XML file

- enum `EDisplayNotation` {
`fnAutomatic`,
`fnFixed`,
`fnScientific`,
`_UndefinedEDisplayNotation` }

typedef for float notation

- enum `EInterfaceType` {
`intflValue`,
`intflBase`,
`intflInteger`,
`intflBoolean`,
`intflCommand`,
`intflFloat`,
`intflString`,
`intflRegister`,
`intflCategory`,
`intflEnumeration`,
`intflEnumEntry`,
`intflPort` }

typedef for interface type

- enum `ELinkType` {
`ctParentNodes`,
`ctReadingChildren`,
`ctWritingChildren`,
`ctInvalidatingChildren`,
`ctDependingNodes`,
`ctTerminalNodes` }

typedef for link type

- enum [EIncMode](#) {
[noIncrement](#),
[fixedIncrement](#),
[listIncrement](#) }
typedef for increment mode
- enum [EInputDirection](#) {
[idFrom](#),
[idTo](#),
[idNone](#) }
typedef for link type
- enum [EGenApiSchemaVersion](#) {
[v1_0](#) = 1,
[v1_1](#) = 2,
[_Undefined](#) = -1 }
GenApi schema version.

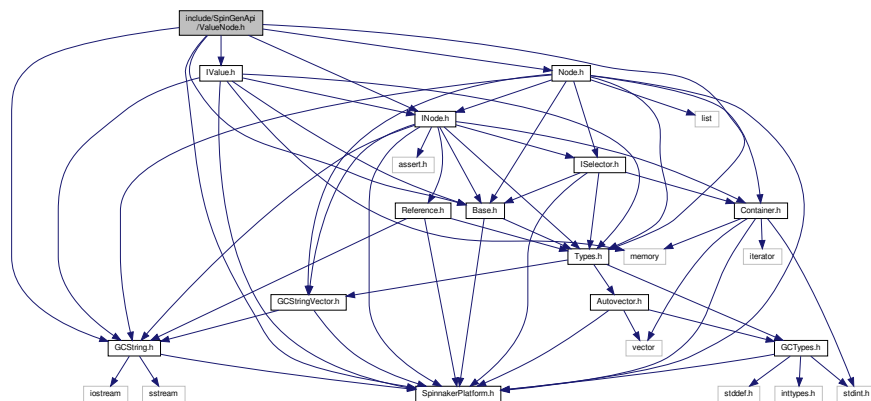
16.139.1 Macro Definition Documentation

16.139.1.1 interface

```
#define interface struct
```

16.140 include/SpinGenApi/ValueNode.h File Reference

Include dependency graph for ValueNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ValueNode](#)
Interface for value properties.

Namespaces

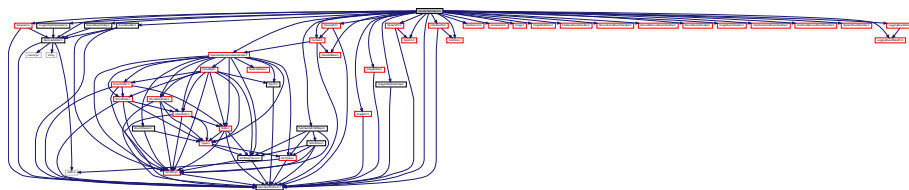
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

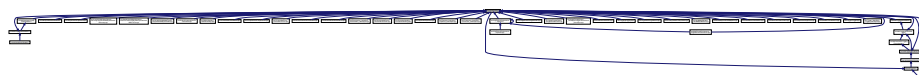
- typedef ValueNode [CValueRef](#)

16.141 include/Spinnaker.h File Reference

Include dependency graph for Spinnaker.h:

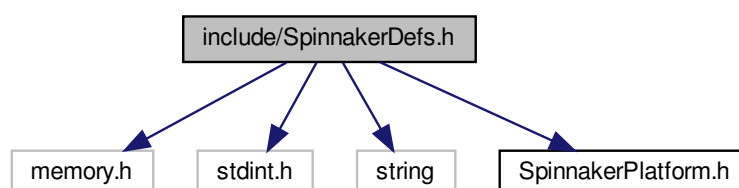


This graph shows which files directly or indirectly include this file:

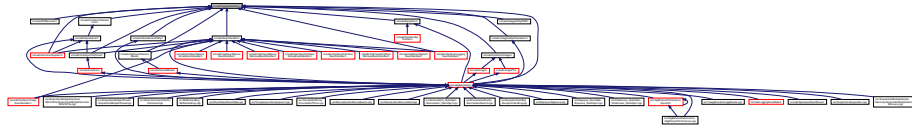


16.142 include/SpinnakerDefs.h File Reference

Include dependency graph for SpinnakerDefs.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [PNGOption](#)
Options for saving PNG images.
- struct [PPMOption](#)
Options for saving PPM images.
- struct [PGMOption](#)
Options for saving PGM images.
- struct [TIFFOption](#)
Options for saving TIFF images.
- struct [JPEGOption](#)
Options for saving JPEG image.
- struct [JPG2Option](#)
Options for saving JPEG2000 image.
- struct [BMPOption](#)
Options for saving Bitmap image.
- struct [LibraryVersion](#)
Provides easier access to the current version of [Spinnaker](#).
- struct [ActionCommandResult](#)
Action Command Result.
- struct [CCMSettings](#)
- struct [DeviceEventInferenceData](#)
Data Fields for Device Event payload for EventInference.
- struct [DeviceEventExposureEndData](#)
Data Fields for Device Event payload for EventExposureEnd.

Namespaces

- [Spinnaker](#)

Enumerations

- enum [Error](#) {
[SPINNAKER_ERR_SUCCESS](#) = 0,
[SPINNAKER_ERR_ERROR](#) = -1001,
[SPINNAKER_ERR_NOT_INITIALIZED](#) = -1002,
[SPINNAKER_ERR_NOT_IMPLEMENTED](#) = -1003,
[SPINNAKER_ERR_RESOURCE_IN_USE](#) = -1004,
[SPINNAKER_ERR_ACCESS_DENIED](#) = -1005,
[SPINNAKER_ERR_INVALID_HANDLE](#) = -1006,
[SPINNAKER_ERR_INVALID_ID](#) = -1007,
[SPINNAKER_ERR_NO_DATA](#) = -1008,

```

SPINNAKER_ERR_INVALID_PARAMETER = -1009,
SPINNAKER_ERR_IO = -1010,
SPINNAKER_ERR_TIMEOUT = -1011,
SPINNAKER_ERR_ABORT = -1012,
SPINNAKER_ERR_INVALID_BUFFER = -1013,
SPINNAKER_ERR_NOT_AVAILABLE = -1014,
SPINNAKER_ERR_INVALID_ADDRESS = -1015,
SPINNAKER_ERR_BUFFER_TOO_SMALL = -1016,
SPINNAKER_ERR_INVALID_INDEX = -1017,
SPINNAKER_ERR_PARSING_CHUNK_DATA = -1018,
SPINNAKER_ERR_INVALID_VALUE = -1019,
SPINNAKER_ERR_RESOURCE_EXHAUSTED = -1020,
SPINNAKER_ERR_OUT_OF_MEMORY = -1021,
SPINNAKER_ERR_BUSY = -1022,
GENICAM_ERR_INVALID_ARGUMENT = -2001,
GENICAM_ERR_OUT_OF_RANGE = -2002,
GENICAM_ERR_PROPERTY = -2003,
GENICAM_ERR_RUN_TIME = -2004,
GENICAM_ERR_LOGICAL = -2005,
GENICAM_ERR_ACCESS = -2006,
GENICAM_ERR_TIMEOUT = -2007,
GENICAM_ERR_DYNAMIC_CAST = -2008,
GENICAM_ERR_GENERIC = -2009,
GENICAM_ERR_BAD_ALLOCATION = -2010,
SPINNAKER_ERR_IM_CONVERT = -3001,
SPINNAKER_ERR_IM_COPY = -3002,
SPINNAKER_ERR_IM_MALLOC = -3003,
SPINNAKER_ERR_IM_NOT_SUPPORTED = -3004,
SPINNAKER_ERR_IM_HISTOGRAM_RANGE = -3005,
SPINNAKER_ERR_IM_HISTOGRAM_MEAN = -3006,
SPINNAKER_ERR_IM_MIN_MAX = -3007,
SPINNAKER_ERR_IM_COLOR_CONVERSION = -3008,
SPINNAKER_ERR_IM_DECOMPRESSION = -3009,
SPINNAKER_ERR_CUSTOM_ID = -10000 }

```

Spinnaker enum definitions.

- enum `EventType` {


```

SPINNAKER_EVENT_ARRIVAL_REMOVAL,
SPINNAKER_EVENT_DEVICE,
SPINNAKER_EVENT_DEVICE_SPECIFIC,
SPINNAKER_EVENT_NEW_BUFFER,
SPINNAKER_EVENT_LOGGING_EVENT,
SPINNAKER_EVENT_UNKNOWN,
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL }

```

Event types in Spinnaker.

- enum `PixelFormatNamespaceID` {


```

SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }

```

This enum represents the namespace in which the TL specific pixel format resides.

- enum `ColorProcessingAlgorithm` {


```

DEFAULT,
NO_COLOR_PROCESSING,
NEAREST_NEIGHBOR,
NEAREST_NEIGHBOR_AVG,
BILINEAR,

```

```

EDGE_SENSING,
HQ_LINEAR,
IPP,
DIRECTIONAL_FILTER,
RIGOROUS,
WEIGHTED_DIRECTIONAL_FILTER }

```

Color processing algorithms.

- enum `ImageFileFormat` {
`FROM_FILE_EXT` = -1,
`PGM`,
`PPM`,
`BMP`,
`JPEG`,
`JPEG2000`,
`TIFF`,
`PNG`,
`RAW`,
`JPEG12_C`,
`IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

File formats to be used for saving images to disk.

- enum `ImageStatus` {
`IMAGE_UNKNOWN_ERROR` = -1,
`IMAGE_NO_ERROR` = 0,
`IMAGE_CRC_CHECK_FAILED` = 1,
`IMAGE_DATA_OVERFLOW` = 2,
`IMAGE_MISSING_PACKETS`,
`IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT`,
`IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT`,
`IMAGE_PACKETID_INCONSISTENT`,
`IMAGE_MISSING_LEADER` = 7,
`IMAGE_MISSING_TRAILER`,
`IMAGE_DATA_INCOMPLETE`,
`IMAGE_INFO_INCONSISTENT`,
`IMAGE_CHUNK_DATA_INVALID` = 11,
`IMAGE_NO_SYSTEM_RESOURCES` = 12 }

Status of images returned from `GetNextImage()` call.

- enum `StatisticsChannel` {
`GREY`,
`RED`,
`GREEN`,
`BLUE`,
`HUE`,
`SATURATION`,
`LIGHTNESS`,
`NUM_STATISTICS_CHANNELS` }

Channels that allow statistics to be calculated.

- enum `SpinnakerLogLevel` {
`LOG_LEVEL_OFF` = -1,
`LOG_LEVEL_FATAL` = 0,
`LOG_LEVEL_ALERT` = 100,
`LOG_LEVEL_CRIT` = 200,
`LOG_LEVEL_ERROR` = 300,
`LOG_LEVEL_WARN` = 400,
`LOG_LEVEL_NOTICE` = 500,
`LOG_LEVEL_INFO` = 600,
`LOG_LEVEL_DEBUG` = 700,
`LOG_LEVEL_NOTSET` = 800 }

log levels

- enum `PayloadTypeInfoIds` {
`PAYLOAD_TYPE_UNKNOWN` = 0,
`PAYLOAD_TYPE_IMAGE` = 1,
`PAYLOAD_TYPE_RAW_DATA` = 2,
`PAYLOAD_TYPE_FILE` = 3,
`PAYLOAD_TYPE_CHUNK_DATA` = 4,
`PAYLOAD_TYPE_JPEG` = 5,
`PAYLOAD_TYPE_JPEG2000` = 6,
`PAYLOAD_TYPE_H264` = 7,
`PAYLOAD_TYPE_CHUNK_ONLY` = 8,
`PAYLOAD_TYPE_DEVICE_SPECIFIC` = 9,
`PAYLOAD_TYPE_MULTI_PART` = 10,
`PAYLOAD_TYPE_CUSTOM_ID` = 1000,
`PAYLOAD_TYPE_EXTENDED_CHUNK` = 1001,
`PAYLOAD_TYPE_LOSSLESS_COMPRESSED` = 1002,
`PAYLOAD_TYPE_LOSSY_COMPRESSED` = 1003,
`PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED` = 1004,
`PAYLOAD_TYPE_CHUNK_DATA_LOSSLESS_COMPRESSED` = 1005,
`PAYLOAD_TYPE_CHUNK_DATA_LOSSY_COMPRESSED` = 1006 }
- enum `ActionCommandStatus` {
`ACTION_COMMAND_STATUS_OK` = 0,
`ACTION_COMMAND_STATUS_NO_REF_TIME`,
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,
`ACTION_COMMAND_STATUS_ACTION_LATE`,
`ACTION_COMMAND_STATUS_ERROR` }

Possible Status Codes Returned from Action Command.

- enum `PixelFormatIntType` {
`IntType_UINT8`,
`IntType_INT8`,
`IntType_UINT10`,
`IntType_UINT10p`,
`IntType_UINT10P`,
`IntType_UINT12`,
`IntType_UINT12p`,
`IntType_UINT12P`,
`IntType_UINT14`,
`IntType_UINT16`,
`IntType_INT16`,
`IntType_FLOAT32`,
`IntType_UNKNOWN` }

Possible integer types and packing used in a pixel format.

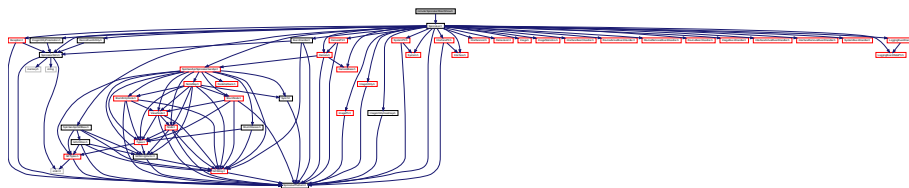
- enum `BufferOwnership` {
`BUFFER_OWNERSHIP_SYSTEM`,
`BUFFER_OWNERSHIP_USER` }
- enum `CCMColorTemperature` {
`TUNGSTEN_2800K`,
`WARM_FLUORESCENT_3000K`,
`COOL_FLUORESCENT_4000K`,
`SUNNY_5000K`,
`CLOUDY_6500K`,
`SHADE_8000K` }
- enum `CCMType` {
`LINEAR_3X3`,
`POLYNOMIAL_9X3` }
- enum `CCMSensor` { `IMX250` }

Variables

- const uint64_t [EVENT_TIMEOUT_NONE](#) = 0
Timeout values for getting next image, device, or interface event.
- const uint64_t [EVENT_TIMEOUT_INFINITE](#) = 0xFFFFFFFFFFFFFFFF

16.143 include/SpinnakerDirectShow.h File Reference

Include dependency graph for SpinnakerDirectShow.h:



Functions

- STDMETHOD() [GetSelectedCameraIndex](#) (unsigned int *selectedIndex)=0
Retrieves an integer index to the currently selected camera.
- STDMETHOD() [SetSelectedCameraIndex](#) (unsigned int index, bool needsRelease=true)=0
Sets the currently selected camera to the index specified.
- STDMETHOD() [GetCameraInfo](#) (unsigned int index, char *model, char *type, char *serial, size_t bufferSize)=0
Retrieves general information about the camera at the specified index.
- STDMETHOD() [IsStreaming](#) (bool *isStreaming)=0
Checks if the selected camera is currently streaming.
- STDMETHOD() [BeginAcquisition](#) ()=0
Starts the image acquisition engine for the currently selected camera.
- STDMETHOD() [EndAcquisition](#) ()=0
Stops the image acquisition engine for the currently selected camera.

Spinnaker GenAPI Functions

These functions deal with [Spinnaker](#) GenAPI NodeMap and Node Accesses for setting camera properties.

- STDMETHOD() [NodeMapGetNumNodes](#) (size_t *numNodes)=0
Retrieves the number of nodes available in the node map for the currently selected camera.
- STDMETHOD() [NodeMapGetNodeAtIndex](#) (size_t index, char *nodeName, size_t bufferSize)=0
Retrieves the string representation of the node at the specified index.
- STDMETHOD() [NodesImplemented](#) (const char *nodeName, bool *isImplemented)=0
Checks if a node is implemented.
- STDMETHOD() [NodesAvailable](#) (const char *nodeName, bool *isAvailable)=0
Checks if a node is available.
- STDMETHOD() [NodesReadable](#) (const char *nodeName, bool *isReadable)=0
Checks if a node is readable.
- STDMETHOD() [NodesWritable](#) (const char *nodeName, bool *isWritable)=0
Checks if a node is writable.
- STDMETHOD() [NodeGetType](#) (const char *nodeName, char *typeName, size_t bufferSize)=0
Gets the type of the node.

- STDMETHOD() [NodeToString](#) (const char *nodeName, char *valueAsString, size_t bufferSize)=0
Gets content of the node as string.
- STDMETHOD() [NodeGetDisplayName](#) (const char *nodeName, char *displayName, size_t bufferSize)=0
Gets a name string for display.

Integer Node Functions

These functions deal with [Spinnaker](#) GenAPI Integer Node Accesses

- STDMETHOD() [IntegerGetValue](#) (const char *nodeName, int64_t *value)=0
Gets integer node value for the specified feature.
- STDMETHOD() [IntegerSetValue](#) (const char *nodeName, int64_t value)=0
Sets integer node value for the specified feature.
- STDMETHOD() [IntegerGetMax](#) (const char *nodeName, int64_t *maxValue)=0
Gets maximum integer value allowed for the specified feature.
- STDMETHOD() [IntegerGetMin](#) (const char *nodeName, int64_t *minValue)=0
Gets minimum integer value allowed for the specified feature.
- STDMETHOD() [IntegerGetIncMode](#) (const char *nodeName, [Spinnaker::GenApi::ElncMode](#) *incMode)=0
Gets integer increment mode for the specified feature.
- STDMETHOD() [IntegerGetInc](#) (const char *nodeName, int64_t *increment)=0
Gets integer step increment for the specified feature.

Float Node Functions

These functions deal with [Spinnaker](#) GenAPI Float Node Accesses

- STDMETHOD() [FloatGetValue](#) (const char *nodeName, double *value)=0
Gets float node value for the specified feature.
- STDMETHOD() [FloatSetValue](#) (const char *nodeName, double value)=0
Sets float node value for the specified feature.
- STDMETHOD() [FloatGetMax](#) (const char *nodeName, double *floatMax)=0
Gets maximum float value allowed for the specified feature.
- STDMETHOD() [FloatGetMin](#) (const char *nodeName, double *floatMin)=0
Gets minimum float value allowed for the specified feature.
- STDMETHOD() [FloatGetIncMode](#) (const char *nodeName, [Spinnaker::GenApi::ElncMode](#) *incMode)=0
Gets float increment mode for the specified feature.
- STDMETHOD() [FloatGetInc](#) (const char *nodeName, double *increment)=0
Gets float step increment for the specified feature.

Boolean Node Functions

These functions deal with [Spinnaker](#) GenAPI Boolean Node Accesses

- STDMETHOD() [BooleanGetValue](#) (const char *nodeName, bool *value)=0
Gets boolean node value for the specified feature.
- STDMETHOD() [BooleanSetValue](#) (const char *nodeName, bool value)=0
Sets boolean node value for the specified feature.

String Node Functions

These functions deal with [Spinnaker](#) GenAPI String Node Accesses

- STDMETHOD() [StringGetValue](#) (const char *nodeName, char *value, size_t bufferSize)=0
Gets string node value for the specified feature.
- STDMETHOD() [StringSetValue](#) (const char *nodeName, const char *value)=0
Sets string node value for the specified feature.

Command Node Functions

These functions deal with [Spinnaker](#) GenAPI Command Node Accesses

- STDMETHOD() [CommandExecute](#) (const char *nodeName)=0
Executes the command for the specified feature.

Variables

- const unsigned int [MAX_LENGTH](#) = 256
- static const GUID [IID_ISpinnakerInterface](#)

This is the Interface that allows users to get and set device node properties on the camera.

Enumeration Node Functions

These functions deal with [Spinnaker](#) GenAPI Enumeration and EnumeratEntry Node Accesses

- const char * [enumerationName](#)
Checks if an enumeration entry exists for the specified enumeration feature.
- const char const char * [enumerationEntryName](#)
- const char const char bool * [entryExist](#) = 0
- const char unsigned int [entryIndex](#)
- const char unsigned int char size_t [bufferSize](#) = 0
- STDMETHODCALLTYPE [EnumerationGetEntry](#) (const char *nodeName, char *value, size_t [bufferSize](#))=0
Gets enumeration entry string for the specified enumeration feature.
- STDMETHODCALLTYPE [EnumerationSetEntry](#) (const char *nodeName, const char *value)=0
Sets enumeration entry for the specified enumeration feature.
- STDMETHODCALLTYPE [EnumerationGetNumEntries](#) (const char *[enumerationName](#), size_t *numEntries)=0
Retrieves the number of enumeration entry nodes available for the specified enumeration feature.

16.143.1 Variable Documentation

16.143.1.1 MAX_LENGTH

```
const unsigned int MAX_LENGTH = 256
```

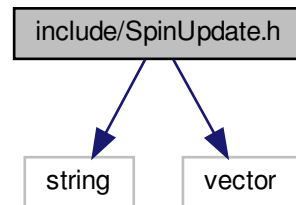
16.144 include/SpinnakerPlatform.h File Reference

Macros

- #define [SPINNAKER_API_ABSTRACT](#) /*nothing*/
- #define [SPINNAKER_API](#) __attribute__((visibility("default")))
- #define [SPINNAKER_LOCAL](#) __attribute__((visibility("hidden")))

16.145 include/SpinUpdate.h File Reference

Include dependency graph for SpinUpdate.h:



Macros

- `#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT`

Functions

- `SPINUPDATE_API int UpdateFirmwareConsole` (unsigned int numArgs, char **argList)
Updates the firmware for the device.
- `SPINUPDATE_API int UpdateFirmwareGUI` (std::string args)
- `SPINUPDATE_API int UpdateFirmware` (std::vector< std::string > args)
- `SPINUPDATE_API void SetMessageCallback` (UpdaterMessageCallback messageCallbackFunction)
- `SPINUPDATE_API void SetProgressCallback` (UpdaterProgressCallback progressCallbackFunction)
- `SPINUPDATE_API const char * GetErrorMessage` ()

Variables

- `SPINUPDATE_API typedef int(* UpdatorMessageCallback)` (const char *message)
- `SPINUPDATE_API typedef int(* UpdatorProgressCallback)` (const char *action, unsigned int address, int globalPercent, int currPercent)

16.145.1 Macro Definition Documentation

16.145.1.1 SPINUPDATE_API

```
#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT
```


16.145.2 Function Documentation

16.145.2.1 GetErrorMessage()

```
SPINUPDATE_API const char* GetErrorMessage ( )
```

16.145.2.2 SetMessageCallback()

```
SPINUPDATE_API void SetMessageCallback (
    UpdatorMessageCallback messageCallbackFunction )
```

16.145.2.3 SetProgressCallback()

```
SPINUPDATE_API void SetProgressCallback (
    UpdatorProgressCallback progressCallbackFunction )
```

16.145.2.4 UpdateFirmware()

```
SPINUPDATE_API int UpdateFirmware (
    std::vector< std::string > args )
```

16.145.2.5 UpdateFirmwareConsole()

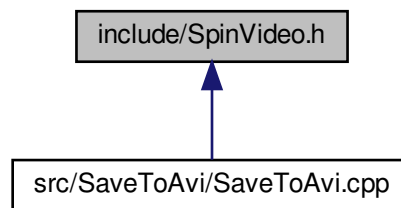
```
SPINUPDATE_API int UpdateFirmwareConsole (
    unsigned int numArgs,
    char ** argList )
```

Updates the firmware for the device.

Parameters

<i>numArgs</i>	Number of strings pointed to by argv
<i>argList</i>	Pointer to list of string options for the firmware update

This graph shows which files directly or indirectly include this file:



Classes

- class [SpinVideo](#)

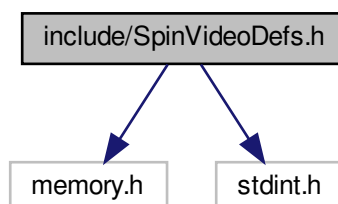
Provides the functionality for the user to record images to an AVI/MP4 file.

Namespaces

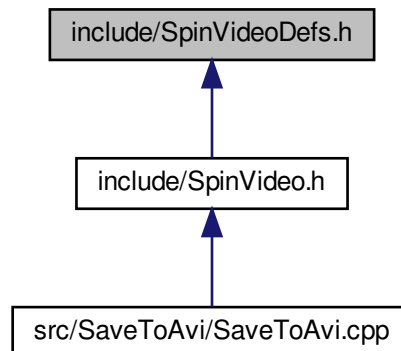
- [Spinnaker](#)
- [Spinnaker::Video](#)

16.147 include/SpinVideoDefs.h File Reference

Include dependency graph for `SpinVideoDefs.h`:



This graph shows which files directly or indirectly include this file:



Classes

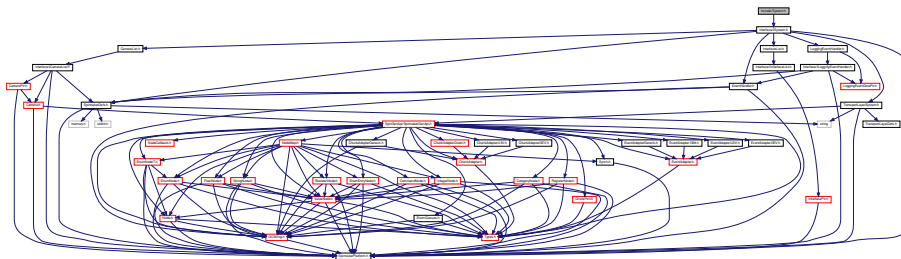
- struct [MJPGOption](#)
Options for saving MJPG files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [AVIOption](#)
Options for saving AVI files.

Namespaces

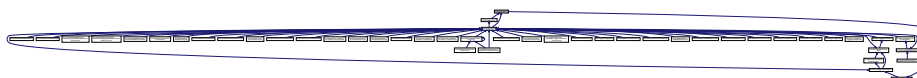
- [Spinnaker](#)
- [Spinnaker::Video](#)

16.148 include/System.h File Reference

Include dependency graph for System.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [System](#)

The system object is used to retrieve the list of interfaces and cameras available.

Namespaces

- [Spinnaker](#)

Macros

- `#define FLIR_SPINNAKER_VERSION_MAJOR 2`
- `#define FLIR_SPINNAKER_VERSION_MINOR 4`
- `#define FLIR_SPINNAKER_VERSION_TYPE 0`
- `#define FLIR_SPINNAKER_VERSION_BUILD 143`

16.148.1 Macro Definition Documentation

16.148.1.1 FLIR_SPINNAKER_VERSION_BUILD

```
#define FLIR_SPINNAKER_VERSION_BUILD 143
```

16.148.1.2 FLIR_SPINNAKER_VERSION_MAJOR

```
#define FLIR_SPINNAKER_VERSION_MAJOR 2
```

16.148.1.3 FLIR_SPINNAKER_VERSION_MINOR

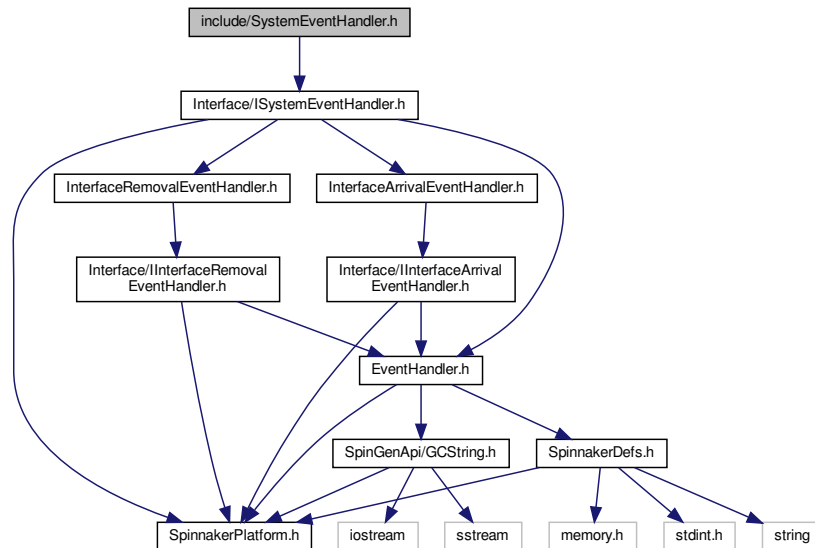
```
#define FLIR_SPINNAKER_VERSION_MINOR 4
```

16.148.1.4 FLIR_SPINNAKER_VERSION_TYPE

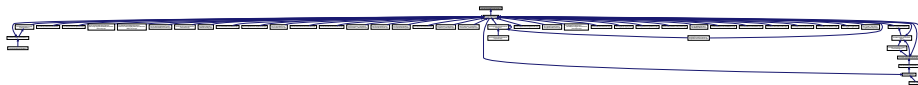
```
#define FLIR_SPINNAKER_VERSION_TYPE 0
```

16.149 include/SystemEventHandler.h File Reference

Include dependency graph for SystemEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

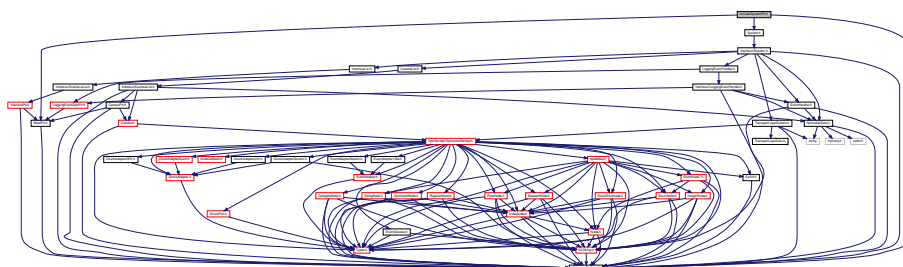
- class [SystemEventHandler](#)
A handler to interface arrival and removal events on the system.

Namespaces

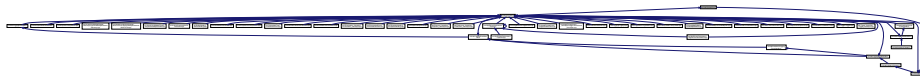
- [Spinnaker](#)

16.150 include/SystemPtr.h File Reference

Include dependency graph for SystemPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

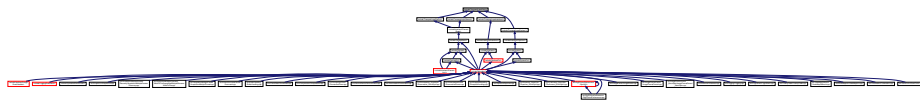
- class [SystemPtr](#)
A reference tracked pointer to a system object.

Namespaces

- [Spinnaker](#)

16.151 include/TransportLayerDefs.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)

Enumerations

- enum [StreamTypeEnum](#) {
[StreamType_GigEVision](#),
[StreamType_CameraLink](#),
[StreamType_CameraLinkHS](#),
[StreamType_CoaXPress](#),
[StreamType_USB3Vision](#),
[StreamType_Custom](#),
[NUMSTREAMTYPE](#) }
The enum definitions for TL Device nodes from the transport layer .xml files.
- enum [StreamBufferCountModeEnum](#) {
[StreamBufferCountMode_Manual](#),
[StreamBufferCountMode_Auto](#),
[NUMSTREAMBUFFERCOUNTMODE](#) }
- enum [StreamBufferHandlingModeEnum](#) {
[StreamBufferHandlingMode_OldestFirst](#),
[StreamBufferHandlingMode_OldestFirstOverwrite](#),
[StreamBufferHandlingMode_NewestOnly](#),
[StreamBufferHandlingMode_NewestFirst](#),
[NUMSTREAMBUFFERHANDLINGMODE](#) }

- enum DeviceTypeEnum {
DeviceType_GigEVision,
DeviceType_CameraLink,
DeviceType_CameraLinkHS,
DeviceType_CoaXPress,
DeviceType_USB3Vision,
DeviceType_Custom,
NUMDEVICETYPE }
- enum DeviceAccessStatusEnum {
DeviceAccessStatus_Unknown,
DeviceAccessStatus_ReadWrite,
DeviceAccessStatus_ReadOnly,
DeviceAccessStatus_NoAccess,
DeviceAccessStatus_Busy,
DeviceAccessStatus_OpenReadWrite,
DeviceAccessStatus_OpenReadOnly,
NUMDEVICEACCESSSTATUS }
- enum GevCCPEnum {
GevCCP_EnumEntry_GevCCP_OpenAccess,
GevCCP_EnumEntry_GevCCP_ExclusiveAccess,
GevCCP_EnumEntry_GevCCP_ControlAccess,
NUMGEVCCP }
- enum GUIXMLLocationEnum {
GUIXMLLocation_Device,
GUIXMLLocation_Host,
NUMGUIXMLLOCATION }
- enum GenICamXMLLocationEnum {
GenICamXMLLocation_Device,
GenICamXMLLocation_Host,
NUMGENICAMXMLLOCATION }
- enum DeviceEndiannessMechanismEnum {
DeviceEndiannessMechanism_Legacy,
DeviceEndiannessMechanism_Standard,
NUMDEVICEENDIANESSMECHANISM }
- enum DeviceCurrentSpeedEnum {
DeviceCurrentSpeed_UnknownSpeed,
DeviceCurrentSpeed_LowSpeed,
DeviceCurrentSpeed_FullSpeed,
DeviceCurrentSpeed_HighSpeed,
DeviceCurrentSpeed_SuperSpeed,
NUMDEVICECURRENTSPEED }
- enum InterfaceTypeEnum {
InterfaceType_GigEVision,
InterfaceType_CameraLink,
InterfaceType_CameraLinkHS,
InterfaceType_CoaXPress,
InterfaceType_USB3Vision,
InterfaceType_Custom,
NUMINTERFACETYPE }
- enum POEStatusEnum {
POEStatus_NotSupported,
POEStatus_PowerOff,
POEStatus_PowerOn,
NUMPOESTATUS }
- enum FilterDriverStatusEnum {
FilterDriverStatus_NotSupported,
FilterDriverStatus_Disabled,

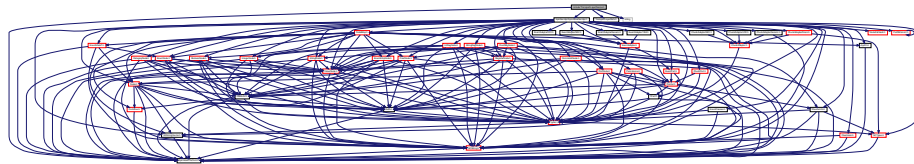

```

FilterDriverStatus_Enabled,
NUMFILTERDRIVERSTATUS }
• enum TLTypeEnum {
    TLType_GigEVision,
    TLType_CameraLink,
    TLType_CameraLinkHS,
    TLType_CoaXPress,
    TLType_USB3Vision,
    TLType_Mixed,
    TLType_Custom,
    NUMTLTYPE }

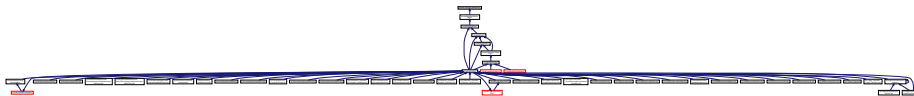
```

16.152 include/TransportLayerDevice.h File Reference

Include dependency graph for TransportLayerDevice.h:



This graph shows which files directly or indirectly include this file:



Classes

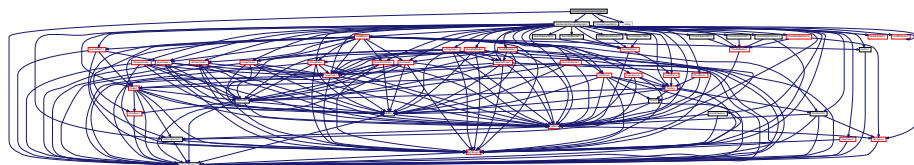
- class [TransportLayerDevice](#)
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

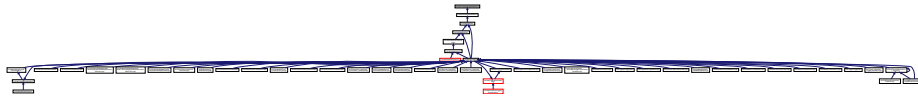
- [Spinnaker](#)

16.153 include/TransportLayerInterface.h File Reference

Include dependency graph for TransportLayerInterface.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TransportLayerInterface](#)

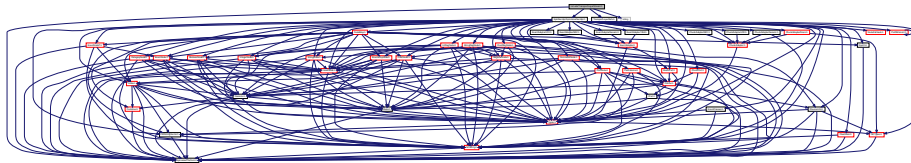
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

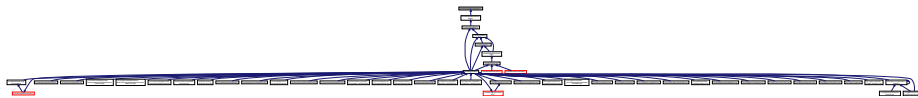
- [Spinnaker](#)

16.154 include/TransportLayerStream.h File Reference

Include dependency graph for TransportLayerStream.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TransportLayerStream](#)

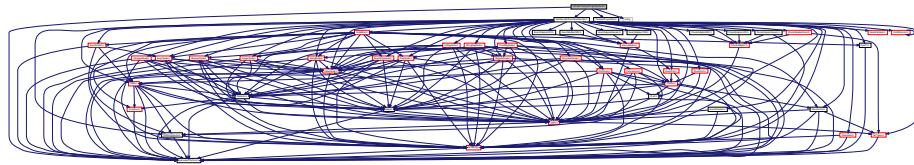
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

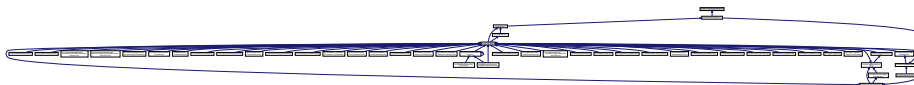
- [Spinnaker](#)

16.155 include/TransportLayerSystem.h File Reference

Include dependency graph for TransportLayerSystem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TransportLayerSystem](#)

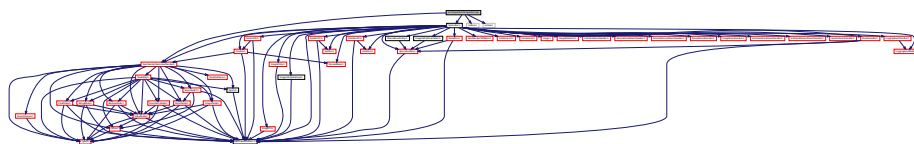
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

- [Spinnaker](#)

16.156 src/Acquisition/Acquisition.cpp File Reference

Include dependency graph for Acquisition.cpp:



Functions

- int [AcquireImages](#) ([CameraPtr](#) pCam, [INodeMap](#) &nodeMap, [INodeMap](#) &nodeMapTLDevice)
- int [PrintDeviceInfo](#) ([INodeMap](#) &nodeMap)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.156.1 Function Documentation

16.156.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTTLDevice )
```

16.156.1.2 main()

```
int main (
    int ,
    char ** )
```

16.156.1.3 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.156.1.4 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.157 src/Acquisition/resource.h File Reference**16.158 src/AcquisitionMultipleCameraRecovery/resource.h File Reference****16.159 src/AcquisitionMultipleCamerasWriteToFile/resource.h File Reference****16.160 src/AcquisitionMultipleThread/resource.h File Reference****16.161 src/ActionCommand/resource.h File Reference****16.162 src/BufferHandling/resource.h File Reference****16.163 src/ChunkData/resource.h File Reference**

- 16.164 src/Compression/resource.h File Reference
- 16.165 src/CounterAndTimer/resource.h File Reference
- 16.166 src/DeviceEvents/resource.h File Reference
- 16.167 src/Enumeration/resource.h File Reference
- 16.168 src/Enumeration_QuickSpin/resource.h File Reference
- 16.169 src/EnumerationEvents/resource.h File Reference
- 16.170 src/ExceptionHandling/resource.h File Reference
- 16.171 src/Exposure/resource.h File Reference
- 16.172 src/Exposure_QuickSpin/resource.h File Reference
- 16.173 src/FileAccess_QuickSpin/resource.h File Reference
- 16.174 src/GigEVisionPerformance/resource.h File Reference
- 16.175 src/ImageEvents/resource.h File Reference
- 16.176 src/ImageFormatControl/resource.h File Reference
- 16.177 src/ImageFormatControl_QuickSpin/resource.h File Reference
- 16.178 src/Inference/resource.h File Reference
- 16.179 src/Logging/resource.h File Reference
- 16.180 src/LogicBlock/resource.h File Reference

16.181 [src/LookupTable/resource.h](#) File Reference

16.182 [src/NodeMapCallback/resource.h](#) File Reference

16.183 [src/NodeMapInfo/resource.h](#) File Reference

16.184 [src/Polarization/resource.h](#) File Reference

16.185 [src/SaveToAvi/resource.h](#) File Reference

16.186 [src/Sequencer/resource.h](#) File Reference

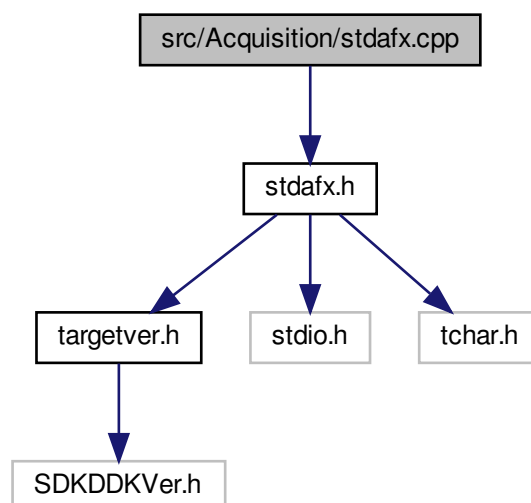
16.187 [src/SerialRxTx/resource.h](#) File Reference

16.188 [src/Trigger/resource.h](#) File Reference

16.189 [src/Trigger_QuickSpin/resource.h](#) File Reference

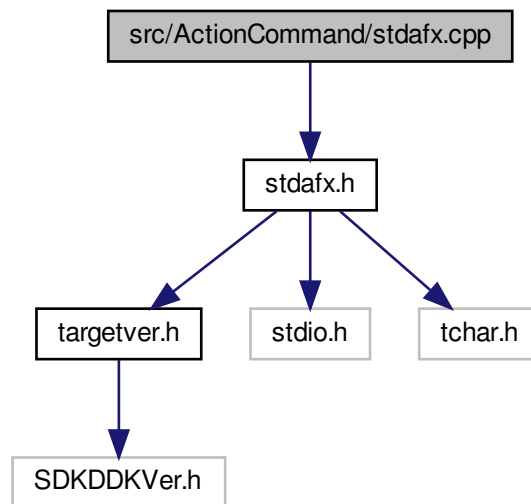
16.190 [src/Acquisition/stdafx.cpp](#) File Reference

Include dependency graph for stdafx.cpp:



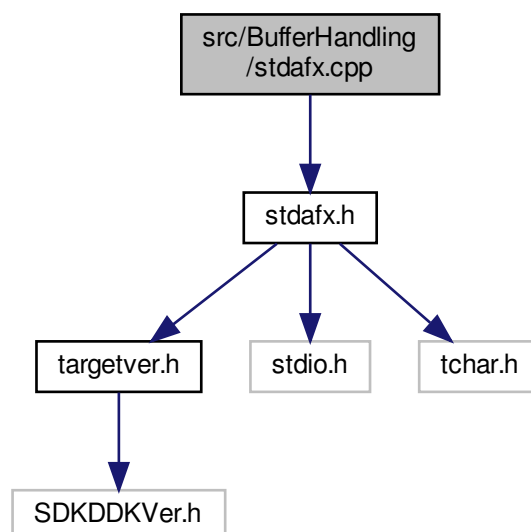
16.191 src/ActionCommand/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



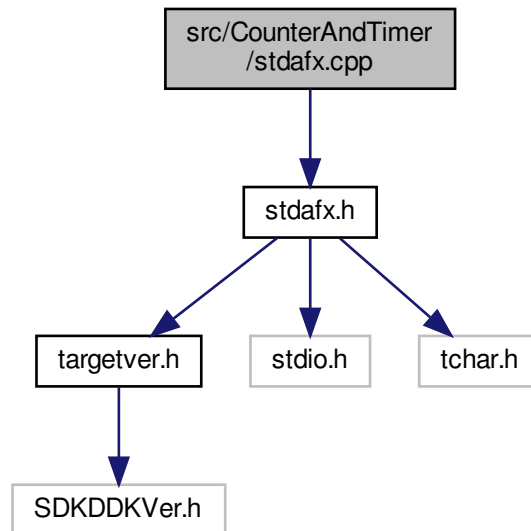
16.192 src/BufferHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



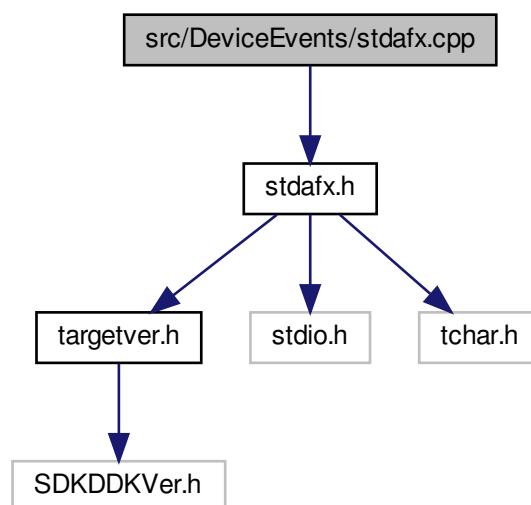
16.193 src/CounterAndTimer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



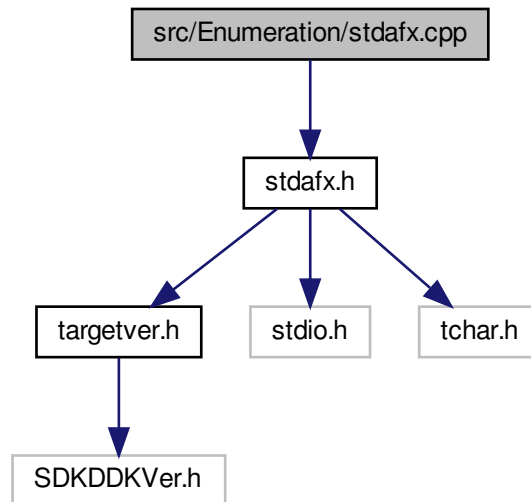
16.194 src/DeviceEvents/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



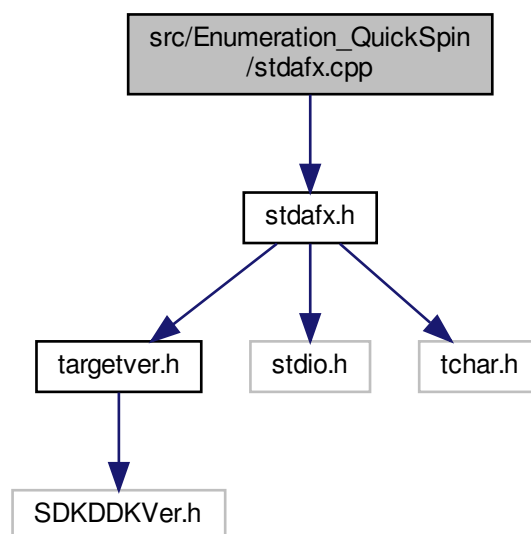
16.195 src/Enumeration/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



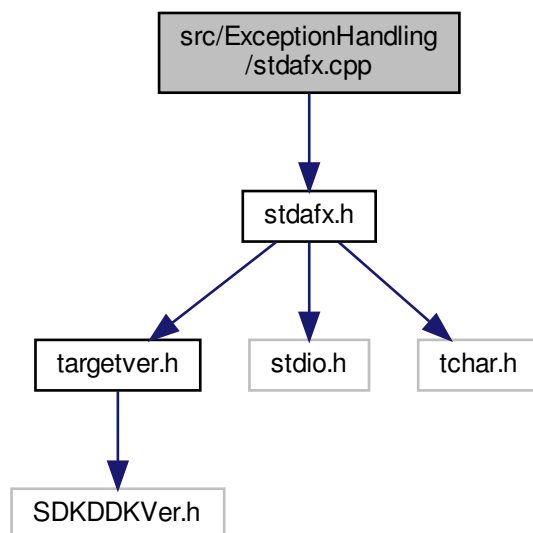
16.196 src/Enumeration_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



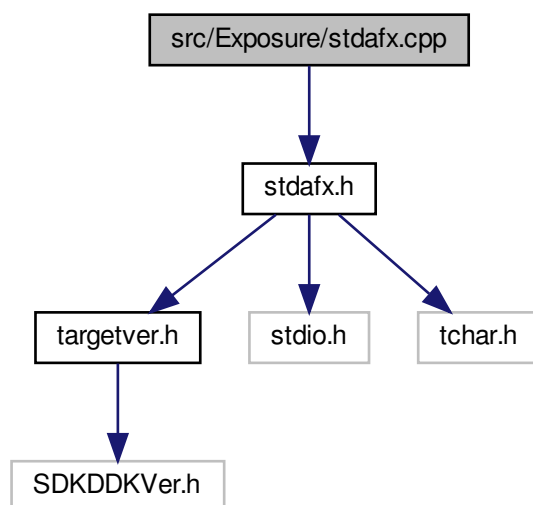
16.197 src/ExceptionHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



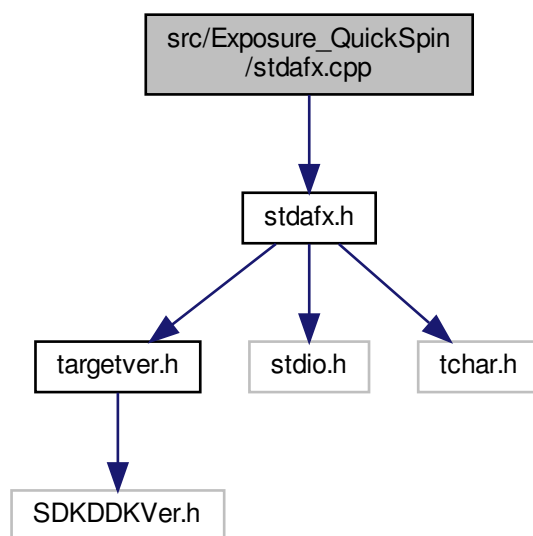
16.198 src/Exposure/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



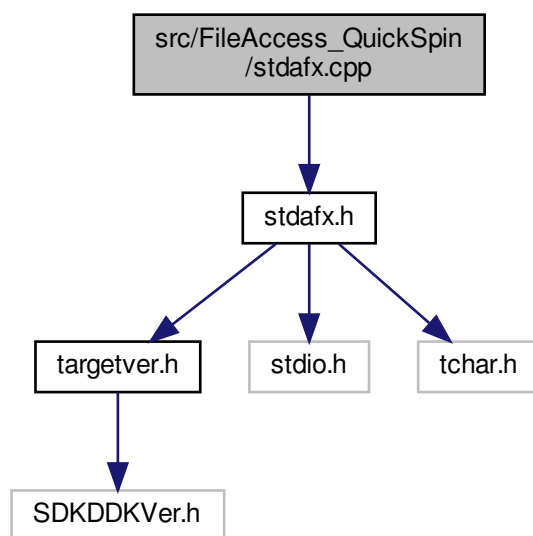
16.199 src/Exposure_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



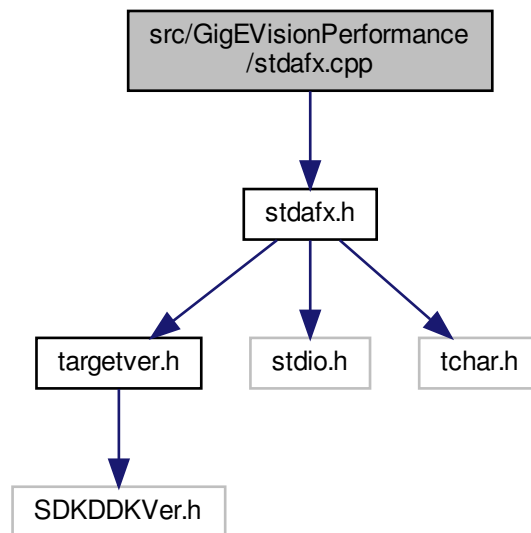
16.200 src/FileAccess_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



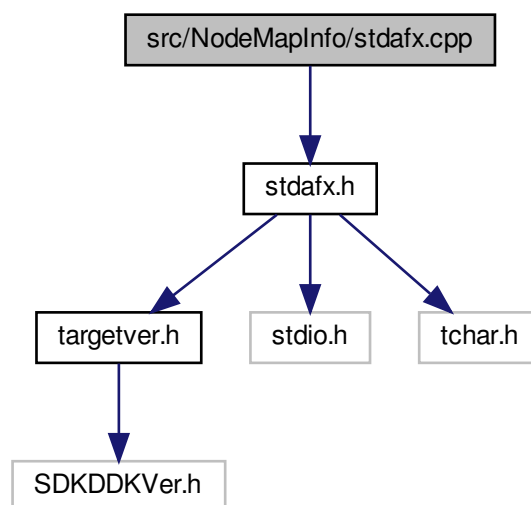
16.201 src/GigEVisionPerformance/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



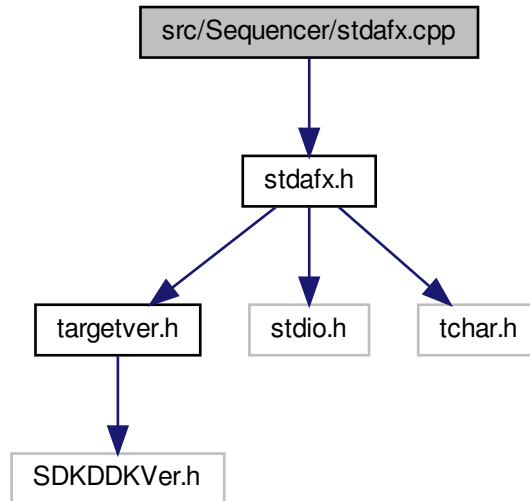
16.202 src/NodeMapInfo/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



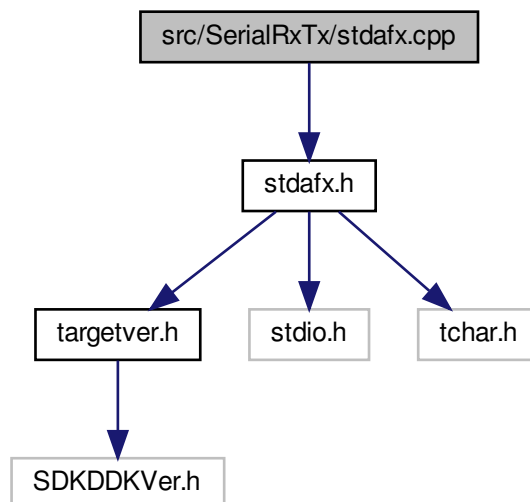
16.203 src/Sequencer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



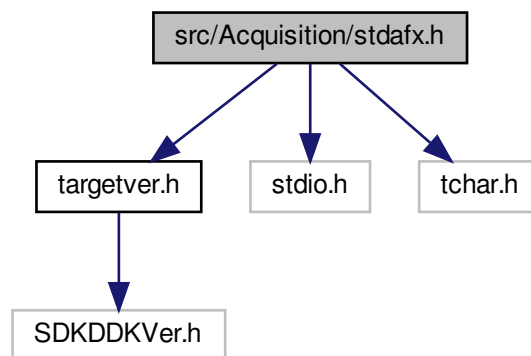
16.204 src/SerialRxTx/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

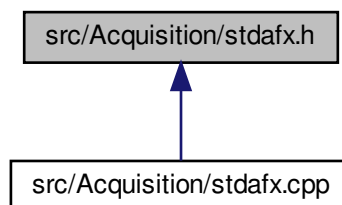


16.205 src/Acquisition/stdafx.h File Reference

Include dependency graph for stdafx.h:

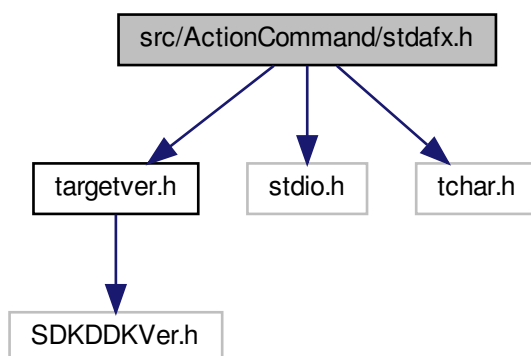


This graph shows which files directly or indirectly include this file:

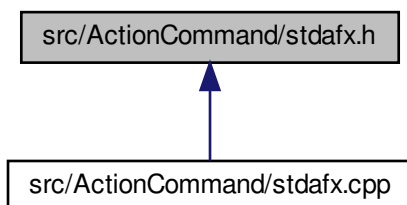


16.206 src/ActionCommand/stdafx.h File Reference

Include dependency graph for stdafx.h:

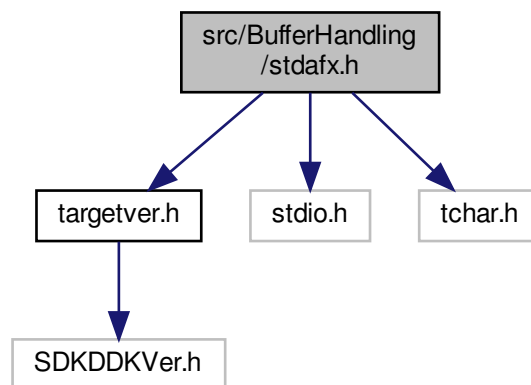


This graph shows which files directly or indirectly include this file:

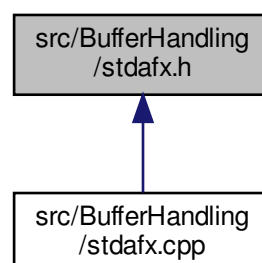


16.207 src/BufferHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

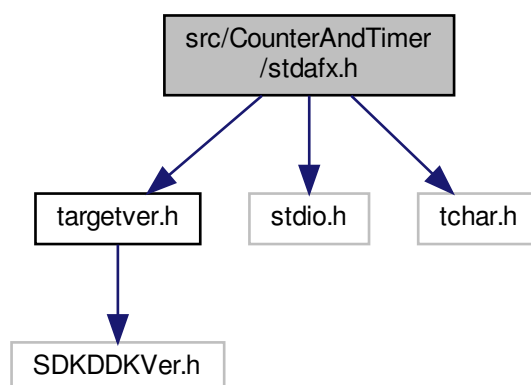


This graph shows which files directly or indirectly include this file:

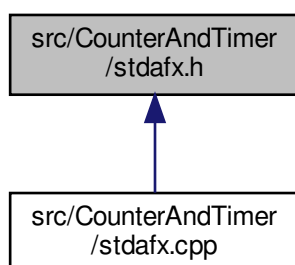


16.208 src/CounterAndTimer/stdafx.h File Reference

Include dependency graph for stdafx.h:

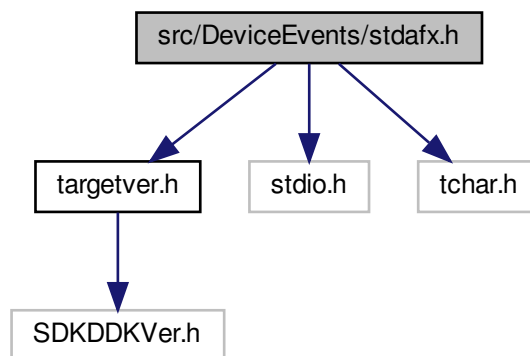


This graph shows which files directly or indirectly include this file:

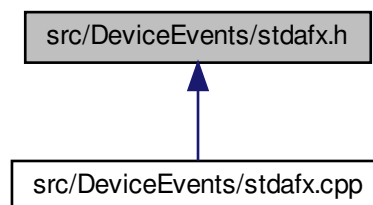


16.209 src/DeviceEvents/stdafx.h File Reference

Include dependency graph for stdafx.h:

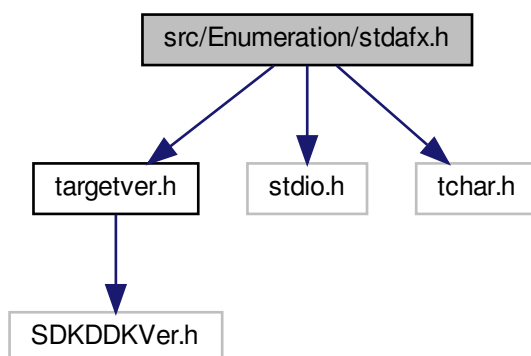


This graph shows which files directly or indirectly include this file:

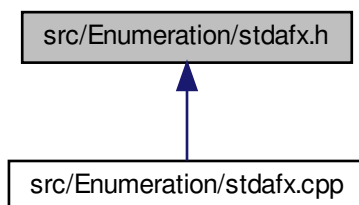


16.210 src/Enumeration/stdafx.h File Reference

Include dependency graph for stdafx.h:

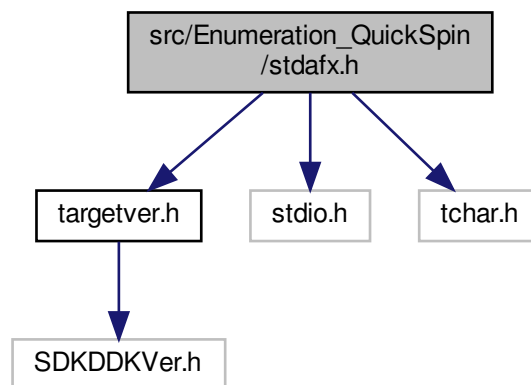


This graph shows which files directly or indirectly include this file:

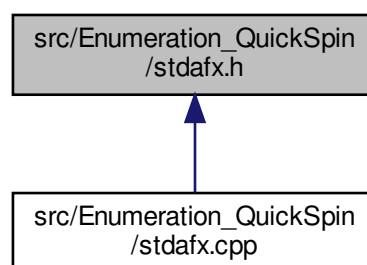


16.211 src/Enumeration_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

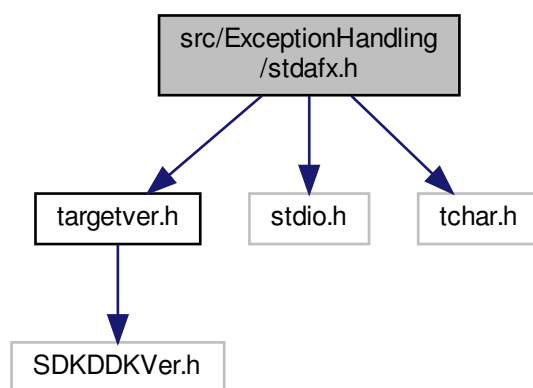


This graph shows which files directly or indirectly include this file:

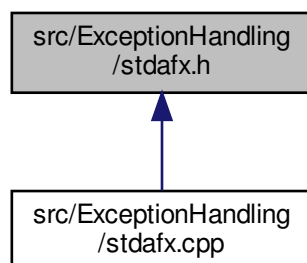


16.212 src/ExceptionHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

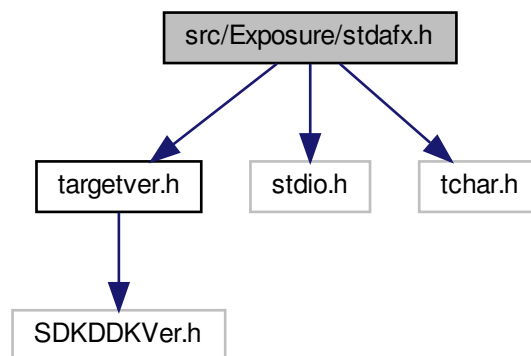


This graph shows which files directly or indirectly include this file:

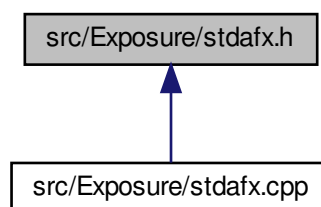


16.213 src/Exposure/stdafx.h File Reference

Include dependency graph for stdafx.h:

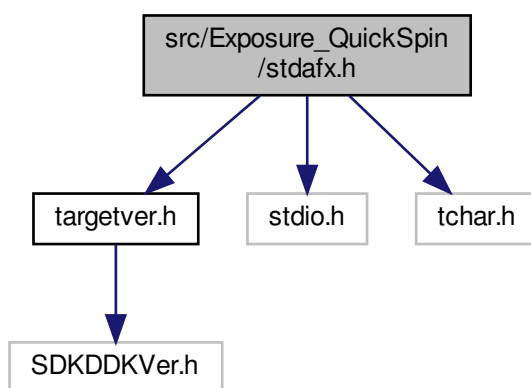


This graph shows which files directly or indirectly include this file:

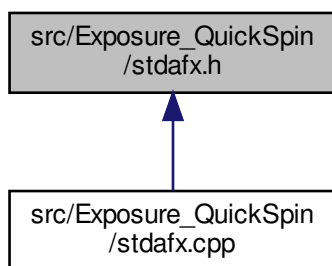


16.214 src/Exposure_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

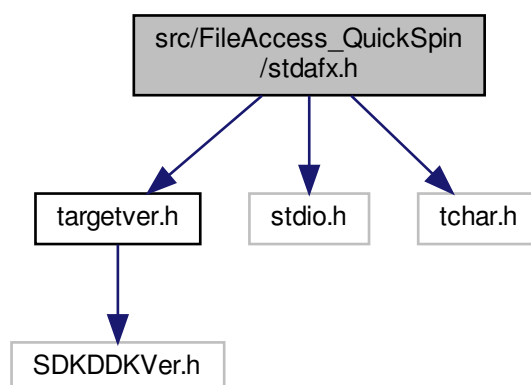


This graph shows which files directly or indirectly include this file:

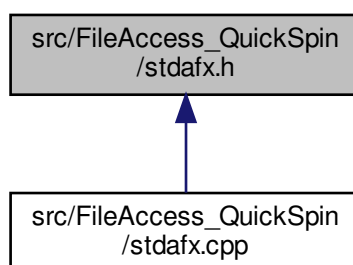


16.215 src/FileAccess_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

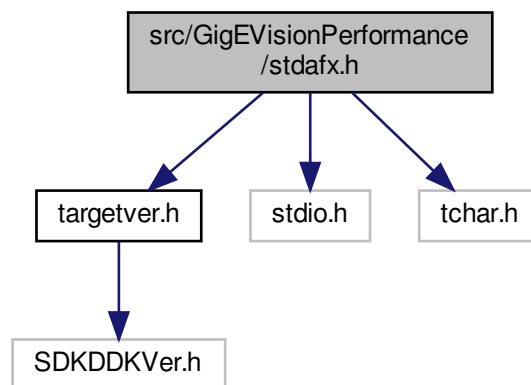


This graph shows which files directly or indirectly include this file:

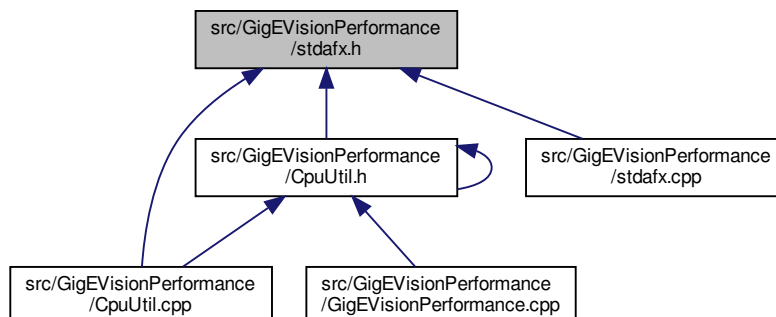


16.216 src/GigEVisionPerformance/stdafx.h File Reference

Include dependency graph for stdafx.h:

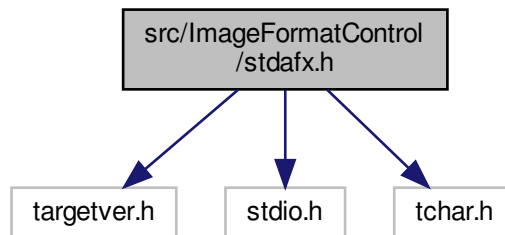


This graph shows which files directly or indirectly include this file:



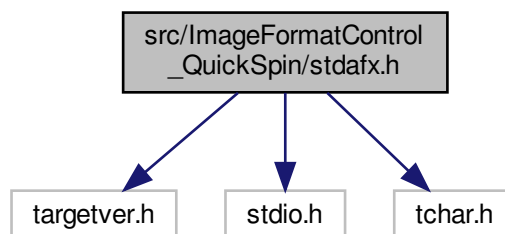
16.217 src/ImageFormatControl/stdafx.h File Reference

Include dependency graph for stdafx.h:



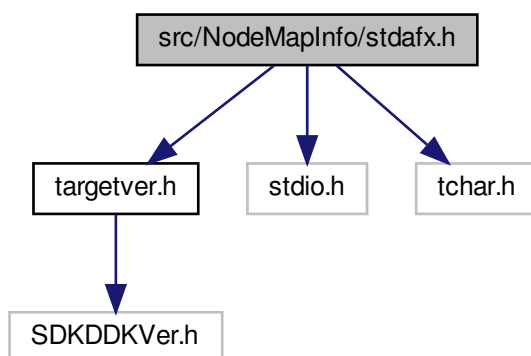
16.218 src/ImageFormatControl_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

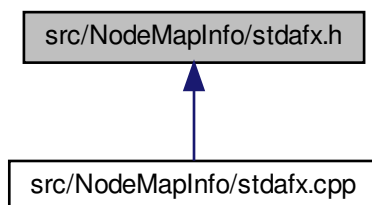


16.219 src/NodeMapInfo/stdafx.h File Reference

Include dependency graph for stdafx.h:

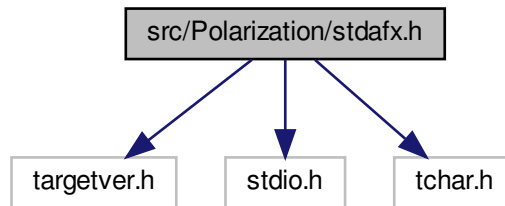


This graph shows which files directly or indirectly include this file:



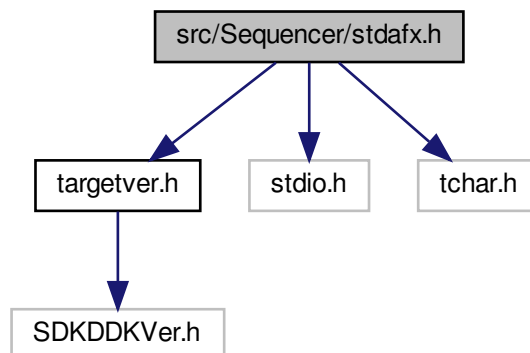
16.220 src/Polarization/stdafx.h File Reference

Include dependency graph for stdafx.h:

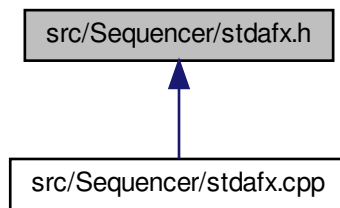


16.221 src/Sequencer/stdafx.h File Reference

Include dependency graph for stdafx.h:

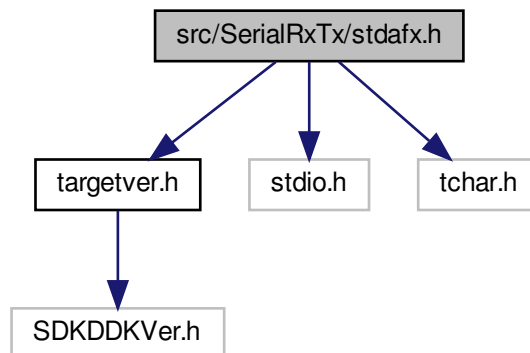


This graph shows which files directly or indirectly include this file:

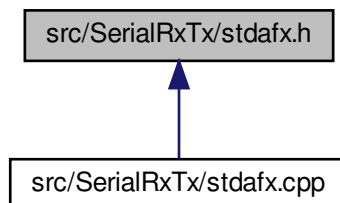


16.222 src/SerialRxTx/stdafx.h File Reference

Include dependency graph for stdafx.h:

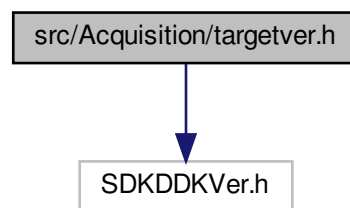


This graph shows which files directly or indirectly include this file:

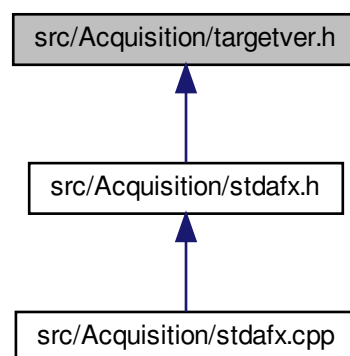


16.223 src/Acquisition/targetver.h File Reference

Include dependency graph for targetver.h:

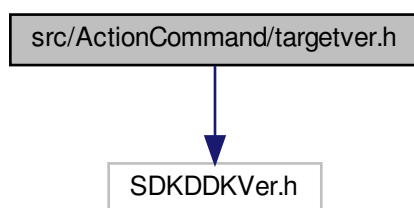


This graph shows which files directly or indirectly include this file:

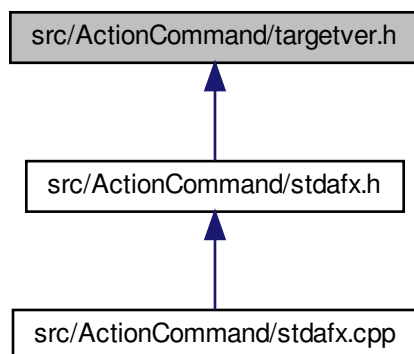


16.224 src/ActionCommand/targetver.h File Reference

Include dependency graph for targetver.h:

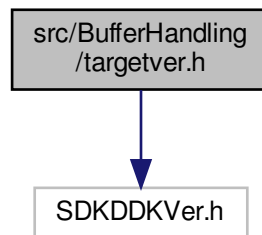


This graph shows which files directly or indirectly include this file:

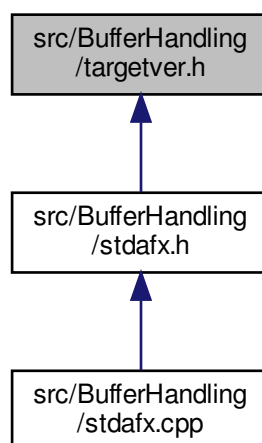


16.225 src/BufferHandling/targetver.h File Reference

Include dependency graph for targetver.h:

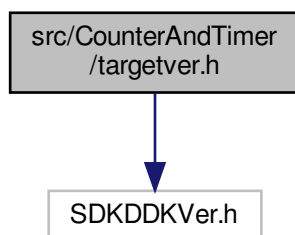


This graph shows which files directly or indirectly include this file:

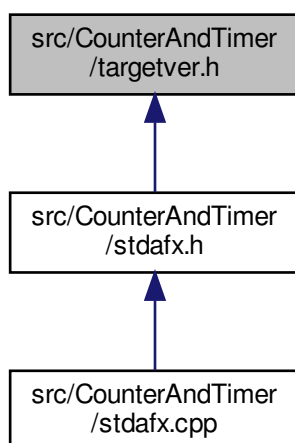


16.226 src/CounterAndTimer/targetver.h File Reference

Include dependency graph for targetver.h:

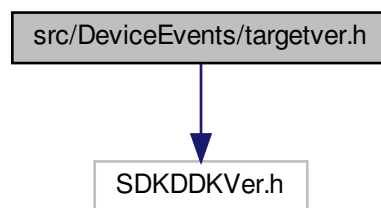


This graph shows which files directly or indirectly include this file:

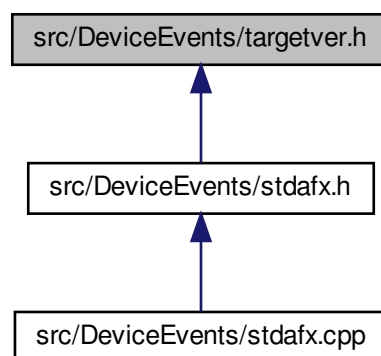


16.227 src/DeviceEvents/targetver.h File Reference

Include dependency graph for targetver.h:

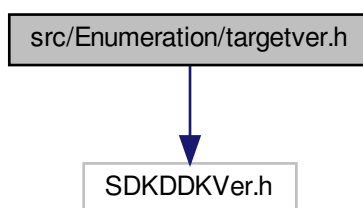


This graph shows which files directly or indirectly include this file:

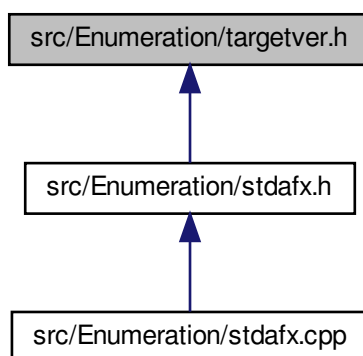


16.228 src/Enumeration/targetver.h File Reference

Include dependency graph for targetver.h:

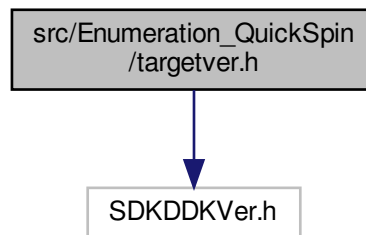


This graph shows which files directly or indirectly include this file:

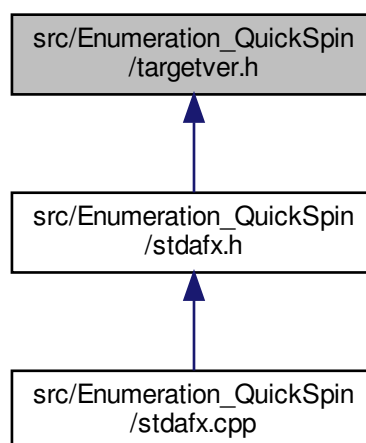


16.229 src/Enumeration_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

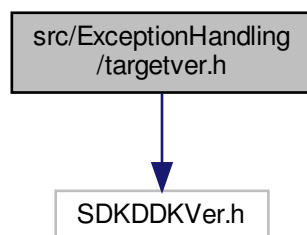


This graph shows which files directly or indirectly include this file:

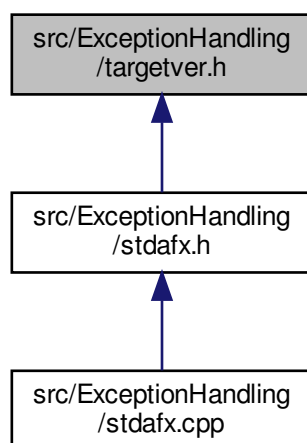


16.230 src/ExceptionHandling/targetver.h File Reference

Include dependency graph for targetver.h:

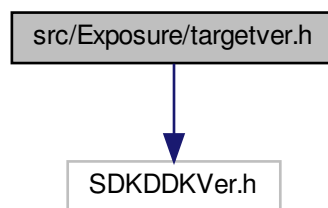


This graph shows which files directly or indirectly include this file:

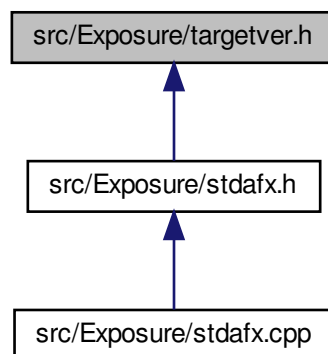


16.231 src/Exposure/targetver.h File Reference

Include dependency graph for targetver.h:

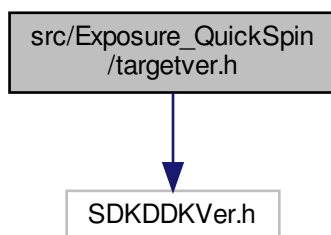


This graph shows which files directly or indirectly include this file:

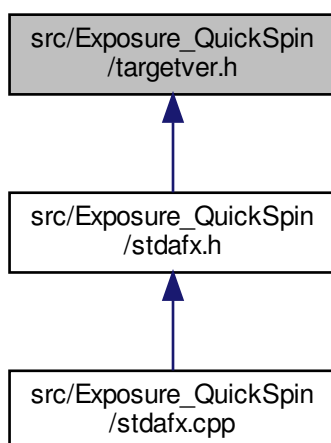


16.232 src/Exposure_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

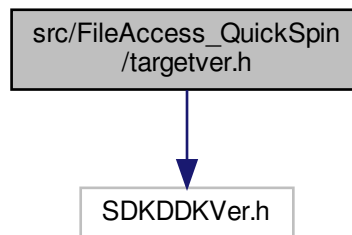


This graph shows which files directly or indirectly include this file:

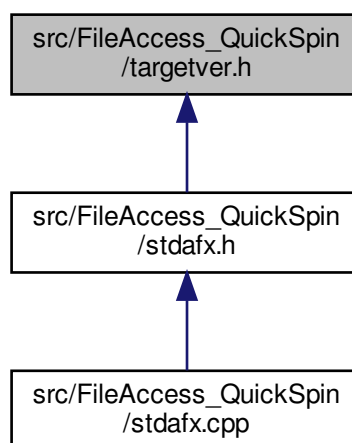


16.233 src/FileAccess_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

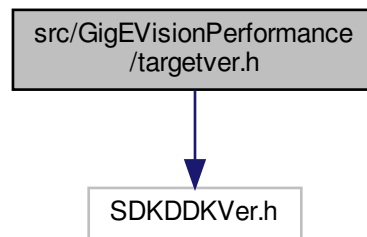


This graph shows which files directly or indirectly include this file:

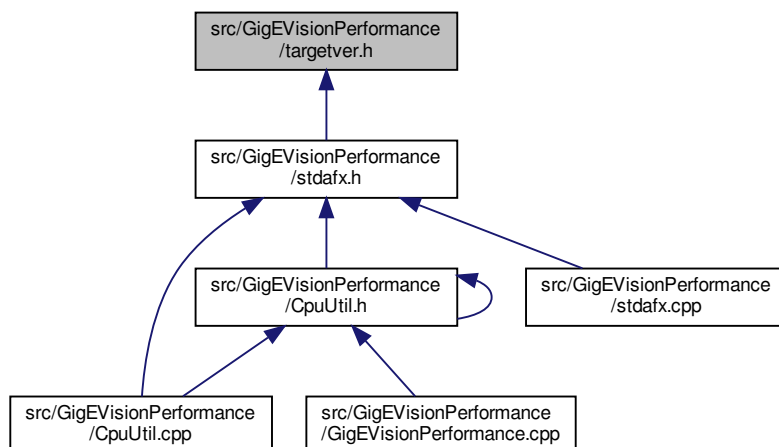


16.234 src/GigEVisionPerformance/targetver.h File Reference

Include dependency graph for targetver.h:

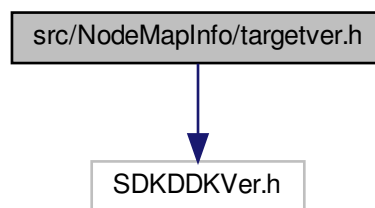


This graph shows which files directly or indirectly include this file:

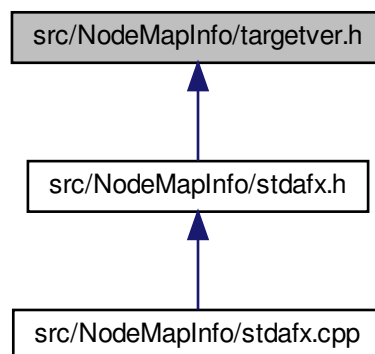


16.235 src/NodeMapInfo/targetver.h File Reference

Include dependency graph for targetver.h:

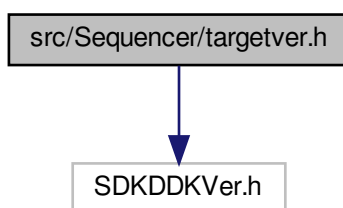


This graph shows which files directly or indirectly include this file:

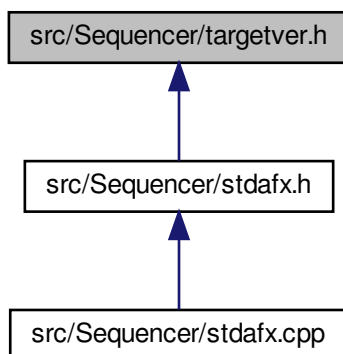


16.236 src/Sequencer/targetver.h File Reference

Include dependency graph for targetver.h:

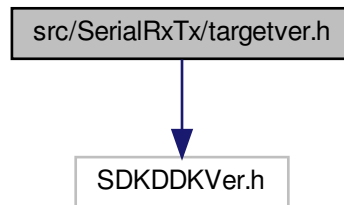


This graph shows which files directly or indirectly include this file:

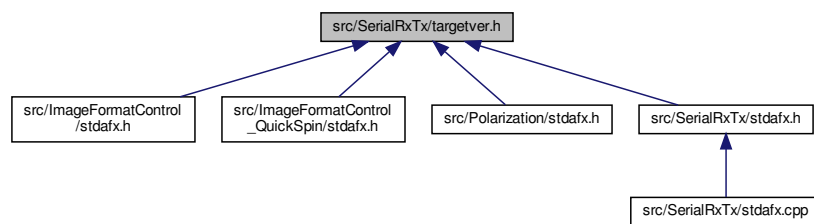


16.237 src/SerialRxTx/targetver.h File Reference

Include dependency graph for targetver.h:

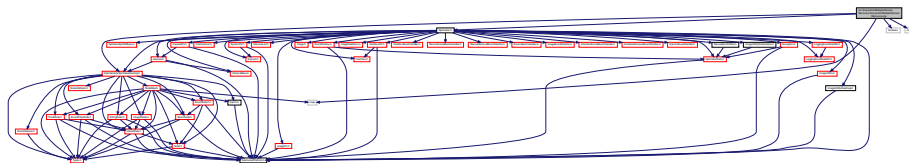


This graph shows which files directly or indirectly include this file:



16.238 src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp File Reference

Include dependency graph for AcquisitionMultipleCameraRecovery.cpp:



Classes

- class [ImageEventHandlerImpl](#)
- struct [GrabInfo](#)
- class [InterfaceEventHandlerImpl](#)

Functions

- void [SleepyWrapper](#) (int milliseconds)
- void [RefreshCameraList](#) ([SystemPtr](#) system)
- bool [ConfigureCamera](#) ([CameraPtr](#) pCam)
- bool [ConfigureUserSet1](#) ([CameraPtr](#) pCam)
- void [ResetCameraUserSetToDefault](#) ([CameraPtr](#) pCam)
- string [GetDeviceSerial](#) ([CameraPtr](#) pCam)
- void [PrintExampleStatistics](#) ()
- int [main](#) (int, char **)

Variables

- std::map< std::string, [GrabInfo](#) > [cameraGrabInfoMap](#)
- [CameraList](#) [globalCamList](#)

16.238.1 Function Documentation

16.238.1.1 [ConfigureCamera\(\)](#)

```
bool ConfigureCamera (  
    CameraPtr pCam )
```

16.238.1.2 [ConfigureUserSet1\(\)](#)

```
bool ConfigureUserSet1 (  
    CameraPtr pCam )
```

16.238.1.3 [GetDeviceSerial\(\)](#)

```
string GetDeviceSerial (  
    CameraPtr pCam )
```

16.238.1.4 [main\(\)](#)

```
int main (  
    int ,  
    char ** )
```

16.238.1.5 PrintExampleStatistics()

```
void PrintExampleStatistics ( )
```

16.238.1.6 RefreshCameraList()

```
void RefreshCameraList (
    SystemPtr system )
```

16.238.1.7 ResetCameraUserSetToDefault()

```
void ResetCameraUserSetToDefault (
    CameraPtr pCam )
```

16.238.1.8 SleepyWrapper()

```
void SleepyWrapper (
    int milliseconds )
```

16.238.2 Variable Documentation**16.238.2.1 cameraGrabInfoMap**

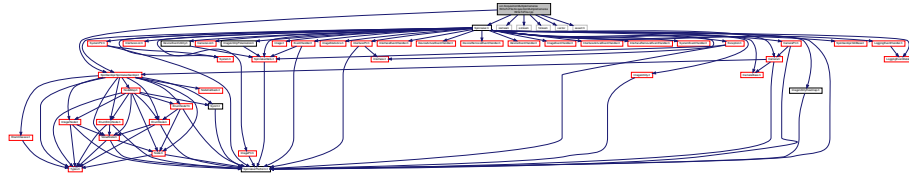
```
std::map<std::string, GrabInfo> cameraGrabInfoMap
```

16.238.2.2 globalCamList

```
CameraList globalCamList
```

16.239 src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp File Reference

Include dependency graph for AcquisitionMultipleCamerasWriteToFile.cpp:



Classes

- struct [ImageInfo](#)

Functions

- bool [CreateFiles](#) (unsigned int numCameras)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- bool [ConfigureCameras](#) (CameraList &camList, unsigned int numCameras)
- bool [AcquireImagesAndSaveToFile](#) (CameraList &camList, unsigned int numCameras)
- bool [RetrieveImagesFromFiles](#) (unsigned int numCameras, string fileFormat="bmp")
- int [RunCameras](#) (CameraList &camList, unsigned int numCameras)
- int [main](#) (int, char **)

Variables

- const string [kDestinationDirectory](#) = ""
- const unsigned int [k_numImages](#) = 30
- vector< [ImageInfo](#) > [imageInfos](#)

16.239.1 Function Documentation

16.239.1.1 AcquireImagesAndSaveToFile()

```
bool AcquireImagesAndSaveToFile (
    CameraList & camList,
    unsigned int numCameras )
```


16.239.1.2 ConfigureCameras()

```
bool ConfigureCameras (
    CameraList & camList,
    unsigned int numCameras )
```

16.239.1.3 CreateFiles()

```
bool CreateFiles (
    unsigned int numCameras )
```

16.239.1.4 main()

```
int main (
    int ,
    char ** )
```

16.239.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.239.1.6 RetrieveImagesFromFiles()

```
bool RetrieveImagesFromFiles (
    unsigned int numCameras,
    string fileFormat = "bmp" )
```

16.239.1.7 RunCameras()

```
int RunCameras (
    CameraList & camList,
    unsigned int numCameras )
```

16.239.2 Variable Documentation

16.239.2.1 imageInfos

```
vector<ImageInfo> imageInfos
```

16.239.2.2 k_numImages

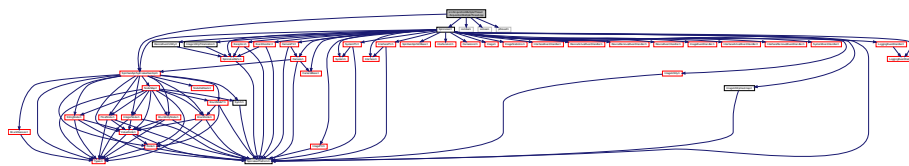
```
const unsigned int k_numImages = 30
```

16.239.2.3 kDestinationDirectory

```
const string kDestinationDirectory = ""
```

16.240 src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference

Include dependency graph for AcquisitionMultipleThread.cpp:



Functions

- int [PrintDeviceInfo](#) (INodeMap &nodeMap, std::string camSerial)
- void * [AcquireImages](#) (void *arg)
- int [RunMultipleCameras](#) (CameraList camList)
- int [main](#) (int, char **)

16.240.1 Function Documentation

16.240.1.1 AcquireImages()

```
void* AcquireImages (
    void * arg )
```

16.240.1.2 main()

```
int main (
    int ,
    char ** )
```

16.240.1.3 PrintDeviceInfo()

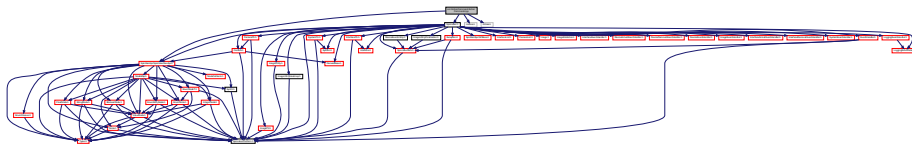
```
int PrintDeviceInfo (
    INodeMap & nodeMap,
    std::string camSerial )
```

16.240.1.4 RunMultipleCameras()

```
int RunMultipleCameras (
    CameraList camList )
```

16.241 src/ActionCommand/ActionCommand.cpp File Reference

Include dependency graph for ActionCommand.cpp:



Functions

- void [SleepyWrapper](#) (int milliseconds)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap, const unsigned int camNum)
- int [ConfigureInterface](#) (const [InterfaceList](#) &interfaceList)
- int [ConfigureIEEE1588](#) (const [CameraList](#) &camList)
- int [ConfigureActionControl](#) (const [CameraList](#) &camList)
- int [ConfigureOtherNodes](#) (const [CameraList](#) &camList)
- int [ConfigureTrigger](#) (const [CameraList](#) camList)
- int [ConfigureChunkData](#) (const [CameraList](#) &camList)
- int [AcquireImages](#) (const [SystemPtr](#) &system, const [InterfaceList](#) &interfaceList, [CameraList](#) camList)
- int [RunMultipleCameras](#) (const [SystemPtr](#) &system, const [InterfaceList](#) &interfaceList, const [CameraList](#) &camList)
- int [main](#) (int, char **)

16.241.1 Function Documentation

16.241.1.1 AcquireImages()

```
int AcquireImages (
    const SystemPtr & system,
    const InterfaceList & interfaceList,
    CameraList camList )
```

16.241.1.2 ConfigureActionControl()

```
int ConfigureActionControl (
    const CameraList & camList )
```

16.241.1.3 ConfigureChunkData()

```
int ConfigureChunkData (
    const CameraList & camList )
```

16.241.1.4 ConfigureIEEE1588()

```
int ConfigureIEEE1588 (
    const CameraList & camList )
```

16.241.1.5 ConfigureInterface()

```
int ConfigureInterface (
    const InterfaceList & interfaceList )
```

16.241.1.6 ConfigureOtherNodes()

```
int ConfigureOtherNodes (
    const CameraList & camList )
```

16.241.1.7 ConfigureTrigger()

```
int ConfigureTrigger (
    const CameraList camList )
```

16.241.1.8 main()

```
int main (
    int ,
    char ** )
```

16.241.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap,
    const unsigned int camNum )
```

16.241.1.10 RunMultipleCameras()

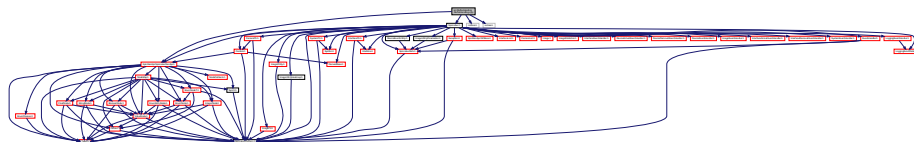
```
int RunMultipleCameras (
    const SystemPtr & system,
    const InterfaceList & interfaceList,
    const CameraList & camList )
```

16.241.1.11 SleepyWrapper()

```
void SleepyWrapper (
    int milliseconds )
```

16.242 src/BufferHandling/BufferHandling.cpp File Reference

Include dependency graph for BufferHandling.cpp:



Macros

- `#define numBuffers 3`
- `#define z_numTriggers 6`
- `#define k_numLoops 9`

Functions

- `void SleepyWrapper` (int milliseconds)
- `int ConfigureTrigger` (INodeMap &nodeMap)
- `int GrabNextImageByTrigger` (INodeMap &nodeMap)
- `int ResetTrigger` (INodeMap &nodeMap)
- `int PrintDeviceInfo` (INodeMap &nodeMap)
- `int AcquireImages` (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- `int RunSingleCamera` (CameraPtr pCam)
- `int main` (int, char **)

16.242.1 Macro Definition Documentation

16.242.1.1 k_numLoops

```
#define k_numLoops 9
```

16.242.1.2 numBuffers

```
#define numBuffers 3
```

16.242.1.3 z_numTriggers

```
#define z_numTriggers 6
```

16.242.2 Function Documentation

16.242.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.242.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.242.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
    INodeMap & nodeMap )
```

16.242.2.4 main()

```
int main (
    int ,
    char ** )
```

16.242.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.242.2.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.242.2.7 RunSingleCamera()

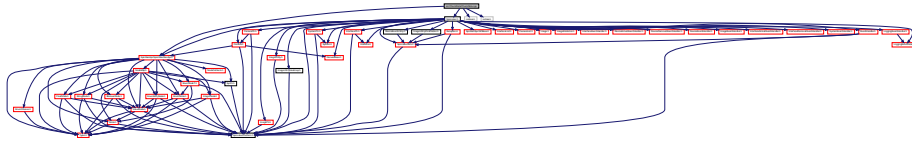
```
int RunSingleCamera (
    CameraPtr pCam )
```

16.242.2.8 SleepyWrapper()

```
void SleepyWrapper (
    int milliseconds )
```

16.243 src/ChunkData/ChunkData.cpp File Reference

Include dependency graph for ChunkData.cpp:



Enumerations

- enum `chunkDataType` {
 `IMAGE`,
 `NODEMAP` }

Functions

- int `ConfigureChunkData` (INodeMap &nodeMap)
- int `DisplayChunkData` (ImagePtr pImage)
- int `DisplayChunkData` (INodeMap &nodeMap)
- int `PrintDeviceInfo` (INodeMap &nodeMap)
- int `AcquireImages` (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int `DisableChunkData` (INodeMap &nodeMap)
- int `RunSingleCamera` (CameraPtr pCam)
- int `main` (int, char **)

Variables

- const `chunkDataType` `chosenChunkData` = `IMAGE`

16.243.1 Enumeration Type Documentation

16.243.1.1 chunkDataType

enum `chunkDataType`

Enumerator

<code>IMAGE</code>	
<code>NODEMAP</code>	

16.243.2 Function Documentation

16.243.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.243.2.2 ConfigureChunkData()

```
int ConfigureChunkData (
    INodeMap & nodeMap )
```

16.243.2.3 DisableChunkData()

```
int DisableChunkData (
    INodeMap & nodeMap )
```

16.243.2.4 DisplayChunkData() [1/2]

```
int DisplayChunkData (
    ImagePtr pImage )
```

16.243.2.5 DisplayChunkData() [2/2]

```
int DisplayChunkData (
    INodeMap & nodeMap )
```

16.243.2.6 main()

```
int main (
    int ,
    char ** )
```

16.243.2.7 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.243.2.8 RunSingleCamera()

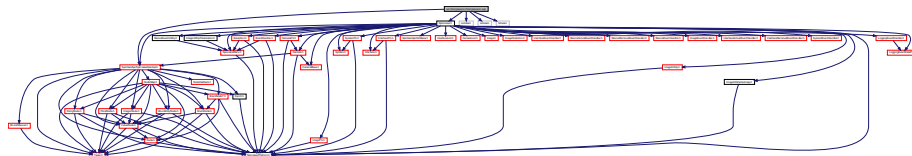
```
int RunSingleCamera (
    CameraPtr pCam )
```

16.243.3 Variable Documentation**16.243.3.1 chosenChunkData**

```
const chunkDataType chosenChunkData = IMAGE
```

16.244 src/Compression/Compression.cpp File Reference

Include dependency graph for Compression.cpp:

**Classes**

- struct [CompressedImageInfo](#)

Functions

- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice, vector< [CompressedImageInfo](#) > &compressedImageInfos)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- bool [EnableImageChunkData](#) (INodeMap &nodeMap)
- bool [DisableImageChunkData](#) (INodeMap &nodeMap)
- bool [EnableImageCompression](#) (INodeMap &nodeMap)
- bool [DisableImageCompression](#) (INodeMap &nodeMap)
- int [ProcessCompressedImagesFromFile](#) (const vector< [CompressedImageInfo](#) > &compressedImageInfos)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

Variables

- const bool `enableChunkData` = false

16.244.1 Function Documentation

16.244.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    vector< CompressedImageInfo > & compressedImageInfos )
```

16.244.1.2 DisableImageChunkData()

```
bool DisableImageChunkData (
    INodeMap & nodeMap )
```

16.244.1.3 DisableImageCompression()

```
bool DisableImageCompression (
    INodeMap & nodeMap )
```

16.244.1.4 EnableImageChunkData()

```
bool EnableImageChunkData (
    INodeMap & nodeMap )
```

16.244.1.5 EnableImageCompression()

```
bool EnableImageCompression (
    INodeMap & nodeMap )
```

16.244.1.6 main()

```
int main (
    int ,
    char ** )
```

16.244.1.7 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.244.1.8 ProcessCompressedImagesFromFile()

```
int ProcessCompressedImagesFromFile (
    const vector< CompressedImageInfo > & compressedImageInfos )
```

16.244.1.9 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

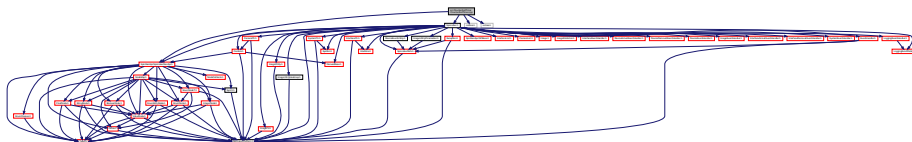
16.244.2 Variable Documentation

16.244.2.1 enableChunkData

```
const bool enableChunkData = false
```

16.245 src/CounterAndTimer/CounterAndTimer.cpp File Reference

Include dependency graph for CounterAndTimer.cpp:



Functions

- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [SetupCounterAndTimer](#) (INodeMap &nodeMap)
- int [ConfigureDigitalIO](#) (INodeMap &nodeMap)
- int [ConfigureExposureandTrigger](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [ResetTrigger](#) (INodeMap &nodeMap)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.245.1 Function Documentation

16.245.1.1 [AcquireImages\(\)](#)

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.245.1.2 [ConfigureDigitalIO\(\)](#)

```
int ConfigureDigitalIO (
    INodeMap & nodeMap )
```

16.245.1.3 [ConfigureExposureandTrigger\(\)](#)

```
int ConfigureExposureandTrigger (
    INodeMap & nodeMap )
```

16.245.1.4 [main\(\)](#)

```
int main (
    int ,
    char ** )
```

16.245.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.245.1.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.245.1.7 RunSingleCamera()

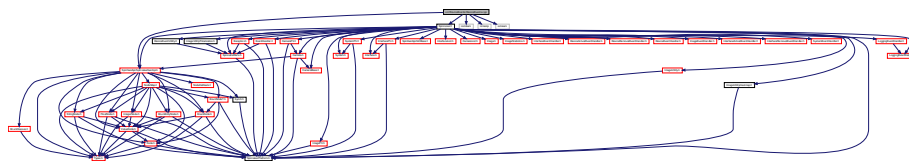
```
int RunSingleCamera (
    CameraPtr pCam )
```

16.245.1.8 SetupCounterAndTimer()

```
int SetupCounterAndTimer (
    INodeMap & nodeMap )
```

16.246 src/DeviceEvents/DeviceEvents.cpp File Reference

Include dependency graph for DeviceEvents.cpp:

**Classes**

- class [DeviceEventHandlerImpl](#)

Enumerations

- enum [eventType](#) {
[GENERIC](#),
[SPECIFIC](#) }

Functions

- int [ConfigureDeviceEvents](#) (INodeMap &nodeMap, [CameraPtr](#) pCam, [DeviceEventHandlerImpl](#) *&deviceEvent↵
EventHandler, const [gcstring](#) &deviceEventName)
- int [ResetDeviceEvents](#) ([CameraPtr](#) pCam, [DeviceEventHandlerImpl](#) *&deviceEventHandler)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- bool [InferenceAvailable](#) (INodeMap &nodeMap)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

Variables

- const [eventType](#) [chosenEvent](#) = [GENERIC](#)

16.246.1 Enumeration Type Documentation

16.246.1.1 eventType

enum [eventType](#)

Enumerator

GENERIC	
SPECIFIC	

16.246.2 Function Documentation

16.246.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.246.2.2 ConfigureDeviceEvents()

```
int ConfigureDeviceEvents (
    INodeMap & nodeMap,
    CameraPtr pCam,
    DeviceEventHandlerImpl *& deviceEventHandler,
    const gcstring & deviceEventName )
```

16.246.2.3 InferenceAvailable()

```
bool InferenceAvailable (
    INodeMap & nodeMap )
```

16.246.2.4 main()

```
int main (
    int ,
    char ** )
```

16.246.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.246.2.6 ResetDeviceEvents()

```
int ResetDeviceEvents (
    CameraPtr pCam,
    DeviceEventHandlerImpl *& deviceEventHandler )
```

16.246.2.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

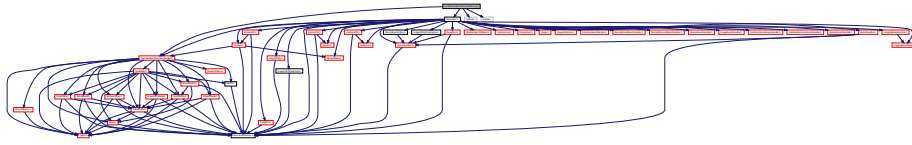
16.246.3 Variable Documentation

16.246.3.1 chosenEvent

```
const eventType chosenEvent = GENERIC
```


16.247 src/Enumeration/Enumeration.cpp File Reference

Include dependency graph for Enumeration.cpp:



Functions

- int [QueryInterface](#) ([InterfacePtr](#) pInterface)
- int [main](#) (int, char **)

16.247.1 Function Documentation

16.247.1.1 main()

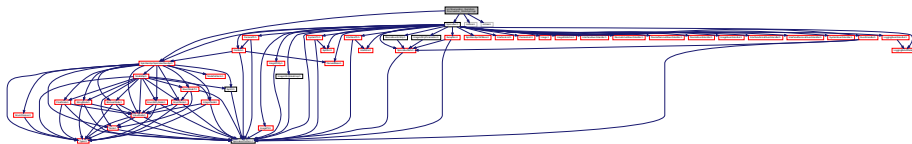
```
int main (
    int ,
    char ** )
```

16.247.1.2 QueryInterface()

```
int QueryInterface (
    InterfacePtr pInterface )
```

16.248 src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp File Reference

Include dependency graph for Enumeration_QuickSpin.cpp:



Functions

- int [QueryInterface](#) ([InterfacePtr](#) pInterface)
- int [main](#) (int, char **)

16.248.1 Function Documentation

16.248.1.1 main()

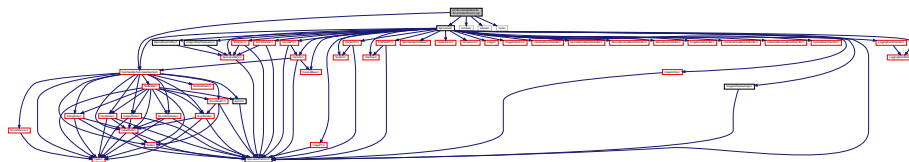
```
int main (
    int ,
    char ** )
```

16.248.1.2 QueryInterface()

```
int QueryInterface (
    InterfacePtr pInterface )
```

16.249 src/EnumerationEvents/EnumerationEvents.cpp File Reference

Include dependency graph for EnumerationEvents.cpp:



Classes

- class [InterfaceEventHandlerImpl](#)
- class [SystemEventHandlerImpl](#)

Functions

- void [CheckGevEnabled](#) ([SystemPtr](#) &pSystem)
- int [main](#) (int, char **)

16.249.1 Function Documentation

16.249.1.1 CheckGevEnabled()

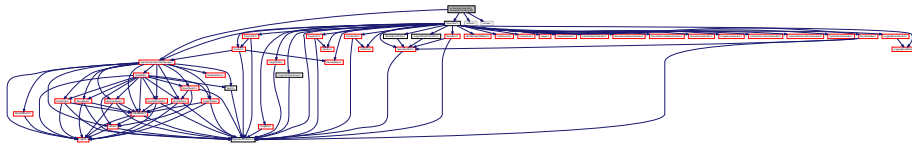
```
void CheckGevEnabled (
    SystemPtr & pSystem )
```

16.249.1.2 main()

```
int main (
    int ,
    char ** )
```

16.250 src/ExceptionHandling/ExceptionHandling.cpp File Reference

Include dependency graph for ExceptionHandling.cpp:

**Enumerations**

- enum `exceptionType` {
 SPINNAKER_EXCEPTION,
 STANDARD_EXCEPTION,
 STANDARD_CAST_TO_SPINNAKER }

Functions

- void `causeSpinnakerException` ()
- void `causeStandardException` ()
- int `main` (int, char **)

Variables

- const `exceptionType chosenException` = SPINNAKER_EXCEPTION

16.250.1 Enumeration Type Documentation**16.250.1.1 exceptionType**

```
enum exceptionType
```

Enumerator

SPINNAKER_EXCEPTION	
STANDARD_EXCEPTION	
STANDARD_CAST_TO_SPINNAKER	

16.250.2 Function Documentation

16.250.2.1 causeSpinnakerException()

```
void causeSpinnakerException ( )
```

16.250.2.2 causeStandardException()

```
void causeStandardException ( )
```

16.250.2.3 main()

```
int main (
    int ,
    char ** )
```

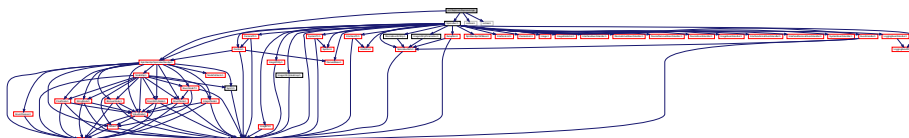
16.250.3 Variable Documentation

16.250.3.1 chosenException

```
const exceptionType chosenException = SPINNAKER_EXCEPTION
```

16.251 src/Exposure/Exposure.cpp File Reference

Include dependency graph for Exposure.cpp:



Functions

- int [ConfigureExposure](#) (INodeMap &nodeMap)
- int [ResetExposure](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.251.1 Function Documentation

16.251.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.251.1.2 ConfigureExposure()

```
int ConfigureExposure (
    INodeMap & nodeMap )
```

16.251.1.3 main()

```
int main (
    int ,
    char ** )
```

16.251.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.251.1.5 ResetExposure()

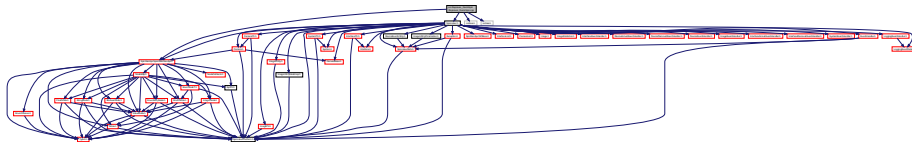
```
int ResetExposure (
    INodeMap & nodeMap )
```

16.251.1.6 RunSingleCamera()

```
int RunSingleCamera (  
    CameraPtr pCam )
```

16.252 src/Exposure_QuickSpin/Exposure_QuickSpin.cpp File Reference

Include dependency graph for Exposure_QuickSpin.cpp:



Functions

- int [ConfigureExposure](#) ([CameraPtr](#) pCam)
- int [ResetExposure](#) ([CameraPtr](#) pCam)
- int [PrintDeviceInfo](#) ([CameraPtr](#) pCam)
- int [AcquireImages](#) ([CameraPtr](#) pCam)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.252.1 Function Documentation

16.252.1.1 AcquireImages()

```
int AcquireImages (  
    CameraPtr pCam )
```

16.252.1.2 ConfigureExposure()

```
int ConfigureExposure (  
    CameraPtr pCam )
```

16.252.1.3 main()

```
int main (
    int ,
    char ** )
```

16.252.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    CameraPtr pCam )
```

16.252.1.5 ResetExposure()

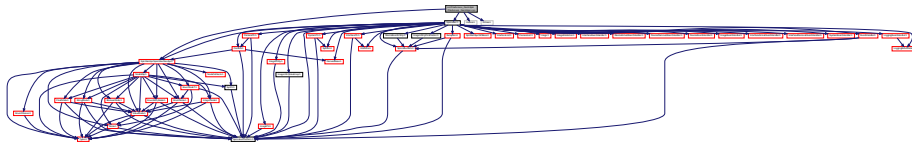
```
int ResetExposure (
    CameraPtr pCam )
```

16.252.1.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.253 src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp File Reference

Include dependency graph for FileAccess_QuickSpin.cpp:

**Functions**

- static void `PrintResultMessage` (bool result)
- int `PrintDeviceInfo` (INodeMap &nodeMap)
- bool `InitializeSystem` (SystemPtr &system, CameraList &camList, CameraPtr &pCam)
- static void `PrintDebugMessage` (string msg)
- bool `AcquireImages` (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice, ImagePtr pReferenceImage)
- bool `ExecuteDeleteCommand` (CameraPtr pCam)
- bool `OpenFileToWrite` (CameraPtr pCam)
- bool `ExecuteWriteCommand` (CameraPtr pCam)
- bool `CloseFile` (CameraPtr pCam)
- bool `UploadImage` ()
- bool `OpenFileToRead` (CameraPtr pCamera)
- bool `ExecuteReadCommand` (CameraPtr pCamera)
- bool `DownloadImage` ()
- void `PrintUsage` ()
- int `main` (int argc, char *argv[])

Variables

- static bool `_enableDebug` = false
- static `gcstring` `_fileSelector` = "UserFile1"

16.253.1 Function Documentation

16.253.1.1 AcquireImages()

```
bool AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    ImagePtr pReferenceImage )
```

16.253.1.2 CloseFile()

```
bool CloseFile (
    CameraPtr pCam )
```

16.253.1.3 DownloadImage()

```
bool DownloadImage ( )
```

16.253.1.4 ExecuteDeleteCommand()

```
bool ExecuteDeleteCommand (
    CameraPtr pCam )
```

16.253.1.5 ExecuteReadCommand()

```
bool ExecuteReadCommand (
    CameraPtr pCamera )
```


16.253.1.6 ExecuteWriteCommand()

```
bool ExecuteWriteCommand (
    CameraPtr pCam )
```

16.253.1.7 InitializeSystem()

```
bool InitializeSystem (
    SystemPtr & system,
    CameraList & camList,
    CameraPtr & pCam )
```

16.253.1.8 main()

```
int main (
    int argc,
    char * argv[] )
```

16.253.1.9 OpenFileToRead()

```
bool OpenFileToRead (
    CameraPtr pCamera )
```

16.253.1.10 OpenFileToWrite()

```
bool OpenFileToWrite (
    CameraPtr pCam )
```

16.253.1.11 PrintDebugMessage()

```
static void PrintDebugMessage (
    string msg ) [static]
```

16.253.1.12 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.253.1.13 PrintResultMessage()

```
static void PrintResultMessage (
    bool result ) [static]
```

16.253.1.14 PrintUsage()

```
void PrintUsage ( )
```

16.253.1.15 UploadImage()

```
bool UploadImage ( )
```

16.253.2 Variable Documentation**16.253.2.1 _enableDebug**

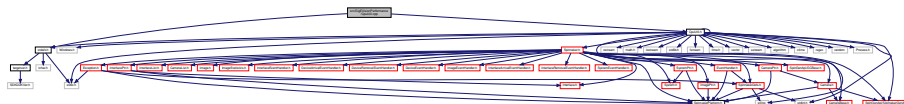
```
bool _enableDebug = false [static]
```

16.253.2.2 _fileSelector

```
gcstring _fileSelector = "UserFile1" [static]
```

16.254 src/GigEVisionPerformance/CpuUtil.cpp File Reference

Include dependency graph for CpuUtil.cpp:



Namespaces

- [CpuUtil](#)
- [PerformanceCounter](#)
- [SecondsCounter](#)
- [Conversion](#)

Functions

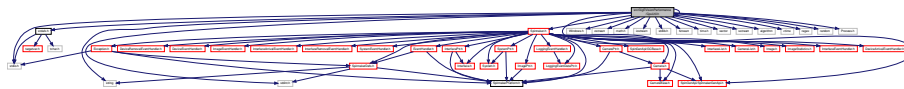
- bool [StartCpuTracing](#) (CpuUsageInfo *cpuUsage)
- bool [StopCpuTracing](#) (CpuUsageInfo *cpuUsage)
- std::string [GetCpuStats](#) (CpuUsageInfo *cpuUsage)
- void [StartPerformanceCounter](#) ()
- double [GetPerformanceCounter](#) ()
- void [StartSecondsCounter](#) ()
- int [GetSecondsCounter](#) ()
- string [NumToCString](#) (int number)
- string [NumToCString](#) (double number)
- string [NumToCString](#) (float number)

Variables

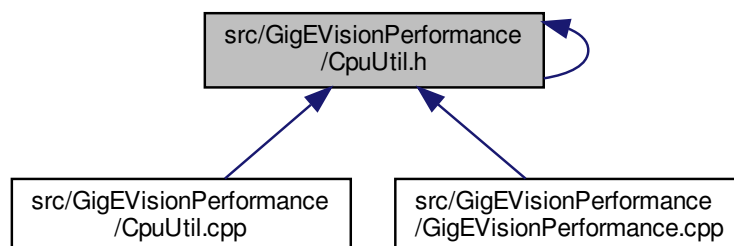
- double [PCFreq](#)
- __int64 [CounterStart](#)
- time_t [startTime](#)
- time_t [endTime](#)
- double [timeDiff](#)

16.255 src/GigEVisionPerformance/CpuUtil.h File Reference

Include dependency graph for CpuUtil.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [CpuUsageInfo](#)

Namespaces

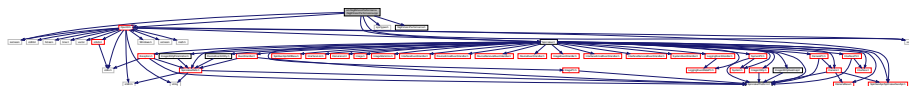
- [CpuUtil](#)
- [PerformanceCounter](#)
- [SecondsCounter](#)
- [Conversion](#)

Functions

- bool [StartCpuTracing](#) (CpuUsageInfo *cpuUsage)
- bool [StopCpuTracing](#) (CpuUsageInfo *cpuUsage)
- std::string [GetCpuStats](#) (CpuUsageInfo *cpuUsage)
- void [StartPerformanceCounter](#) ()
- double [GetPerformanceCounter](#) ()
- void [StartSecondsCounter](#) ()
- int [GetSecondsCounter](#) ()
- string [NumToCString](#) (int number)
- string [NumToCString](#) (double number)

16.256 src/GigEVisionPerformance/GigEVisionPerformance.cpp File Reference

Include dependency graph for GigEVisionPerformance.cpp:



Functions

- void [PrintUsage](#) ()
- bool [ParseArguments](#) (int argc, char *argv[])
- void [getCameraCategory](#) (INodeMap &nodeMap, string categoryString)
- void [PrintDataStreamInfo](#) (const [Spinnaker::CameraPtr](#) pCamera)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapGenTL, int numImages, ToAcquire, int iteration)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- void [PrintCPUUsage](#) ()
- void [PrintAllNodes](#) ([CameraPtr](#) pCam)
- bool [EnableManualFramerate](#) ([CameraPtr](#) pCam)
- bool [SetFrameRate](#) ([CameraPtr](#) pCam)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int argc, char *argv[])

Variables

- [CpuUtil::CpuUsageInfo](#) `cpuUsageInfo`
- `int` `TestDuration` = 0
- `char *` `PixelFormatToSet` = `nullptr`
- `int` `PacketSizeToSet` = 9000
- `int` `PacketDelayToSet` = 0
- `bool` `IsRelease` = false
- `bool` `UseDuration` = false
- `bool` `UseMaxFramerate` = false
- `float` `UserSetFramerate` = 0.0
- `int` `NumImagesToGrab` = 100
- `const char *` `argNumImages` = "-numimages"
- `const char *` `argDuration` = "-duration"
- `const char *` `argRelease` = "-callrelease"
- `const char *` `argBayerRG` = "-bayerrg"
- `const char *` `argPacketSize` = "-packetsize"
- `const char *` `argPacketDelay` = "-packetdelay"
- `const char *` `argMaxFrames` = "-maxfps"
- `const char *` `argUserSetFrames` = "-fps"
- `const char *` `argPrintUsage` = "-?"

16.256.1 Function Documentation

16.256.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapGenTL,
    int numImagesToAcquire,
    int iteration )
```

16.256.1.2 EnableManualFramerate()

```
bool EnableManualFramerate (
    CameraPtr pCam )
```

16.256.1.3 getCameraCategory()

```
void getCameraCategory (
    INodeMap & nodeMap,
    string categoryString )
```

16.256.1.4 main()

```
int main (
    int argc,
    char * argv[] )
```

16.256.1.5 ParseArguments()

```
bool ParseArguments (
    int argc,
    char * argv[] )
```

16.256.1.6 PrintAllNodes()

```
void PrintAllNodes (
    CameraPtr pCam )
```

16.256.1.7 PrintCPUUsage()

```
void PrintCPUUsage ( )
```

16.256.1.8 PrintDataStreamInfo()

```
void PrintDataStreamInfo (
    const Spinnaker::CameraPtr pCamera )
```

16.256.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.256.1.10 PrintUsage()

```
void PrintUsage ( )
```

16.256.1.11 RunSingleCamera()

```
int RunSingleCamera (  
    CameraPtr pCam )
```

16.256.1.12 SetFrameRate()

```
bool SetFrameRate (  
    CameraPtr pCam )
```

16.256.2 Variable Documentation

16.256.2.1 argBayerRG

```
const char* argBayerRG = "-bayerrg"
```

16.256.2.2 argDuration

```
const char* argDuration = "-duration"
```

16.256.2.3 argMaxFrames

```
const char* argMaxFrames = "-maxfps"
```

16.256.2.4 argNumImages

```
const char* argNumImages = "-numimages"
```

16.256.2.5 argPacketDelay

```
const char* argPacketDelay = "-packetdelay"
```

16.256.2.6 argPacketSize

```
const char* argPacketSize = "-packetsize"
```

16.256.2.7 argPrintUsage

```
const char* argPrintUsage = "-?"
```

16.256.2.8 argRelease

```
const char* argRelease = "-callrelease"
```

16.256.2.9 argUserSetFrames

```
const char* argUserSetFrames = "-fps"
```

16.256.2.10 cpuUsageInfo

```
CpuUtil::CpuUsageInfo cpuUsageInfo
```

16.256.2.11 IsRelease

```
bool IsRelease = false
```

16.256.2.12 NumImagesToGrab

```
int NumImagesToGrab = 100
```

16.256.2.13 PacketDelayToSet

```
int PacketDelayToSet = 0
```


16.256.2.14 PacketSizeToSet

```
int PacketSizeToSet = 9000
```

16.256.2.15 PixelFormatToSet

```
char* PixelFormatToSet = nullptr
```

16.256.2.16 TestDuration

```
int TestDuration = 0
```

16.256.2.17 UseDuration

```
bool UseDuration = false
```

16.256.2.18 UseMaxFramerate

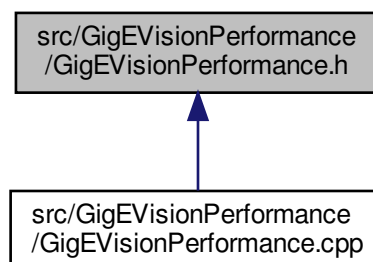
```
bool UseMaxFramerate = false
```

16.256.2.19 UserSetFramerate

```
float UserSetFramerate = 0.0
```

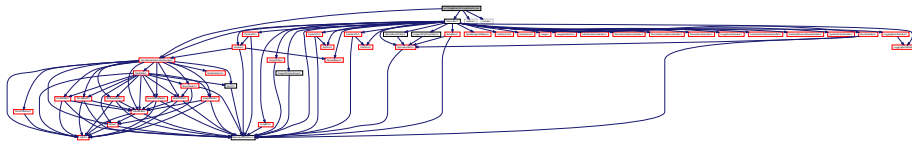
16.257 src/GigEVisionPerformance/GigEVisionPerformance.h File Reference

This graph shows which files directly or indirectly include this file:



16.258 src/ImageEvents/ImageEvents.cpp File Reference

Include dependency graph for ImageEvents.cpp:



Classes

- class [ImageEventHandlerImpl](#)

Functions

- void [SleepyWrapper](#) (int milliseconds)
- int [ConfigureImageEvents](#) ([CameraPtr](#) pCam, [ImageEventHandlerImpl](#) *&imageEventHandler)
- int [WaitForImages](#) ([ImageEventHandlerImpl](#) *&imageEventHandler)
- int [ResetImageEvents](#) ([CameraPtr](#) pCam, [ImageEventHandlerImpl](#) *&imageEventHandler)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice, [ImageEventHandlerImpl](#) *&imageEventHandler)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.258.1 Function Documentation

16.258.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    ImageEventHandlerImpl *& imageEventHandler )
```

16.258.1.2 ConfigureImageEvents()

```
int ConfigureImageEvents (
    CameraPtr pCam,
    ImageEventHandlerImpl *& imageEventHandler )
```

16.258.1.3 main()

```
int main (
    int ,
    char ** )
```

16.258.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.258.1.5 ResetImageEvents()

```
int ResetImageEvents (
    CameraPtr pCam,
    ImageEventHandlerImpl *& imageEventHandler )
```

16.258.1.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.258.1.7 SleepyWrapper()

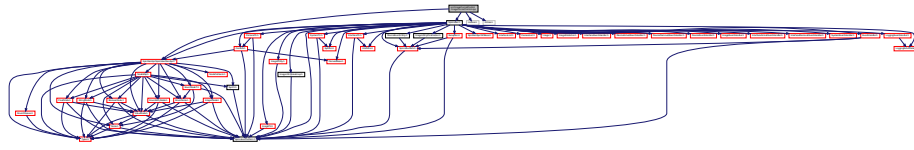
```
void SleepyWrapper (
    int milliseconds )
```

16.258.1.8 WaitForImages()

```
int WaitForImages (
    ImageEventHandlerImpl *& imageEventHandler )
```

16.259 src/ImageFormatControl/ImageFormatControl.cpp File Reference

Include dependency graph for ImageFormatControl.cpp:



Functions

- int [ConfigureCustomImageSettings](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.259.1 Function Documentation

16.259.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.259.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (
    INodeMap & nodeMap )
```

16.259.1.3 main()

```
int main (
    int ,
    char ** )
```

16.259.1.4 PrintDeviceInfo()

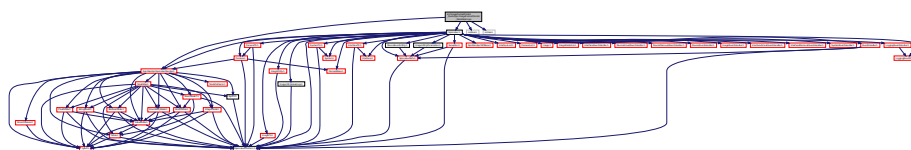
```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.259.1.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.260 src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp File Reference

Include dependency graph for ImageFormatControl_QuickSpin.cpp:



Functions

- int [ConfigureCustomImageSettings](#) (CameraPtr pCam)
- int [PrintDeviceInfo](#) (CameraPtr pCam)
- int [AcquireImages](#) (CameraPtr pCam)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.260.1 Function Documentation

16.260.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam )
```

16.260.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (
    CameraPtr pCam )
```

16.260.1.3 main()

```
int main (
    int ,
    char ** )
```

16.260.1.4 PrintDeviceInfo()

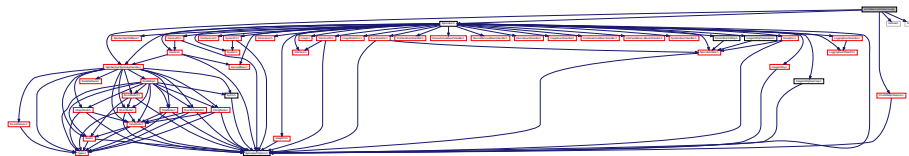
```
int PrintDeviceInfo (
    CameraPtr pCam )
```

16.260.1.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.261 src/Inference/Inference.cpp File Reference

Include dependency graph for Inference.cpp:



Enumerations

- enum [InferenceNetworkType](#) {
DETECTION,
CLASSIFICATION }
- enum [FileUploadPersistence](#) {
FLASH,
DDR }

Functions

- `const std::vector< std::string > labelClassification (arrayLabelClassification, end(arrayLabelClassification))`
- `const std::vector< std::string > labelDetection (arrayLabelDetection, end(arrayLabelDetection))`
- `int PrintDeviceInfo (INodeMap &nodeMap)`
- `bool CameraDeleteFile (INodeMap &nodeMap)`
- `bool CameraOpenFile (INodeMap &nodeMap)`
- `bool CameraWriteToFile (INodeMap &nodeMap)`
- `bool CameraCloseFile (INodeMap &nodeMap)`
- `std::vector< char > LoadFileIntoMemory (const string &filename)`
- `int UploadFileToCamera (INodeMap &nodeMap, const std::string &fileSelectorEntryName, const std::string &filePath)`
- `int DeleteFileOnCamera (INodeMap &nodeMap, const std::string &fileSelectorEntryName)`
- `int SetChunkEnable (INodeMap &nodeMap, const gcstring &entryName, const bool enable)`
- `int ConfigureChunkData (INodeMap &nodeMap)`
- `int DisableChunkData (INodeMap &nodeMap)`
- `int DisplayChunkData (const ImagePtr pImage)`
- `int DisableTrigger (INodeMap &nodeMap)`
- `int ConfigureTrigger (INodeMap &nodeMap)`
- `int ConfigureInference (INodeMap &nodeMap, bool isEnabled)`
- `int ConfigureTestPattern (INodeMap &nodeMap, bool isEnabled)`
- `int AcquireImages (const CameraPtr &pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)`
- `int RunSingleCamera (const CameraPtr &pCam)`
- `int main (int, char **)`

Variables

- `const InferenceNetworkType chosenInferenceNetworkType = CLASSIFICATION`
- `const FileUploadPersistence chosenFileUploadPersistence = DDR`
- `const std::string networkFilePath`
- `const std::string injectedImageFilePath`
- `const unsigned int injectedImageWidth = (chosenInferenceNetworkType == CLASSIFICATION ? 640 : 720)`
- `const unsigned int injectedImageHeight = (chosenInferenceNetworkType == CLASSIFICATION ? 400 : 540)`
- `const char * arrayLabelClassification [] = {"daisy", "dandelion", "roses", "sunflowers", "tulips"}`
- `const char * arrayLabelDetection []`

16.261.1 Enumeration Type Documentation

16.261.1.1 FileUploadPersistence

enum `FileUploadPersistence`

Enumerator

FLASH	
DDR	

16.261.1.2 InferenceNetworkType

enum [InferenceNetworkType](#)

Enumerator

DETECTION	This network determines the most likely class given a set of predetermined, trained options. Object detection can also provide a location within the image (in the form of a "bounding box" surrounding the class), and can detect multiple objects.
CLASSIFICATION	This network determines the best option from a list of predetermined options; the camera gives a percentage that determines the likelihood of the currently perceived image being one of the classes it has been trained to recognize.

16.261.2 Function Documentation

16.261.2.1 AcquireImages()

```
int AcquireImages (
    const CameraPtr & pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.261.2.2 CameraCloseFile()

```
bool CameraCloseFile (
    INodeMap & nodeMap )
```

16.261.2.3 CameraDeleteFile()

```
bool CameraDeleteFile (
    INodeMap & nodeMap )
```

16.261.2.4 CameraOpenFile()

```
bool CameraOpenFile (
    INodeMap & nodeMap )
```


16.261.2.5 CameraWriteToFile()

```
bool CameraWriteToFile (
    INodeMap & nodeMap )
```

16.261.2.6 ConfigureChunkData()

```
int ConfigureChunkData (
    INodeMap & nodeMap )
```

16.261.2.7 ConfigureInference()

```
int ConfigureInference (
    INodeMap & nodeMap,
    bool isEnabled )
```

16.261.2.8 ConfigureTestPattern()

```
int ConfigureTestPattern (
    INodeMap & nodeMap,
    bool isEnabled )
```

16.261.2.9 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.261.2.10 DeleteFileOnCamera()

```
int DeleteFileOnCamera (
    INodeMap & nodeMap,
    const std::string & fileSelectorEntryName )
```

16.261.2.11 DisableChunkData()

```
int DisableChunkData (
    INodeMap & nodeMap )
```

16.261.2.12 DisableTrigger()

```
int DisableTrigger (
    INodeMap & nodeMap )
```

16.261.2.13 DisplayChunkData()

```
int DisplayChunkData (
    const ImagePtr pImage )
```

16.261.2.14 labelClassification()

```
const std::vector<std::string> labelClassification (
    arrayLabelClassification ,
    end(arrayLabelClassification) )
```

16.261.2.15 labelDetection()

```
const std::vector<std::string> labelDetection (
    arrayLabelDetection ,
    end(arrayLabelDetection) )
```

16.261.2.16 LoadFileIntoMemory()

```
std::vector<char> LoadFileIntoMemory (
    const string & filename )
```

16.261.2.17 main()

```
int main (
    int ,
    char ** )
```

16.261.2.18 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.261.2.19 RunSingleCamera()

```
int RunSingleCamera (
    const CameraPtr & pCam )
```

16.261.2.20 SetChunkEnable()

```
int SetChunkEnable (
    INodeMap & nodeMap,
    const gcstring & entryName,
    const bool enable )
```

16.261.2.21 UploadFileToCamera()

```
int UploadFileToCamera (
    INodeMap & nodeMap,
    const std::string & fileSelectorEntryName,
    const std::string & filePath )
```

16.261.3 Variable Documentation

16.261.3.1 arrayLabelClassification

```
const char* arrayLabelClassification[] = {"daisy", "dandelion", "roses", "sunflowers", "tulips"}
```

16.261.3.2 arrayLabelDetection

```
const char* arrayLabelDetection[]
```

Initial value:

```
= { "background", "aeroplane", "bicycle", "bird", "boat", "bottle",
    "bus", "car", "cat", "chair", "cow", "diningtable",
    "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
    "sofa", "train", "monitor" }
```

16.261.3.3 chosenFileUploadPersistence

```
const FileUploadPersistence chosenFileUploadPersistence = DDR
```

16.261.3.4 chosenInferenceNetworkType

```
const InferenceNetworkType chosenInferenceNetworkType = CLASSIFICATION
```

16.261.3.5 injectedImageFilePath

```
const std::string injectedImageFilePath
```

Initial value:

```
=
(chosenInferenceNetworkType == CLASSIFICATION ? "
    Injected_Image_Classification.raw"
    : "Injected_Image_Detection.raw")
```

16.261.3.6 injectedImageHeight

```
const unsigned int injectedImageHeight = (chosenInferenceNetworkType == CLASSIFICATION ? 400
: 540)
```

16.261.3.7 injectedImageWidth

```
const unsigned int injectedImageWidth = (chosenInferenceNetworkType == CLASSIFICATION ? 640 :
720)
```

16.261.3.8 networkFilePath

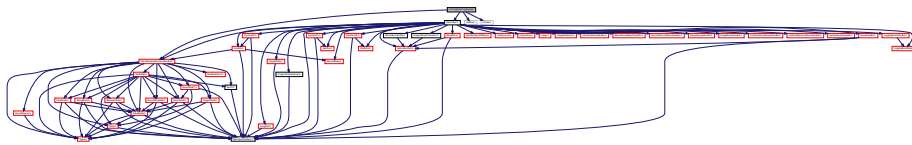
```
const std::string networkFilePath
```

Initial value:

```
=
(chosenInferenceNetworkType == CLASSIFICATION ? "
  Network_Classification" : "Network_Detection")
```

16.262 src/Logging/Logging.cpp File Reference

Include dependency graph for Logging.cpp:



Classes

- class [LoggingEventHandlerImpl](#)

Functions

- int [main](#) (int, char **)

Variables

- const [SpinnakerLogLevel](#) [k_LoggingLevel](#) = LOG_LEVEL_DEBUG

16.262.1 Function Documentation

16.262.1.1 main()

```
int main (
    int ,
    char ** )
```

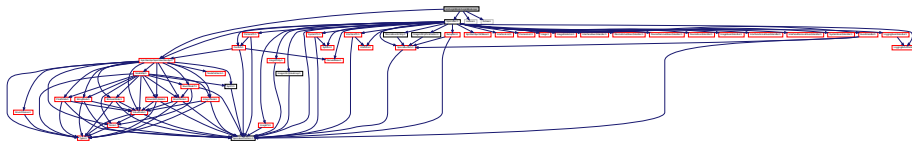
16.262.2 Variable Documentation

16.262.2.1 k_LoggingLevel

```
const SpinnakerLogLevel k_LoggingLevel = LOG_LEVEL_DEBUG
```

16.263 src/LogicBlock/LogicBlock.cpp File Reference

Include dependency graph for LogicBlock.cpp:



Functions

- int [ConfigureTrigger](#) (INodeMap &nodeMap)
- int [ConfigureLogicBlock](#) (INodeMap &nodeMap)
- int [GrabTwoImages](#) (INodeMap &nodeMap, [CameraPtr](#) pCam)
- int [ResetTrigger](#) (INodeMap &nodeMap)
- int [ResetExposure](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.263.1 Function Documentation

16.263.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.263.1.2 ConfigureLogicBlock()

```
int ConfigureLogicBlock (
    INodeMap & nodeMap )
```

16.263.1.3 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.263.1.4 GrabTwoImages()

```
int GrabTwoImages (
    INodeMap & nodeMap,
    CameraPtr pCam )
```

16.263.1.5 main()

```
int main (
    int ,
    char ** )
```

16.263.1.6 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.263.1.7 ResetExposure()

```
int ResetExposure (
    INodeMap & nodeMap )
```

16.263.1.8 ResetTrigger()

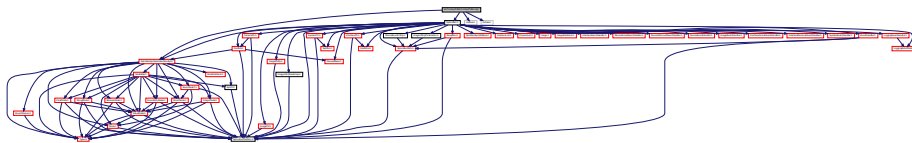
```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.263.1.9 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.264 src/LookupTable/LookupTable.cpp File Reference

Include dependency graph for LookupTable.cpp:



Functions

- void [PrintRetrieveNodeFailure](#) (string node, string name)
- int [ConfigureLookupTables](#) (INodeMap &nodeMap)
- int [ResetLookupTables](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.264.1 Function Documentation

16.264.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.264.1.2 ConfigureLookupTables()

```
int ConfigureLookupTables (
    INodeMap & nodeMap )
```


16.264.1.3 main()

```
int main (
    int ,
    char ** )
```

16.264.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.264.1.5 PrintRetrieveNodeFailure()

```
void PrintRetrieveNodeFailure (
    string node,
    string name )
```

16.264.1.6 ResetLookupTables()

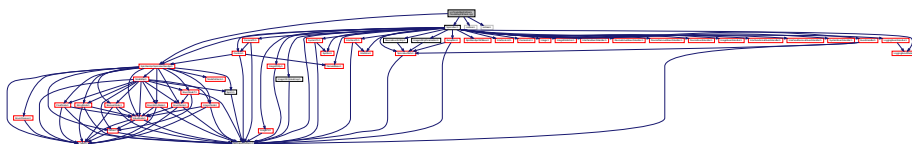
```
int ResetLookupTables (
    INodeMap & nodeMap )
```

16.264.1.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.265 src/NodeMapCallback/NodeMapCallback.cpp File Reference

Include dependency graph for NodeMapCallback.cpp:



Functions

- void [OnHeightNodeUpdate](#) (INode *node)
- void [OnGainNodeUpdate](#) (INode *node)
- int [ConfigureCallbacks](#) (INodeMap &nodeMap, [CallbackHandleType](#) &callbackHeight, [CallbackHandleType](#) &callbackGain)
- int [ChangeHeightAndGain](#) (INodeMap &nodeMap)
- int [ResetCallbacks](#) (INodeMap &nodeMap, [CallbackHandleType](#) callbackHeight, [CallbackHandleType](#) callbackGain)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.265.1 Function Documentation

16.265.1.1 ChangeHeightAndGain()

```
int ChangeHeightAndGain (
    INodeMap & nodeMap )
```

16.265.1.2 ConfigureCallbacks()

```
int ConfigureCallbacks (
    INodeMap & nodeMap,
    CallbackHandleType & callbackHeight,
    CallbackHandleType & callbackGain )
```

16.265.1.3 main()

```
int main (
    int ,
    char ** )
```

16.265.1.4 OnGainNodeUpdate()

```
void OnGainNodeUpdate (
    INode * node )
```

16.265.1.5 OnHeightNodeUpdate()

```
void OnHeightNodeUpdate (
    INode * node )
```

16.265.1.6 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.265.1.7 ResetCallbacks()

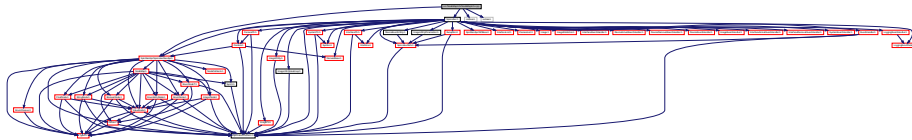
```
int ResetCallbacks (
    INodeMap & nodeMap,
    CallbackHandleType callbackHeight,
    CallbackHandleType callbackGain )
```

16.265.1.8 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.266 src/NodeMapInfo/NodeMapInfo.cpp File Reference

Include dependency graph for NodeMapInfo.cpp:



Enumerations

- enum readType {
 VALUE,
 INDIVIDUAL }

Functions

- int [PrintEnumerationSelector](#) ([CNodePtr](#) node, unsigned int level)
- void [Indent](#) (unsigned int level)
- int [PrintValueNode](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintStringNode](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintIntegerNode](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintFloatNode](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintBooleanNode](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintCommandNode](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintEnumerationNodeAndCurrentEntry](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintNode](#) ([CNodePtr](#) node, unsigned int level)
- int [PrintCategoryNodeAndAllFeatures](#) ([CNodePtr](#) node, unsigned int level)
- int [RunSingleCamera](#) ([CameraPtr](#) cam)
- int [main](#) (int, char **)

Variables

- const unsigned int [maxChars](#) = 35
- const [readType](#) [chosenRead](#) = [VALUE](#)

16.266.1 Enumeration Type Documentation

16.266.1.1 readType

enum [readType](#)

Enumerator

VALUE	
INDIVIDUAL	

16.266.2 Function Documentation

16.266.2.1 Indent()

```
void Indent (  
    unsigned int level )
```

16.266.2.2 main()

```
int main (
    int ,
    char ** )
```

16.266.2.3 PrintBooleanNode()

```
int PrintBooleanNode (
    CNodePtr node,
    unsigned int level )
```

16.266.2.4 PrintCategoryNodeAndAllFeatures()

```
int PrintCategoryNodeAndAllFeatures (
    CNodePtr node,
    unsigned int level )
```

16.266.2.5 PrintCommandNode()

```
int PrintCommandNode (
    CNodePtr node,
    unsigned int level )
```

16.266.2.6 PrintEnumerationNodeAndCurrentEntry()

```
int PrintEnumerationNodeAndCurrentEntry (
    CNodePtr node,
    unsigned int level )
```

16.266.2.7 PrintEnumerationSelector()

```
int PrintEnumerationSelector (
    CNodePtr node,
    unsigned int level )
```

16.266.2.8 PrintFloatNode()

```
int PrintFloatNode (
    CNodePtr node,
    unsigned int level )
```

16.266.2.9 PrintIntegerNode()

```
int PrintIntegerNode (
    CNodePtr node,
    unsigned int level )
```

16.266.2.10 PrintNode()

```
int PrintNode (
    CNodePtr node,
    unsigned int level )
```

16.266.2.11 PrintStringNode()

```
int PrintStringNode (
    CNodePtr node,
    unsigned int level )
```

16.266.2.12 PrintValueNode()

```
int PrintValueNode (
    CNodePtr node,
    unsigned int level )
```

16.266.2.13 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr cam )
```

16.266.3 Variable Documentation

16.266.3.1 chosenRead

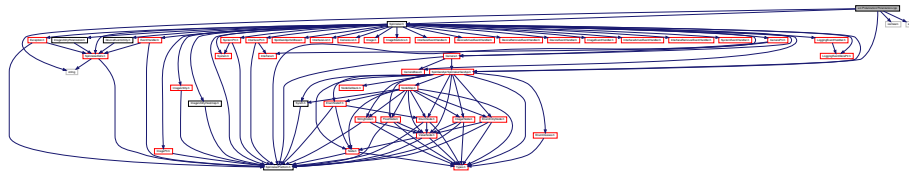
```
const readType chosenRead = VALUE
```

16.266.3.2 maxChars

```
const unsigned int maxChars = 35
```

16.267 src/Polarization/Polarization.cpp File Reference

Include dependency graph for Polarization.cpp:



Functions

- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [ConfigureStream](#) (INodeMap &nodeMap)
- int [SaveImage](#) (const [ImagePtr](#) &pImage, const string filename, [gcstring](#) &serialNumber)
- std::string [GetQuadFileNameAppendage](#) (const [ImageUtilityPolarization::PolarizationQuadrant](#) quadrant)
- int [CreateHeatmapImages](#) (const [ImagePtr](#) &mono8Image, const string baseFilename, [gcstring](#) &deviceSerialNumber)
- int [ExtractAndSavePolarQuadImages](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [CreateAndSaveGlareReducedImage](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [CreateNormalizedImage](#) (const [ImagePtr](#) &imageToNormalize, const string baseFilename, [gcstring](#) &deviceSerialNumber, [ImageUtility::SourceDataRange](#) srcDataRange=[ImageUtility::IMAGE_DATA_RANGE](#))
- int [CreateAndSaveStokesImages](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [CreateAndSaveAolpDolpImages](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

Variables

- static bool [isPixelFormatColor](#) = false

16.267.1 Function Documentation

16.267.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMapTLDevice )
```

16.267.1.2 ConfigureStream()

```
int ConfigureStream (
    INodeMap & nodeMap )
```

16.267.1.3 CreateAndSaveAolpDolpImages()

```
int CreateAndSaveAolpDolpImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.267.1.4 CreateAndSaveGlareReducedImage()

```
int CreateAndSaveGlareReducedImage (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.267.1.5 CreateAndSaveStokesImages()

```
int CreateAndSaveStokesImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.267.1.6 CreateHeatmapImages()

```
int CreateHeatmapImages (
    const ImagePtr & mono8Image,
    const string baseFilename,
    gcstring & deviceSerialNumber )
```


16.267.1.7 CreateNormalizedImage()

```
int CreateNormalizedImage (
    const ImagePtr & imageToNormalize,
    const string baseFilename,
    gcstring & deviceSerialNumber,
    ImageUtility::SourceDataRange srcDataRange = ImageUtility::IMAGE\_DATA\_RANGE )
```

16.267.1.8 ExtractAndSavePolarQuadImages()

```
int ExtractAndSavePolarQuadImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.267.1.9 GetQuadFileNameAppendage()

```
std::string GetQuadFileNameAppendage (
    const ImageUtilityPolarization::PolarizationQuadrant quadrant )
```

16.267.1.10 main()

```
int main (
    int ,
    char ** )
```

16.267.1.11 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.267.1.12 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.267.1.13 SaveImage()

```
int SaveImage (
    const ImagePtr & pImage,
    const string filename,
    gcstring & serialNumber )
```

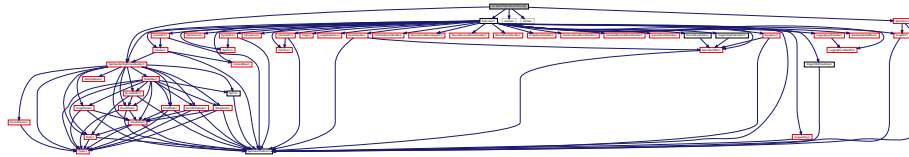
16.267.2 Variable Documentation

16.267.2.1 isPixelFormatColor

```
bool isPixelFormatColor = false [static]
```

16.268 src/SaveToAvi/SaveToAvi.cpp File Reference

Include dependency graph for SaveToAvi.cpp:



Enumerations

- enum [videoType](#) {
[UNCOMPRESSED](#),
[MJPG](#),
[H264](#) }

Functions

- int [SaveVectorToVideo](#) (INodeMap &nodeMap, INodeMap &nodeMapTLDevice, vector< [ImagePtr](#) > &images)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, vector< [ImagePtr](#) > &images)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

Variables

- const [videoType](#) chosenVideoType = [UNCOMPRESSED](#)

16.268.1 Enumeration Type Documentation

16.268.1.1 videoType

```
enum videoType
```

Enumerator

UNCOMPRESSED	
MJPEG	
H264	

16.268.2 Function Documentation

16.268.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    vector< ImagePtr > & images )
```

16.268.2.2 main()

```
int main (
    int ,
    char ** )
```

16.268.2.3 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.268.2.4 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.268.2.5 SaveVectorToVideo()

```
int SaveVectorToVideo (
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    vector< ImagePtr > & images )
```

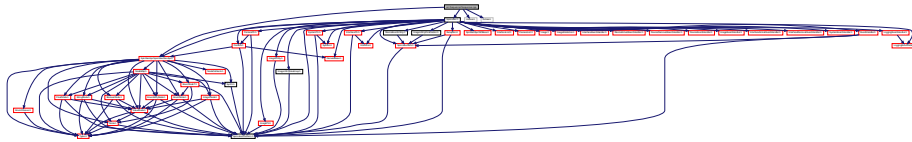
16.268.3 Variable Documentation

16.268.3.1 chosenVideoType

```
const videoType chosenVideoType = UNCOMPRESSED
```

16.269 src/Sequencer/Sequencer.cpp File Reference

Include dependency graph for Sequencer.cpp:



Functions

- void [PrintRetrieveNodeFailure](#) (string node, string name)
- int [ConfigureSequencerPartOne](#) (INodeMap &nodeMap)
- int [SetSingleState](#) (INodeMap &nodeMap, unsigned int sequenceNumber, int64_t widthToSet, int64_t heightToSet, double exposureTimeToSet, double gainToSet)
- int [ConfigureSequencerPartTwo](#) (INodeMap &nodeMap)
- int [ResetSequencer](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapGenTL, uint64_t timeout)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.269.1 Function Documentation

16.269.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapGenTL,
    uint64_t timeout )
```

16.269.1.2 ConfigureSequencerPartOne()

```
int ConfigureSequencerPartOne (
    INodeMap & nodeMap )
```

16.269.1.3 ConfigureSequencerPartTwo()

```
int ConfigureSequencerPartTwo (
    INodeMap & nodeMap )
```

16.269.1.4 main()

```
int main (
    int ,
    char ** )
```

16.269.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.269.1.6 PrintRetrieveNodeFailure()

```
void PrintRetrieveNodeFailure (
    string node,
    string name )
```

16.269.1.7 ResetSequencer()

```
int ResetSequencer (
    INodeMap & nodeMap )
```

16.269.1.8 RunSingleCamera()

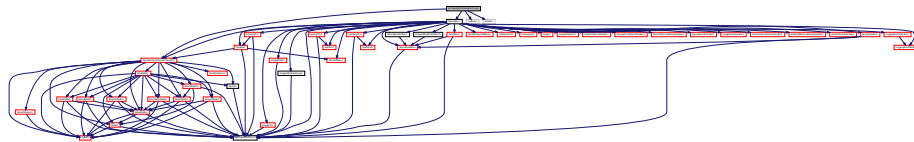
```
int RunSingleCamera (
    CameraPtr pCam )
```

16.269.1.9 SetSingleState()

```
int SetSingleState (
    INodeMap & nodeMap,
    unsigned int sequenceNumber,
    int64_t widthToSet,
    int64_t heightToSet,
    double exposureTimeToSet,
    double gainToSet )
```

16.270 src/SerialRxTx/SerialRxTx.cpp File Reference

Include dependency graph for SerialRxTx.cpp:



Macros

- `#define COM_PORT_COUNT_MAX 256`
- `#define TWO_SECOND_DELAY 2000`
- `#define SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND 1000`
- `#define SERIAL_PORT_BAUD_RATE 19200`
- `#define SERIAL_PORT_STOP_BITS 0`
- `#define SERIAL_PORT_PARITY_BITS 0`
- `#define SERIAL_PORT_DELAY 1500`
- `#define DATA_BITS 8`
- `#define MILLISECOND 1000`

Functions

- `int PrintDeviceInfo (INodeMap &nodeMap)`
- `int ConfigureDevice (CameraPtr pCam, HANDLE &hFileHandle)`
- `int SerialRx (CameraPtr pCam, INodeMap &nodeMap, HANDLE &hFileHandle)`
- `int SerialTx (CameraPtr pCam, INodeMap &nodeMap, HANDLE &hFileHandle)`
- `int CleanUp (INodeMap &nodeMap, HANDLE &hFileHandle)`
- `int RunSingleCamera (CameraPtr pCam)`
- `int main (int, char **)`

16.270.1 Macro Definition Documentation

16.270.1.1 COM_PORT_COUNT_MAX

```
#define COM_PORT_COUNT_MAX 256
```

16.270.1.2 DATA_BITS

```
#define DATA_BITS 8
```

16.270.1.3 MILLISECOND

```
#define MILLISECOND 1000
```

16.270.1.4 SERIAL_PORT_BAUD_RATE

```
#define SERIAL_PORT_BAUD_RATE 19200
```

16.270.1.5 SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND

```
#define SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND 1000
```

16.270.1.6 SERIAL_PORT_DELAY

```
#define SERIAL_PORT_DELAY 1500
```

16.270.1.7 SERIAL_PORT_PARITY_BITS

```
#define SERIAL_PORT_PARITY_BITS 0
```

16.270.1.8 SERIAL_PORT_STOP_BITS

```
#define SERIAL_PORT_STOP_BITS 0
```

16.270.1.9 TWO_SECOND_DELAY

```
#define TWO_SECOND_DELAY 2000
```

16.270.2 Function Documentation

16.270.2.1 CleanUp()

```
int CleanUp (
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

16.270.2.2 ConfigureDevice()

```
int ConfigureDevice (
    CameraPtr pCam,
    HANDLE & hFileHandle )
```

16.270.2.3 main()

```
int main (
    int ,
    char ** )
```

16.270.2.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.270.2.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```


16.270.2.6 SerialRx()

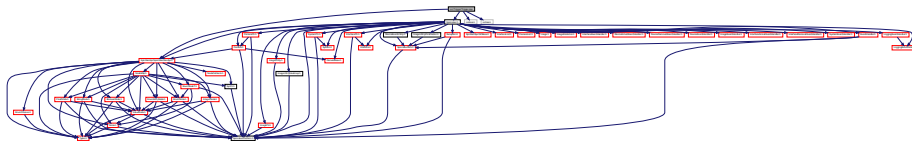
```
int SerialRx (
    CameraPtr pCam,
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

16.270.2.7 SerialTx()

```
int SerialTx (
    CameraPtr pCam,
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

16.271 src/Trigger/Trigger.cpp File Reference

Include dependency graph for Trigger.cpp:



Enumerations

- enum triggerType {
SOFTWARE,
HARDWARE,
SOFTWARE,
HARDWARE }

Functions

- int ConfigureTrigger (INodeMap &nodeMap)
- int GrabNextImageByTrigger (INodeMap &nodeMap, CameraPtr pCam)
- int ResetTrigger (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const triggerType chosenTrigger = SOFTWARE

16.271.1 Enumeration Type Documentation

16.271.1.1 triggerType

enum `triggerType`

Enumerator

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

16.271.2 Function Documentation

16.271.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.271.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.271.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
    INodeMap & nodeMap,
    CameraPtr pCam )
```

16.271.2.4 main()

```
int main (
    int ,
    char ** )
```

16.271.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.271.2.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.271.2.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

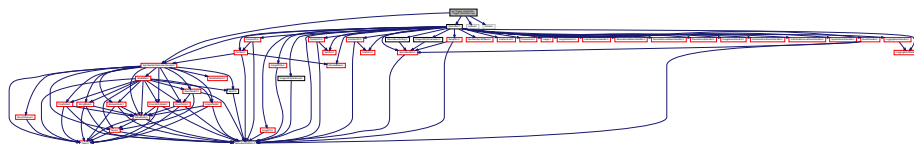
16.271.3 Variable Documentation

16.271.3.1 chosenTrigger

```
const triggerType chosenTrigger = SOFTWARE
```

16.272 src/Trigger_QuickSpin/Trigger_QuickSpin.cpp File Reference

Include dependency graph for Trigger_QuickSpin.cpp:



Enumerations

- enum triggerType {
 SOFTWARE,
 HARDWARE,
 SOFTWARE,
 HARDWARE }

Functions

- int ConfigureTrigger (CameraPtr pCam)
- int GrabNextImageByTrigger (CameraPtr pCam, ImagePtr &pResultImage)
- int ResetTrigger (CameraPtr pCam)
- int PrintDeviceInfo (CameraPtr pCam)
- int AcquireImages (CameraPtr pCam)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const `triggerType` `chosenTrigger` = SOFTWARE

16.272.1 Enumeration Type Documentation

16.272.1.1 `triggerType`

enum `triggerType`

Enumerator

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

16.272.2 Function Documentation

16.272.2.1 `AcquireImages()`

```
int AcquireImages (
    CameraPtr pCam )
```

16.272.2.2 `ConfigureTrigger()`

```
int ConfigureTrigger (
    CameraPtr pCam )
```

16.272.2.3 `GrabNextImageByTrigger()`

```
int GrabNextImageByTrigger (
    CameraPtr pCam,
    ImagePtr & pResultImage )
```

16.272.2.4 main()

```
int main (
    int ,
    char ** )
```

16.272.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    CameraPtr pCam )
```

16.272.2.6 ResetTrigger()

```
int ResetTrigger (
    CameraPtr pCam )
```

16.272.2.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.272.3 Variable Documentation**16.272.3.1 chosenTrigger**

```
const triggerType chosenTrigger = SOFTWARE
```

Chapter 17

Example Documentation

17.1 Acquisition.cpp

[Acquisition.cpp](#) shows how to acquire images. It relies on information provided in the Enumeration example. Also, check out the ExceptionHandling and NodeMapInfo examples if you haven't already. ExceptionHandling shows the handling of standard and [Spinnaker](#) exceptions while NodeMapInfo explores retrieving information from various node types.

This example touches on the preparation and cleanup of a camera just before and just after the acquisition of images. Image retrieval and conversion, grabbing image data, and saving images are all covered as well.

Once comfortable with Acquisition, we suggest checking out AcquisitionMultipleCamera, NodeMapCallback, or SaveToAvi. AcquisitionMultipleCamera demonstrates simultaneously acquiring images from a number of cameras, NodeMapCallback serves as a good introduction to programming with callbacks and events, and SaveToAvi exhibits video creation.

17.2 AcquisitionMultipleCameraRecovery.cpp

[AcquisitionMultipleCameraRecovery.cpp](#) shows how to continuously acquire images from multiple cameras using image events. It demonstrates the use of User Set Control to save persistent camera configurations, allowing for smooth camera recovery through interface events. This example relies on information provided in the ImageEvents, EnumerationEvents, ImageFormatControl, and Acquisition examples.

This example uses a global map to retain image information, including the number of images grabbed, the number of incomplete images and the number of removals for each camera over the duration of the example. Cameras may be added or removed after the example has started.

The example assumes each camera has a unique serial number and is capable of configuring User Set 1. Note that if a camera was configured and is disconnected before the example ends, it will not be reconfigured to use the default User Set.

17.3 AcquisitionMultipleCamerasWriteToFiles.cpp

AcquisitionMultipleCamerasWriteToFiles.cpp shows how to acquire images from one or more cameras and write the images to a binary file. Thereafter, the acquired images can be retrieved from the file and saved using a desired file format supported by the current [Spinnaker](#) SDK.

This example covers most of the basics for getting started with the [Spinnaker](#) API including acquiring system objects, camera list acquisition and initialization of cameras, acquisition of images and writing to a file, image retrieval from a file, conversion and saving with desired file format is covered as well.

17.4 AcquisitionMultipleThread.cpp

[AcquisitionMultipleThread.cpp](#) shows how to capture images from multiple cameras simultaneously using threads. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example is similar to the Acquisition example, except that threads are used to allow for simultaneous acquisitions.

17.5 ActionCommand.cpp

[ActionCommand.cpp](#) shows how to send action commands to the camera(s). And it also can perform frame synchronization using multiple cameras simultaneously. It uses IEEE 1588 timestamp, triggers and action commands.

One camera will be master and the other camera(s) will be slave(s). All cameras have to be on the same network.

17.6 BufferHandling.cpp

[BufferHandling.cpp](#) shows how the different buffer handling modes work. It relies on information provided in the Acquisition and Trigger examples.

Buffer handling determines the ordering in which images are retrieved, and what occurs when an image is transmitted while the buffer is full. There are four different buffer handling modes available; NewestFirst, NewestOnly, OldestFirst and OldestFirstOverwrite.

This example explores retrieving images in a set pattern; triggering the camera while not retrieving an image (letting the buffer fill up), and retrieving images while not triggering. We cycle through the different buffer handling modes to see which images are retrieved, confirming their identities via their Frame ID values.

17.7 ChunkData.cpp

[ChunkData.cpp](#) shows how to get chunk data on an image, either from the nodemap or from the image itself. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

Chunk data provides information on various traits of an image. This includes identifiers such as frame ID, properties such as black level, and more. This information can be acquired from either the nodemap or the image itself.

It may be preferable to grab chunk data from each individual image, as it can be hard to verify whether data is coming from the correct image when using the nodemap. This is because chunk data retrieved from the nodemap is only valid for the current image; when `GetNextImage()` is called, chunk data will be updated to that of the new current image.

17.8 Compression.cpp

[Compression.cpp](#) shows how to utilize the compression features on a supported camera and in the [Spinnaker](#) SDK. It relies on information provided in the Acquisition, Enumeration, ChunkData, and NodeMapInfo examples.

This example covers all of the following: the preparation of a camera to acquire compressed images (or compressed chunk images), image retrieval, image saving, loading compressed images from disk, reconstructing compressed images, and converting compressed images.

17.9 CounterAndTimer.cpp

[CounterAndTimer.cpp](#) shows how to setup a Pulse Width Modulation (PWM) signal using counters and timers. The camera will output the PWM signal via strobe, and capture images at a rate defined by the PWM signal as well. Users should take care to use a PWM signal within the camera's max framerate (by default, the PWM signal is set to 50 Hz).

Counter and Timer functionality is only available for BFS and Oryx Cameras. For details on the hardware setup, see our kb article, "Using Counter and Timer Control"; <https://www.flir.com/support-center/iis/machine-vision/application-note/using-counter-and-timer-control>

17.10 DeviceEvents.cpp

[DeviceEvents.cpp](#) shows how to create a handler to access device events. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

Device events can be thought of as camera-related events. This example creates a user-defined class, [DeviceEventHandlerImpl](#), which allows the user to define any properties, parameters, and the event handler itself while [DeviceEventHandler](#), the parent class, allows the child class to appropriately interface with the [Spinnaker](#) SDK.

17.11 Enumeration.cpp

[Enumeration.cpp](#) shows how to enumerate interfaces and cameras. Knowing this is mandatory for doing anything with the [Spinnaker](#) SDK, and is therefore the best place to start learning how to use the SDK.

This example introduces the preparation, use, and cleanup of the system object, interface and camera lists, interfaces, and cameras. It also touches on retrieving both nodes from nodemaps and information from nodes.

Once comfortable with enumeration, we suggest checking out the [Acquisition](#), [ExceptionHandling](#), or [NodeMapInfo](#) examples. [Acquisition](#) demonstrates using a camera to acquire images, [ExceptionHandling](#) explores the use of standard and [Spinnaker](#) exceptions, and [NodeMapInfo](#) demonstrates retrieving information from various node types.

17.12 Enumeration_QuickSpin.cpp

[Enumeration_QuickSpin.cpp](#) shows how to enumerate interfaces and cameras using the QuickSpin API. QuickSpin is a subset of the [Spinnaker](#) library that allows for simpler node access and control. This is a great example to start learning about QuickSpin.

This example introduces the preparation, use, and cleanup of the system object, interface and camera lists, interfaces, and cameras. It also touches on retrieving information from pre-fetched nodes using QuickSpin. Retrieving node information is the only portion of the example that differs from [Enumeration](#).

A much wider range of topics is covered in the full [Spinnaker](#) examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full [Spinnaker](#) examples for further or specific knowledge on a topic.

17.13 EnumerationEvents.cpp

[EnumerationEvents.cpp](#) explores arrival and removal events on interfaces and the system. It relies on information provided in the [Enumeration](#), [Acquisition](#), and [NodeMapInfo](#) examples.

It can also be helpful to familiarize yourself with the [NodeMapCallback](#) example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates two user-defined classes: [InterfaceEventHandlerImpl](#) and [SystemEventHandlerImpl](#). These child classes allow the user to define properties, parameters, and the event itself while the parent classes - [DeviceArrivalEventHandler](#), [DeviceRemovalEventHandler](#), and [InterfaceEventHandler](#) - allow the child classes to interface with [Spinnaker](#).

17.14 ExceptionHandling.cpp

[ExceptionHandling.cpp](#) shows the catching of an exception in [Spinnaker](#). Following this, check out the [Acquisition](#) or [NodeMapInfo](#) examples if you haven't already. [Acquisition](#) demonstrates image acquisition while [NodeMapInfo](#) explores retrieving information from various node types.

This example shows three typical paths of exception handling in [Spinnaker](#): catching the exception as a [Spinnaker](#) exception, as a standard exception, or as a standard exception which is then cast to a [Spinnaker](#) exception.

Once comfortable with [Acquisition](#), [ExceptionHandling](#), and [NodeMapInfo](#), we suggest checking out [AcquisitionMultipleCamera](#), [NodeMapCallback](#), or [SaveToAvi](#). [AcquisitionMultipleCamera](#) demonstrates simultaneously acquiring images from a number of cameras, [NodeMapCallback](#) serves as a good introduction to programming with callbacks and events, and [SaveToAvi](#) exhibits video creation.

17.15 Exposure.cpp

[Exposure.cpp](#) shows how to set a custom exposure time on a device. It relies on information provided in the [Enumeration](#), [Acquisition](#), and [NodeMapInfo](#) examples.

This example shows the processes of preparing the camera, setting a custom exposure time, and restoring the camera to its default state (without power cycling). Ensuring custom values do not fall out of range is also touched on.

Following this, we suggest familiarizing yourself with the [ImageFormatControl](#) example if you haven't already. [ImageFormatControl](#) is another example on camera customization that is shorter and simpler than many of the others. Once comfortable with [Exposure](#) and [ImageFormatControl](#), we suggest checking out any of the longer, more complicated examples related to camera configuration: [ChunkData](#), [LookupTable](#), [Sequencer](#), or [Trigger](#).

17.16 Exposure_QuickSpin.cpp

[Exposure_QuickSpin.cpp](#) shows how to customize image exposure time using the QuickSpin API. QuickSpin is a subset of the [Spinnaker](#) library that allows for simpler node access and control.

This example prepares the camera, sets a new exposure time, and restores the camera to its default state. Ensuring custom values fall within an acceptable range is also touched on. Retrieving and setting information is the only portion of the example that differs from [Exposure](#).

A much wider range of topics is covered in the full [Spinnaker](#) examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full [Spinnaker](#) examples for further or specific knowledge on a topic.

17.17 FileAccess_Quickspin.cpp

[FileAccess_Quickspin.cpp](#) shows how to read and write images using camera File Access function. This example uploads an image to the camera File Access storage and also download the image from the camera File Access storage and saves it to the disk. It also provides debug message when debug mode is turned on giving more detail status of the progress and error messages to the users.

It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

17.18 GigEVisionPerformance.cpp

[GigEVisionPerformance.cpp](#) measures GigE Vision performance. It is built on top of Acquisition example.

This example measures CPU related performance statistics and print them out at the end.

17.19 ImageEvents.cpp

[ImageEvents.cpp](#) shows how to acquire images using the image event handler. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates a user-defined class, [ImageEventHandlerImpl](#), that inherits from the [Spinnaker](#) class, [ImageEventHandler](#). [ImageEventHandlerImpl](#) allows the user to define any properties, parameters, and the event itself while [ImageEventHandler](#) allows the child class to appropriately interface with [Spinnaker](#).

17.20 ImageFormatControl.cpp

[ImageFormatControl.cpp](#) shows how to apply custom image settings to the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example demonstrates setting minimums to offsets, X and Y, and maximums to width and height. It also shows the setting of a new pixel format, which is an enumeration type node.

Following this, we suggest familiarizing yourself with the Exposure example if you haven't already. Exposure is another example on camera customization that is shorter and simpler than many of the others. Once comfortable with Exposure and ImageFormatControl, we suggest checking out any of the longer, more complicated examples related to camera configuration: ChunkData, LookupTable, Sequencer, or Trigger.

17.21 ImageFormatControl_QuickSpin.cpp

[ImageFormatControl_QuickSpin.cpp](#) shows how to apply custom image settings to the camera using the QuickSpin API. QuickSpin is a subset of the [Spinnaker](#) library that allows for simpler node access and control.

This example demonstrates customizing offsets X and Y, width and height, and the pixel format. Ensuring custom values fall within an acceptable range is also touched on. Retrieving and setting node values using QuickSpin is the only portion of the example that differs from ImageFormatControl.

A much wider range of topics is covered in the full [Spinnaker](#) examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full [Spinnaker](#) examples for further or specific knowledge on a topic.

17.22 Inference.cpp

[Inference.cpp](#) shows how to perform the following:

- Upload custom inference neural networks to the camera (DDR or Flash)
- Inject sample test image
- Enable/Configure chunk data
- Enable/Configure trigger inference ready sync
- Acquire images
- Display inference data from acquired image chunk data
- Disable previously configured camera configurations

Inference is only available for Firefly deep learning cameras. See the related content section on the Firefly DL product page for relevant documentation.

<https://www.flir.com/products/firefly-dl/>

It can also be helpful to familiarize yourself with the Acquisition, ChunkData and FileAccess_QuickSpin examples.

17.23 Logging.cpp

[Logging.cpp](#) shows how to create a handler to access logging events. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates a user-defined class, [LoggingEventHandlerImpl](#), that inherits from the [Spinnaker](#) class, [LoggingEventHandler](#). The child class allows the user to define any properties, parameters, and the event itself while [LoggingEventHandler](#) allows the child class to appropriately interface with the [Spinnaker](#) SDK.

17.24 LogicBlock.cpp

[LogicBlock.cpp](#) shows how to use logic blocks to detect missing triggers and refire. It relies on information provided in the Acquisition and Trigger examples.

A logic block is a collection of combinatorial logic and latches that allows users to create new, custom signals inside the camera. These custom signals can be used by the camera (for example to trigger exposure) or sent out to integrate with external systems.

Logic Block functionality is only available for BFS and Oryx Cameras. For details on logic blocks and how this example works, see our kb article, "Using Logic Blocks with Blackfly S and Oryx"; <https://www.flir.com/support-center/iis/machine-vision/application-note/using-logic-blocks-with-blackfly->

17.25 LookupTable.cpp

[LookupTable.cpp](#) shows how to configure lookup tables on the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

Lookup tables allow for the customization and control of individual pixels. This can be a very powerful and deeply useful tool; however, because use cases are context dependent, this example only explores lookup table configuration.

17.26 NodeMapCallback.cpp

[NodeMapCallback.cpp](#) shows how to use nodemap callbacks. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples. As callbacks are very similar to events, it may be a good idea to explore this example prior to tackling the events examples.

This example focuses on creating, registering, using, and unregistering callbacks. A callback requires a function signature, which allows it to be registered to and access a node. Events, while slightly more complex, follow this same pattern.

Once comfortable with NodeMapCallback, we suggest checking out any of the events examples: DeviceEvents, EnumerationEvents, ImageEvents, or Logging.

17.27 NodeMapInfo.cpp

[NodeMapInfo.cpp](#) shows how to retrieve node map information. It relies on information provided in the Enumeration example. Also, check out the Acquisition and ExceptionHandling examples if you haven't already. Acquisition demonstrates image acquisition while ExceptionHandling shows the handling of standard and [Spinnaker](#) exceptions.

This example explores retrieving information from all major node types on the camera. This includes string, integer, float, boolean, command, enumeration, category, and value types. Looping through multiple child nodes is also covered. A few node types are not covered - base, port, and register - as they are not fundamental. The final node type - enumeration entry - is explored and printed for nodes whose parent node is a selector node.

Once comfortable with NodeMapInfo, we suggest checking out ImageFormatControl and Exposure. ImageFormatControl explores customizing image settings on a camera while Exposure introduces the standard structure of configuring a device, acquiring some images, and then returning the device to a default state.

17.28 Polarization.cpp

[Polarization.cpp](#) shows how to extract and create images from a source image of Polarized8 or BayerRGPolarized8 pixel format using methods from the ImageUtilityPolarization, ImageUtility and ImageUtilityHeatmap classes. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example demonstrates some of the methods that can be used to extract polarization quadrant images and create Stokes', AoLP, and DoLP images from the ImageUtilityPolarization class. It then demonstrates how to use some of the available methods in the ImageUtility and ImageUtilityHeatmap classes to create normalized and heatmap images.

Polarization is only available for polarized cameras. For more information please visit our website; <https://www.flir.com/discover/iis/machine-vision/imaging-reflective-surfaces-sonys-first-pola>

17.29 SaveToAvi.cpp

[SaveToAvi.cpp](#) shows how to create a video from a vector of images. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example introduces the SpinVideo class, which is used to quickly and easily create various types of video files. It demonstrates the creation of three types: uncompressed, MJPG, and H264.

17.30 Sequencer.cpp

[Sequencer.cpp](#) shows how to use the sequencer to grab images with various settings. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples as these examples provide a strong introduction to camera customization.

The sequencer is another very powerful tool, which can be used to create and store multiple states of customized image settings. A very useful application of the sequencer is creating high dynamic range images.

This example is probably the most complex and definitely the longest. As such, the configuration has been split between three functions. The first prepares the camera to set the sequences, the second sets the settings for a single state (it is run five times), and the third configures the camera to use the sequencer when it acquires images.

17.31 SerialRxTx.cpp

[SerialRxTx.cpp](#) shows how to communicate using Serial ports. It sets serial port settings in [Spinnaker](#), opens and operates on FileAccess nodes and creates COM Port handle. After the setup, it transmits and receives simple data. It verifies the communication by transmitting data and reading data to/from the COM Port. Basic understanding of serial communication and knowledge of the installation process for a serial port is required.

THIS EXAMPLE ONLY WORKS IN WINDOWS OS

Machine setup steps to run the example:

1. Your PC needs a serial port. If one does not already exist, purchase a USB to Serial port cable or adapter. After installing this, your PC will see a serial port in device manager. Note the COM port assigned to this serial port.
2. From your PC's serial port, you need a RS232 to TTL converter
3. To wire up from the RS232-TTL converter to the camera, use a FLIR supplied GPIO cable or make your own from the part numbers supplied in our technical reference manual. Wire up the TX/RX/GND pins between the adapter and the camera.
4. Run the example. It should find the COM port, set up the serial settings, read anything sent to the camera, and then transmit ABCD over the serial port

The final setup of the machine to run the example should be: camera -> GPIO -> TTL-RS232 -> RS232 port

More information on how to configure and test serial port can be found here: <https://www.flir.com/support-center/iis/machine-vision/application-note/configuring-and-testing-the-rs-232>

17.32 Trigger.cpp

[Trigger.cpp](#) shows how to trigger the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

This example shows the process of configuring, using, and cleaning up a camera for use with both a software and a hardware trigger.

17.33 Trigger_QuickSpin.cpp

[Trigger_QuickSpin.cpp](#) shows how to capture images with the trigger using the QuickSpin API. QuickSpin is a subset of the [Spinnaker](#) library that allows for simpler node access and control.

This example demonstrates how to prepare, execute, and clean up the camera in regards to using both software and hardware triggers. Retrieving and setting node values using QuickSpin is the only portion of the example that differs from Trigger.

A much wider range of topics is covered in the full Spinnaker examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full Spinnaker examples for further or specific knowledge on a topic.

Index

- [_ClearXMLCache](#)
 - Spinnaker GenApi Classes, [268](#)
 - Spinnaker::GenApi::CNodeMapRefT, [763](#)
 - [_Connect](#)
 - Spinnaker GenApi Classes, [268](#), [269](#)
 - Spinnaker::GenApi::CNodeMapRefT, [763](#)
 - [_Destroy](#)
 - Spinnaker GenApi Classes, [269](#)
 - [_GetDeviceName](#)
 - Spinnaker GenApi Classes, [269](#)
 - Spinnaker::GenApi::CNodeMapRefT, [763](#)
 - [_GetNode](#)
 - Spinnaker GenApi Classes, [269](#)
 - Spinnaker::GenApi::CNodeMapRefT, [763](#)
 - [_GetNodes](#)
 - Spinnaker GenApi Classes, [269](#)
 - Spinnaker::GenApi::CNodeMapRefT, [764](#)
 - [_GetSupportedSchemaVersions](#)
 - Spinnaker GenApi Classes, [269](#)
 - Spinnaker::GenApi::CNodeMapRefT, [764](#)
 - [_Initialize](#)
 - Spinnaker::GenApi::CGeneric_XMLLoader↔
Params, [722](#)
 - [_InvalidateNodes](#)
 - Spinnaker GenApi Classes, [269](#)
 - Spinnaker::GenApi::CNodeMapRefT, [764](#)
 - [_LoadXMLFromFile](#)
 - Spinnaker GenApi Classes, [270](#)
 - Spinnaker::GenApi::CNodeMapRefT, [764](#)
 - [_LoadXMLFromFileInject](#)
 - Spinnaker GenApi Classes, [270](#)
 - Spinnaker::GenApi::CNodeMapRefT, [764](#)
 - [_LoadXMLFromString](#)
 - Spinnaker GenApi Classes, [270](#)
 - Spinnaker::GenApi::CNodeMapRefT, [765](#)
 - [_LoadXMLFromStringInject](#)
 - Spinnaker GenApi Classes, [270](#)
 - Spinnaker::GenApi::CNodeMapRefT, [765](#)
 - [_LoadXMLFromZIPData](#)
 - Spinnaker GenApi Classes, [270](#)
 - Spinnaker::GenApi::CNodeMapRefT, [765](#)
 - [_LoadXMLFromZIPFile](#)
 - Spinnaker GenApi Classes, [270](#)
 - Spinnaker::GenApi::CNodeMapRefT, [765](#)
 - [_Poll](#)
 - Spinnaker GenApi Classes, [270](#)
 - Spinnaker::GenApi::CNodeMapRefT, [765](#)
 - [_Ptr](#)
 - Spinnaker::GenApi::CNodeMapRefT, [766](#)
- Spinnaker::GenApi::NodeMap, [1104](#)
- [_TO_STRING](#)
 - GCUtilities.h, [1314](#)
- [_UndefinedRepresentation](#)
 - Types Enums, [402](#)
- [__ERR__](#)
 - GCUtilities.h, [1313](#)
- [__LINE_STR__](#)
 - GCUtilities.h, [1314](#)
- [__LOCATION__](#)
 - GCUtilities.h, [1314](#)
- [__OUTPUT_FORMATER__](#)
 - GCUtilities.h, [1314](#)
- [__STDC_CONSTANT_MACROS](#)
 - GCTypes.h, [1310](#)
- [__STDC_LIMIT_MACROS](#)
 - GCTypes.h, [1310](#)
- [__TODO__](#)
 - GCUtilities.h, [1314](#)
- [__WARN__](#)
 - GCUtilities.h, [1314](#)
- [_enableDebug](#)
 - FileAccess_QuickSpin.cpp, [1458](#)
- [_fileSelector](#)
 - FileAccess_QuickSpin.cpp, [1458](#)
- [_npos](#)
 - Spinnaker::GenICam::gcstring, [860](#)
- [_pCount](#)
 - Spinnaker::GenApi::double_autovector_t, [811](#)
 - Spinnaker::GenApi::int64_autovector_t, [1024](#)
- [_pv](#)
 - Spinnaker::GenApi::double_autovector_t, [811](#)
 - Spinnaker::GenApi::int64_autovector_t, [1025](#)
- [~AutoLock](#)
 - Spinnaker::GenApi::AutoLock, [489](#)
 - Spinnaker::GenICam::AutoLock, [488](#)
- [~BasePtr](#)
 - Spinnaker::BasePtr, [491](#)
- [~BooleanNode](#)
 - Spinnaker::GenApi::BooleanNode, [497](#)
- [~CChunkAdapter](#)
 - Spinnaker::GenApi::CChunkAdapter, [677](#)
- [~CChunkAdapterDcam](#)
 - Spinnaker::GenApi::CChunkAdapterDcam, [680](#)
- [~CChunkAdapterGEV](#)
 - Spinnaker::GenApi::CChunkAdapterGEV, [685](#)
- [~CChunkAdapterGeneric](#)
 - Spinnaker::GenApi::CChunkAdapterGeneric, [683](#)
- [~CChunkAdapterU3V](#)

- Spinnaker::GenApi::CChunkAdapterU3V, 687
- ~CChunkPort
 - Spinnaker::GenApi::CChunkPort, 690
- ~CEnumerationTRef
 - Spinnaker::GenApi::CEnumerationTRef, 698
- ~CEventAdapter
 - Spinnaker::GenApi::CEventAdapter, 702
- ~CEventAdapter1394
 - Spinnaker::GenApi::CEventAdapter1394, 704
- ~CEventAdapterGEV
 - Spinnaker::GenApi::CEventAdapterGEV, 709
- ~CEventAdapterGeneric
 - Spinnaker::GenApi::CEventAdapterGeneric, 706
- ~CEventAdapterU3V
 - Spinnaker::GenApi::CEventAdapterU3V, 711
- ~CEventPort
 - Spinnaker::GenApi::CEventPort, 714
- ~CFeatureBag
 - Spinnaker::GenApi::CFeatureBag, 718
- ~CGlobalLock
 - Spinnaker::GenICam::CGlobalLock, 724
- ~CGlobalLockUnlocker
 - Spinnaker::GenICam::CGlobalLockUnlocker, 726
- ~CLock
 - Spinnaker::GenApi::CLock, 743
 - Spinnaker::GenICam::CLock, 741
- ~CNodeCallback
 - Spinnaker::GenApi::CNodeCallback, 748
- ~CNodeMapFactory
 - Spinnaker::GenApi::CNodeMapFactory, 752
- ~CNodeMapRefT
 - Spinnaker GenApi Classes, 272
- ~CPointer
 - Spinnaker::GenApi::CPointer, 774
- ~CPortImpl
 - Spinnaker::GenApi::CPortImpl, 779
- ~CPortWriteList
 - Spinnaker::GenApi::CPortWriteList, 782
- ~CRegisterPortImpl
 - Spinnaker::GenApi::CRegisterPortImpl, 786
- ~CSelectorSet
 - Spinnaker::GenApi::CSelectorSet, 790
- ~Camera
 - Spinnaker::Camera, 529
- ~CameraBase
 - Spinnaker::CameraBase, 654
- ~CameraList
 - Spinnaker::CameraList, 668
- ~CategoryNode
 - Spinnaker::GenApi::CategoryNode, 675
- ~ChunkData
 - Spinnaker::ChunkData, 730
- ~CommandNode
 - Spinnaker::GenApi::CommandNode, 768
- ~DeviceArrivalEventHandler
 - Spinnaker::DeviceArrivalEventHandler, 798
- ~DeviceEventHandler
 - Spinnaker::DeviceEventHandler, 801
- ~DeviceEventHandlerImpl
 - DeviceEventHandlerImpl, 803
- ~DeviceRemovalEventHandler
 - Spinnaker::DeviceRemovalEventHandler, 808
- ~EnumEntryNode
 - Spinnaker::GenApi::EnumEntryNode, 820
- ~EnumNode
 - Spinnaker::GenApi::EnumNode, 824
- ~EventHandler
 - Spinnaker::EventHandler, 833
- ~Exception
 - Spinnaker::Exception, 839
- ~FileProtocolAdapter
 - Spinnaker::GenApi::FileProtocolAdapter, 843
- ~FloatNode
 - Spinnaker::GenApi::FloatNode, 849
- ~FloatRegNode
 - Spinnaker::GenApi::FloatRegNode, 855
- ~ICameraBase
 - Spinnaker::ICameraBase, 882
- ~ICameraList
 - Spinnaker::ICameraList, 890
- ~IChunkData
 - Spinnaker::IChunkData, 894
- ~IDataStream
 - Spinnaker::IDataStream, 902
- ~IDevFileStreamBuf
 - Spinnaker::GenApi::IDevFileStreamBuf, 912
- ~IDeviceArrivalEventHandler
 - Spinnaker::IDeviceArrivalEventHandler, 914
- ~IDeviceEventHandler
 - Spinnaker::IDeviceEventHandler, 916
- ~IDeviceRemovalEventHandler
 - Spinnaker::IDeviceRemovalEventHandler, 918
- ~IImage
 - Spinnaker::IImage, 921
- ~IImageEventHandler
 - Spinnaker::IImageEventHandler, 932
- ~IImageStatistics
 - Spinnaker::IImageStatistics, 934
- ~IInterface
 - Spinnaker::IInterface, 939
- ~IInterfaceArrivalEventHandler
 - Spinnaker::IInterfaceArrivalEventHandler, 943
- ~IInterfaceEventHandler
 - Spinnaker::IInterfaceEventHandler, 945
- ~IInterfaceList
 - Spinnaker::IInterfaceList, 948
- ~IInterfaceRemovalEventHandler
 - Spinnaker::IInterfaceRemovalEventHandler, 951
- ~ILoggingEventHandler
 - Spinnaker::ILoggingEventHandler, 953
- ~ISystem
 - Spinnaker::ISystem, 1059
- ~ISystemEventHandler
 - Spinnaker::ISystemEventHandler, 1065
- ~Image
 - Spinnaker::Image, 958

- ~ImageEventHandler
 - Spinnaker::ImageEventHandler, [984](#)
- ~ImageEventHandlerImpl
 - ImageEventHandlerImpl, [986](#), [987](#)
- ~ImagePtr
 - Spinnaker::ImagePtr, [991](#)
- ~ImageStatistics
 - Spinnaker::ImageStatistics, [994](#)
- ~InferenceBoundingBoxResult
 - Chunk Data Inference Class, [174](#)
- ~IntRegNode
 - Spinnaker::GenApi::IntRegNode, [1056](#)
- ~IntegerNode
 - Spinnaker::GenApi::IntegerNode, [1027](#)
- ~Interface
 - Spinnaker::Interface, [1032](#)
- ~InterfaceArrivalEventHandler
 - Spinnaker::InterfaceArrivalEventHandler, [1038](#)
- ~InterfaceEventHandler
 - Spinnaker::InterfaceEventHandler, [1040](#)
- ~InterfaceEventHandlerImpl
 - InterfaceEventHandlerImpl, [1044](#)
- ~InterfaceList
 - Spinnaker::InterfaceList, [1047](#)
- ~InterfaceRemovalEventHandler
 - Spinnaker::InterfaceRemovalEventHandler, [1053](#)
- ~Lock
 - Spinnaker::GenICam::LockableObject::Lock, [1070](#)
- ~LoggingEventData
 - Spinnaker::LoggingEventData, [1073](#)
- ~LoggingEventHandler
 - Spinnaker::LoggingEventHandler, [1079](#)
- ~Node
 - Spinnaker::GenApi::Node, [1088](#)
- ~NodeMap
 - Spinnaker::GenApi::NodeMap, [1099](#)
- ~ODevFileStreamBuf
 - Spinnaker::GenApi::ODevFileStreamBuf, [1109](#)
- ~PortNode
 - Spinnaker::GenApi::PortNode, [1115](#)
- ~PortRecorder
 - Spinnaker::GenApi::PortRecorder, [1120](#)
- ~PortReplay
 - Spinnaker::GenApi::PortReplay, [1124](#)
- ~RegisterNode
 - Spinnaker::GenApi::RegisterNode, [1129](#)
- ~SpinVideo
 - Spinnaker::Video::SpinVideo, [1133](#)
- ~StringNode
 - Spinnaker::GenApi::StringNode, [1138](#)
- ~StringRegNode
 - Spinnaker::GenApi::StringRegNode, [1143](#)
- ~System
 - Spinnaker::System, [1146](#)
- ~SystemEventHandler
 - Spinnaker::SystemEventHandler, [1156](#)
- ~SystemEventHandlerImpl
 - SystemEventHandlerImpl, [1158](#)
- ~SystemPtr
 - Spinnaker::SystemPtr, [1162](#)
- ~TransportLayerDevice
 - Spinnaker::TransportLayerDevice, [1167](#)
- ~TransportLayerInterface
 - Spinnaker::TransportLayerInterface, [1178](#)
- ~TransportLayerStream
 - Spinnaker::TransportLayerStream, [1188](#)
- ~TransportLayerSystem
 - Spinnaker::TransportLayerSystem, [1196](#)
- ~ValueNode
 - Spinnaker::GenApi::ValueNode, [1207](#)
- ~double_autovector_t
 - Spinnaker::GenApi::double_autovector_t, [809](#)
- ~gcstring
 - Spinnaker::GenICam::gcstring, [860](#)
- ~int64_autovector_t
 - Spinnaker::GenApi::int64_autovector_t, [1023](#)
- ADAPTERCONFIG_API
 - AdapterConfig.h, [1213](#)
- aPAUSEMACCtrlFramesReceived
 - Spinnaker::Camera, [535](#)
- aPAUSEMACCtrlFramesTransmitted
 - Spinnaker::Camera, [535](#)
- AVI Recorder Class, [46](#)
 - DEPRECATED_CLASS, [46](#)
- AVIOption, [489](#)
 - Spinnaker::Video::AVIOption, [489](#)
- AasRoiEnable
 - Spinnaker::Camera, [529](#)
- AasRoiHeight
 - Spinnaker::Camera, [530](#)
- AasRoiOffsetX
 - Spinnaker::Camera, [530](#)
- AasRoiOffsetY
 - Spinnaker::Camera, [530](#)
- AasRoiWidth
 - Spinnaker::Camera, [530](#)
- AcquireImages
 - Acquisition.cpp, [1387](#)
 - AcquisitionMultipleThread.cpp, [1434](#)
 - ActionCommand.cpp, [1436](#)
 - BufferHandling.cpp, [1438](#)
 - ChunkData.cpp, [1441](#)
 - Compression.cpp, [1443](#)
 - CounterAndTimer.cpp, [1445](#)
 - DeviceEvents.cpp, [1447](#)
 - Exposure.cpp, [1453](#)
 - Exposure_QuickSpin.cpp, [1454](#)
 - FileAccess_QuickSpin.cpp, [1456](#)
 - GigEVisionPerformance.cpp, [1461](#)
 - ImageEvents.cpp, [1466](#)
 - ImageFormatControl.cpp, [1468](#)
 - ImageFormatControl_QuickSpin.cpp, [1469](#)
 - Inference.cpp, [1472](#)
 - LogicBlock.cpp, [1478](#)
 - LookupTable.cpp, [1480](#)
 - Polarization.cpp, [1488](#)

- SaveToAvi.cpp, [1491](#)
- Sequencer.cpp, [1492](#)
- Trigger.cpp, [1499](#)
- Trigger_QuickSpin.cpp, [1501](#)
- AcquireImagesAndSaveToFile
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1432](#)
- Acquisition.cpp
 - AcquireImages, [1387](#)
 - main, [1388](#)
 - PrintDeviceInfo, [1388](#)
 - RunSingleCamera, [1388](#)
- AcquisitionAbort
 - Spinnaker::Camera, [531](#)
- AcquisitionArm
 - Spinnaker::Camera, [531](#)
- AcquisitionBurstFrameCount
 - Spinnaker::Camera, [531](#)
- AcquisitionFrameCount
 - Spinnaker::Camera, [531](#)
- AcquisitionFrameRate
 - Spinnaker::Camera, [532](#)
- AcquisitionFrameRateEnable
 - Spinnaker::Camera, [532](#)
- AcquisitionLineRate
 - Spinnaker::Camera, [532](#)
- AcquisitionMode
 - Spinnaker::Camera, [532](#)
- AcquisitionModeEnums
 - CameraDefs Class, [84](#)
- AcquisitionMultipleCameraRecovery.cpp
 - cameraGrabInfoMap, [1431](#)
 - ConfigureCamera, [1430](#)
 - ConfigureUserSet1, [1430](#)
 - GetDeviceSerial, [1430](#)
 - globalCamList, [1431](#)
 - main, [1430](#)
 - PrintExampleStatistics, [1430](#)
 - RefreshCameraList, [1431](#)
 - ResetCameraUserSetToDefault, [1431](#)
 - SleepyWrapper, [1431](#)
- AcquisitionMultipleCamerasWriteToFile.cpp
 - AcquireImagesAndSaveToFile, [1432](#)
 - ConfigureCameras, [1432](#)
 - CreateFiles, [1433](#)
 - imageInfos, [1433](#)
 - k_numImages, [1434](#)
 - kDestinationDirectory, [1434](#)
 - main, [1433](#)
 - PrintDeviceInfo, [1433](#)
 - RetrieveImagesFromFiles, [1433](#)
 - RunCameras, [1433](#)
- AcquisitionMultipleThread.cpp
 - AcquireImages, [1434](#)
 - main, [1434](#)
 - PrintDeviceInfo, [1435](#)
 - RunMultipleCameras, [1435](#)
- AcquisitionResultingFrameRate
 - Spinnaker::Camera, [532](#)
- AcquisitionStart
 - Spinnaker::Camera, [532](#)
- AcquisitionStatus
 - Spinnaker::Camera, [533](#)
- AcquisitionStatusSelector
 - Spinnaker::Camera, [533](#)
- AcquisitionStatusSelectorEnums
 - CameraDefs Class, [84](#)
- AcquisitionStop
 - Spinnaker::Camera, [533](#)
- ActionCommand
 - Spinnaker::TransportLayerInterface, [1178](#)
- ActionCommand.cpp
 - AcquireImages, [1436](#)
 - ConfigureActionControl, [1436](#)
 - ConfigureChunkData, [1436](#)
 - ConfigureIEEE1588, [1436](#)
 - ConfigureInterface, [1436](#)
 - ConfigureOtherNodes, [1436](#)
 - ConfigureTrigger, [1436](#)
 - main, [1437](#)
 - PrintDeviceInfo, [1437](#)
 - RunMultipleCameras, [1437](#)
 - SleepyWrapper, [1437](#)
- ActionCommandResult, [481](#)
- ActionCommandStatus
 - Spinnaker Definitions, [224](#)
- ActionDeviceKey
 - Spinnaker::Camera, [533](#)
- ActionGroupKey
 - Spinnaker::Camera, [533](#)
- ActionGroupMask
 - Spinnaker::Camera, [533](#)
- ActionQueueSize
 - Spinnaker::Camera, [534](#)
- ActionSelector
 - Spinnaker::Camera, [534](#)
- ActionUnconditionalMode
 - Spinnaker::Camera, [534](#)
- ActionUnconditionalModeEnums
 - CameraDefs Class, [85](#)
- AdapterConfig, [413](#)
 - AdapterConfigErr, [414](#)
 - AutoPopulateAdapterInfo, [414](#)
 - AutoPopulateAdvancedProperties, [414](#)
 - ConfigureAdapter, [414](#)
 - GetAuto10GDesc, [415](#)
 - GetAutoGigabitDesc, [415](#)
 - GetAutoStartIp, [415](#)
 - GetAutoSubnetMask, [415](#)
 - GetAutoSubnetMaskLength, [415](#)
 - GetConfigLogFileName, [415](#)
 - GetEnumerationLogFileName, [415](#)
 - GetMaxIpAddress, [416](#)
 - GetMinIpAddress, [416](#)
 - GetSubnetMaskLength, [416](#)
 - IsOnSameSubnet, [416](#)
 - IsValidIpAddress, [416](#)

- IsValidSubnetMask, [416](#)
- PopulateAdapterIpInfo, [416](#)
- RetrieveAllAdapters, [417](#)
- ValidateIpAddress, [417](#)
- AdapterConfig.h
 - ADAPTERCONFIG_API, [1213](#)
- AdapterConfig::AdapterInfo
 - adapterDescription, [484](#)
 - adapterGUID, [484](#)
 - AdapterInfo, [484](#)
 - adapterMACAddress, [484](#)
 - adapterName, [485](#)
 - dhcpEnabled, [485](#)
 - ipInfo, [485](#)
 - jumboPacketValidValues, [485](#)
 - jumboPackets, [485](#)
 - jumboPacketsRegKey, [485](#)
 - receiveBuffers, [485](#)
 - receiveBuffersMax, [485](#)
 - receiveBuffersMin, [486](#)
 - receiveBuffersRegKey, [486](#)
 - receiveBuffersStep, [486](#)
 - transmitBuffers, [486](#)
 - transmitBuffersMax, [486](#)
 - transmitBuffersMin, [486](#)
 - transmitBuffersRegKey, [486](#)
 - transmitBuffersStep, [486](#)
- AdapterConfig::IpInfo
 - gateway, [1057](#)
 - ipAddress, [1057](#)
 - IpInfo, [1057](#)
 - subnetLength, [1057](#)
 - subnetMask, [1057](#)
- AdapterConfigErr
 - AdapterConfig, [414](#)
- AdapterConfigException, [482](#)
 - AdapterConfigException, [482](#)
 - ErrCode, [483](#)
 - GetParamStr, [483](#)
- adapterDescription
 - AdapterConfig::AdapterInfo, [484](#)
- adapterGUID
 - AdapterConfig::AdapterInfo, [484](#)
- AdapterInfo, [483](#)
 - AdapterConfig::AdapterInfo, [484](#)
- adapterMACAddress
 - AdapterConfig::AdapterInfo, [484](#)
- adapterName
 - AdapterConfig::AdapterInfo, [485](#)
- AdaptiveCompressionEnable
 - Spinnaker::Camera, [534](#)
- AdcBitDepth
 - Spinnaker::Camera, [534](#)
- AdcBitDepthEnums
 - CameraDefs Class, [85](#)
- AddInjectionData
 - Spinnaker::GenApi::CNodeMapFactory, [754](#)
- Address
 - IPort Interface, [356](#)
- AnnouncelImage
 - Spinnaker::IDataStream, [903](#)
- Append
 - Spinnaker::CameraList, [669](#)
 - Spinnaker::ICameraList, [890](#)
 - Spinnaker::IInterfaceList, [948](#)
 - Spinnaker::InterfaceList, [1048](#)
 - Spinnaker::Video::SpinVideo, [1134](#)
- append
 - Spinnaker::GenICam::gcstring, [860](#)
- ApplyStyleSheet
 - Spinnaker::GenApi::CNodeMapFactory, [754](#)
- argBayerRG
 - GigEVisionPerformance.cpp, [1463](#)
- argDuration
 - GigEVisionPerformance.cpp, [1463](#)
- argMaxFrames
 - GigEVisionPerformance.cpp, [1463](#)
- argNumImages
 - GigEVisionPerformance.cpp, [1463](#)
- argPacketDelay
 - GigEVisionPerformance.cpp, [1463](#)
- argPacketSize
 - GigEVisionPerformance.cpp, [1463](#)
- argPrintUsage
 - GigEVisionPerformance.cpp, [1464](#)
- argRelease
 - GigEVisionPerformance.cpp, [1464](#)
- argUserSetFrames
 - GigEVisionPerformance.cpp, [1464](#)
- arrayLabelClassification
 - Inference.cpp, [1475](#)
- arrayLabelDetection
 - Inference.cpp, [1475](#)
- assign
 - Spinnaker::GenICam::gcstring, [860](#), [861](#)
- attach
 - Spinnaker::GenApi::FileProtocolAdapter, [843](#)
- AttachBuffer
 - Spinnaker::GenApi::CChunkAdapter, [677](#)
 - Spinnaker::GenApi::CChunkAdapterDcam, [680](#)
 - Spinnaker::GenApi::CChunkAdapterGEV, [685](#)
 - Spinnaker::GenApi::CChunkAdapterGeneric, [683](#)
 - Spinnaker::GenApi::CChunkAdapterU3V, [688](#)
 - Spinnaker::IDataStream, [903](#)
- AttachChunk
 - Spinnaker::GenApi::CChunkPort, [691](#)
- AttachEvent
 - Spinnaker::GenApi::CEventPort, [714](#)
- AttachNode
 - Spinnaker::GenApi::CEventPort, [714](#)
- AttachNodeMap
 - Spinnaker::GenApi::CChunkAdapter, [677](#)
 - Spinnaker::GenApi::CEventAdapter, [702](#)
- AttachPort
 - Spinnaker::GenApi::CChunkPort, [691](#)
- AttachStatistics_t, [487](#)

- NumAttachedChunks, [487](#)
- NumChunkPorts, [487](#)
- NumChunks, [487](#)
- AutoAlgorithmSelector
 - Spinnaker::Camera, [535](#)
- AutoAlgorithmSelectorEnums
 - CameraDefs Class, [85](#)
- AutoExposureControlLoopDamping
 - Spinnaker::Camera, [535](#)
- AutoExposureControlPriority
 - Spinnaker::Camera, [535](#)
- AutoExposureControlPriorityEnums
 - CameraDefs Class, [86](#)
- AutoExposureEVCompensation
 - Spinnaker::Camera, [536](#)
- AutoExposureExposureTimeLowerLimit
 - Spinnaker::Camera, [536](#)
- AutoExposureExposureTimeUpperLimit
 - Spinnaker::Camera, [536](#)
- AutoExposureGainLowerLimit
 - Spinnaker::Camera, [536](#)
- AutoExposureGainUpperLimit
 - Spinnaker::Camera, [537](#)
- AutoExposureGreyValueLowerLimit
 - Spinnaker::Camera, [537](#)
- AutoExposureGreyValueUpperLimit
 - Spinnaker::Camera, [537](#)
- AutoExposureLightingMode
 - Spinnaker::Camera, [537](#)
- AutoExposureLightingModeEnums
 - CameraDefs Class, [86](#)
- AutoExposureMeteringMode
 - Spinnaker::Camera, [538](#)
- AutoExposureMeteringModeEnums
 - CameraDefs Class, [87](#)
- AutoExposureTargetGreyValue
 - Spinnaker::Camera, [538](#)
- AutoExposureTargetGreyValueAuto
 - Spinnaker::Camera, [538](#)
- AutoExposureTargetGreyValueAutoEnums
 - CameraDefs Class, [87](#)
- AutoLock, [488](#)
 - Spinnaker::GenApi::AutoLock, [489](#)
 - Spinnaker::GenICam::AutoLock, [488](#)
- AutoPopulateAdapterInfo
 - AdapterConfig, [414](#)
- AutoPopulateAdvancedProperties
 - AdapterConfig, [414](#)
- AutoVector Class, [273](#)
- Automatic
 - Types Enums, [408](#)
- BMPOption, [494](#)
 - Spinnaker::BMPOption, [495](#)
- BalanceRatio
 - Spinnaker::Camera, [539](#)
- BalanceRatioSelector
 - Spinnaker::Camera, [539](#)
- BalanceRatioSelectorEnums
 - CameraDefs Class, [87](#)
- BalanceWhiteAuto
 - Spinnaker::Camera, [539](#)
- BalanceWhiteAutoDamping
 - Spinnaker::Camera, [539](#)
- BalanceWhiteAutoEnums
 - CameraDefs Class, [88](#)
- BalanceWhiteAutoLowerLimit
 - Spinnaker::Camera, [540](#)
- BalanceWhiteAutoProfile
 - Spinnaker::Camera, [540](#)
- BalanceWhiteAutoProfileEnums
 - CameraDefs Class, [88](#)
- BalanceWhiteAutoUpperLimit
 - Spinnaker::Camera, [540](#)
- BasePtr
 - Spinnaker::BasePtr, [491](#)
- BasePtr Class, [49](#)
 - operator==, [49](#)
- BasePtr< T, B >, [490](#)
- BeginAcquisition
 - Spinnaker::CameraBase, [655](#)
 - Spinnaker::ICameraBase, [882](#)
 - SpinnakerDirectShow.h, [232](#)
- Beginner
 - Types Enums, [409](#)
- binaryFile
 - Spinnaker::PGMOption, [1111](#)
 - Spinnaker::PPMOption, [1126](#)
- BinningHorizontal
 - Spinnaker::Camera, [540](#)
- BinningHorizontalMode
 - Spinnaker::Camera, [541](#)
- BinningHorizontalModeEnums
 - CameraDefs Class, [88](#)
- BinningSelector
 - Spinnaker::Camera, [541](#)
- BinningSelectorEnums
 - CameraDefs Class, [89](#)
- BinningVertical
 - Spinnaker::Camera, [541](#)
- BinningVerticalMode
 - Spinnaker::Camera, [541](#)
- BinningVerticalModeEnums
 - CameraDefs Class, [89](#)
- bitrate
 - Spinnaker::Video::H264Option, [879](#)
- BlackLevel
 - Spinnaker::Camera, [542](#)
- BlackLevelAuto
 - Spinnaker::Camera, [542](#)
- BlackLevelAutoBalance
 - Spinnaker::Camera, [542](#)
- BlackLevelAutoBalanceEnums
 - CameraDefs Class, [89](#)
- BlackLevelAutoEnums
 - CameraDefs Class, [90](#)
- BlackLevelClampingEnable

- Spinnaker::Camera, [542](#)
- BlackLevelRaw
 - Spinnaker::Camera, [542](#)
- BlackLevelSelector
 - Spinnaker::Camera, [543](#)
- BlackLevelSelectorEnums
 - CameraDefs Class, [90](#)
- BlockId
 - GVCP_EVENT_ITEM_EXTENDED_ID, [871](#)
 - GVCP_EVENT_ITEM, [869](#)
- BlockId64High
 - GVCP_EVENT_ITEM_EXTENDED_ID, [871](#)
- BlockId64Low
 - GVCP_EVENT_ITEM_EXTENDED_ID, [872](#)
- Boolean
 - Types Enums, [407](#)
- BooleanGetValue
 - Spinnaker Headers, [205](#)
- BooleanNode, [496](#)
 - Spinnaker::GenApi::BooleanNode, [497](#)
- BooleanNode Class, [277](#)
 - CBooleanRef, [277](#)
- BooleanSetValue
 - Spinnaker Headers, [205](#)
- bottomRightXCoord
 - Chunk Data Inference Class, [175](#)
- bottomRightYCoord
 - Chunk Data Inference Class, [175](#)
- boxType
 - Chunk Data Inference Class, [175](#)
- BufferHandling.cpp
 - AcquireImages, [1438](#)
 - ConfigureTrigger, [1438](#)
 - GrabNextImageByTrigger, [1439](#)
 - k_numLoops, [1438](#)
 - main, [1439](#)
 - numBuffers, [1438](#)
 - PrintDeviceInfo, [1439](#)
 - ResetTrigger, [1439](#)
 - RunSingleCamera, [1439](#)
 - SleepyWrapper, [1439](#)
 - z_numTriggers, [1438](#)
- BufferOwnership
 - Spinnaker Definitions, [224](#)
- bufferSize
 - Spinnaker Headers, [217](#)
- build
 - Spinnaker::LibraryVersion, [1069](#)
- c_str
 - Spinnaker::GenICam::gcstring, [861](#)
- CBasePtr
 - Pointer Class, [380](#)
- CBooleanPtr
 - Pointer Class, [380](#)
- CBooleanRef
 - BooleanNode Class, [277](#)
- CCMColorTemperature
 - Spinnaker Definitions, [224](#)
- CCMSensor
 - Spinnaker Definitions, [225](#)
- CCMSettings, [694](#)
 - Spinnaker::CCMSettings, [694](#)
- CCMType
 - Spinnaker Definitions, [225](#)
- CCategoryPtr
 - Pointer Class, [381](#)
- CCategoryRef
 - CategoryNode Class, [278](#)
- CChunkAdapter, [676](#)
 - Spinnaker::GenApi::CChunkAdapter, [677](#)
- CChunkAdapterDcam, [679](#)
 - Spinnaker::GenApi::CChunkAdapterDcam, [680](#)
- CChunkAdapterGEV, [684](#)
 - Spinnaker::GenApi::CChunkAdapterGEV, [685](#)
- CChunkAdapterGeneric, [682](#)
 - Spinnaker::GenApi::CChunkAdapterGeneric, [682](#)
- CChunkAdapterU3V, [686](#)
 - Spinnaker::GenApi::CChunkAdapterU3V, [687](#)
- CChunkPort, [689](#)
 - Spinnaker::GenApi::CChunkPort, [690](#)
- CChunkPortPtr
 - Pointer Class, [381](#)
- CCommandPtr
 - Pointer Class, [381](#)
- CCommandRef
 - CommandNode Class, [284](#)
- CDeviceInfoPtr
 - Pointer Class, [381](#)
- CEnumEntryPtr
 - Pointer Class, [381](#)
- CEnumEntryRef
 - EnumEntryNode Class, [289](#)
- CEnumerationPtr
 - Pointer Class, [381](#)
- CEnumerationRef
 - EnumNode Class, [290](#)
- CEnumerationTRef
 - Spinnaker::GenApi::CEnumerationTRef, [698](#)
- CEnumerationTRef< EnumT >, [695](#)
- CEventAdapter, [701](#)
 - Spinnaker::GenApi::CEventAdapter, [702](#)
- CEventAdapter1394, [703](#)
 - Spinnaker::GenApi::CEventAdapter1394, [704](#)
- CEventAdapterGEV, [708](#)
 - Spinnaker::GenApi::CEventAdapterGEV, [709](#)
- CEventAdapterGeneric, [705](#)
 - Spinnaker::GenApi::CEventAdapterGeneric, [706](#)
- CEventAdapterU3V, [710](#)
 - Spinnaker::GenApi::CEventAdapterU3V, [711](#)
- CEventPort, [712](#)
 - Spinnaker::GenApi::CEventPort, [713](#)
- CFeatureBag, [717](#)
 - Spinnaker::GenApi::CFeatureBag, [718](#)
- CFloatPtr, [720](#)
 - Spinnaker::GenApi::CFloatPtr, [721](#)
- CFloatRef

- FloatNode Class, [299](#)
- CGeneric_XMLLoaderParams, [722](#)
- CGlobalLock, [723](#)
 - Spinnaker::GenICam::CGlobalLock, [723](#), [724](#)
- CGlobalLockUnlocker, [725](#)
 - Spinnaker::GenICam::CGlobalLockUnlocker, [726](#)
- CHUNK_BASE_ADDRESS_REGISTER_LEN
 - ICChunkPort Interface, [314](#)
- CHUNK_BASE_ADDRESS_REGISTER
 - ICChunkPort Interface, [314](#)
- CHUNK_LENGTH_REGISTER_LEN
 - ICChunkPort Interface, [315](#)
- CHUNK_LENGTH_REGISTER
 - ICChunkPort Interface, [315](#)
- CIntegerPtr
 - Pointer Class, [382](#)
- CIntegerRef
 - IntegerNode Class, [354](#)
- CLock, [740](#), [742](#)
 - Spinnaker::GenApi::CLock, [743](#)
 - Spinnaker::GenICam::CLock, [741](#)
- CLockEx, [745](#), [746](#)
- CNodeCallback, [747](#)
 - Spinnaker::GenApi::CNodeCallback, [748](#)
- CNodeMapDynPtr
 - Pointer Class, [382](#)
- CNodeMapFactory, [750](#)
 - Spinnaker::GenApi::CNodeMapFactory, [752](#), [753](#)
- CNodeMapFactory::NodeStatistics_t, [1105](#)
- CNodeMapPtr
 - Pointer Class, [382](#)
- CNodeMapRef, [758](#)
 - Spinnaker GenApi Classes, [268](#)
 - Spinnaker::GenApi::CNodeMapRef, [759](#), [760](#)
- CNodeMapRefT< TCameraParams >, [761](#)
- CNodeMapRefT
 - Spinnaker GenApi Classes, [271](#)
- CNodePtr
 - Pointer Class, [382](#)
- CNodeRef
 - Spinnaker GenApi Classes, [268](#)
- COM_PORT_COUNT_MAX
 - SerialRxTx.cpp, [1494](#)
- COMMAND_MAGIC
 - Spinnaker::GenApi, [476](#)
- CPointer
 - Spinnaker::GenApi::CPointer, [774](#)
- CPointer< T, B >, [773](#)
- CPortConstructPtr
 - Pointer Class, [382](#)
- CPortImpl, [778](#)
 - Spinnaker::GenApi::CPortImpl, [779](#)
- CPortPtr
 - Pointer Class, [382](#)
- CPortRecorderPtr
 - Pointer Class, [383](#)
- CPortRecorderRef
 - PortRecorder Class, [387](#)
- CPortRef
 - PortNode Class, [386](#)
- CPortReplayPtr
 - Pointer Class, [383](#)
- CPortWriteList, [781](#)
 - Spinnaker::GenApi::CPortWriteList, [782](#)
- CPortWriteListPtr
 - Pointer Class, [383](#)
- CRCChecksum
 - DCAM_CHECKSUM, [795](#)
- CRegisterPortImpl, [785](#)
 - Spinnaker::GenApi::CRegisterPortImpl, [786](#)
- CRegisterPtr
 - Pointer Class, [383](#)
- CRegisterRef
 - RegisterNode Class, [391](#)
- CSelectorPtr
 - Pointer Class, [383](#)
- CSelectorRef
 - Spinnaker GenApi Classes, [268](#)
- CSelectorSet, [788](#)
 - Spinnaker::GenApi::CSelectorSet, [789](#)
- CStringPtr
 - Pointer Class, [383](#)
- CStringRef
 - StringNode Class, [395](#)
- CTestPortStruct
 - Spinnaker::GenApi::CTestPortStruct, [793](#)
- CTestPortStruct< CDataStruct >, [791](#)
- CValuePtr
 - Pointer Class, [384](#)
- CValueRef
 - ValueNode Class, [410](#)
- CacheChunkData
 - ICChunkPort Interface, [315](#)
 - Spinnaker::GenApi::PortNode, [1115](#)
- CalculateStatistics
 - Spinnaker::Image, [921](#)
 - Spinnaker::Image, [959](#)
- CallbackHandleType
 - Spinnaker GenApi Interfaces, [275](#)
- Camera, [499](#)
 - Spinnaker::Camera, [529](#)
- Camera Base Class, [51](#)
- Camera Base Interface Class, [255](#)
- Camera Class, [50](#)
- Camera List Class, [168](#)
- CameraBase, [652](#)
 - Spinnaker::CameraBase, [654](#), [655](#)
 - Spinnaker::TransportLayerDevice, [1167](#)
 - Spinnaker::TransportLayerStream, [1189](#)
- CameraCloseFile
 - Inference.cpp, [1472](#)
- CameraDefs Class, [52](#)
 - AcquisitionModeEnums, [84](#)
 - AcquisitionStatusSelectorEnums, [84](#)
 - ActionUnconditionalModeEnums, [85](#)
 - AdcBitDepthEnums, [85](#)

- AutoAlgorithmSelectorEnums, 85
- AutoExposureControlPriorityEnums, 86
- AutoExposureLightingModeEnums, 86
- AutoExposureMeteringModeEnums, 87
- AutoExposureTargetGreyValueAutoEnums, 87
- BalanceRatioSelectorEnums, 87
- BalanceWhiteAutoEnums, 88
- BalanceWhiteAutoProfileEnums, 88
- BinningHorizontalModeEnums, 88
- BinningSelectorEnums, 89
- BinningVerticalModeEnums, 89
- BlackLevelAutoBalanceEnums, 89
- BlackLevelAutoEnums, 90
- BlackLevelSelectorEnums, 90
- ChunkBlackLevelSelectorEnums, 90
- ChunkCounterSelectorEnums, 91
- ChunkEncoderSelectorEnums, 91
- ChunkEncoderStatusEnums, 91
- ChunkExposureTimeSelectorEnums, 92
- ChunkGainSelectorEnums, 92
- ChunkImageComponentEnums, 92
- ChunkPixelFormatEnums, 93
- ChunkRegionIDEnums, 93
- ChunkScan3dCoordinateReferenceSelector↔
Enums, 94
- ChunkScan3dCoordinateSelectorEnums, 94
- ChunkScan3dCoordinateSystemEnums, 94
- ChunkScan3dCoordinateSystemReferenceEnums,
95
- ChunkScan3dCoordinateTransformSelector↔
Enums, 95
- ChunkScan3dDistanceUnitEnums, 95
- ChunkScan3dOutputModeEnums, 96
- ChunkSelectorEnums, 96
- ChunkSourceIDEnums, 97
- ChunkTimerSelectorEnums, 97
- ChunkTransferStreamIDEnums, 98
- CICConfigurationEnums, 98
- CITimeSlotsCountEnums, 98
- ColorTransformationSelectorEnums, 99
- ColorTransformationValueSelectorEnums, 99
- CompressionSaturationPriorityEnums, 100
- CounterEventActivationEnums, 100
- CounterEventSourceEnums, 100
- CounterResetActivationEnums, 101
- CounterResetSourceEnums, 101
- CounterSelectorEnums, 102
- CounterStatusEnums, 102
- CounterTriggerActivationEnums, 103
- CounterTriggerSourceEnums, 103
- CxpConnectionTestModeEnums, 104
- CxpLinkConfigurationEnums, 104
- CxpLinkConfigurationPreferredEnums, 105
- CxpLinkConfigurationStatusEnums, 106
- CxpPoCxpStatusEnums, 107
- DecimationHorizontalModeEnums, 107
- DecimationSelectorEnums, 107
- DecimationVerticalModeEnums, 108
- DefectCorrectionModeEnums, 108
- DeinterlacingEnums, 108
- DeviceCharacterSetEnums, 109
- DeviceClockSelectorEnums, 109
- DeviceConnectionStatusEnums, 109
- DeviceIndicatorModeEnums, 110
- DeviceLinkHeartbeatModeEnums, 110
- DeviceLinkThroughputLimitModeEnums, 110
- DevicePowerSupplySelectorEnums, 110
- DeviceRegistersEndiannessEnums, 111
- DeviceScanTypeEnums, 111
- DeviceSerialPortBaudRateEnums, 111
- DeviceSerialPortSelectorEnums, 112
- DeviceStreamChannelEndiannessEnums, 112
- DeviceStreamChannelTypeEnums, 112
- DeviceTLTypeEnums, 114
- DeviceTapGeometryEnums, 113
- DeviceTemperatureSelectorEnums, 114
- DeviceTypeEnums, 115
- EncoderModeEnums, 115
- EncoderOutputModeEnums, 115
- EncoderResetActivationEnums, 116
- EncoderResetSourceEnums, 116
- EncoderSelectorEnums, 117
- EncoderSourceAEnums, 118
- EncoderSourceBEnums, 118
- EncoderStatusEnums, 118
- EventNotificationEnums, 119
- EventSelectorEnums, 119
- ExposureActiveModeEnums, 119
- ExposureAutoEnums, 119
- ExposureModeEnums, 120
- ExposureTimeModeEnums, 120
- ExposureTimeSelectorEnums, 121
- FileOpenModeEnums, 121
- FileOperationSelectorEnums, 121
- FileOperationStatusEnums, 122
- FileSelectorEnums, 122
- GainAutoBalanceEnums, 122
- GainAutoEnums, 124
- GainSelectorEnums, 124
- GevCCPEnums, 124
- GevCurrentPhysicalLinkConfigurationEnums, 125
- GevGVCPExtendedStatusCodesSelectorEnums,
125
- GevGVSPExtendedIDModeEnums, 125
- GevIEEE1588ClockAccuracyEnums, 126
- GevIEEE1588ModeEnums, 126
- GevIEEE1588StatusEnums, 126
- GevIPConfigurationStatusEnums, 127
- GevPhysicalLinkConfigurationEnums, 127
- GevSupportedOptionSelectorEnums, 127
- ImageComponentSelectorEnums, 128
- ImageCompressionJPEGFormatOptionEnums,
129
- ImageCompressionModeEnums, 129
- ImageCompressionRateOptionEnums, 130
- LUTSelectorEnums, 134

- LineFormatEnums, [130](#)
- LineInputFilterSelectorEnums, [130](#)
- LineModeEnums, [131](#)
- LineSelectorEnums, [131](#)
- LineSourceEnums, [131](#)
- LogicBlockLUTInputActivationEnums, [132](#)
- LogicBlockLUTInputSelectorEnums, [132](#)
- LogicBlockLUTInputSourceEnums, [133](#)
- LogicBlockLUTSelectorEnums, [133](#)
- LogicBlockSelectorEnums, [134](#)
- PixelColorFilterEnums, [134](#)
- PixelFormatEnums, [135](#)
- PixelFormatInfoSelectorEnums, [140](#)
- PixelSizeEnums, [146](#)
- RegionDestinationEnums, [147](#)
- RegionModeEnums, [147](#)
- RegionSelectorEnums, [147](#)
- RgbTransformLightSourceEnums, [147](#)
- Scan3dCoordinateReferenceSelectorEnums, [148](#)
- Scan3dCoordinateSelectorEnums, [148](#)
- Scan3dCoordinateSystemEnums, [149](#)
- Scan3dCoordinateSystemReferenceEnums, [149](#)
- Scan3dCoordinateTransformSelectorEnums, [149](#)
- Scan3dDistanceUnitEnums, [150](#)
- Scan3dOutputModeEnums, [150](#)
- SensorDigitizationTapsEnums, [151](#)
- SensorShutterModeEnums, [151](#)
- SensorTapsEnums, [152](#)
- SequencerConfigurationModeEnums, [152](#)
- SequencerConfigurationValidEnums, [153](#)
- SequencerModeEnums, [153](#)
- SequencerSetValidEnums, [153](#)
- SequencerTriggerActivationEnums, [153](#)
- SequencerTriggerSourceEnums, [154](#)
- SerialPortBaudRateEnums, [154](#)
- SerialPortParityEnums, [155](#)
- SerialPortSelectorEnums, [155](#)
- SerialPortSourceEnums, [155](#)
- SerialPortStopBitsEnums, [156](#)
- SoftwareSignalSelectorEnums, [156](#)
- SourceSelectorEnums, [156](#)
- TestPatternEnums, [156](#)
- TestPatternGeneratorSelectorEnums, [157](#)
- TimerSelectorEnums, [157](#)
- TimerStatusEnums, [157](#)
- TimerTriggerActivationEnums, [158](#)
- TimerTriggerSourceEnums, [158](#)
- TransferComponentSelectorEnums, [160](#)
- TransferControlModeEnums, [160](#)
- TransferOperationModeEnums, [160](#)
- TransferQueueModeEnums, [161](#)
- TransferSelectorEnums, [161](#)
- TransferStatusSelectorEnums, [161](#)
- TransferTriggerActivationEnums, [162](#)
- TransferTriggerModeEnums, [162](#)
- TransferTriggerSelectorEnums, [162](#)
- TransferTriggerSourceEnums, [163](#)
- TriggerActivationEnums, [164](#)
- TriggerModeEnums, [164](#)
- TriggerOverlapEnums, [165](#)
- TriggerSelectorEnums, [165](#)
- TriggerSourceEnums, [165](#)
- UserOutputSelectorEnums, [166](#)
- UserSetDefaultEnums, [166](#)
- UserSetSelectorEnums, [167](#)
- WhiteClipSelectorEnums, [167](#)
- CameraDeleteFile
 - Inference.cpp, [1472](#)
- cameraGrabInfoMap
 - AcquisitionMultipleCameraRecovery.cpp, [1431](#)
- CameraInternal
 - Spinnaker::ICameraBase, [888](#)
 - Spinnaker::TransportLayerDevice, [1167](#)
 - Spinnaker::TransportLayerStream, [1189](#)
- CameraList, [667](#)
 - Spinnaker::CameraList, [668](#)
- CameraListImpl
 - Spinnaker::ICameraList, [892](#)
- CameraOpenFile
 - Inference.cpp, [1472](#)
- CameraPtr, [672](#)
 - CameraPtr Class, [169](#), [170](#)
- CameraPtr Class, [169](#)
 - CameraPtr, [169](#), [170](#)
- CameraWriteToFile
 - Inference.cpp, [1472](#)
- CastToIDestroy
 - Spinnaker GenApi Classes, [271](#)
- CategoryNode, [674](#)
 - Spinnaker::GenApi::CategoryNode, [675](#)
- CategoryNode Class, [278](#)
 - CCategoryRef, [278](#)
- causeSpinnakerException
 - ExceptionHandling.cpp, [1452](#)
- causeStandardException
 - ExceptionHandling.cpp, [1452](#)
- centerXCoord
 - Chunk Data Inference Class, [175](#)
- centerYCoord
 - Chunk Data Inference Class, [175](#)
- ChangeHeightAndGain
 - NodeMapCallback.cpp, [1482](#)
- CheckBufferLayout
 - Spinnaker::GenApi::CChunkAdapter, [678](#)
 - Spinnaker::GenApi::CChunkAdapterDcam, [681](#)
 - Spinnaker::GenApi::CChunkAdapterGEV, [686](#)
 - Spinnaker::GenApi::CChunkAdapterGeneric, [683](#)
 - Spinnaker::GenApi::CChunkAdapterU3V, [688](#)
- CheckCRC
 - Spinnaker::GenApi::CChunkAdapterDcam, [681](#)
 - Spinnaker::Image, [921](#)
 - Spinnaker::Image, [959](#)
- CheckChunkID
 - Spinnaker::GenApi::CChunkPort, [691](#)
- CheckEventID
 - Spinnaker::GenApi::CEventPort, [714](#)

- CheckGevEnabled
 - EnumerationEvents.cpp, [1450](#)
- chosenChunkData
 - ChunkData.cpp, [1442](#)
- chosenEvent
 - DeviceEvents.cpp, [1448](#)
- chosenException
 - ExceptionHandling.cpp, [1452](#)
- chosenFileUploadPersistence
 - Inference.cpp, [1476](#)
- chosenInferenceNetworkType
 - Inference.cpp, [1476](#)
- chosenRead
 - NodeMapInfo.cpp, [1486](#)
- chosenTrigger
 - Trigger.cpp, [1500](#)
 - Trigger_QuickSpin.cpp, [1502](#)
- chosenVideoType
 - SaveToAvi.cpp, [1492](#)
- Chunk Data Inference Class, [172](#)
 - ~InferenceBoundingBoxResult, [174](#)
 - bottomRightXCoord, [175](#)
 - bottomRightYCoord, [175](#)
 - boxType, [175](#)
 - centerXCoord, [175](#)
 - centerYCoord, [175](#)
 - circle, [176](#)
 - classId, [176](#)
 - confidence, [176](#)
 - GetBoxAt, [173](#)
 - GetBoxCount, [173](#)
 - GetBoxSize, [173](#)
 - GetVersion, [173](#)
 - InferenceBoundingBoxResult, [174](#)
 - operator=, [174](#)
 - radius, [176](#)
 - rect, [176](#)
 - rotatedRect, [176](#)
 - rotationAngle, [176](#)
 - topLeftXCoord, [176](#), [177](#)
 - topLeftYCoord, [177](#)
- ChunkAdapter Class, [279](#)
- ChunkAdapterDcam Class, [280](#)
- ChunkAdapterGEV Class, [282](#)
- ChunkAdapterGeneric Class, [281](#)
- ChunkAdapterU3V Class, [411](#)
- ChunkBlackLevel
 - Spinnaker::Camera, [543](#)
- ChunkBlackLevelSelector
 - Spinnaker::Camera, [543](#)
- ChunkBlackLevelSelectorEnums
 - CameraDefs Class, [90](#)
- ChunkCRC
 - Spinnaker::Camera, [544](#)
- ChunkCompressionMode
 - Spinnaker::Camera, [543](#)
- ChunkCompressionRatio
 - Spinnaker::Camera, [543](#)
- ChunkCounterSelector
 - Spinnaker::Camera, [544](#)
- ChunkCounterSelectorEnums
 - CameraDefs Class, [91](#)
- ChunkCounterValue
 - Spinnaker::Camera, [544](#)
- ChunkData, [727](#)
 - Spinnaker::ChunkData, [729](#), [730](#)
- ChunkData Class, [171](#)
- ChunkData.cpp
 - AcquireImages, [1441](#)
 - chosenChunkData, [1442](#)
 - chunkDataType, [1440](#)
 - ConfigureChunkData, [1441](#)
 - DisableChunkData, [1441](#)
 - DisplayChunkData, [1441](#)
 - main, [1441](#)
 - PrintDeviceInfo, [1441](#)
 - RunSingleCamera, [1442](#)
- chunkDataType
 - ChunkData.cpp, [1440](#)
- ChunkEnable
 - Spinnaker::Camera, [544](#)
- ChunkEncoderSelector
 - Spinnaker::Camera, [544](#)
- ChunkEncoderSelectorEnums
 - CameraDefs Class, [91](#)
- ChunkEncoderStatus
 - Spinnaker::Camera, [544](#)
- ChunkEncoderStatusEnums
 - CameraDefs Class, [91](#)
- ChunkEncoderValue
 - Spinnaker::Camera, [545](#)
- ChunkExposureEndLineStatusAll
 - Spinnaker::Camera, [545](#)
- ChunkExposureTime
 - Spinnaker::Camera, [545](#)
- ChunkExposureTimeSelector
 - Spinnaker::Camera, [545](#)
- ChunkExposureTimeSelectorEnums
 - CameraDefs Class, [92](#)
- ChunkFrameID
 - Spinnaker::Camera, [545](#)
- ChunkGain
 - Spinnaker::Camera, [545](#)
- ChunkGainSelector
 - Spinnaker::Camera, [546](#)
- ChunkGainSelectorEnums
 - CameraDefs Class, [92](#)
- ChunkHeight
 - Spinnaker::Camera, [546](#)
- ChunkID
 - DCAM_CHUNK_TRAILER, [796](#)
 - GVCP_CHUNK_TRAILER, [868](#)
 - SingleChunkData_t, [1131](#)
 - SingleChunkDataStr_t, [1131](#)
 - U3V_CHUNK_TRAILER, [1202](#)
- ChunkImage

- Spinnaker::Camera, [546](#)
- ChunkImageComponent
 - Spinnaker::Camera, [546](#)
- ChunkImageComponentEnums
 - CameraDefs Class, [92](#)
- ChunkInferenceBoundingBoxResult
 - Spinnaker::Camera, [546](#)
- ChunkInferenceConfidence
 - Spinnaker::Camera, [546](#)
- ChunkInferenceFrameld
 - Spinnaker::Camera, [547](#)
- ChunkInferenceResult
 - Spinnaker::Camera, [547](#)
- ChunkLength
 - DCAM_CHUNK_TRAILER, [796](#)
 - GVCP_CHUNK_TRAILER, [868](#)
 - SingleChunkData_t, [1131](#)
 - SingleChunkDataStr_t, [1131](#)
 - U3V_CHUNK_TRAILER, [1202](#)
- ChunkLinePitch
 - Spinnaker::Camera, [547](#)
- ChunkLineStatusAll
 - Spinnaker::Camera, [547](#)
- ChunkModeActive
 - Spinnaker::Camera, [547](#)
- ChunkOffset
 - SingleChunkData_t, [1131](#)
 - SingleChunkDataStr_t, [1132](#)
- ChunkOffsetX
 - Spinnaker::Camera, [547](#)
- ChunkOffsetY
 - Spinnaker::Camera, [548](#)
- ChunkPartSelector
 - Spinnaker::Camera, [548](#)
- ChunkPixelDynamicRangeMax
 - Spinnaker::Camera, [548](#)
- ChunkPixelDynamicRangeMin
 - Spinnaker::Camera, [548](#)
- ChunkPixelFormat
 - Spinnaker::Camera, [548](#)
- ChunkPixelFormatEnums
 - CameraDefs Class, [93](#)
- ChunkPort Class, [283](#)
- ChunkRegionIDEnums
 - CameraDefs Class, [93](#)
- ChunkRegionID
 - Spinnaker::Camera, [548](#)
- ChunkScan3dAxisMax
 - Spinnaker::Camera, [549](#)
- ChunkScan3dAxisMin
 - Spinnaker::Camera, [549](#)
- ChunkScan3dCoordinateOffset
 - Spinnaker::Camera, [549](#)
- ChunkScan3dCoordinateReferenceSelector
 - Spinnaker::Camera, [549](#)
- ChunkScan3dCoordinateReferenceSelectorEnums
 - CameraDefs Class, [94](#)
- ChunkScan3dCoordinateReferenceValue
 - Spinnaker::Camera, [549](#)
- ChunkScan3dCoordinateScale
 - Spinnaker::Camera, [549](#)
- ChunkScan3dCoordinateSelector
 - Spinnaker::Camera, [550](#)
- ChunkScan3dCoordinateSelectorEnums
 - CameraDefs Class, [94](#)
- ChunkScan3dCoordinateSystem
 - Spinnaker::Camera, [550](#)
- ChunkScan3dCoordinateSystemEnums
 - CameraDefs Class, [94](#)
- ChunkScan3dCoordinateSystemReference
 - Spinnaker::Camera, [550](#)
- ChunkScan3dCoordinateSystemReferenceEnums
 - CameraDefs Class, [95](#)
- ChunkScan3dCoordinateTransformSelector
 - Spinnaker::Camera, [550](#)
- ChunkScan3dCoordinateTransformSelectorEnums
 - CameraDefs Class, [95](#)
- ChunkScan3dDistanceUnit
 - Spinnaker::Camera, [550](#)
- ChunkScan3dDistanceUnitEnums
 - CameraDefs Class, [95](#)
- ChunkScan3dInvalidDataFlag
 - Spinnaker::Camera, [550](#)
- ChunkScan3dInvalidDataValue
 - Spinnaker::Camera, [551](#)
- ChunkScan3dOutputMode
 - Spinnaker::Camera, [551](#)
- ChunkScan3dOutputModeEnums
 - CameraDefs Class, [96](#)
- ChunkScan3dTransformValue
 - Spinnaker::Camera, [551](#)
- ChunkScanLineSelector
 - Spinnaker::Camera, [551](#)
- ChunkSelector
 - Spinnaker::Camera, [551](#)
- ChunkSelectorEnums
 - CameraDefs Class, [96](#)
- ChunkSequencerSetActive
 - Spinnaker::Camera, [551](#)
- ChunkSerialData
 - Spinnaker::Camera, [552](#)
- ChunkSerialDataLength
 - Spinnaker::Camera, [552](#)
- ChunkSerialReceiveOverflow
 - Spinnaker::Camera, [552](#)
- ChunkSourceIDEnums
 - CameraDefs Class, [97](#)
- ChunkSourceID
 - Spinnaker::Camera, [552](#)
- ChunkStreamChannelID
 - Spinnaker::Camera, [552](#)
- ChunkTimerSelector
 - Spinnaker::Camera, [552](#)
- ChunkTimerSelectorEnums
 - CameraDefs Class, [97](#)
- ChunkTimerValue

- Spinnaker::Camera, [553](#)
- ChunkTimestamp
 - Spinnaker::Camera, [553](#)
- ChunkTimestampLatchValue
 - Spinnaker::Camera, [553](#)
- ChunkTransferBlockID
 - Spinnaker::Camera, [553](#)
- ChunkTransferQueueCurrentBlockCount
 - Spinnaker::Camera, [553](#)
- ChunkTransferStreamIDEnums
 - CameraDefs Class, [98](#)
- ChunkTransferStreamID
 - Spinnaker::Camera, [553](#)
- ChunkWidth
 - Spinnaker::Camera, [554](#)
- circle
 - Chunk Data Inference Class, [176](#)
- CL
 - Types Enums, [408](#)
- CIConfiguration
 - Spinnaker::Camera, [554](#)
- CIConfigurationEnums
 - CameraDefs Class, [98](#)
- CITimeSlotsCount
 - Spinnaker::Camera, [554](#)
- CITimeSlotsCountEnums
 - CameraDefs Class, [98](#)
- classId
 - Chunk Data Inference Class, [176](#)
- CleanUp
 - SerialRxTx.cpp, [1496](#)
- CleanupChunkAdapter
 - Spinnaker::IDataStream, [903](#)
- Clear
 - Spinnaker::CameraList, [669](#)
 - Spinnaker::ICameraList, [890](#)
 - Spinnaker::IInterfaceList, [948](#)
 - Spinnaker::InterfaceList, [1048](#)
- ClearCache
 - Spinnaker::GenApi::CChunkPort, [691](#)
 - Spinnaker::GenApi::CNodeMapFactory, [754](#)
- ClearCaches
 - Spinnaker::GenApi::CChunkAdapter, [678](#)
- ClearXMLCache
 - Spinnaker::GenApi::NodeMap, [1099](#)
- Close
 - Spinnaker::Video::SpinVideo, [1134](#)
- close
 - Spinnaker::GenApi::IDevFileStreamBase, [910](#)
 - Spinnaker::GenApi::IDevFileStreamBuf, [912](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [1107](#)
 - Spinnaker::GenApi::ODevFileStreamBuf, [1109](#)
- CloseFile
 - FileAccess_QuickSpin.cpp, [1456](#)
- closeFile
 - Spinnaker::GenApi::FileProtocolAdapter, [844](#)
- ColorProcessingAlgorithm
 - Spinnaker Definitions, [225](#)
- ColorTemperature
 - Spinnaker::CCMSettings, [695](#)
- ColorTransformationEnable
 - Spinnaker::Camera, [554](#)
- ColorTransformationSelector
 - Spinnaker::Camera, [554](#)
- ColorTransformationSelectorEnums
 - CameraDefs Class, [99](#)
- ColorTransformationValue
 - Spinnaker::Camera, [555](#)
- ColorTransformationValueSelector
 - Spinnaker::Camera, [555](#)
- ColorTransformationValueSelectorEnums
 - CameraDefs Class, [99](#)
- Combine
 - INode Interface, [337](#)
- Command
 - GVCP_REQUEST_HEADER, [877](#)
- CommandExecute
 - Spinnaker Headers, [206](#)
- CommandHeader
 - U3V_EVENT_MESSAGE, [1205](#)
- CommandId
 - U3V_COMMAND_HEADER, [1202](#)
- CommandNode, [766](#)
 - Spinnaker::GenApi::CommandNode, [767](#), [768](#)
- CommandNode Class, [284](#)
 - CCommandRef, [284](#)
- compare
 - Spinnaker::GenICam::gcstring, [861](#)
- Compatibility.h
 - FMT_I64, [1292](#)
- CompressedImageInfo, [769](#)
 - CompressedImageInfo, [769](#)
 - compressedImageSize, [770](#)
 - fileName, [770](#)
 - imageHeight, [770](#)
 - imagePixelFormat, [770](#)
 - imageWidth, [770](#)
 - imageXOffset, [770](#)
 - imageYOffset, [771](#)
- compressedImageSize
 - CompressedImageInfo, [770](#)
- compression
 - Spinnaker::TIFFOption, [1164](#)
- Compression.cpp
 - AcquireImages, [1443](#)
 - DisableImageChunkData, [1443](#)
 - DisableImageCompression, [1443](#)
 - enableChunkData, [1444](#)
 - EnableImageChunkData, [1443](#)
 - EnableImageCompression, [1443](#)
 - main, [1443](#)
 - PrintDeviceInfo, [1444](#)
 - ProcessCompressedImagesFromFile, [1444](#)
 - RunSingleCamera, [1444](#)
- compressionLevel
 - Spinnaker::PNGOption, [1112](#)

- CompressionMethod
 - Spinnaker::TIFFOption, [1163](#)
- CompressionRatio
 - Spinnaker::Camera, [555](#)
- CompressionSaturationPriority
 - Spinnaker::Camera, [555](#)
- CompressionSaturationPriorityEnums
 - CameraDefs Class, [100](#)
- confidence
 - Chunk Data Inference Class, [176](#)
 - Spinnaker::DeviceEventInferenceData, [804](#)
- ConfigureActionControl
 - ActionCommand.cpp, [1436](#)
- ConfigureAdapter
 - AdapterConfig, [414](#)
- ConfigureCallbacks
 - NodeMapCallback.cpp, [1482](#)
- ConfigureCamera
 - AcquisitionMultipleCameraRecovery.cpp, [1430](#)
- ConfigureCameras
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1432](#)
- ConfigureChunkData
 - ActionCommand.cpp, [1436](#)
 - ChunkData.cpp, [1441](#)
 - Inference.cpp, [1473](#)
- ConfigureCustomImageSettings
 - ImageFormatControl.cpp, [1468](#)
 - ImageFormatControl_QuickSpin.cpp, [1469](#)
- ConfigureDevice
 - SerialRxTx.cpp, [1496](#)
- ConfigureDeviceEvents
 - DeviceEvents.cpp, [1447](#)
- ConfigureDigitalIO
 - CounterAndTimer.cpp, [1445](#)
- ConfigureExposure
 - Exposure.cpp, [1453](#)
 - Exposure_QuickSpin.cpp, [1454](#)
- ConfigureExposureandTrigger
 - CounterAndTimer.cpp, [1445](#)
- ConfigureIEEE1588
 - ActionCommand.cpp, [1436](#)
- ConfigureImageEvents
 - ImageEvents.cpp, [1466](#)
- ConfigureInference
 - Inference.cpp, [1473](#)
- ConfigureInterface
 - ActionCommand.cpp, [1436](#)
- ConfigureLogicBlock
 - LogicBlock.cpp, [1478](#)
- ConfigureLookupTables
 - LookupTable.cpp, [1480](#)
- ConfigureOtherNodes
 - ActionCommand.cpp, [1436](#)
- ConfigureSequencerPartOne
 - Sequencer.cpp, [1492](#)
- ConfigureSequencerPartTwo
 - Sequencer.cpp, [1493](#)
- ConfigureStream
 - Polarization.cpp, [1488](#)
- ConfigureTestPattern
 - Inference.cpp, [1473](#)
- ConfigureTrigger
 - ActionCommand.cpp, [1436](#)
 - BufferHandling.cpp, [1438](#)
 - Inference.cpp, [1473](#)
 - LogicBlock.cpp, [1479](#)
 - Trigger.cpp, [1499](#)
 - Trigger_QuickSpin.cpp, [1501](#)
- ConfigureUserSet1
 - AcquisitionMultipleCameraRecovery.cpp, [1430](#)
- Connect
 - INodeMap Interface, [346](#), [347](#)
 - Spinnaker::GenApi::NodeMap, [1099](#)
- Container Class, [285](#)
- Conversion, [417](#)
 - NumToCString, [417](#)
- Convert
 - Spinnaker::IImage, [921](#), [922](#)
 - Spinnaker::Image, [959–961](#)
- Counter, [771](#)
 - Spinnaker::GenApi::Counter, [771](#)
- Counter Class, [286](#)
- CounterAndTimer.cpp
 - AcquireImages, [1445](#)
 - ConfigureDigitalIO, [1445](#)
 - ConfigureExposureandTrigger, [1445](#)
 - main, [1445](#)
 - PrintDeviceInfo, [1445](#)
 - ResetTrigger, [1446](#)
 - RunSingleCamera, [1446](#)
 - SetupCounterAndTimer, [1446](#)
- CounterDelay
 - Spinnaker::Camera, [555](#)
- CounterDuration
 - Spinnaker::Camera, [556](#)
- CounterEventActivation
 - Spinnaker::Camera, [556](#)
- CounterEventActivationEnums
 - CameraDefs Class, [100](#)
- CounterEventSource
 - Spinnaker::Camera, [556](#)
- CounterEventSourceEnums
 - CameraDefs Class, [100](#)
- CounterReset
 - Spinnaker::Camera, [556](#)
- CounterResetActivation
 - Spinnaker::Camera, [556](#)
- CounterResetActivationEnums
 - CameraDefs Class, [101](#)
- CounterResetSource
 - Spinnaker::Camera, [556](#)
- CounterResetSourceEnums
 - CameraDefs Class, [101](#)
- CounterSelector
 - Spinnaker::Camera, [557](#)
- CounterSelectorEnums

- CameraDefs Class, [102](#)
- CounterStart
 - PerformanceCounter, [419](#)
- CounterStatus
 - Spinnaker::Camera, [557](#)
- CounterStatusEnums
 - CameraDefs Class, [102](#)
- CounterTriggerActivation
 - Spinnaker::Camera, [557](#)
- CounterTriggerActivationEnums
 - CameraDefs Class, [103](#)
- CounterTriggerSource
 - Spinnaker::Camera, [557](#)
- CounterTriggerSourceEnums
 - CameraDefs Class, [103](#)
- CounterValue
 - Spinnaker::Camera, [557](#)
- CounterValueAtReset
 - Spinnaker::Camera, [557](#)
- CpuUsageInfo, [784](#)
- cpuUsageInfo
 - GigEVisionPerformance.cpp, [1464](#)
- CpuUtil, [418](#)
 - GetCpuStats, [418](#)
 - StartCpuTracing, [418](#)
 - StopCpuTracing, [418](#)
- CpuUtil::CpuUsageInfo
 - dummy, [784](#)
- Create
 - Spinnaker::Image, [961](#), [962](#)
- CreateAndSaveAolpDolpImages
 - Polarization.cpp, [1488](#)
- CreateAndSaveGlareReducedImage
 - Polarization.cpp, [1488](#)
- CreateAndSaveStokesImages
 - Polarization.cpp, [1488](#)
- CreateAolp
 - Spinnaker::ImageUtilityPolarization, [1012](#)
- CreateColorCorrected
 - Spinnaker::ImageUtilityCCM, [1004](#), [1005](#)
- CreateDolp
 - Spinnaker::ImageUtilityPolarization, [1014](#)
- CreateEmptyNodeMap
 - Spinnaker::GenApi::CNodeMapFactory, [755](#)
- CreateFiles
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1433](#)
- CreateGlareReduced
 - Spinnaker::ImageUtilityPolarization, [1014](#), [1015](#)
- CreateHeatmap
 - Spinnaker::ImageUtilityHeatmap, [1007](#), [1008](#)
- CreateHeatmapImages
 - Polarization.cpp, [1488](#)
- CreateNodeDataFromNodeMap
 - Spinnaker::GenApi::CNodeMapFactory, [755](#)
- CreateNodeMap
 - Spinnaker::GenApi::CNodeMapFactory, [755](#)
- CreateNormalized
 - Spinnaker::ImageUtility, [1001](#)–[1003](#)
- CreateNormalizedImage
 - Polarization.cpp, [1488](#)
- CreateScaled
 - Spinnaker::ImageUtility, [1003](#)
- CreateShared
 - Spinnaker::Image, [963](#)
- CreateStokesS0
 - Spinnaker::ImageUtilityPolarization, [1015](#), [1016](#)
- CreateStokesS1
 - Spinnaker::ImageUtilityPolarization, [1016](#), [1017](#)
- CreateStokesS2
 - Spinnaker::ImageUtilityPolarization, [1017](#)
- Custom
 - Types Enums, [407](#)
- CustomCCMCode
 - Spinnaker::CCMSettings, [695](#)
- CxpConnectionSelector
 - Spinnaker::Camera, [558](#)
- CxpConnectionTestErrorCount
 - Spinnaker::Camera, [558](#)
- CxpConnectionTestMode
 - Spinnaker::Camera, [558](#)
- CxpConnectionTestModeEnums
 - CameraDefs Class, [104](#)
- CxpConnectionTestPacketCount
 - Spinnaker::Camera, [558](#)
- CxpLinkConfiguration
 - Spinnaker::Camera, [558](#)
- CxpLinkConfigurationEnums
 - CameraDefs Class, [104](#)
- CxpLinkConfigurationPreferred
 - Spinnaker::Camera, [558](#)
- CxpLinkConfigurationPreferredEnums
 - CameraDefs Class, [105](#)
- CxpLinkConfigurationStatus
 - Spinnaker::Camera, [559](#)
- CxpLinkConfigurationStatusEnums
 - CameraDefs Class, [106](#)
- CxpPoCxpAuto
 - Spinnaker::Camera, [559](#)
- CxpPoCxpStatus
 - Spinnaker::Camera, [559](#)
- CxpPoCxpStatusEnums
 - CameraDefs Class, [107](#)
- CxpPoCxpTripReset
 - Spinnaker::Camera, [559](#)
- CxpPoCxpTurnOff
 - Spinnaker::Camera, [559](#)
- DATA_BITS
 - SerialRxTx.cpp, [1495](#)
- DCAM_CHECKSUM, [795](#)
 - CRCChecksum, [795](#)
- DCAM_CHUNK_TRAILER, [795](#)
 - ChunkID, [796](#)
 - ChunkLength, [796](#)
 - InverseChunkLength, [796](#)
- DEPRECATED_CLASS
 - AVI Recorder Class, [46](#)

- Data
 - GVCP_EVENTDATA_REQUEST_EXTENDED_ID, [876](#)
 - GVCP_EVENTDATA_REQUEST, [875](#)
- DelInit
 - Spinnaker::CameraBase, [655](#)
 - Spinnaker::ICameraBase, [882](#)
- DecimationHorizontal
 - Spinnaker::Camera, [559](#)
- DecimationHorizontalMode
 - Spinnaker::Camera, [560](#)
- DecimationHorizontalModeEnums
 - CameraDefs Class, [107](#)
- DecimationSelector
 - Spinnaker::Camera, [560](#)
- DecimationSelectorEnums
 - CameraDefs Class, [107](#)
- DecimationVertical
 - Spinnaker::Camera, [560](#)
- DecimationVerticalMode
 - Spinnaker::Camera, [560](#)
- DecimationVerticalModeEnums
 - CameraDefs Class, [108](#)
- Decreasing
 - Types Enums, [408](#)
- DeepCopy
 - Spinnaker::Image, [922](#)
 - Spinnaker::Image, [963](#)
- DefectCorrectStaticEnable
 - Spinnaker::Camera, [561](#)
- DefectCorrectionMode
 - Spinnaker::Camera, [561](#)
- DefectCorrectionModeEnums
 - CameraDefs Class, [108](#)
- DefectTableApply
 - Spinnaker::Camera, [561](#)
- DefectTableCoordinateX
 - Spinnaker::Camera, [561](#)
- DefectTableCoordinateY
 - Spinnaker::Camera, [561](#)
- DefectTableFactoryRestore
 - Spinnaker::Camera, [562](#)
- DefectTableIndex
 - Spinnaker::Camera, [562](#)
- DefectTablePixelCount
 - Spinnaker::Camera, [562](#)
- DefectTableSave
 - Spinnaker::Camera, [562](#)
- Deinterlacing
 - Spinnaker::Camera, [563](#)
- DeinterlacingEnums
 - CameraDefs Class, [108](#)
- deleteFile
 - Spinnaker::GenApi::FileProtocolAdapter, [844](#)
- DeleteFileOnCamera
 - Inference.cpp, [1473](#)
- DeliverEventMessage
 - Spinnaker::GenApi::CEventAdapter1394, [705](#)
 - Spinnaker::GenApi::CEventAdapterGEV, [709](#)
 - Spinnaker::GenApi::CEventAdapterU3V, [711](#)
- DeliverMessage
 - Spinnaker::GenApi::CEventAdapter, [702](#)
 - Spinnaker::GenApi::CEventAdapter1394, [705](#)
 - Spinnaker::GenApi::CEventAdapterGEV, [709](#)
 - Spinnaker::GenApi::CEventAdapterGeneric, [707](#)
 - Spinnaker::GenApi::CEventAdapterU3V, [711](#)
- Deregister
 - NodeCallback Class, [372](#)
- DeregisterCallback
 - INode Interface, [337](#)
 - Spinnaker::GenApi::Node, [1088](#)
- Destroy
 - Spinnaker::GenApi::CNodeCallback, [748](#)
 - Spinnaker::GenApi::Function_NodeCallback, [857](#)
 - Spinnaker::GenApi::Member_NodeCallback, [1083](#)
 - Spinnaker::GenApi::NodeMap, [1099](#)
- DetachBuffer
 - Spinnaker::GenApi::CChunkAdapter, [678](#)
- DetachChunk
 - Spinnaker::GenApi::CChunkPort, [692](#)
- DetachEvent
 - Spinnaker::GenApi::CEventPort, [715](#)
- DetachNode
 - Spinnaker::GenApi::CEventPort, [715](#)
- DetachNodeMap
 - Spinnaker::GenApi::CChunkAdapter, [678](#)
 - Spinnaker::GenApi::CEventAdapter, [703](#)
- DetachPort
 - Spinnaker::GenApi::CChunkPort, [692](#)
- DeviceAccessStatus
 - Spinnaker::TransportLayerDevice, [1168](#)
 - Spinnaker::TransportLayerInterface, [1179](#)
- DeviceAccessStatusEnum
 - TransportLayerDefs Class, [244](#)
- DeviceAddress
 - Spinnaker::ActionCommandResult, [481](#)
- DeviceArrivalEventHandler, [796](#)
 - Spinnaker::DeviceArrivalEventHandler, [797](#)
- DeviceArrivalEventHandler Class, [180](#)
- DeviceCharacterSet
 - Spinnaker::Camera, [563](#)
- DeviceCharacterSetEnums
 - CameraDefs Class, [109](#)
- DeviceClockFrequency
 - Spinnaker::Camera, [563](#)
- DeviceClockSelector
 - Spinnaker::Camera, [563](#)
- DeviceClockSelectorEnums
 - CameraDefs Class, [109](#)
- DeviceConnectionSelector
 - Spinnaker::Camera, [563](#)
- DeviceConnectionSpeed
 - Spinnaker::Camera, [564](#)
- DeviceConnectionStatus
 - Spinnaker::Camera, [564](#)
- DeviceConnectionStatusEnums

- CameraDefs Class, [109](#)
- DeviceCount
 - Spinnaker::TransportLayerInterface, [1179](#)
- DeviceCurrentSpeed
 - Spinnaker::TransportLayerDevice, [1168](#)
- DeviceCurrentSpeedEnum
 - TransportLayerDefs Class, [245](#)
- DeviceDisplayName
 - Spinnaker::TransportLayerDevice, [1168](#)
- DeviceDriverVersion
 - Spinnaker::TransportLayerDevice, [1168](#)
- DeviceEndianessMechanism
 - Spinnaker::TransportLayerDevice, [1168](#)
- DeviceEndianessMechanismEnum
 - TransportLayerDefs Class, [245](#)
- DeviceEventChannelCount
 - Spinnaker::Camera, [564](#)
- DeviceEventExposureEndData, [798](#)
- DeviceEventHandler, [799](#)
 - Spinnaker::DeviceEventHandler, [800](#)
- DeviceEventHandler Class, [181](#)
- DeviceEventHandlerImpl, [802](#)
 - ~DeviceEventHandlerImpl, [803](#)
 - DeviceEventHandlerImpl, [803](#)
 - OnDeviceEvent, [804](#)
- DeviceEventInferenceData, [804](#)
- DeviceEventUtility, [805](#)
- DeviceEvents.cpp
 - AcquireImages, [1447](#)
 - chosenEvent, [1448](#)
 - ConfigureDeviceEvents, [1447](#)
 - eventType, [1447](#)
 - InferenceAvailable, [1447](#)
 - main, [1448](#)
 - PrintDeviceInfo, [1448](#)
 - ResetDeviceEvents, [1448](#)
 - RunSingleCamera, [1448](#)
- DeviceFamilyName
 - Spinnaker::Camera, [564](#)
- DeviceFeaturePersistenceEnd
 - Spinnaker::Camera, [564](#)
- DeviceFeaturePersistenceStart
 - Spinnaker::Camera, [564](#)
- DeviceFirmwareVersion
 - Spinnaker::Camera, [565](#)
- DeviceGenCPVersionMajor
 - Spinnaker::Camera, [565](#)
- DeviceGenCPVersionMinor
 - Spinnaker::Camera, [565](#)
- DeviceID
 - Spinnaker::Camera, [565](#)
 - Spinnaker::TransportLayerDevice, [1169](#)
 - Spinnaker::TransportLayerInterface, [1179](#)
- DeviceIndicatorMode
 - Spinnaker::Camera, [565](#)
- DeviceIndicatorModeEnums
 - CameraDefs Class, [110](#)
- DeviceInstanceId
 - Spinnaker::TransportLayerDevice, [1169](#)
- DevicesUpdater
 - Spinnaker::TransportLayerDevice, [1169](#)
- DeviceLinkBandwidthReserve
 - Spinnaker::Camera, [565](#)
- DeviceLinkCommandTimeout
 - Spinnaker::Camera, [566](#)
- DeviceLinkConnectionCount
 - Spinnaker::Camera, [566](#)
- DeviceLinkCurrentThroughput
 - Spinnaker::Camera, [566](#)
- DeviceLinkHeartbeatMode
 - Spinnaker::Camera, [566](#)
- DeviceLinkHeartbeatModeEnums
 - CameraDefs Class, [110](#)
- DeviceLinkHeartbeatTimeout
 - Spinnaker::Camera, [566](#)
- DeviceLinkSelector
 - Spinnaker::Camera, [566](#)
- DeviceLinkSpeed
 - Spinnaker::Camera, [567](#)
 - Spinnaker::TransportLayerDevice, [1169](#)
- DeviceLinkThroughputLimit
 - Spinnaker::Camera, [567](#)
- DeviceLinkThroughputLimitMode
 - Spinnaker::Camera, [567](#)
- DeviceLinkThroughputLimitModeEnums
 - CameraDefs Class, [110](#)
- DeviceLocation
 - Spinnaker::TransportLayerDevice, [1169](#)
- DeviceManifestEntrySelector
 - Spinnaker::Camera, [567](#)
- DeviceManifestPrimaryURL
 - Spinnaker::Camera, [568](#)
- DeviceManifestSchemaMajorVersion
 - Spinnaker::Camera, [568](#)
- DeviceManifestSchemaMinorVersion
 - Spinnaker::Camera, [568](#)
- DeviceManifestSecondaryURL
 - Spinnaker::Camera, [568](#)
- DeviceManifestXMLMajorVersion
 - Spinnaker::Camera, [568](#)
- DeviceManifestXMLMinorVersion
 - Spinnaker::Camera, [568](#)
- DeviceManifestXMLSubMinorVersion
 - Spinnaker::Camera, [569](#)
- DeviceManufacturerInfo
 - Spinnaker::Camera, [569](#)
- DeviceMaxThroughput
 - Spinnaker::Camera, [569](#)
- DeviceModelName
 - Spinnaker::Camera, [569](#)
 - Spinnaker::TransportLayerDevice, [1169](#)
 - Spinnaker::TransportLayerInterface, [1179](#)
- DeviceMulticastMonitorMode
 - Spinnaker::TransportLayerDevice, [1170](#)
- DevicePortId
 - Spinnaker::TransportLayerDevice, [1170](#)

- DevicePowerSupplySelector
 - Spinnaker::Camera, [569](#)
- DevicePowerSupplySelectorEnums
 - CameraDefs Class, [110](#)
- DeviceRegistersCheck
 - Spinnaker::Camera, [570](#)
- DeviceRegistersEndianness
 - Spinnaker::Camera, [570](#)
- DeviceRegistersEndiannessEnums
 - CameraDefs Class, [111](#)
- DeviceRegistersStreamingEnd
 - Spinnaker::Camera, [570](#)
- DeviceRegistersStreamingStart
 - Spinnaker::Camera, [570](#)
- DeviceRegistersValid
 - Spinnaker::Camera, [570](#)
- DeviceRemovalEventHandler, [806](#)
 - Spinnaker::DeviceRemovalEventHandler, [807](#)
- DeviceRemovalEventHandler Class, [182](#)
- DeviceReset
 - Spinnaker::Camera, [570](#)
- DeviceSFNCVersionMajor
 - Spinnaker::Camera, [571](#)
- DeviceSFNCVersionMinor
 - Spinnaker::Camera, [572](#)
- DeviceSFNCVersionSubMinor
 - Spinnaker::Camera, [572](#)
- DeviceScanType
 - Spinnaker::Camera, [571](#)
- DeviceScanTypeEnums
 - CameraDefs Class, [111](#)
- DeviceSelector
 - Spinnaker::TransportLayerInterface, [1179](#)
- DeviceSerialNumber
 - Spinnaker::Camera, [571](#)
 - Spinnaker::TransportLayerDevice, [1170](#)
 - Spinnaker::TransportLayerInterface, [1179](#)
- DeviceSerialPortBaudRate
 - Spinnaker::Camera, [571](#)
- DeviceSerialPortBaudRateEnums
 - CameraDefs Class, [111](#)
- DeviceSerialPortSelector
 - Spinnaker::Camera, [571](#)
- DeviceSerialPortSelectorEnums
 - CameraDefs Class, [112](#)
- DeviceStreamChannelCount
 - Spinnaker::Camera, [572](#)
- DeviceStreamChannelEndianness
 - Spinnaker::Camera, [572](#)
- DeviceStreamChannelEndiannessEnums
 - CameraDefs Class, [112](#)
- DeviceStreamChannelLink
 - Spinnaker::Camera, [572](#)
- DeviceStreamChannelPacketSize
 - Spinnaker::Camera, [573](#)
- DeviceStreamChannelSelector
 - Spinnaker::Camera, [573](#)
- DeviceStreamChannelType
 - Spinnaker::Camera, [573](#)
- DeviceStreamChannelTypeEnums
 - CameraDefs Class, [112](#)
- DeviceTLType
 - Spinnaker::Camera, [574](#)
- DeviceTLTypeEnums
 - CameraDefs Class, [114](#)
- DeviceTLVersionMajor
 - Spinnaker::Camera, [574](#)
- DeviceTLVersionMinor
 - Spinnaker::Camera, [574](#)
- DeviceTLVersionSubMinor
 - Spinnaker::Camera, [574](#)
- DeviceTapGeometry
 - Spinnaker::Camera, [573](#)
- DeviceTapGeometryEnums
 - CameraDefs Class, [113](#)
- DeviceTemperature
 - Spinnaker::Camera, [573](#)
- DeviceTemperatureSelector
 - Spinnaker::Camera, [573](#)
- DeviceTemperatureSelectorEnums
 - CameraDefs Class, [114](#)
- DeviceType
 - Spinnaker::Camera, [574](#)
 - Spinnaker::TransportLayerDevice, [1170](#)
- DeviceTypeEnum
 - TransportLayerDefs Class, [246](#)
- DeviceTypeEnums
 - CameraDefs Class, [115](#)
- DeviceU3VProtocol
 - Spinnaker::TransportLayerDevice, [1170](#)
- DeviceUnlock
 - Spinnaker::TransportLayerInterface, [1180](#)
- DeviceUpdateList
 - Spinnaker::TransportLayerInterface, [1180](#)
- DeviceUptime
 - Spinnaker::Camera, [575](#)
- DeviceUserID
 - Spinnaker::Camera, [575](#)
 - Spinnaker::TransportLayerDevice, [1170](#)
- DeviceVendorName
 - Spinnaker::Camera, [575](#)
 - Spinnaker::TransportLayerDevice, [1171](#)
 - Spinnaker::TransportLayerInterface, [1180](#)
- DeviceVersion
 - Spinnaker::Camera, [575](#)
 - Spinnaker::TransportLayerDevice, [1171](#)
- dhcpEnabled
 - AdapterConfig::AdapterInfo, [485](#)
- DisableAll
 - Spinnaker::ImageStatistics, [935](#)
 - Spinnaker::ImageStatistics, [994](#)
- DisableChunkData
 - ChunkData.cpp, [1441](#)
 - Inference.cpp, [1473](#)
- DisableImageChunkData
 - Compression.cpp, [1443](#)

- DisableImageCompression
 - Compression.cpp, [1443](#)
- DisableTrigger
 - Inference.cpp, [1474](#)
- DiscoverMaxPacketSize
 - Spinnaker::CameraBase, [655](#)
 - Spinnaker::lCameraBase, [883](#)
- DisplayChunkData
 - ChunkData.cpp, [1441](#)
 - Inference.cpp, [1474](#)
- doc/spindocs/C++/GettingStarted.dox, [1211](#)
- doc/spindocs/C++/ProgrammerGuide.dox, [1211](#)
- doc/spindocs/shared/Benefits.dox, [1211](#)
- doc/spindocs/shared/FlyCapture2Comparison.dox, [1211](#)
- doc/spindocs/shared/GenlCamGenTL.dox, [1211](#)
- doc/spindocs/shared/Licensing.dox, [1211](#)
- doc/spindocs/shared/Maintenance.dox, [1211](#)
- DoesEnvironmentVariableExist
 - GCUtilities Utility, [306](#)
- double_autovector_t, [808](#)
 - Spinnaker::GenApi::double_autovector_t, [809](#)
- DownloadImage
 - FileAccess_QuickSpin.cpp, [1456](#)
- dummy
 - CpuUtil::CpuUsageInfo, [784](#)
- EAccessMode
 - Types Enums, [403](#)
- EAccessModeClass, [811](#)
- ECacheUsage_t
 - NodeMapFactory Class, [375](#)
- ECachingMode
 - Types Enums, [403](#)
- ECachingModeClass, [812](#)
- ECallbackType
 - NodeCallback Class, [372](#)
- EContentType_t
 - NodeMapFactory Class, [376](#)
- EDisplayNotation
 - Types Enums, [403](#)
- EDisplayNotationClass, [813](#)
- EEndianess
 - Types Enums, [405](#)
- EEndianessClass, [814](#)
- EGenApiSchemaVersion
 - Types Enums, [405](#)
- EGenApiSchemaVersionClass, [816](#)
- ElncMode
 - Types Enums, [405](#)
- EInputDirection
 - Types Enums, [406](#)
- EInputDirectionClass, [817](#)
- EInterfaceType
 - Types Enums, [406](#)
- ELinkType
 - Types Enums, [406](#)
- ENamespace
 - Types Enums, [407](#)
- ENamespaceClass, [818](#)
- ERepresentation
 - Types Enums, [407](#)
- ERepresentationClass, [827](#)
- ESign
 - Types Enums, [407](#)
- ESignClass, [828](#)
- ESlope
 - Types Enums, [408](#)
- ESlopeClass, [829](#)
- EStandardNameSpace
 - Types Enums, [408](#)
- EStandardNameSpaceClass, [830](#)
- EVENT_TIMEOUT_INFINITE
 - Spinnaker Headers, [218](#)
- EVENT_TIMEOUT_NONE
 - Spinnaker Headers, [218](#)
- EVisibility
 - Types Enums, [408](#)
- EVisibilityClass, [835](#)
- EXMLValidation
 - Types Enums, [409](#)
- EXPAND_TO_STRINGISE
 - GCUtilities.h, [1314](#)
- EYesNo
 - Types Enums, [409](#)
- EYesNoClass, [841](#)
- EatComments
 - Spinnaker GenApi Classes, [271](#)
- empty
 - Spinnaker::GenlCam::gcstring, [861](#)
- EnableAll
 - Spinnaker::lImageStatistics, [935](#)
 - Spinnaker::ImageStatistics, [994](#)
- enableChunkData
 - Compression.cpp, [1444](#)
- EnableGreyOnly
 - Spinnaker::lImageStatistics, [935](#)
 - Spinnaker::ImageStatistics, [994](#)
- EnableHSLOnly
 - Spinnaker::lImageStatistics, [935](#)
 - Spinnaker::ImageStatistics, [995](#)
- EnableImageChunkData
 - Compression.cpp, [1443](#)
- EnableImageCompression
 - Compression.cpp, [1443](#)
- EnableManualFramerate
 - GigEVisionPerformance.cpp, [1461](#)
- EnableRGBOnly
 - Spinnaker::lImageStatistics, [935](#)
 - Spinnaker::ImageStatistics, [995](#)
- EncoderDivider
 - Spinnaker::Camera, [575](#)
- EncoderMode
 - Spinnaker::Camera, [575](#)
- EncoderModeEnums
 - CameraDefs Class, [115](#)
- EncoderOutputMode

- Spinnaker::Camera, 576
- EncoderOutputModeEnums
 - CameraDefs Class, 115
- EncoderReset
 - Spinnaker::Camera, 576
- EncoderResetActivation
 - Spinnaker::Camera, 576
- EncoderResetActivationEnums
 - CameraDefs Class, 116
- EncoderResetSource
 - Spinnaker::Camera, 576
- EncoderResetSourceEnums
 - CameraDefs Class, 116
- EncoderSelector
 - Spinnaker::Camera, 576
- EncoderSelectorEnums
 - CameraDefs Class, 117
- EncoderSourceAEnums
 - CameraDefs Class, 118
- EncoderSourceBEnums
 - CameraDefs Class, 118
- EncoderSourceA
 - Spinnaker::Camera, 576
- EncoderSourceB
 - Spinnaker::Camera, 577
- EncoderStatus
 - Spinnaker::Camera, 577
- EncoderStatusEnums
 - CameraDefs Class, 118
- EncoderTimeout
 - Spinnaker::Camera, 577
- EncoderValue
 - Spinnaker::Camera, 577
- EncoderValueAtReset
 - Spinnaker::Camera, 577
- EndAcquisition
 - Spinnaker::CameraBase, 656
 - Spinnaker::ICameraBase, 883
 - SpinnakerDirectShow.h, 233
- endTime
 - SecondsCounter, 420
- entryExist
 - Spinnaker Headers, 217
- entryIndex
 - Spinnaker Headers, 217
- EnumClasses Class, 287
- EnumEntryNode, 819
 - Spinnaker::GenApi::EnumEntryNode, 820
- EnumEntryNode Class, 289
 - CEnumEntryRef, 289
- EnumNode, 822
 - Spinnaker::GenApi::EnumNode, 824
- EnumNode Class, 290
 - CEnumerationRef, 290
- EnumNodeT Class, 291
- EnumerateGEVInterfaces
 - Spinnaker::TransportLayerSystem, 1197
- EnumerateUSBInterfaces
 - Spinnaker::TransportLayerSystem, 1197
- Enumeration.cpp
 - main, 1449
 - QueryInterface, 1449
- Enumeration_QuickSpin.cpp
 - main, 1450
 - QueryInterface, 1450
- EnumerationCount
 - Spinnaker::Camera, 577
- enumerationEntryName
 - Spinnaker Headers, 217
- EnumerationEvents.cpp
 - CheckGevEnabled, 1450
 - main, 1451
- EnumerationGetEntry
 - Spinnaker Headers, 206
- EnumerationGetNumEntries
 - Spinnaker Headers, 206
- enumerationName
 - Spinnaker Headers, 218
- EnumerationSetEntry
 - Spinnaker Headers, 207
- ErrCode
 - AdapterConfigException, 483
- Error
 - Spinnaker Definitions, 226
- Event
 - GVCP_EVENTDATA_REQUEST_EXTENDED_↔ ID, 876
 - GVCP_EVENTDATA_REQUEST, 875
- EventAcquisitionEnd
 - Spinnaker::Camera, 578
- EventAcquisitionEndFrameID
 - Spinnaker::Camera, 578
- EventAcquisitionEndTimestamp
 - Spinnaker::Camera, 578
- EventAcquisitionError
 - Spinnaker::Camera, 578
- EventAcquisitionErrorFrameID
 - Spinnaker::Camera, 578
- EventAcquisitionErrorTimestamp
 - Spinnaker::Camera, 578
- EventAcquisitionStart
 - Spinnaker::Camera, 579
- EventAcquisitionStartFrameID
 - Spinnaker::Camera, 579
- EventAcquisitionStartTimestamp
 - Spinnaker::Camera, 579
- EventAcquisitionTransferEnd
 - Spinnaker::Camera, 579
- EventAcquisitionTransferEndFrameID
 - Spinnaker::Camera, 579
- EventAcquisitionTransferEndTimestamp
 - Spinnaker::Camera, 579
- EventAcquisitionTransferStart
 - Spinnaker::Camera, 580
- EventAcquisitionTransferStartFrameID
 - Spinnaker::Camera, 580

EventAcquisitionTransferStartTimestamp
 Spinnaker::Camera, [580](#)

EventAcquisitionTrigger
 Spinnaker::Camera, [580](#)

EventAcquisitionTriggerFrameID
 Spinnaker::Camera, [580](#)

EventAcquisitionTriggerTimestamp
 Spinnaker::Camera, [580](#)

EventActionLate
 Spinnaker::Camera, [581](#)

EventActionLateFrameID
 Spinnaker::Camera, [581](#)

EventActionLateTimestamp
 Spinnaker::Camera, [581](#)

EventAdapter Class, [292](#)

EventAdapter1394 Class, [293](#)

EventAdapterGEV Class, [295](#)

EventAdapterGeneric Class, [294](#)

EventAdapterU3V Class, [296](#)

EventCounter0End
 Spinnaker::Camera, [581](#)

EventCounter0EndFrameID
 Spinnaker::Camera, [581](#)

EventCounter0EndTimestamp
 Spinnaker::Camera, [581](#)

EventCounter0Start
 Spinnaker::Camera, [582](#)

EventCounter0StartFrameID
 Spinnaker::Camera, [582](#)

EventCounter0StartTimestamp
 Spinnaker::Camera, [582](#)

EventCounter1End
 Spinnaker::Camera, [582](#)

EventCounter1EndFrameID
 Spinnaker::Camera, [582](#)

EventCounter1EndTimestamp
 Spinnaker::Camera, [582](#)

EventCounter1Start
 Spinnaker::Camera, [583](#)

EventCounter1StartFrameID
 Spinnaker::Camera, [583](#)

EventCounter1StartTimestamp
 Spinnaker::Camera, [583](#)

EventData
 U3V_EVENT_MESSAGE, [1205](#)

EventEncoder0Restarted
 Spinnaker::Camera, [583](#)

EventEncoder0RestartedFrameID
 Spinnaker::Camera, [583](#)

EventEncoder0RestartedTimestamp
 Spinnaker::Camera, [583](#)

EventEncoder0Stopped
 Spinnaker::Camera, [584](#)

EventEncoder0StoppedFrameID
 Spinnaker::Camera, [584](#)

EventEncoder0StoppedTimestamp
 Spinnaker::Camera, [584](#)

EventEncoder1Restarted
 Spinnaker::Camera, [584](#)

EventEncoder1RestartedFrameID
 Spinnaker::Camera, [584](#)

EventEncoder1RestartedTimestamp
 Spinnaker::Camera, [584](#)

EventEncoder1Stopped
 Spinnaker::Camera, [585](#)

EventEncoder1StoppedFrameID
 Spinnaker::Camera, [585](#)

EventEncoder1StoppedTimestamp
 Spinnaker::Camera, [585](#)

EventError
 Spinnaker::Camera, [585](#)

EventErrorCode
 Spinnaker::Camera, [585](#)

EventErrorFrameID
 Spinnaker::Camera, [585](#)

EventErrorTimestamp
 Spinnaker::Camera, [586](#)

EventExposureEnd
 Spinnaker::Camera, [586](#)

EventExposureEndFrameID
 Spinnaker::Camera, [586](#)

EventExposureEndTimestamp
 Spinnaker::Camera, [586](#)

EventExposureStart
 Spinnaker::Camera, [586](#)

EventExposureStartFrameID
 Spinnaker::Camera, [586](#)

EventExposureStartTimestamp
 Spinnaker::Camera, [587](#)

EventFrameBurstEnd
 Spinnaker::Camera, [587](#)

EventFrameBurstEndFrameID
 Spinnaker::Camera, [587](#)

EventFrameBurstEndTimestamp
 Spinnaker::Camera, [587](#)

EventFrameBurstStart
 Spinnaker::Camera, [587](#)

EventFrameBurstStartFrameID
 Spinnaker::Camera, [587](#)

EventFrameBurstStartTimestamp
 Spinnaker::Camera, [588](#)

EventFrameEnd
 Spinnaker::Camera, [588](#)

EventFrameEndFrameID
 Spinnaker::Camera, [588](#)

EventFrameEndTimestamp
 Spinnaker::Camera, [588](#)

EventFrameStart
 Spinnaker::Camera, [588](#)

EventFrameStartFrameID
 Spinnaker::Camera, [588](#)

EventFrameStartTimestamp
 Spinnaker::Camera, [589](#)

EventFrameTransferEnd
 Spinnaker::Camera, [589](#)

EventFrameTransferEndFrameID

- Spinnaker::Camera, [589](#)
- EventFrameTransferEndTimestamp
 - Spinnaker::Camera, [589](#)
- EventFrameTransferStart
 - Spinnaker::Camera, [589](#)
- EventFrameTransferStartFrameID
 - Spinnaker::Camera, [589](#)
- EventFrameTransferStartTimestamp
 - Spinnaker::Camera, [590](#)
- EventFrameTrigger
 - Spinnaker::Camera, [590](#)
- EventFrameTriggerFrameID
 - Spinnaker::Camera, [590](#)
- EventFrameTriggerTimestamp
 - Spinnaker::Camera, [590](#)
- EventHandler, [832](#)
 - Spinnaker::EventHandler, [833](#)
- EventHandler Class, [183](#)
- EventId
 - GVCP_EVENT_ITEM_BASIC, [870](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [872](#)
 - GVCP_EVENT_ITEM, [869](#)
 - U3V_EVENT_DATA, [1204](#)
- EventLine0AnyEdge
 - Spinnaker::Camera, [590](#)
- EventLine0AnyEdgeFrameID
 - Spinnaker::Camera, [590](#)
- EventLine0AnyEdgeTimestamp
 - Spinnaker::Camera, [591](#)
- EventLine0FallingEdge
 - Spinnaker::Camera, [591](#)
- EventLine0FallingEdgeFrameID
 - Spinnaker::Camera, [591](#)
- EventLine0FallingEdgeTimestamp
 - Spinnaker::Camera, [591](#)
- EventLine0RisingEdge
 - Spinnaker::Camera, [591](#)
- EventLine0RisingEdgeFrameID
 - Spinnaker::Camera, [591](#)
- EventLine0RisingEdgeTimestamp
 - Spinnaker::Camera, [592](#)
- EventLine1AnyEdge
 - Spinnaker::Camera, [592](#)
- EventLine1AnyEdgeFrameID
 - Spinnaker::Camera, [592](#)
- EventLine1AnyEdgeTimestamp
 - Spinnaker::Camera, [592](#)
- EventLine1FallingEdge
 - Spinnaker::Camera, [592](#)
- EventLine1FallingEdgeFrameID
 - Spinnaker::Camera, [592](#)
- EventLine1FallingEdgeTimestamp
 - Spinnaker::Camera, [593](#)
- EventLine1RisingEdge
 - Spinnaker::Camera, [593](#)
- EventLine1RisingEdgeFrameID
 - Spinnaker::Camera, [593](#)
- EventLine1RisingEdgeTimestamp
 - Spinnaker::Camera, [593](#)
- EventLinkSpeedChange
 - Spinnaker::Camera, [593](#)
- EventLinkSpeedChangeFrameID
 - Spinnaker::Camera, [593](#)
- EventLinkSpeedChangeTimestamp
 - Spinnaker::Camera, [594](#)
- EventLinkTrigger0
 - Spinnaker::Camera, [594](#)
- EventLinkTrigger0FrameID
 - Spinnaker::Camera, [594](#)
- EventLinkTrigger0Timestamp
 - Spinnaker::Camera, [594](#)
- EventLinkTrigger1
 - Spinnaker::Camera, [594](#)
- EventLinkTrigger1FrameID
 - Spinnaker::Camera, [594](#)
- EventLinkTrigger1Timestamp
 - Spinnaker::Camera, [595](#)
- EventNotification
 - Spinnaker::Camera, [595](#)
- EventNotificationEnums
 - CameraDefs Class, [119](#)
- EventPort Class, [297](#)
- EventProcessor
 - Spinnaker::EventHandler, [834](#)
- EventSelector
 - Spinnaker::Camera, [595](#)
- EventSelectorEnums
 - CameraDefs Class, [119](#)
- EventSequencerSetChange
 - Spinnaker::Camera, [595](#)
- EventSequencerSetChangeFrameID
 - Spinnaker::Camera, [595](#)
- EventSequencerSetChangeTimestamp
 - Spinnaker::Camera, [595](#)
- EventSerialData
 - Spinnaker::Camera, [596](#)
- EventSerialDataLength
 - Spinnaker::Camera, [596](#)
- EventSerialPortReceive
 - Spinnaker::Camera, [596](#)
- EventSerialPortReceiveTimestamp
 - Spinnaker::Camera, [596](#)
- EventSerialReceiveOverflow
 - Spinnaker::Camera, [596](#)
- EventStream0TransferBlockEnd
 - Spinnaker::Camera, [596](#)
- EventStream0TransferBlockEndFrameID
 - Spinnaker::Camera, [597](#)
- EventStream0TransferBlockEndTimestamp
 - Spinnaker::Camera, [597](#)
- EventStream0TransferBlockStart
 - Spinnaker::Camera, [597](#)
- EventStream0TransferBlockStartFrameID
 - Spinnaker::Camera, [597](#)
- EventStream0TransferBlockStartTimestamp
 - Spinnaker::Camera, [597](#)

- EventStream0TransferBlockTrigger
 - Spinnaker::Camera, [597](#)
- EventStream0TransferBlockTriggerFrameID
 - Spinnaker::Camera, [598](#)
- EventStream0TransferBlockTriggerTimestamp
 - Spinnaker::Camera, [598](#)
- EventStream0TransferBurstEnd
 - Spinnaker::Camera, [598](#)
- EventStream0TransferBurstEndFrameID
 - Spinnaker::Camera, [598](#)
- EventStream0TransferBurstEndTimestamp
 - Spinnaker::Camera, [598](#)
- EventStream0TransferBurstStart
 - Spinnaker::Camera, [598](#)
- EventStream0TransferBurstStartFrameID
 - Spinnaker::Camera, [599](#)
- EventStream0TransferBurstStartTimestamp
 - Spinnaker::Camera, [599](#)
- EventStream0TransferEnd
 - Spinnaker::Camera, [599](#)
- EventStream0TransferEndFrameID
 - Spinnaker::Camera, [599](#)
- EventStream0TransferEndTimestamp
 - Spinnaker::Camera, [599](#)
- EventStream0TransferOverflow
 - Spinnaker::Camera, [599](#)
- EventStream0TransferOverflowFrameID
 - Spinnaker::Camera, [600](#)
- EventStream0TransferOverflowTimestamp
 - Spinnaker::Camera, [600](#)
- EventStream0TransferPause
 - Spinnaker::Camera, [600](#)
- EventStream0TransferPauseFrameID
 - Spinnaker::Camera, [600](#)
- EventStream0TransferPauseTimestamp
 - Spinnaker::Camera, [600](#)
- EventStream0TransferResume
 - Spinnaker::Camera, [600](#)
- EventStream0TransferResumeFrameID
 - Spinnaker::Camera, [601](#)
- EventStream0TransferResumeTimestamp
 - Spinnaker::Camera, [601](#)
- EventStream0TransferStart
 - Spinnaker::Camera, [601](#)
- EventStream0TransferStartFrameID
 - Spinnaker::Camera, [601](#)
- EventStream0TransferStartTimestamp
 - Spinnaker::Camera, [601](#)
- EventTest
 - Spinnaker::Camera, [601](#)
- EventTestTimestamp
 - Spinnaker::Camera, [602](#)
- EventTimer0End
 - Spinnaker::Camera, [602](#)
- EventTimer0EndFrameID
 - Spinnaker::Camera, [602](#)
- EventTimer0EndTimestamp
 - Spinnaker::Camera, [602](#)
- EventTimer0Start
 - Spinnaker::Camera, [602](#)
- EventTimer0StartFrameID
 - Spinnaker::Camera, [602](#)
- EventTimer0StartTimestamp
 - Spinnaker::Camera, [603](#)
- EventTimer1End
 - Spinnaker::Camera, [603](#)
- EventTimer1EndFrameID
 - Spinnaker::Camera, [603](#)
- EventTimer1EndTimestamp
 - Spinnaker::Camera, [603](#)
- EventTimer1Start
 - Spinnaker::Camera, [603](#)
- EventTimer1StartFrameID
 - Spinnaker::Camera, [603](#)
- EventTimer1StartTimestamp
 - Spinnaker::Camera, [604](#)
- EventType
 - Spinnaker Definitions, [227](#)
- eventType
 - DeviceEvents.cpp, [1447](#)
- Exception, [836](#)
 - Spinnaker::Exception, [838](#), [839](#)
- Exception Class, [184](#)
- ExceptionHandling.cpp
 - causeSpinnakerException, [1452](#)
 - causeStandardException, [1452](#)
 - chosenException, [1452](#)
 - exceptionType, [1451](#)
 - main, [1452](#)
- exceptionType
 - ExceptionHandling.cpp, [1451](#)
- Execute
 - Spinnaker::GenApi::CommandNode, [768](#)
- ExecuteDeleteCommand
 - FileAccess_QuickSpin.cpp, [1456](#)
- ExecuteReadCommand
 - FileAccess_QuickSpin.cpp, [1456](#)
- ExecuteWriteCommand
 - FileAccess_QuickSpin.cpp, [1456](#)
- Expert
 - Types Enums, [409](#)
- Exposure.cpp
 - AcquireImages, [1453](#)
 - ConfigureExposure, [1453](#)
 - main, [1453](#)
 - PrintDeviceInfo, [1453](#)
 - ResetExposure, [1453](#)
 - RunSingleCamera, [1453](#)
- Exposure_QuickSpin.cpp
 - AcquireImages, [1454](#)
 - ConfigureExposure, [1454](#)
 - main, [1454](#)
 - PrintDeviceInfo, [1455](#)
 - ResetExposure, [1455](#)
 - RunSingleCamera, [1455](#)
- ExposureActiveMode

- Spinnaker::Camera, 604
- ExposureActiveModeEnums
 - CameraDefs Class, 119
- ExposureAuto
 - Spinnaker::Camera, 604
- ExposureAutoEnums
 - CameraDefs Class, 119
- ExposureMode
 - Spinnaker::Camera, 604
- ExposureModeEnums
 - CameraDefs Class, 120
- ExposureTime
 - Spinnaker::Camera, 604
- ExposureTimeMode
 - Spinnaker::Camera, 604
- ExposureTimeModeEnums
 - CameraDefs Class, 120
- ExposureTimeSelector
 - Spinnaker::Camera, 605
- ExposureTimeSelectorEnums
 - CameraDefs Class, 121
- ExtractAndSavePolarQuadImages
 - Polarization.cpp, 1489
- ExtractIndependentSubtree
 - INodeMapDyn Interface, 350
- ExtractPolarQuadrant
 - Spinnaker::ImageUtilityPolarization, 1018
- ExtractSubtree
 - Spinnaker::GenApi::CNodeMapFactory, 755
- FLIR_SPINNAKER_VERSION_BUILD
 - System.h, 1381
- FLIR_SPINNAKER_VERSION_MAJOR
 - System.h, 1381
- FLIR_SPINNAKER_VERSION_MINOR
 - System.h, 1381
- FLIR_SPINNAKER_VERSION_TYPE
 - System.h, 1381
- FMT_I64
 - Compatibility.h, 1292
- FactoryReset
 - Spinnaker::Camera, 605
- FileAccess_QuickSpin.cpp
 - _enableDebug, 1458
 - _fileSelector, 1458
 - AcquireImages, 1456
 - CloseFile, 1456
 - DownloadImage, 1456
 - ExecuteDeleteCommand, 1456
 - ExecuteReadCommand, 1456
 - ExecuteWriteCommand, 1456
 - InitializeSystem, 1457
 - main, 1457
 - OpenFileToRead, 1457
 - OpenFileToWrite, 1457
 - PrintDebugMessage, 1457
 - PrintDeviceInfo, 1457
 - PrintResultMessage, 1458
 - PrintUsage, 1458
 - UploadImage, 1458
- FileAccessBuffer
 - Spinnaker::Camera, 605
- FileAccessLength
 - Spinnaker::Camera, 605
- FileAccessOffset
 - Spinnaker::Camera, 605
- fileName
 - CompressedImageInfo, 770
- FileOpenMode
 - Spinnaker::Camera, 605
- FileOpenModeEnums
 - CameraDefs Class, 121
- FileOperationExecute
 - Spinnaker::Camera, 606
- FileOperationResult
 - Spinnaker::Camera, 606
- FileOperationSelector
 - Spinnaker::Camera, 606
- FileOperationSelectorEnums
 - CameraDefs Class, 121
- FileOperationStatus
 - Spinnaker::Camera, 606
- FileOperationStatusEnums
 - CameraDefs Class, 122
- FileProtocolAdapter, 842
 - Spinnaker::GenApi::FileProtocolAdapter, 843
- FileSelector
 - Spinnaker::Camera, 606
- FileSelectorEnums
 - CameraDefs Class, 122
- FileSize
 - Spinnaker::Camera, 607
- FileUploadPersistence
 - Inference.cpp, 1471
- filebuf_type
 - Spinnaker::GenApi::IDevFileStreamBase, 909
 - Spinnaker::GenApi::ODevFileStreamBase, 1107
- Filestream Class, 298
- FilterDriverStatus
 - Spinnaker::TransportLayerInterface, 1180
- FilterDriverStatusEnum
 - TransportLayerDefs Class, 246
- find
 - Spinnaker::GenICam::gcstring, 861, 862
- find_first_not_of
 - Spinnaker::GenICam::gcstring, 862
- find_first_of
 - Spinnaker::GenICam::gcstring, 862
- Flags
 - GVCP_REQUEST_HEADER, 877
 - U3V_COMMAND_HEADER, 1203
- float32_t
 - GCTypes Class, 303
- float64_t
 - GCTypes Class, 303
- FloatGetInc
 - Spinnaker Headers, 207

- FloatGetIncMode
 - Spinnaker Headers, [207](#)
- FloatGetMax
 - Spinnaker Headers, [208](#)
- FloatGetMin
 - Spinnaker Headers, [208](#)
- FloatGetValue
 - Spinnaker Headers, [209](#)
- FloatNode, [846](#)
 - Spinnaker::GenApi::FloatNode, [849](#)
- FloatNode Class, [299](#)
 - CFloatRef, [299](#)
- FloatRegNode, [853](#)
 - Spinnaker::GenApi::FloatRegNode, [854](#), [855](#)
- FloatRegNode Class, [300](#)
- FloatSetValue
 - Spinnaker Headers, [209](#)
- FlushQueueAllDiscard
 - Spinnaker::IDataStream, [904](#)
- ForceIP
 - Spinnaker::CameraBase, [656](#)
 - Spinnaker::ICameraBase, [883](#)
- frameID
 - Spinnaker::DeviceEventExposureEndData, [799](#)
 - Spinnaker::DeviceEventInferenceData, [804](#)
- frameRate
 - Spinnaker::Video::AVIOption, [490](#)
 - Spinnaker::Video::H264Option, [879](#)
 - Spinnaker::Video::MJPGOption, [1084](#)
- FromString
 - IValue Class, [368](#)
 - Spinnaker::GenApi::EAccessModeClass, [811](#)
 - Spinnaker::GenApi::ECachingModeClass, [813](#)
 - Spinnaker::GenApi::EDisplayNotationClass, [814](#)
 - Spinnaker::GenApi::EEndianessClass, [815](#)
 - Spinnaker::GenApi::EGenApiSchemaVersion↔
Class, [816](#)
 - Spinnaker::GenApi::EInputDirectionClass, [817](#)
 - Spinnaker::GenApi::ENameSpaceClass, [818](#)
 - Spinnaker::GenApi::ERepresentationClass, [827](#)
 - Spinnaker::GenApi::ESignClass, [829](#)
 - Spinnaker::GenApi::ESlopeClass, [830](#)
 - Spinnaker::GenApi::EStandardNameSpaceClass,
[831](#)
 - Spinnaker::GenApi::EVisibilityClass, [835](#)
 - Spinnaker::GenApi::EYesNoClass, [842](#)
 - Spinnaker::GenApi::ValueNode, [1207](#)
- Function_NodeCallback
 - Spinnaker::GenApi::Function_NodeCallback, [857](#)
- Function_NodeCallback< Function >, [856](#)
- GC_COUNTOF
 - GCUtilities.h, [1315](#)
- GC_INT32_MAX
 - GCTypes.h, [1310](#)
- GC_INT32_MIN
 - GCTypes.h, [1310](#)
- GC_INT64_MAX
 - GCTypes.h, [1310](#)
- GC_INT64_MIN
 - GCTypes.h, [1310](#)
- GC_INT8_MAX
 - GCTypes.h, [1311](#)
- GC_INT8_MIN
 - GCTypes.h, [1311](#)
- GC_UINT32_MAX
 - GCTypes.h, [1311](#)
- GC_UINT64_MAX
 - GCTypes.h, [1311](#)
- GC_UINT8_MAX
 - GCTypes.h, [1311](#)
- GCSTRING_NPOS
 - GCString.h, [1306](#)
- GCString Class, [301](#)
- GCString.h
 - GCSTRING_NPOS, [1306](#)
 - operator<<, [1306](#)
 - operator>>, [1307](#)
- GCSynch Class, [302](#)
- GCTypes Class, [303](#)
 - float32_t, [303](#)
 - float64_t, [303](#)
- GCTypes.h
 - __STDC_CONSTANT_MACROS, [1310](#)
 - __STDC_LIMIT_MACROS, [1310](#)
 - GC_INT32_MAX, [1310](#)
 - GC_INT32_MIN, [1310](#)
 - GC_INT64_MAX, [1310](#)
 - GC_INT64_MIN, [1310](#)
 - GC_INT8_MAX, [1311](#)
 - GC_INT8_MIN, [1311](#)
 - GC_UINT32_MAX, [1311](#)
 - GC_UINT64_MAX, [1311](#)
 - GC_UINT8_MAX, [1311](#)
- GCUtilities Utility, [305](#)
 - DoesEnvironmentVariableExist, [306](#)
 - GetFiles, [306](#)
 - GetGenICamCLProtocolFolder, [307](#)
 - GetGenICamCacheFolder, [306](#)
 - GetGenICamLogConfig, [307](#)
 - GetModulePathFromFunction, [307](#)
 - GetValueOfEnvironmentVariable, [307](#), [308](#)
 - INTEGRAL_CAST2, [308](#)
 - INTEGRAL_CAST, [308](#)
 - ReplaceEnvironmentVariables, [308](#)
 - SetGenICamCLProtocolFolder, [309](#)
 - SetGenICamCacheFolder, [308](#)
 - SetGenICamLogConfig, [309](#)
 - Tokenize, [309](#)
 - UrlDecode, [309](#)
 - UrlEncode, [310](#)
- GCUtilities.h
 - _TO_STRING, [1314](#)
 - __ERR__, [1313](#)
 - __LINE_STR__, [1314](#)
 - __LOCATION__, [1314](#)
 - __OUTPUT_FORMATER__, [1314](#)

- [__TODO__, 1314](#)
- [__WARN__, 1314](#)
- [EXPAND_TO_STRINGISE, 1314](#)
- [GC_COUNTOF, 1315](#)
- [GENICAM_DEPRECATED, 1315](#)
- [GENICAM_UNUSED, 1315](#)
- [USE_TEMP_CACHE_FILE, 1315](#)
- [GENCP_COMMAND_HEADER_SIZE](#)
 - [Spinnaker::GenApi, 476](#)
- [GENCP_EVENT_BASIC_SIZE](#)
 - [Spinnaker::GenApi, 476](#)
- [GENCP_EVENT_CMD_ID](#)
 - [Spinnaker::GenApi, 477](#)
- [GENICAM_DEPRECATED](#)
 - [GCUilities.h, 1315](#)
- [GENICAM_UNUSED](#)
 - [GCUilities.h, 1315](#)
- [GUIXMLLocation](#)
 - [Spinnaker::TransportLayerDevice, 1174](#)
- [GUIXMLLocationEnum](#)
 - [TransportLayerDefs Class, 247](#)
- [GUIXMLPath](#)
 - [Spinnaker::TransportLayerDevice, 1175](#)
- [GVCP_CHUNK_TRAILER, 868](#)
 - [ChunkID, 868](#)
 - [ChunkLength, 868](#)
- [GVCP_EVENT_ITEM_BASIC, 870](#)
 - [EventId, 870](#)
 - [ReservedOrEventSize, 871](#)
- [GVCP_EVENT_ITEM_EXTENDED_ID, 871](#)
 - [BlockId, 871](#)
 - [BlockId64High, 871](#)
 - [BlockId64Low, 872](#)
 - [EventId, 872](#)
 - [ReservedOrEventSize, 872](#)
 - [StreamChannelId, 872](#)
 - [TimestampHigh, 872](#)
 - [TimestampLow, 872](#)
- [GVCP_EVENT_ITEM, 869](#)
 - [BlockId, 869](#)
 - [EventId, 869](#)
 - [ReservedOrEventSize, 869](#)
 - [StreamChannelId, 870](#)
 - [TimestampHigh, 870](#)
 - [TimestampLow, 870](#)
- [GVCP_EVENT_REQUEST_EXTENDED_ID, 874](#)
 - [Header, 874](#)
 - [Items, 874](#)
- [GVCP_EVENT_REQUEST, 873](#)
 - [Header, 873](#)
 - [Items, 873](#)
- [GVCP_EVENTDATA_REQUEST_EXTENDED_ID, 876](#)
 - [Data, 876](#)
 - [Event, 876](#)
 - [Header, 877](#)
- [GVCP_EVENTDATA_REQUEST, 875](#)
 - [Data, 875](#)
 - [Event, 875](#)
- [Header, 875](#)
- [GVCP_MESSAGE_TAGS](#)
 - [Spinnaker::GenApi, 475](#)
- [GVCP_REQUEST_HEADER, 877](#)
 - [Command, 877](#)
 - [Flags, 877](#)
 - [Length, 877](#)
 - [Magic, 878](#)
 - [ReqId, 878](#)
- [Gain](#)
 - [Spinnaker::Camera, 607](#)
- [GainAuto](#)
 - [Spinnaker::Camera, 607](#)
- [GainAutoBalance](#)
 - [Spinnaker::Camera, 607](#)
- [GainAutoBalanceEnums](#)
 - [CameraDefs Class, 122](#)
- [GainAutoEnums](#)
 - [CameraDefs Class, 124](#)
- [GainSelector](#)
 - [Spinnaker::Camera, 608](#)
- [GainSelectorEnums](#)
 - [CameraDefs Class, 124](#)
- [Gamma](#)
 - [Spinnaker::Camera, 608](#)
- [GammaEnable](#)
 - [Spinnaker::Camera, 608](#)
- [gateway](#)
 - [AdapterConfig::IpInfo, 1057](#)
- [gcstring, 858](#)
 - [Spinnaker::GenICam::gcstring, 859](#)
- [GenICamXMLLocation](#)
 - [Spinnaker::TransportLayerDevice, 1171](#)
- [GenICamXMLLocationEnum](#)
 - [TransportLayerDefs Class, 246](#)
- [GenICamXMLPath](#)
 - [Spinnaker::TransportLayerDevice, 1171](#)
- [GenTLSFNCVersionMajor](#)
 - [Spinnaker::TransportLayerSystem, 1197](#)
- [GenTLSFNCVersionMinor](#)
 - [Spinnaker::TransportLayerSystem, 1197](#)
- [GenTLSFNCVersionSubMinor](#)
 - [Spinnaker::TransportLayerSystem, 1198](#)
- [GenTLVersionMajor](#)
 - [Spinnaker::TransportLayerSystem, 1198](#)
- [GenTLVersionMinor](#)
 - [Spinnaker::TransportLayerSystem, 1198](#)
- [Get](#)
 - [IRegister Interfaces, 361](#)
 - [Spinnaker::GenApi::RegisterNode, 1129](#)
- [get](#)
 - [Spinnaker::BasePtr, 492](#)
- [GetAccessMode](#)
 - [Spinnaker::CameraBase, 656](#)
 - [Spinnaker::GenApi::CChunkPort, 692](#)
 - [Spinnaker::GenApi::CEventPort, 715](#)
 - [Spinnaker::GenApi::CPortImpl, 779](#)
 - [Spinnaker::GenApi::CRegisterPortImpl, 786](#)

- Spinnaker::GenApi::CTestPortStruct, [793](#)
- Spinnaker::GenApi::Node, [1088](#)
- Spinnaker::GenApi::PortRecorder, [1120](#)
- Spinnaker::GenApi::PortReplay, [1124](#)
- Spinnaker::ICameraBase, [883](#)
- GetAddress
 - IRegister Interfaces, [362](#)
 - Spinnaker::GenApi::RegisterNode, [1129](#)
- GetAlias
 - INode Interface, [338](#)
 - Spinnaker::GenApi::Node, [1089](#)
- GetAuto10GDesc
 - AdapterConfig, [415](#)
- GetAutoGigabitDesc
 - AdapterConfig, [415](#)
- GetAutoStartIp
 - AdapterConfig, [415](#)
- GetAutoSubnetMask
 - AdapterConfig, [415](#)
- GetAutoSubnetMaskLength
 - AdapterConfig, [415](#)
- GetBitsPerPixel
 - Spinnaker::Image, [922](#)
 - Spinnaker::Image, [963](#)
- GetBlackLevel
 - Spinnaker::ChunkData, [730](#)
 - Spinnaker::IChunkData, [895](#)
- GetBoxAt
 - Chunk Data Inference Class, [173](#)
- GetBoxCount
 - Chunk Data Inference Class, [173](#)
- GetBoxSize
 - Chunk Data Inference Class, [173](#)
- getBufSize
 - Spinnaker::GenApi::FileProtocolAdapter, [844](#)
- GetBufferChunkData
 - Spinnaker::IDataStream, [904](#)
- GetBufferInfoBool8Type
 - Spinnaker::IDataStream, [904](#)
- GetBufferInfoPtrType
 - Spinnaker::IDataStream, [904](#)
- GetBufferInfoSizeType
 - Spinnaker::IDataStream, [904](#)
- GetBufferInfoUInt64Type
 - Spinnaker::IDataStream, [904](#)
- GetBufferOwnership
 - Spinnaker::CameraBase, [657](#)
 - Spinnaker::ICameraBase, [883](#)
- GetBufferSize
 - Spinnaker::Image, [922](#)
 - Spinnaker::Image, [963](#)
- GetBuildDate
 - Spinnaker::Exception, [839](#)
- GetBuildTime
 - Spinnaker::Exception, [839](#)
- GetByDeviceId
 - Spinnaker::CameraList, [669](#)
 - Spinnaker::ICameraList, [891](#)
- GetByIndex
 - Spinnaker::CameraList, [670](#)
 - Spinnaker::ICameraList, [891](#)
 - Spinnaker::InterfaceList, [948](#)
 - Spinnaker::InterfaceList, [1048](#)
- GetBySerial
 - Spinnaker::CameraList, [670](#)
 - Spinnaker::ICameraList, [891](#)
- GetCRC
 - Spinnaker::ChunkData, [731](#)
 - Spinnaker::IChunkData, [895](#)
- GetCachingMode
 - INode Interface, [338](#)
 - Spinnaker::GenApi::Node, [1089](#)
- GetCallbackType
 - Spinnaker::GenApi::CNodeCallback, [748](#)
- getCameraCategory
 - GigEVisionPerformance.cpp, [1461](#)
- GetCameraInfo
 - SpinnakerDirectShow.h, [233](#)
- GetCameras
 - Spinnaker::Interface, [939](#)
 - Spinnaker::ISystem, [1060](#)
 - Spinnaker::Interface, [1032](#)
 - Spinnaker::System, [1146](#)
- GetCastAlias
 - INode Interface, [338](#)
 - Spinnaker::GenApi::Node, [1089](#)
- GetCategoryName
 - Spinnaker::LoggingEventData, [1074](#)
- GetChannelStatus
 - Spinnaker::ImageStatistics, [935](#)
 - Spinnaker::ImageStatistics, [995](#)
- GetChildren
 - INode Interface, [338](#)
 - Spinnaker::GenApi::Node, [1089](#)
- GetChunkData
 - Spinnaker::Image, [922](#)
 - Spinnaker::Image, [964](#)
- GetChunkIDLength
 - Spinnaker::GenApi::CChunkPort, [692](#)
- GetChunkID
 - Spinnaker::GenApi::PortNode, [1115](#)
- GetChunkLayoutId
 - Spinnaker::Image, [923](#)
 - Spinnaker::Image, [964](#)
- GetColorProcessing
 - Spinnaker::Image, [923](#)
 - Spinnaker::Image, [964](#)
- GetCompressionMode
 - Spinnaker::ChunkData, [730](#)
 - Spinnaker::IChunkData, [895](#)
- GetCompressionRatio
 - Spinnaker::ChunkData, [730](#)
 - Spinnaker::IChunkData, [895](#)
- GetConfigLogFileName
 - AdapterConfig, [415](#)
- GetCookie

- IPortRecorder Interface, [359](#)
- Spinnaker::GenApi::CPortWriteList, [783](#)
- GetCounterValue
 - Spinnaker::ChunkData, [731](#)
 - Spinnaker::IChunkData, [895](#)
- GetCpuStats
 - CpuUtil, [418](#)
- GetCurrentEntry
 - IEnumeration Interface, [324](#)
 - Spinnaker::GenApi::CEnumerationTRef, [698](#)
 - Spinnaker::GenApi::EnumNode, [824](#)
- GetData
 - Spinnaker::IImage, [923](#)
 - Spinnaker::Image, [965](#)
- GetDataAbsoluteMax
 - Spinnaker::IImage, [923](#)
 - Spinnaker::Image, [965](#)
- GetDataAbsoluteMin
 - Spinnaker::IImage, [923](#)
 - Spinnaker::Image, [965](#)
- GetDefaultColorProcessing
 - Spinnaker::Image, [966](#)
- GetDescription
 - INode Interface, [339](#)
 - Spinnaker::GenApi::Node, [1090](#)
- GetDeviceEventId
 - Spinnaker::DeviceEventHandler, [801](#)
 - Spinnaker::IDeviceEventHandler, [916](#)
- GetDeviceEventName
 - Spinnaker::DeviceEventHandler, [801](#)
 - Spinnaker::IDeviceEventHandler, [916](#)
- GetDeviceName
 - INodeMap Interface, [347](#)
 - Spinnaker::GenApi::Node, [1090](#)
 - Spinnaker::GenApi::NodeMap, [1100](#)
- GetDeviceNodeMap
 - Spinnaker::IDataStream, [905](#)
- GetDeviceSerial
 - AcquisitionMultipleCameraRecovery.cpp, [1430](#)
- GetDeviceVersion
 - IDeviceInfo Interface, [319](#)
 - Spinnaker::GenApi::NodeMap, [1100](#)
- GetDisplayName
 - INode Interface, [339](#)
 - Spinnaker::GenApi::Node, [1090](#)
- GetDisplayNotation
 - IFloat Interface, [330](#)
 - Spinnaker::GenApi::FloatNode, [849](#)
- GetDisplayPrecision
 - IFloat Interface, [330](#)
 - Spinnaker::GenApi::FloatNode, [849](#)
- GetDocuURL
 - INode Interface, [339](#)
 - Spinnaker::GenApi::Node, [1090](#)
- GetEncoderValue
 - Spinnaker::ChunkData, [731](#)
 - Spinnaker::IChunkData, [895](#)
- GetEntries
 - IEnumeration Interface, [324](#)
 - Spinnaker::GenApi::EnumNode, [824](#)
- GetEntry
 - IEnumeration Interface, [325](#)
 - IEnumerationT Interface, [327](#)
 - Spinnaker::GenApi::CEnumerationTRef, [699](#)
 - Spinnaker::GenApi::EnumNode, [825](#)
- GetEntryByName
 - IEnumeration Interface, [325](#)
 - Spinnaker::GenApi::EnumNode, [825](#)
- GetEnumAlias
 - Spinnaker::GenApi::CFloatPtr, [721](#)
 - Spinnaker::GenApi::FloatNode, [849](#)
- GetEnumerationLogFileName
 - AdapterConfig, [415](#)
- GetError
 - Spinnaker::Exception, [839](#)
- GetErrorMessage
 - SpinUpdate.h, [1377](#)
 - Spinnaker::Exception, [839](#)
- GetEventIdLength
 - Spinnaker::GenApi::CEventPort, [715](#)
- GetEventId
 - INode Interface, [339](#)
 - Spinnaker::GenApi::Node, [1090](#)
- GetEventPayloadData
 - Spinnaker::EventHandler, [833](#)
- GetEventPayloadDataSize
 - Spinnaker::EventHandler, [833](#)
- GetEventType
 - Spinnaker::EventHandler, [833](#)
- GetExposureEndLineStatusAll
 - Spinnaker::ChunkData, [731](#)
 - Spinnaker::IChunkData, [896](#)
- GetExposureTime
 - Spinnaker::ChunkData, [732](#)
 - Spinnaker::IChunkData, [896](#)
- GetFeatureBagHandle
 - Spinnaker::GenApi::CFeatureBag, [718](#)
- GetFeatures
 - Spinnaker::GenApi::CategoryNode, [675](#)
- GetFileName
 - Spinnaker::Exception, [840](#)
- GetFiles
 - GCUtilities Utility, [306](#)
- GetFloatAlias
 - Spinnaker::GenApi::IntegerNode, [1027](#)
- GetFrameId
 - Spinnaker::ChunkData, [732](#)
 - Spinnaker::IChunkData, [896](#)
 - Spinnaker::IImage, [923](#)
 - Spinnaker::Image, [966](#)
- GetFullErrorMessage
 - Spinnaker::Exception, [840](#)
- GetFunctionName
 - Spinnaker::Exception, [840](#)
- GetGain
 - Spinnaker::ChunkData, [732](#)

- Spinnaker::IChunkData, 896
- GetGenApiVersion
 - IDeviceInfo Interface, 319
 - Spinnaker::GenApi::NodeMap, 1100
- GetGenICamCLProtocolFolder
 - GCUtilities Utility, 307
- GetGenICamCacheFolder
 - GCUtilities Utility, 306
- GetGenICamLogConfig
 - GCUtilities Utility, 307
- GetGuiXml
 - Spinnaker::CameraBase, 657
 - Spinnaker::ICameraBase, 883
- GetHeatmapColorGradient
 - Spinnaker::ImageUtilityHeatmap, 1008
- GetHeatmapRange
 - Spinnaker::ImageUtilityHeatmap, 1009
- GetHeight
 - Spinnaker::ChunkData, 732
 - Spinnaker::IChunkData, 896
 - Spinnaker::Image, 924
 - Spinnaker::Image, 966
- GetHistogram
 - Spinnaker::ImageStatistics, 936
 - Spinnaker::ImageStatistics, 995
- GetID
 - Spinnaker::Image, 924
 - Spinnaker::Image, 967
- GetImage
 - Spinnaker::ChunkData, 733
 - Spinnaker::IChunkData, 896
- getImageCount
 - ImageEventHandlerImpl, 987
- GetImageData
 - Spinnaker::Image, 924
 - Spinnaker::Image, 967
- GetImageSize
 - Spinnaker::Image, 924
 - Spinnaker::Image, 967
- GetImageStatus
 - Spinnaker::Image, 924
 - Spinnaker::Image, 967
- GetImageStatusDescription
 - Spinnaker::Image, 968
- GetInc
 - IFloat Interface, 330
 - Spinnaker::GenApi::FloatNode, 849
 - Spinnaker::GenApi::IntegerNode, 1028
- GetIncMode
 - IFloat Interface, 330
 - Spinnaker::GenApi::FloatNode, 850
 - Spinnaker::GenApi::IntegerNode, 1028
- GetInferenceBoundingBoxResult
 - Spinnaker::ChunkData, 733
 - Spinnaker::IChunkData, 897
- GetInferenceConfidence
 - Spinnaker::ChunkData, 733
 - Spinnaker::IChunkData, 897
- GetInferenceFrameId
 - Spinnaker::ChunkData, 733
 - Spinnaker::IChunkData, 897
- GetInferenceResult
 - Spinnaker::ChunkData, 734
 - Spinnaker::IChunkData, 897
- GetInstance
 - Spinnaker::System, 1147
- GetIntAlias
 - Spinnaker::GenApi::CFloatPtr, 721
 - Spinnaker::GenApi::FloatNode, 850
- GetIntValue
 - IEnumeration Interface, 325
 - Spinnaker::GenApi::EnumNode, 825
- GetInterfaceId
 - InterfaceEventHandlerImpl, 1044
- GetInterfaceName
 - Pointer Class, 384
- GetInterfaces
 - Spinnaker::ISystem, 1060
 - Spinnaker::System, 1147
- GetLength
 - IRegister Interfaces, 362
 - Spinnaker::GenApi::RegisterNode, 1130
- GetLibraryVersion
 - Spinnaker::ISystem, 1060
 - Spinnaker::System, 1147
- GetLineNumber
 - Spinnaker::Exception, 840
- GetLinePitch
 - Spinnaker::ChunkData, 734
 - Spinnaker::IChunkData, 897
- GetLineStatusAll
 - Spinnaker::ChunkData, 734
 - Spinnaker::IChunkData, 897
- GetListOfValidValues
 - IFloat Interface, 330
 - Spinnaker::GenApi::FloatNode, 850
 - Spinnaker::GenApi::IntegerNode, 1028
- GetLock
 - INodeMap Interface, 347
 - Spinnaker::GenApi::NodeMap, 1100
 - Spinnaker::GenICam::LockableObject, 1072
- GetLogMessage
 - Spinnaker::LoggingEventData, 1074
- GetLoggingEventPriorityLevel
 - Spinnaker::ISystem, 1060
 - Spinnaker::System, 1148
- GetMax
 - IFloat Interface, 330
 - Spinnaker::GenApi::FloatNode, 850
 - Spinnaker::GenApi::IntegerNode, 1028
- getMaxImages
 - ImageEventHandlerImpl, 987
- GetMaxIpAddress
 - AdapterConfig, 416
- GetMaxLength
 - IString Class, 367

- Spinnaker::GenApi::StringNode, 1139
- GetMean
 - Spinnaker::ImageStatistics, 936
 - Spinnaker::ImageStatistics, 996
- GetMin
 - IFloat Interface, 331
 - Spinnaker::GenApi::FloatNode, 850
 - Spinnaker::GenApi::IntegerNode, 1028
- GetMinIpAddress
 - AdapterConfig, 416
- GetModelName
 - Spinnaker::GenApi::NodeMap, 1100
- GetModulePathFromFunction
 - GCUtilities Utility, 307
- GetNDC
 - Spinnaker::LoggingEventData, 1074
- GetName
 - Spinnaker::GenApi::Node, 1090
- GetNameSpace
 - INode Interface, 339
 - Spinnaker::GenApi::Node, 1090
- GetNextImage
 - Spinnaker::CameraBase, 657
 - Spinnaker::ICameraBase, 884
 - Spinnaker::IDataStream, 905
- GetNextImageInternal
 - Spinnaker::IDataStream, 905
- GetNode
 - INodeMap Interface, 347
 - Spinnaker::GenApi::CNodeCallback, 749
 - Spinnaker::GenApi::NodeMap, 1100
 - Spinnaker::GenApi::ValueNode, 1207
- GetNodeHandle
 - Spinnaker::GenApi::Node, 1091
- GetNodeMap
 - INode Interface, 339
 - Spinnaker::CameraBase, 658
 - Spinnaker::GenApi::Node, 1091
 - Spinnaker::ICameraBase, 884
 - Spinnaker::IDataStream, 905
- GetNodeMapHandle
 - Spinnaker::GenApi::NodeMap, 1101
- GetNodeStatistics
 - Spinnaker::GenApi::CNodeMapFactory, 756
- GetNodes
 - Spinnaker::GenApi::NodeMap, 1101
- GetNumChannels
 - Spinnaker::Image, 924
 - Spinnaker::Image, 968
- GetNumDataStreams
 - Spinnaker::CameraBase, 658
 - Spinnaker::ICameraBase, 884
- GetNumDecompressionThreads
 - Spinnaker::Image, 968
- GetNumImagesInUse
 - Spinnaker::CameraBase, 659
 - Spinnaker::ICameraBase, 884
 - Spinnaker::IDataStream, 905
- GetNumNodes
 - INodeMap Interface, 347
 - Spinnaker::GenApi::NodeMap, 1101
- GetNumPixelValues
 - Spinnaker::ImageStatistics, 936
 - Spinnaker::ImageStatistics, 996
- GetNumReads
 - Spinnaker::GenApi::CTestPortStruct, 793
- GetNumWrites
 - Spinnaker::GenApi::CTestPortStruct, 793
- GetNumericValue
 - IEnumEntry Interface, 322
 - Spinnaker::GenApi::EnumEntryNode, 820
- GetOffsetX
 - Spinnaker::ChunkData, 734
 - Spinnaker::IChunkData, 898
- GetOffsetY
 - Spinnaker::ChunkData, 735
 - Spinnaker::IChunkData, 898
- GetParamStr
 - AdapterConfigException, 483
- GetParents
 - INode Interface, 339
 - Spinnaker::GenApi::Node, 1091
- GetPartSelector
 - Spinnaker::ChunkData, 735
 - Spinnaker::IChunkData, 898
- GetPayloadType
 - Spinnaker::Image, 925
 - Spinnaker::Image, 969
- GetPerformanceCounter
 - PerformanceCounter, 419
- GetPixelDynamicRangeMax
 - Spinnaker::ChunkData, 735
 - Spinnaker::IChunkData, 898
- GetPixelDynamicRangeMin
 - Spinnaker::ChunkData, 735
 - Spinnaker::IChunkData, 898
- GetPixelFormat
 - Spinnaker::Image, 925
 - Spinnaker::Image, 969
- GetPixelFormatIntType
 - Spinnaker::Image, 925
 - Spinnaker::Image, 969
- GetPixelFormatName
 - Spinnaker::Image, 925
 - Spinnaker::Image, 970
- GetPixelValueRange
 - Spinnaker::ImageStatistics, 936
 - Spinnaker::ImageStatistics, 996
- GetPollingTime
 - INode Interface, 340
 - Spinnaker::GenApi::Node, 1091
- GetPort
 - Spinnaker::IDataStream, 905
- GetPortHandle
 - Spinnaker::GenApi::PortNode, 1116
- GetPortReplayHandle

- Spinnaker::GenApi::PortReplay, [1124](#)
- GetPortWriteListHandle
 - Spinnaker::GenApi::CPortWriteList, [783](#)
- GetPrincipalInterfaceType
 - INode Interface, [340](#)
 - Spinnaker::GenApi::CChunkPort, [692](#)
 - Spinnaker::GenApi::CEventPort, [715](#)
 - Spinnaker::GenApi::CTestPortStruct, [793](#)
 - Spinnaker::GenApi::Node, [1091](#)
- GetPriority
 - Spinnaker::LoggingEventData, [1074](#)
- GetPriorityName
 - Spinnaker::LoggingEventData, [1075](#)
- GetPrivateData
 - Spinnaker::Image, [925](#)
 - Spinnaker::Image, [970](#)
- GetProductGuid
 - IDeviceInfo Interface, [320](#)
 - Spinnaker::GenApi::NodeMap, [1101](#)
- GetProperty
 - INode Interface, [340](#)
 - Spinnaker::GenApi::Node, [1092](#)
- GetPropertyNames
 - INode Interface, [340](#)
 - Spinnaker::GenApi::Node, [1092](#)
- GetQuadFileNameAppendage
 - Polarization.cpp, [1489](#)
- GetRange
 - Spinnaker::ImageStatistics, [936](#)
 - Spinnaker::ImageStatistics, [997](#)
- GetRepresentation
 - IFloat Interface, [331](#)
 - Spinnaker::GenApi::FloatNode, [850](#)
 - Spinnaker::GenApi::IntegerNode, [1028](#)
- GetScan3dAxisMax
 - Spinnaker::ChunkData, [736](#)
 - Spinnaker::IChunkData, [898](#)
- GetScan3dAxisMin
 - Spinnaker::ChunkData, [736](#)
 - Spinnaker::IChunkData, [899](#)
- GetScan3dCoordinateOffset
 - Spinnaker::ChunkData, [736](#)
 - Spinnaker::IChunkData, [899](#)
- GetScan3dCoordinateReferenceValue
 - Spinnaker::ChunkData, [736](#)
 - Spinnaker::IChunkData, [899](#)
- GetScan3dCoordinateScale
 - Spinnaker::ChunkData, [737](#)
 - Spinnaker::IChunkData, [899](#)
- GetScan3dInvalidDataValue
 - Spinnaker::ChunkData, [737](#)
 - Spinnaker::IChunkData, [899](#)
- GetScan3dTransformValue
 - Spinnaker::ChunkData, [737](#)
 - Spinnaker::IChunkData, [899](#)
- GetScanLineSelector
 - Spinnaker::ChunkData, [737](#)
 - Spinnaker::IChunkData, [900](#)
- GetSchemaVersion
 - IDeviceInfo Interface, [320](#)
 - Spinnaker::GenApi::NodeMap, [1101](#)
- GetSecondsCounter
 - SecondsCounter, [420](#)
- GetSelectedCameraIndex
 - SpinnakerDirectShow.h, [234](#)
- GetSelectedFeatures
 - ISelector Interface, [363](#)
 - Spinnaker::GenApi::Node, [1092](#)
- GetSelectingFeatures
 - ISelector Interface, [363](#)
 - Spinnaker::GenApi::Node, [1092](#)
- GetSelectorList
 - ISelectorDigit Interface, [364](#)
 - Spinnaker::GenApi::CSelectorSet, [790](#)
- GetSequencerSetActive
 - Spinnaker::ChunkData, [738](#)
 - Spinnaker::IChunkData, [900](#)
- GetSerialDataLength
 - Spinnaker::ChunkData, [738](#)
 - Spinnaker::IChunkData, [900](#)
- GetSize
 - Spinnaker::CameraList, [671](#)
 - Spinnaker::ICameraList, [891](#)
 - Spinnaker::IInterfaceList, [949](#)
 - Spinnaker::InterfaceList, [1049](#)
- GetStandardNameSpace
 - IDeviceInfo Interface, [320](#)
 - Spinnaker::GenApi::NodeMap, [1101](#)
- GetStatistics
 - Spinnaker::ImageStatistics, [937](#)
 - Spinnaker::ImageStatistics, [997](#)
- GetStreamChannelID
 - Spinnaker::ChunkData, [738](#)
 - Spinnaker::IChunkData, [900](#)
- GetStreamInfoBool8Type
 - Spinnaker::IDataStream, [905](#)
- GetStreamInfoSizeType
 - Spinnaker::IDataStream, [906](#)
- GetStreamType
 - Spinnaker::IDataStream, [906](#)
- GetStride
 - Spinnaker::Image, [925](#)
 - Spinnaker::Image, [970](#)
- GetSubnetMaskLength
 - AdapterConfig, [416](#)
- GetSupportedSchemaVersions
 - INodeMapDyn Interface, [350](#)
 - Spinnaker::GenApi::CNodeMapFactory, [756](#)
 - Spinnaker::GenApi::NodeMap, [1102](#)
- GetSwapEndianness
 - IPortConstruct Interface, [358](#)
 - Spinnaker::GenApi::CChunkPort, [692](#)
 - Spinnaker::GenApi::CEventPort, [715](#)
 - Spinnaker::GenApi::CPortImpl, [779](#)
 - Spinnaker::GenApi::PortNode, [1116](#)
- GetSymbolic

- IEnumEntry Interface, [322](#)
 - Spinnaker::GenApi::EnumEntryNode, [821](#)
- GetSymbolics
 - Spinnaker::GenApi::EnumNode, [825](#)
- GetTLDeviceNodeMap
 - Spinnaker::CameraBase, [659](#)
 - Spinnaker::ICameraBase, [884](#)
- GetTLNodeMap
 - Spinnaker::IInterface, [940](#)
 - Spinnaker::ISystem, [1060](#)
 - Spinnaker::Interface, [1033](#)
 - Spinnaker::System, [1148](#)
- GetTLPayloadType
 - Spinnaker::IImage, [926](#)
 - Spinnaker::Image, [971](#)
- GetTLPixelFormat
 - Spinnaker::IImage, [926](#)
 - Spinnaker::Image, [971](#)
- GetTLPixelFormatNamespace
 - Spinnaker::IImage, [926](#)
 - Spinnaker::Image, [972](#)
- GetTLStreamNodeMap
 - Spinnaker::CameraBase, [659](#)
 - Spinnaker::ICameraBase, [884](#)
- GetThreadName
 - Spinnaker::LoggingEventData, [1075](#)
- GetTimeStamp
 - Spinnaker::IImage, [926](#)
 - Spinnaker::Image, [971](#)
- GetTimerValue
 - Spinnaker::ChunkData, [738](#)
 - Spinnaker::IChunkData, [900](#)
- GetTimestamp
 - Spinnaker::ChunkData, [739](#)
 - Spinnaker::IChunkData, [900](#)
 - Spinnaker::LoggingEventData, [1075](#)
- GetTimestampLatchValue
 - Spinnaker::ChunkData, [739](#)
 - Spinnaker::IChunkData, [901](#)
- GetToolTip
 - IDeviceInfo Interface, [320](#)
 - Spinnaker::GenApi::Node, [1092](#)
 - Spinnaker::GenApi::NodeMap, [1102](#)
- GetTransferBlockID
 - Spinnaker::ChunkData, [739](#)
 - Spinnaker::IChunkData, [901](#)
- GetTransferQueueCurrentBlockCount
 - Spinnaker::ChunkData, [739](#)
 - Spinnaker::IChunkData, [901](#)
- GetUniqueID
 - Spinnaker::CameraBase, [660](#)
 - Spinnaker::ICameraBase, [885](#)
- GetUnit
 - IFloat Interface, [331](#)
 - Spinnaker::GenApi::FloatNode, [851](#)
 - Spinnaker::GenApi::IntegerNode, [1029](#)
- GetUserBufferCount
 - Spinnaker::CameraBase, [660](#)
 - Spinnaker::ICameraBase, [885](#)
- GetUserBufferSize
 - Spinnaker::CameraBase, [660](#)
 - Spinnaker::ICameraBase, [885](#)
- GetUserBufferTotalSize
 - Spinnaker::CameraBase, [661](#)
 - Spinnaker::ICameraBase, [885](#)
- GetValidPayloadSize
 - Spinnaker::IImage, [926](#)
 - Spinnaker::Image, [972](#)
- GetValue
 - IBoolean Interface, [311](#)
 - Spinnaker::GenApi::BooleanNode, [497](#)
 - Spinnaker::GenApi::CEnumerationTRef, [699](#)
 - Spinnaker::GenApi::Counter, [771](#)
 - Spinnaker::GenApi::EnumEntryNode, [821](#)
 - Spinnaker::GenApi::FloatNode, [851](#)
 - Spinnaker::GenApi::IntegerNode, [1029](#)
 - Spinnaker::GenApi::StringNode, [1139](#)
- GetValueOfEnvironmentVariable
 - GCUtilities Utility, [307](#), [308](#)
- GetVendorName
 - IDeviceInfo Interface, [320](#)
 - Spinnaker::GenApi::NodeMap, [1102](#)
- GetVersion
 - Chunk Data Inference Class, [173](#)
- GetVersionGuid
 - IDeviceInfo Interface, [320](#)
 - Spinnaker::GenApi::NodeMap, [1102](#)
- GetVisibility
 - INode Interface, [340](#)
 - Spinnaker::GenApi::Node, [1093](#)
- GetWidth
 - Spinnaker::ChunkData, [740](#)
 - Spinnaker::IChunkData, [901](#)
 - Spinnaker::IImage, [926](#)
 - Spinnaker::Image, [973](#)
- GetXOffset
 - Spinnaker::IImage, [927](#)
 - Spinnaker::Image, [973](#)
- GetXPadding
 - Spinnaker::IImage, [927](#)
 - Spinnaker::Image, [973](#)
- GetYOffset
 - Spinnaker::IImage, [927](#)
 - Spinnaker::Image, [974](#)
- GetYPadding
 - Spinnaker::IImage, [927](#)
 - Spinnaker::Image, [974](#)
- getline
 - Spinnaker::GenICam, [478](#), [479](#)
- GevActionDeviceKey
 - Spinnaker::TransportLayerInterface, [1180](#)
- GevActionGroupKey
 - Spinnaker::TransportLayerInterface, [1180](#)
- GevActionGroupMask
 - Spinnaker::TransportLayerInterface, [1181](#)
- GevActionTime

- Spinnaker::TransportLayerInterface, [1181](#)
- GevActiveLinkCount
 - Spinnaker::Camera, [608](#)
- GevCCPEnum
 - TransportLayerDefs Class, [247](#)
- GevCCPEnums
 - CameraDefs Class, [124](#)
- GevCCP
 - Spinnaker::Camera, [608](#)
 - Spinnaker::TransportLayerDevice, [1171](#)
- GevCurrentDefaultGateway
 - Spinnaker::Camera, [608](#)
- GevCurrentIPAddress
 - Spinnaker::Camera, [609](#)
- GevCurrentIPConfigurationDHCP
 - Spinnaker::Camera, [609](#)
- GevCurrentIPConfigurationLLA
 - Spinnaker::Camera, [609](#)
- GevCurrentIPConfigurationPersistentIP
 - Spinnaker::Camera, [609](#)
- GevCurrentPhysicalLinkConfiguration
 - Spinnaker::Camera, [609](#)
- GevCurrentPhysicalLinkConfigurationEnums
 - CameraDefs Class, [125](#)
- GevCurrentSubnetMask
 - Spinnaker::Camera, [609](#)
- GevDeviceAutoForceIP
 - Spinnaker::TransportLayerDevice, [1171](#)
 - Spinnaker::TransportLayerInterface, [1181](#)
- GevDeviceDiscoverMaximumPacketSize
 - Spinnaker::TransportLayerDevice, [1172](#)
- GevDeviceForceGateway
 - Spinnaker::TransportLayerDevice, [1172](#)
 - Spinnaker::TransportLayerInterface, [1181](#)
- GevDeviceForceIPAddress
 - Spinnaker::TransportLayerDevice, [1172](#)
 - Spinnaker::TransportLayerInterface, [1181](#)
- GevDeviceForceIP
 - Spinnaker::TransportLayerDevice, [1172](#)
 - Spinnaker::TransportLayerInterface, [1181](#)
- GevDeviceForceSubnetMask
 - Spinnaker::TransportLayerDevice, [1172](#)
 - Spinnaker::TransportLayerInterface, [1182](#)
- GevDeviceGateway
 - Spinnaker::TransportLayerDevice, [1172](#)
 - Spinnaker::TransportLayerInterface, [1182](#)
- GevDeviceIPAddress
 - Spinnaker::TransportLayerDevice, [1173](#)
 - Spinnaker::TransportLayerInterface, [1182](#)
- GevDeviceIsWrongSubnet
 - Spinnaker::TransportLayerDevice, [1173](#)
- GevDeviceMACAddress
 - Spinnaker::TransportLayerDevice, [1173](#)
 - Spinnaker::TransportLayerInterface, [1182](#)
- GevDeviceMaximumPacketSize
 - Spinnaker::TransportLayerDevice, [1173](#)
- GevDeviceMaximumRetryCount
 - Spinnaker::TransportLayerDevice, [1173](#)
- GevDeviceModelsBigEndian
 - Spinnaker::TransportLayerDevice, [1173](#)
- GevDevicePort
 - Spinnaker::TransportLayerDevice, [1174](#)
- GevDeviceReadAndWriteTimeout
 - Spinnaker::TransportLayerDevice, [1174](#)
- GevDeviceSubnetMask
 - Spinnaker::TransportLayerDevice, [1174](#)
 - Spinnaker::TransportLayerInterface, [1182](#)
- GevDiscoveryAckDelay
 - Spinnaker::Camera, [610](#)
- GevFailedPacketCount
 - Spinnaker::TransportLayerStream, [1189](#)
- GevFirstURL
 - Spinnaker::Camera, [610](#)
- GevGVCPExtendedStatusCodes
 - Spinnaker::Camera, [610](#)
- GevGVCPExtendedStatusCodesSelector
 - Spinnaker::Camera, [610](#)
- GevGVCPExtendedStatusCodesSelectorEnums
 - CameraDefs Class, [125](#)
- GevGVCPHeartbeatDisable
 - Spinnaker::Camera, [610](#)
- GevGVCPPendingAck
 - Spinnaker::Camera, [610](#)
- GevGVCPPendingTimeout
 - Spinnaker::Camera, [611](#)
- GevGVSPExtendedIDMode
 - Spinnaker::Camera, [611](#)
- GevGVSPExtendedIDModeEnums
 - CameraDefs Class, [125](#)
- GevHeartbeatTimeout
 - Spinnaker::Camera, [611](#)
- GevIEEE1588
 - Spinnaker::Camera, [611](#)
- GevIEEE1588ClockAccuracy
 - Spinnaker::Camera, [611](#)
- GevIEEE1588ClockAccuracyEnums
 - CameraDefs Class, [126](#)
- GevIEEE1588Mode
 - Spinnaker::Camera, [611](#)
- GevIEEE1588ModeEnums
 - CameraDefs Class, [126](#)
- GevIEEE1588Status
 - Spinnaker::Camera, [612](#)
- GevIEEE1588StatusEnums
 - CameraDefs Class, [126](#)
- GevIPConfigurationStatus
 - Spinnaker::Camera, [612](#)
- GevIPConfigurationStatusEnums
 - CameraDefs Class, [127](#)
- GevInterfaceDefaultGateway
 - Spinnaker::TransportLayerSystem, [1198](#)
- GevInterfaceDefaultIPAddress
 - Spinnaker::TransportLayerSystem, [1198](#)
- GevInterfaceDefaultSubnetMask
 - Spinnaker::TransportLayerSystem, [1198](#)
- GevInterfaceGateway

- Spinnaker::TransportLayerInterface, [1182](#)
- GevInterfaceGatewaySelector
 - Spinnaker::TransportLayerInterface, [1183](#)
- GevInterfaceMACAddress
 - Spinnaker::TransportLayerInterface, [1183](#)
 - Spinnaker::TransportLayerSystem, [1199](#)
- GevInterfaceMTU
 - Spinnaker::TransportLayerInterface, [1183](#)
- GevInterfaceReceiveLinkSpeed
 - Spinnaker::TransportLayerInterface, [1183](#)
- GevInterfaceSelector
 - Spinnaker::Camera, [612](#)
- GevInterfaceSubnetIPAddress
 - Spinnaker::TransportLayerInterface, [1183](#)
- GevInterfaceSubnetMask
 - Spinnaker::TransportLayerInterface, [1183](#)
- GevInterfaceSubnetSelector
 - Spinnaker::TransportLayerInterface, [1184](#)
- GevInterfaceTransmitLinkSpeed
 - Spinnaker::TransportLayerInterface, [1184](#)
- GevMACAddress
 - Spinnaker::Camera, [612](#)
- GevMCDA
 - Spinnaker::Camera, [612](#)
- GevMCPHostPort
 - Spinnaker::Camera, [612](#)
- GevMCRC
 - Spinnaker::Camera, [613](#)
- GevMCSP
 - Spinnaker::Camera, [613](#)
- GevMCTT
 - Spinnaker::Camera, [613](#)
- GevMaximumNumberResendRequests
 - Spinnaker::TransportLayerStream, [1189](#)
- GevNumberOfInterfaces
 - Spinnaker::Camera, [613](#)
- GevPAUSEFrameReception
 - Spinnaker::Camera, [613](#)
- GevPAUSEFrameTransmission
 - Spinnaker::Camera, [613](#)
- GevPacketResendMode
 - Spinnaker::TransportLayerStream, [1190](#)
- GevPacketResendTimeout
 - Spinnaker::TransportLayerStream, [1190](#)
- GevPersistentDefaultGateway
 - Spinnaker::Camera, [614](#)
- GevPersistentIPAddress
 - Spinnaker::Camera, [614](#)
- GevPersistentSubnetMask
 - Spinnaker::Camera, [614](#)
- GevPhysicalLinkConfiguration
 - Spinnaker::Camera, [614](#)
- GevPhysicalLinkConfigurationEnums
 - CameraDefs Class, [127](#)
- GevPrimaryApplicationIPAddress
 - Spinnaker::Camera, [614](#)
- GevPrimaryApplicationSocket
 - Spinnaker::Camera, [614](#)
- GevPrimaryApplicationSwitchoverKey
 - Spinnaker::Camera, [615](#)
- GevResendPacketCount
 - Spinnaker::TransportLayerStream, [1190](#)
- GevResendRequestCount
 - Spinnaker::TransportLayerStream, [1190](#)
- GevSCCFGAllInTransmission
 - Spinnaker::Camera, [615](#)
- GevSCCFGExtendedChunkData
 - Spinnaker::Camera, [615](#)
- GevSCCFGPacketResendDestination
 - Spinnaker::Camera, [615](#)
- GevSCCFGUnconditionalStreaming
 - Spinnaker::Camera, [615](#)
- GevSCDA
 - Spinnaker::Camera, [615](#)
- GevSCPDDirection
 - Spinnaker::Camera, [616](#)
- GevSCPHostPort
 - Spinnaker::Camera, [616](#)
- GevSCPInterfaceIndex
 - Spinnaker::Camera, [616](#)
- GevSCPSBigEndian
 - Spinnaker::Camera, [616](#)
- GevSCPSDoNotFragment
 - Spinnaker::Camera, [616](#)
- GevSCPSFireTestPacket
 - Spinnaker::Camera, [617](#)
- GevSCPSPacketSize
 - Spinnaker::Camera, [617](#)
- GevSCPD
 - Spinnaker::Camera, [616](#)
- GevSCSP
 - Spinnaker::Camera, [617](#)
- GevSCZoneConfigurationLock
 - Spinnaker::Camera, [617](#)
- GevSCZoneCount
 - Spinnaker::Camera, [617](#)
- GevSCZoneDirectionAll
 - Spinnaker::Camera, [617](#)
- GevSecondURL
 - Spinnaker::Camera, [618](#)
- GevStreamChannelSelector
 - Spinnaker::Camera, [618](#)
- GevSupportedOption
 - Spinnaker::Camera, [618](#)
- GevSupportedOptionSelector
 - Spinnaker::Camera, [618](#)
- GevSupportedOptionSelectorEnums
 - CameraDefs Class, [127](#)
- GevTimestampTickFrequency
 - Spinnaker::Camera, [618](#)
- GevTotalPacketCount
 - Spinnaker::TransportLayerStream, [1190](#)
- GevVersionMajor
 - Spinnaker::TransportLayerDevice, [1174](#)
 - Spinnaker::TransportLayerSystem, [1199](#)
- GevVersionMinor

- Spinnaker::TransportLayerDevice, [1174](#)
- Spinnaker::TransportLayerSystem, [1199](#)
- GigEVisionPerformance.cpp
 - AcquireImages, [1461](#)
 - argBayerRG, [1463](#)
 - argDuration, [1463](#)
 - argMaxFrames, [1463](#)
 - argNumImages, [1463](#)
 - argPacketDelay, [1463](#)
 - argPacketSize, [1463](#)
 - argPrintUsage, [1464](#)
 - argRelease, [1464](#)
 - argUserSetFrames, [1464](#)
 - cpuUsageInfo, [1464](#)
 - EnableManualFramerate, [1461](#)
 - getCameraCategory, [1461](#)
 - IsRelease, [1464](#)
 - main, [1461](#)
 - NumImagesToGrab, [1464](#)
 - PacketDelayToSet, [1464](#)
 - PacketSizeToSet, [1464](#)
 - ParseArguments, [1462](#)
 - PixelFormatToSet, [1465](#)
 - PrintAllNodes, [1462](#)
 - PrintCPUUsage, [1462](#)
 - PrintDataStreamInfo, [1462](#)
 - PrintDeviceInfo, [1462](#)
 - PrintUsage, [1462](#)
 - RunSingleCamera, [1462](#)
 - SetFrameRate, [1463](#)
 - TestDuration, [1465](#)
 - UseDuration, [1465](#)
 - UseMaxFramerate, [1465](#)
 - UserSetFramerate, [1465](#)
- globalCamList
 - AcquisitionMultipleCameraRecovery.cpp, [1431](#)
- GrabInfo, [867](#)
 - GrabInfo, [867](#)
 - imageEventHandler, [867](#)
 - numImagesGrabbed, [867](#)
 - numIncompleteImages, [868](#)
 - numRemovals, [868](#)
- GrabNextImageByTrigger
 - BufferHandling.cpp, [1439](#)
 - Trigger.cpp, [1499](#)
 - Trigger_QuickSpin.cpp, [1501](#)
- GrabTwoImages
 - LogicBlock.cpp, [1479](#)
- GuiXmlManifestAddress
 - Spinnaker::Camera, [618](#)
- Guru
 - Types Enums, [409](#)
- H264Option, [878](#)
 - Spinnaker::Video::H264Option, [879](#)
- HasCRC
 - Spinnaker::GenApi::CChunkAdapterDcam, [681](#)
 - Spinnaker::Image, [927](#)
 - Spinnaker::Image, [974](#)
- HasInc
 - IFloat Interface, [331](#)
 - Spinnaker::GenApi::FloatNode, [851](#)
- Header
 - GVCP_EVENT_REQUEST_EXTENDED_ID, [874](#)
 - GVCP_EVENT_REQUEST, [873](#)
 - GVCP_EVENTDATA_REQUEST_EXTENDED_ID, [877](#)
 - GVCP_EVENTDATA_REQUEST, [875](#)
- HeatmapColor
 - Spinnaker::ImageUtilityHeatmap, [1007](#)
- Height
 - Spinnaker::Camera, [619](#)
- height
 - Spinnaker::Video::H264Option, [879](#)
- HeightMax
 - Spinnaker::Camera, [619](#)
- HostAdapterDriverVersion
 - Spinnaker::TransportLayerInterface, [1184](#)
- HostAdapterName
 - Spinnaker::TransportLayerInterface, [1184](#)
- HostAdapterVendor
 - Spinnaker::TransportLayerInterface, [1184](#)
- IBase
 - IBase Interface, [276](#)
- IBase Interface, [276](#)
 - IBase, [276](#)
- IBoolean
 - IBoolean Interface, [312](#)
- IBoolean Interface, [311](#)
 - GetValue, [311](#)
 - IBoolean, [312](#)
 - operator(), [312](#)
 - operator=, [312](#)
 - Verify, [312](#)
- ICameraBase, [880](#)
 - Spinnaker::ICameraBase, [882](#)
 - Spinnaker::TransportLayerDevice, [1168](#)
 - Spinnaker::TransportLayerStream, [1189](#)
- ICameraList, [889](#)
 - Spinnaker::ICameraList, [890](#)
- ICategory
 - ICategory Interfaces, [313](#)
- ICategory Interfaces, [313](#)
 - ICategory, [313](#)
- ICChunkData, [893](#)
 - Spinnaker::ICChunkData, [894](#)
- ICChunkData Class, [256](#)
- ICChunkPort
 - ICChunkPort Interface, [315](#)
- ICChunkPort Interface, [314](#)
 - CHUNK_BASE_ADDRESS_REGISTER_LEN, [314](#)
 - CHUNK_BASE_ADDRESS_REGISTER, [314](#)
 - CHUNK_LENGTH_REGISTER_LEN, [315](#)
 - CHUNK_LENGTH_REGISTER, [315](#)
 - CacheChunkData, [315](#)
 - ICChunkPort, [315](#)
- ICommand

- ICommand Interface, [316](#)
- ICommand Interface, [316](#)
 - ICommand, [316](#)
 - IsDone, [316](#)
- IDataStream, [902](#)
 - Spinnaker::EventHandler, [834](#)
 - Spinnaker::IDataStream, [903](#)
 - Spinnaker::Image, [981](#)
- IDestroy
 - IDestroy Interface, [318](#)
- IDestroy Interface, [318](#)
 - IDestroy, [318](#)
- IDevFileStream
 - Spinnaker::GenApi, [475](#)
- IDevFileStreamBase< CharType, Traits >, [908](#)
- IDevFileStreamBuf
 - Spinnaker::GenApi::IDevFileStreamBuf, [912](#)
- IDevFileStreamBuf< CharType, Traits >, [911](#)
- IDeviceArrivalEventHandler, [913](#)
 - Spinnaker::IDeviceArrivalEventHandler, [914](#)
- IDeviceEventHandler, [915](#)
 - Spinnaker::IDeviceEventHandler, [916](#)
- IDeviceInfo
 - IDeviceInfo Interface, [321](#)
- IDeviceInfo Interface, [319](#)
 - GetDeviceVersion, [319](#)
 - GetGenApiVersion, [319](#)
 - GetProductGuid, [320](#)
 - GetSchemaVersion, [320](#)
 - GetStandardNameSpace, [320](#)
 - GetToolTip, [320](#)
 - GetVendorName, [320](#)
 - GetVersionGuid, [320](#)
 - IDeviceInfo, [321](#)
- IDeviceRemovalEventHandler, [917](#)
 - Spinnaker::IDeviceRemovalEventHandler, [918](#)
- IEnumEntry
 - IEnumEntry Interface, [323](#)
- IEnumEntry Interface, [322](#)
 - GetNumericValue, [322](#)
 - GetSymbolic, [322](#)
 - IEnumEntry, [323](#)
 - IsSelfClearing, [322](#)
- IEnumReference
 - IEnumerationT Interface, [328](#)
- IEnumeration
 - IEnumeration Interface, [326](#)
- IEnumeration Interface, [324](#)
 - GetCurrentEntry, [324](#)
 - GetEntries, [324](#)
 - GetEntry, [325](#)
 - GetEntryByName, [325](#)
 - GetIntValue, [325](#)
 - IEnumeration, [326](#)
 - operator*, [325](#)
 - SetIntValue, [326](#)
- IEnumerationT Interface, [327](#)
 - GetEntry, [327](#)
- IEnumReference, [328](#)
- IEnumerationT, [328](#)
 - operator=, [327](#), [328](#)
- IEnumerationT
 - IEnumerationT Interface, [328](#)
- IFloat
 - IFloat Interface, [332](#)
- IFloat Interface, [329](#)
 - GetDisplayNotation, [330](#)
 - GetDisplayPrecision, [330](#)
 - GetInc, [330](#)
 - GetIncMode, [330](#)
 - GetListOfValidValues, [330](#)
 - GetMax, [330](#)
 - GetMin, [331](#)
 - GetRepresentation, [331](#)
 - GetUnit, [331](#)
 - HasInc, [331](#)
 - IFloat, [332](#)
 - ImposeMax, [331](#)
 - ImposeMin, [331](#)
 - operator=, [332](#)
- IID_ISpinnakerInterface
 - SpinnakerDirectShow.h, [235](#)
- Image, [919](#)
 - Spinnaker::Image, [921](#)
- Image Class, [257](#)
- ImageEventHandler, [931](#)
 - Spinnaker::ImageEventHandler, [932](#)
- ImageStatistics, [933](#)
 - Spinnaker::ImageStatistics, [934](#)
- ImageStatistics Class, [258](#)
- Integer
 - Integer Interface, [334](#)
- Integer Interface, [333](#)
 - Integer, [334](#)
 - ImposeMax, [333](#)
 - ImposeMin, [333](#)
 - operator=, [333](#)
- Interface, [937](#)
 - Spinnaker::Interface, [939](#)
 - Spinnaker::TransportLayerInterface, [1178](#)
- Interface Class, [259](#)
- InterfaceArrivalEventHandler, [942](#)
 - Spinnaker::InterfaceArrivalEventHandler, [943](#)
- InterfaceEventHandler, [944](#)
 - Spinnaker::InterfaceEventHandler, [945](#)
- InterfaceList, [947](#)
 - Spinnaker::InterfaceList, [948](#)
- InterfaceList Class, [260](#)
- InterfaceRemovalEventHandler, [950](#)
 - Spinnaker::InterfaceRemovalEventHandler, [951](#)
- ILoggingEventHandler, [952](#)
 - Spinnaker::ILoggingEventHandler, [953](#)
- INTEGRAL_CAST2
 - GCUtilities Utility, [308](#)
- INTEGRAL_CAST
 - GCUtilities Utility, [308](#)

- INode
 - INode Interface, [345](#)
- INode Interface, [335](#)
 - Combine, [337](#)
 - DeregisterCallback, [337](#)
 - GetAlias, [338](#)
 - GetCachingMode, [338](#)
 - GetCastAlias, [338](#)
 - GetChildren, [338](#)
 - GetDescription, [339](#)
 - GetDisplayName, [339](#)
 - GetDocuURL, [339](#)
 - GetEventID, [339](#)
 - GetNameSpace, [339](#)
 - GetNodeMap, [339](#)
 - GetParents, [339](#)
 - GetPollingTime, [340](#)
 - GetPrincipalInterfaceType, [340](#)
 - GetProperty, [340](#)
 - GetPropertyNames, [340](#)
 - GetVisibility, [340](#)
 - INode, [345](#)
 - IRReference, [345](#)
 - ImposeAccessMode, [340](#)
 - ImposeVisibility, [341](#)
 - InvalidateNode, [341](#)
 - IsAccessModeCacheable, [341](#)
 - IsAvailable, [341](#)
 - IsCacheable, [342](#)
 - IsCacheable, [342](#)
 - IsDeprecated, [342](#)
 - IsFeature, [342](#)
 - IsImplemented, [342](#), [343](#)
 - IsReadable, [343](#)
 - IsStreamable, [343](#)
 - IsVisible, [343](#)
 - IsWritable, [344](#)
 - operator!=, [344](#)
 - operator==, [344](#)
 - RegisterCallback, [344](#)
- INodeMap
 - INodeMap Interface, [348](#)
- INodeMap Interface, [346](#)
 - Connect, [346](#), [347](#)
 - GetDeviceName, [347](#)
 - GetLock, [347](#)
 - GetNode, [347](#)
 - GetNumNodes, [347](#)
 - INodeMap, [348](#)
 - InvalidateNodes, [348](#)
 - Poll, [348](#)
- INodeMapDyn
 - INodeMapDyn Interface, [353](#)
- INodeMapDyn Interface, [349](#)
 - ExtractIndependentSubtree, [350](#)
 - GetSupportedSchemaVersions, [350](#)
 - INodeMapDyn, [353](#)
 - LoadXMLFromFile, [350](#)
 - LoadXMLFromFileInject, [350](#)
 - LoadXMLFromString, [351](#)
 - LoadXMLFromStringInject, [351](#)
 - LoadXMLFromZIPData, [351](#)
 - LoadXMLFromZIPFile, [351](#)
 - MergeXMLFiles, [351](#)
 - PreprocessXMLFromFile, [352](#)
 - PreprocessXMLFromZIPFile, [352](#)
- IPersistScript
 - Spinnaker::GenApi, [477](#)
- IPort
 - IPort Interface, [357](#)
- IPort Interface, [356](#)
 - Address, [356](#)
 - IPort, [357](#)
 - Length, [357](#)
 - Write, [356](#)
- IPortConstruct
 - IPortConstruct Interface, [358](#)
- IPortConstruct Interface, [358](#)
 - GetSwapEndianness, [358](#)
 - IPortConstruct, [358](#)
- IPortRecorder
 - IPortRecorder Interface, [360](#)
- IPortRecorder Interface, [359](#)
 - GetCookie, [359](#)
 - IPortRecorder, [360](#)
 - IPortReplay, [360](#)
 - IPortWriteList, [360](#)
 - Invalidate, [360](#)
 - Replay, [359](#)
 - SetCookie, [360](#)
 - StopRecording, [360](#)
- IPortReplay
 - IPortRecorder Interface, [360](#)
- IPortWriteList
 - IPortRecorder Interface, [360](#)
- IRReference
 - INode Interface, [345](#)
- IRRegister
 - IRRegister Interfaces, [362](#)
- IRRegister Interfaces, [361](#)
 - Get, [361](#)
 - GetAddress, [362](#)
 - GetLength, [362](#)
 - IRRegister, [362](#)
- ISelector
 - ISelector Interface, [363](#)
- ISelector Interface, [363](#)
 - GetSelectedFeatures, [363](#)
 - GetSelectingFeatures, [363](#)
 - ISelector, [363](#)
- ISelectorDigit
 - ISelectorDigit Interface, [365](#)
- ISelectorDigit Interface, [364](#)
 - GetSelectorList, [364](#)
 - ISelectorDigit, [365](#)
 - Restore, [365](#)

- SetNext, [365](#)
 - ToString, [365](#)
- IString
 - IString Class, [367](#)
- IString Class, [367](#)
 - GetMaxLength, [367](#)
 - IString, [367](#)
- ISystem, [1058](#)
 - Spinnaker::ISystem, [1059](#)
 - Spinnaker::TransportLayerSystem, [1196](#)
- ISystem Class, [261](#)
- ISystemEventHandler, [1064](#)
 - Spinnaker::ISystemEventHandler, [1065](#)
- IValue
 - IValue Class, [369](#)
- IValue Class, [368](#)
 - FromString, [368](#)
 - IValue, [369](#)
 - IsValueCacheValid, [369](#)
 - ToString, [369](#)
- Image, [954](#)
 - Spinnaker::Image, [958](#)
- Image Class, [185](#)
- Image Utility CCM Class, [190](#)
- Image Utility Class, [189](#)
- Image Utility Heatmap Class, [191](#)
- Image Utility Polarization Class, [192](#)
- ImageComponentEnable
 - Spinnaker::Camera, [619](#)
- ImageComponentSelector
 - Spinnaker::Camera, [619](#)
- ImageComponentSelectorEnums
 - CameraDefs Class, [128](#)
- ImageCompressionBitrate
 - Spinnaker::Camera, [619](#)
- ImageCompressionJPEGFormatOption
 - Spinnaker::Camera, [619](#)
- ImageCompressionJPEGFormatOptionEnums
 - CameraDefs Class, [129](#)
- ImageCompressionMode
 - Spinnaker::Camera, [620](#)
- ImageCompressionModeEnums
 - CameraDefs Class, [129](#)
- ImageCompressionQuality
 - Spinnaker::Camera, [620](#)
- ImageCompressionRateOption
 - Spinnaker::Camera, [620](#)
- ImageCompressionRateOptionEnums
 - CameraDefs Class, [130](#)
- ImageConverter
 - Spinnaker::Image, [981](#)
- ImageConverterIpp
 - Spinnaker::Image, [981](#)
- ImageEventHandler, [983](#)
 - Spinnaker::ImageEventHandler, [984](#)
- imageEventHandler
 - GrabInfo, [867](#)
- ImageEventHandler Class, [186](#)
- ImageEventHandlerImpl, [985](#)
 - ~ImageEventHandlerImpl, [986](#), [987](#)
 - getImageCount, [987](#)
 - getMaxImages, [987](#)
 - ImageEventHandlerImpl, [986](#), [987](#)
 - OnImageEvent, [987](#), [988](#)
- ImageEvents.cpp
 - AcquireImages, [1466](#)
 - ConfigureImageEvents, [1466](#)
 - main, [1466](#)
 - PrintDeviceInfo, [1467](#)
 - ResetImageEvents, [1467](#)
 - RunSingleCamera, [1467](#)
 - SleepyWrapper, [1467](#)
 - WaitForImages, [1467](#)
- imageFile
 - ImageInfo, [988](#)
- ImageFileFormat
 - Spinnaker Definitions, [227](#)
- imageFileName
 - ImageInfo, [989](#)
- ImageFiler
 - Spinnaker::Image, [982](#)
- ImageFormatControl.cpp
 - AcquireImages, [1468](#)
 - ConfigureCustomImageSettings, [1468](#)
 - main, [1468](#)
 - PrintDeviceInfo, [1468](#)
 - RunSingleCamera, [1469](#)
- ImageFormatControl_QuickSpin.cpp
 - AcquireImages, [1469](#)
 - ConfigureCustomImageSettings, [1469](#)
 - main, [1470](#)
 - PrintDeviceInfo, [1470](#)
 - RunSingleCamera, [1470](#)
- imageHeight
 - CompressedImageInfo, [770](#)
 - ImageInfo, [989](#)
- ImageInfo, [988](#)
 - imageFile, [988](#)
 - imageFileName, [989](#)
 - imageHeight, [989](#)
 - ImageInfo, [988](#)
 - imageWidth, [989](#)
 - pixelFormat, [989](#)
- imageInfos
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1433](#)
- imagePixelFormat
 - CompressedImageInfo, [770](#)
- ImagePtr, [990](#)
 - Spinnaker::ImagePtr, [991](#)
- ImagePtr Class, [187](#)
- ImageScalingAlgorithm
 - Spinnaker::ImageUtility, [1000](#)
- ImageStatistics, [992](#)
 - Spinnaker::ImageStatistics, [994](#)
- ImageStatistics Class, [188](#)
- ImageStatsCalculator

- Spinnaker::Image, 982
- Spinnaker::ImageStatistics, 999
- ImageStatus
 - Spinnaker Definitions, 228
- ImageUtility, 999
- ImageUtilityCCM, 1004
- ImageUtilityHeatmap, 1006
- ImageUtilityImpl
 - Spinnaker::Image, 982
- ImageUtilityPolarization, 1010
- ImageUtilityPolarizationImpl
 - Spinnaker::Image, 982
- imageWidth
 - CompressedImageInfo, 770
 - ImageInfo, 989
- imageXOffset
 - CompressedImageInfo, 770
- imageYOffset
 - CompressedImageInfo, 771
- ImposeAccessMode
 - INode Interface, 340
 - Spinnaker::GenApi::Node, 1093
- ImposeMax
 - IFloat Interface, 331
 - Integer Interface, 333
 - Spinnaker::GenApi::FloatNode, 851
 - Spinnaker::GenApi::IntegerNode, 1029
- ImposeMin
 - IFloat Interface, 331
 - Integer Interface, 333
 - Spinnaker::GenApi::FloatNode, 851
 - Spinnaker::GenApi::IntegerNode, 1029
- ImposeVisibility
 - INode Interface, 341
 - Spinnaker::GenApi::Node, 1093
- include/AVIRecorder.h, 1213
- include/AdapterConfig.h, 1211
- include/BasePtr.h, 1214
- include/Camera.h, 1215
- include/CameraBase.h, 1215
- include/CameraDefs.h, 1216
- include/CameraList.h, 1248
- include/CameraPtr.h, 1249
- include/ChunkData.h, 1249
- include/ChunkDataInference.h, 1250
- include/DeviceArrivalEventHandler.h, 1251
- include/DeviceEventHandler.h, 1252
- include/DeviceEventUtility.h, 1253
- include/DeviceRemovalEventHandler.h, 1254
- include/EventHandler.h, 1255
- include/Exception.h, 1255
- include/Image.h, 1256
- include/ImageEventHandler.h, 1257
- include/ImagePtr.h, 1257
- include/ImageStatistics.h, 1258
- include/ImageUtility.h, 1259
- include/ImageUtilityCCM.h, 1260
- include/ImageUtilityHeatmap.h, 1260
- include/ImageUtilityPolarization.h, 1261
- include/Interface.h, 1262
- include/Interface/ICameraBase.h, 1262
- include/Interface/ICameraList.h, 1263
- include/Interface/IChunkData.h, 1264
- include/Interface/IDeviceArrivalEventHandler.h, 1265
- include/Interface/IDeviceEventHandler.h, 1266
- include/Interface/IDeviceRemovalEventHandler.h, 1266
- include/Interface/IImage.h, 1267
- include/Interface/IImageEventHandler.h, 1268
- include/Interface/IImageStatistics.h, 1268
- include/Interface/IInterface.h, 1269
- include/Interface/IInterfaceArrivalEventHandler.h, 1270
- include/Interface/IInterfaceEventHandler.h, 1271
- include/Interface/IInterfaceList.h, 1271
- include/Interface/IInterfaceRemovalEventHandler.h, 1272
- include/Interface/ILoggingEventHandler.h, 1273
- include/Interface/IStream.h, 1273
- include/Interface/ISystem.h, 1274
- include/Interface/ISystemEventHandler.h, 1275
- include/InterfaceArrivalEventHandler.h, 1276
- include/InterfaceEventHandler.h, 1277
- include/InterfaceList.h, 1278
- include/InterfacePtr.h, 1278
- include/InterfaceRemovalEventHandler.h, 1279
- include/LoggingEventData.h, 1280
- include/LoggingEventDataPtr.h, 1280
- include/LoggingEventHandler.h, 1281
- include/SpinGenApi/Autovector.h, 1282
- include/SpinGenApi/Base.h, 1283
- include/SpinGenApi/BooleanNode.h, 1284
- include/SpinGenApi/CategoryNode.h, 1285
- include/SpinGenApi/ChunkAdapter.h, 1286
- include/SpinGenApi/ChunkAdapterDcam.h, 1287
- include/SpinGenApi/ChunkAdapterGEV.h, 1289
- include/SpinGenApi/ChunkAdapterGeneric.h, 1288
- include/SpinGenApi/ChunkAdapterU3V.h, 1290
- include/SpinGenApi/ChunkPort.h, 1291
- include/SpinGenApi/CommandNode.h, 1291
- include/SpinGenApi/Compatibility.h, 1292
- include/SpinGenApi/Container.h, 1293
- include/SpinGenApi/Counter.h, 1293
- include/SpinGenApi/EnumClasses.h, 1294
- include/SpinGenApi/EnumEntryNode.h, 1295
- include/SpinGenApi/EnumNode.h, 1296
- include/SpinGenApi/EnumNodeT.h, 1297
- include/SpinGenApi/EventAdapter.h, 1297
- include/SpinGenApi/EventAdapter1394.h, 1298
- include/SpinGenApi/EventAdapterGEV.h, 1299
- include/SpinGenApi/EventAdapterGeneric.h, 1299
- include/SpinGenApi/EventAdapterU3V.h, 1300
- include/SpinGenApi/EventPort.h, 1301
- include/SpinGenApi/Filestream.h, 1302
- include/SpinGenApi/FloatNode.h, 1303
- include/SpinGenApi/FloatRegNode.h, 1304
- include/SpinGenApi/GCBase.h, 1305
- include/SpinGenApi/GCString.h, 1305

- include/SpinGenApi/GCStringVector.h, [1307](#)
- include/SpinGenApi/GCSynch.h, [1308](#)
- include/SpinGenApi/GCTypes.h, [1309](#)
- include/SpinGenApi/GCUtilities.h, [1312](#)
- include/SpinGenApi/IBoolean.h, [1316](#)
- include/SpinGenApi/ICategory.h, [1317](#)
- include/SpinGenApi/IChunkPort.h, [1318](#)
- include/SpinGenApi/ICommand.h, [1319](#)
- include/SpinGenApi/IDestroy.h, [1320](#)
- include/SpinGenApi/IDeviceInfo.h, [1321](#)
- include/SpinGenApi/IEnumEntry.h, [1322](#)
- include/SpinGenApi/IEnumeration.h, [1323](#)
- include/SpinGenApi/IEnumerationT.h, [1324](#)
- include/SpinGenApi/IFloat.h, [1325](#)
- include/SpinGenApi/IInteger.h, [1327](#)
- include/SpinGenApi/INode.h, [1328](#)
- include/SpinGenApi/INodeMap.h, [1331](#)
- include/SpinGenApi/INodeMapDyn.h, [1333](#)
- include/SpinGenApi/IPort.h, [1336](#)
- include/SpinGenApi/IPortConstruct.h, [1337](#)
- include/SpinGenApi/IPortRecorder.h, [1338](#)
- include/SpinGenApi/IRegister.h, [1339](#)
- include/SpinGenApi/ISelector.h, [1340](#)
- include/SpinGenApi/ISelectorDigit.h, [1341](#)
- include/SpinGenApi/IString.h, [1342](#)
- include/SpinGenApi/IValue.h, [1343](#)
- include/SpinGenApi/IntRegNode.h, [1335](#)
- include/SpinGenApi/IntegerNode.h, [1334](#)
- include/SpinGenApi/Node.h, [1344](#)
- include/SpinGenApi/NodeCallback.h, [1345](#)
- include/SpinGenApi/NodeCallbackImpl.h, [1347](#)
- include/SpinGenApi/NodeMap.h, [1347](#)
- include/SpinGenApi/NodeMapFactory.h, [1348](#)
- include/SpinGenApi/NodeMapRef.h, [1349](#)
- include/SpinGenApi/Persistence.h, [1350](#)
- include/SpinGenApi/Pointer.h, [1351](#)
- include/SpinGenApi/PortImpl.h, [1353](#)
- include/SpinGenApi/PortNode.h, [1354](#)
- include/SpinGenApi/PortRecorder.h, [1355](#)
- include/SpinGenApi/PortReplay.h, [1356](#)
- include/SpinGenApi/PortWriteList.h, [1357](#)
- include/SpinGenApi/Reference.h, [1358](#)
- include/SpinGenApi/RegisterNode.h, [1359](#)
- include/SpinGenApi/RegisterPortImpl.h, [1359](#)
- include/SpinGenApi/SelectorSet.h, [1360](#)
- include/SpinGenApi/SpinTestCamera.h, [1361](#)
- include/SpinGenApi/SpinnakerGenApi.h, [1360](#)
- include/SpinGenApi/StringNode.h, [1361](#)
- include/SpinGenApi/StringRegNode.h, [1362](#)
- include/SpinGenApi/StructPort.h, [1363](#)
- include/SpinGenApi/Synch.h, [1363](#)
- include/SpinGenApi/Types.h, [1364](#)
- include/SpinGenApi/ValueNode.h, [1367](#)
- include/SpinUpdate.h, [1376](#)
- include/SpinVideo.h, [1378](#)
- include/SpinVideoDefs.h, [1379](#)
- include/Spinnaker.h, [1368](#)
- include/SpinnakerDefs.h, [1368](#)
- include/SpinnakerDirectShow.h, [1373](#)
- include/SpinnakerPlatform.h, [1375](#)
- include/System.h, [1380](#)
- include/SystemEventHandler.h, [1382](#)
- include/SystemPtr.h, [1382](#)
- include/TransportLayerDefs.h, [1383](#)
- include/TransportLayerDevice.h, [1385](#)
- include/TransportLayerInterface.h, [1385](#)
- include/TransportLayerStream.h, [1386](#)
- include/TransportLayerSystem.h, [1387](#)
- IncompatibleDeviceCount
 - Spinnaker::TransportLayerInterface, [1184](#)
- IncompatibleDeviceID
 - Spinnaker::TransportLayerInterface, [1185](#)
- IncompatibleDeviceModelName
 - Spinnaker::TransportLayerInterface, [1185](#)
- IncompatibleDeviceSelector
 - Spinnaker::TransportLayerInterface, [1185](#)
- IncompatibleDeviceVendorName
 - Spinnaker::TransportLayerInterface, [1185](#)
- IncompatibleGevDeviceIPAddress
 - Spinnaker::TransportLayerInterface, [1185](#)
- IncompatibleGevDeviceMACAddress
 - Spinnaker::TransportLayerInterface, [1185](#)
- IncompatibleGevDeviceSubnetMask
 - Spinnaker::TransportLayerInterface, [1186](#)
- Increasing
 - Types Enums, [408](#)
- Indent
 - NodeMapInfo.cpp, [1484](#)
- indexedColor_8bit
 - Spinnaker::BMPOption, [495](#)
- Inference.cpp
 - AcquireImages, [1472](#)
 - arrayLabelClassification, [1475](#)
 - arrayLabelDetection, [1475](#)
 - CameraCloseFile, [1472](#)
 - CameraDeleteFile, [1472](#)
 - CameraOpenFile, [1472](#)
 - CameraWriteToFile, [1472](#)
 - chosenFileUploadPersistence, [1476](#)
 - chosenInferenceNetworkType, [1476](#)
 - ConfigureChunkData, [1473](#)
 - ConfigureInference, [1473](#)
 - ConfigureTestPattern, [1473](#)
 - ConfigureTrigger, [1473](#)
 - DeleteFileOnCamera, [1473](#)
 - DisableChunkData, [1473](#)
 - DisableTrigger, [1474](#)
 - DisplayChunkData, [1474](#)
 - FileUploadPersistence, [1471](#)
 - InferenceNetworkType, [1471](#)
 - injectedImageFilePath, [1476](#)
 - injectedImageHeight, [1476](#)
 - injectedImageWidth, [1476](#)
 - labelClassification, [1474](#)
 - labelDetection, [1474](#)
 - LoadFileIntoMemory, [1474](#)

- main, [1474](#)
- networkFilePath, [1477](#)
- PrintDeviceInfo, [1475](#)
- RunSingleCamera, [1475](#)
- SetChunkEnable, [1475](#)
- UploadFileToCamera, [1475](#)
- InferenceAvailable
 - DeviceEvents.cpp, [1447](#)
- InferenceBoundingBox, [1020](#)
- InferenceBoundingBoxResult, [1020](#)
 - Chunk Data Inference Class, [174](#)
- InferenceBoxCircle, [1021](#)
- InferenceBoxRect, [1021](#)
- InferenceBoxRotatedRect, [1022](#)
- InferenceBoxType
 - Spinnaker Classes, [45](#)
- InferenceNetworkType
 - Inference.cpp, [1471](#)
- Init
 - Spinnaker::Camera, [529](#)
 - Spinnaker::CameraBase, [661](#)
 - Spinnaker::ICameraBase, [885](#)
- InitChunkAdapter
 - Spinnaker::IDataStream, [906](#)
- InitializeSystem
 - FileAccess_QuickSpin.cpp, [1457](#)
- injectedImageFilePath
 - Inference.cpp, [1476](#)
- injectedImageHeight
 - Inference.cpp, [1476](#)
- injectedImageWidth
 - Inference.cpp, [1476](#)
- int64_autovector_t, [1022](#)
 - Spinnaker::GenApi::int64_autovector_t, [1023](#)
- IntRegNode, [1054](#)
 - Spinnaker::GenApi::IntRegNode, [1055](#), [1056](#)
- IntRegNode Class, [355](#)
- IntegerGetInc
 - Spinnaker Headers, [209](#)
- IntegerGetIncMode
 - Spinnaker Headers, [210](#)
- IntegerGetMax
 - Spinnaker Headers, [210](#)
- IntegerGetMin
 - Spinnaker Headers, [210](#)
- IntegerGetValue
 - Spinnaker Headers, [211](#)
- IntegerNode, [1025](#)
 - Spinnaker::GenApi::IntegerNode, [1027](#)
- IntegerNode Class, [354](#)
 - CIntegerRef, [354](#)
- IntegerSetValue
 - Spinnaker Headers, [211](#)
- Interface, [1031](#)
 - Spinnaker::TransportLayerInterface, [1178](#)
- interface
 - Types.h, [1367](#)
- Interface Class, [193](#)
- InterfaceArrivalEventHandler, [1036](#)
 - Spinnaker::InterfaceArrivalEventHandler, [1037](#)
- InterfaceArrivalEventHandler Class, [194](#)
- InterfaceDisplayName
 - Spinnaker::TransportLayerInterface, [1186](#)
 - Spinnaker::TransportLayerSystem, [1199](#)
- InterfaceEventHandler, [1039](#)
 - Spinnaker::InterfaceEventHandler, [1040](#)
- InterfaceEventHandler Class, [195](#)
- InterfaceEventHandlerImpl, [1042](#)
 - ~InterfaceEventHandlerImpl, [1044](#)
 - GetInterfaceId, [1044](#)
 - InterfaceEventHandlerImpl, [1043](#), [1044](#)
 - OnDeviceArrival, [1044](#), [1045](#)
 - OnDeviceRemoval, [1045](#)
 - PrintGenericHandlerMessage, [1046](#)
- InterfaceId
 - Spinnaker::TransportLayerInterface, [1186](#)
 - Spinnaker::TransportLayerSystem, [1199](#)
- InterfaceImpl
 - Spinnaker::CameraBase, [666](#)
 - Spinnaker::ICameraBase, [888](#)
 - Spinnaker::ICameraList, [892](#)
- InterfaceInternal
 - Spinnaker::IInterface, [941](#)
 - Spinnaker::Interface, [1036](#)
 - Spinnaker::TransportLayerInterface, [1178](#)
- InterfaceList, [1046](#)
 - Spinnaker::InterfaceList, [1047](#)
- InterfaceList Class, [196](#)
- InterfaceListImpl
 - Spinnaker::IInterfaceList, [949](#)
- InterfacePtr, [1050](#)
 - Spinnaker::InterfacePtr, [1051](#)
- InterfacePtr Class, [197](#)
- InterfaceRemovalEventHandler, [1052](#)
 - Spinnaker::InterfaceRemovalEventHandler, [1053](#)
- InterfaceRemovalEventHandler Class, [198](#)
- InterfaceSelector
 - Spinnaker::TransportLayerSystem, [1199](#)
- InterfaceType
 - Spinnaker::TransportLayerInterface, [1186](#)
- InterfaceTypeEnum
 - TransportLayerDefs Class, [247](#)
- InterfaceUpdateList
 - Spinnaker::TransportLayerSystem, [1200](#)
- interlaced
 - Spinnaker::PNGOption, [1112](#)
- Invalidate
 - IPortRecorder Interface, [360](#)
- InvalidateNode
 - INode Interface, [341](#)
 - Spinnaker::GenApi::CChunkPort, [693](#)
 - Spinnaker::GenApi::CEventPort, [716](#)
 - Spinnaker::GenApi::CPortImpl, [780](#)
 - Spinnaker::GenApi::Node, [1093](#)
- InvalidateNodes
 - INodeMap Interface, [348](#)

- Spinnaker::GenApi::NodeMap, [1103](#)
- InverseChunkLength
 - DCAM_CHUNK_TRAILER, [796](#)
- Invisible
 - Types Enums, [409](#)
- ios_type
 - Spinnaker::GenApi::IDevFileStreamBase, [909](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [1107](#)
- ipAddress
 - AdapterConfig::IpInfo, [1057](#)
- IpInfo, [1056](#)
 - AdapterConfig::IpInfo, [1057](#)
- ipInfo
 - AdapterConfig::AdapterInfo, [485](#)
- is_open
 - Spinnaker::GenApi::IDevFileStreamBase, [910](#)
 - Spinnaker::GenApi::IDevFileStreamBuf, [912](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [1107](#)
 - Spinnaker::GenApi::ODevFileStreamBuf, [1109](#)
- IsAccessModeCacheable
 - INode Interface, [341](#)
 - Spinnaker::GenApi::Node, [1093](#)
- IsAvailable
 - INode Interface, [341](#)
 - Pointer Class, [384](#)
- IsCRCCheckEnabled
 - Spinnaker::IDataStream, [906](#)
- IsCacheable
 - INode Interface, [342](#)
 - Spinnaker::GenApi::Node, [1093](#)
- IsCacheable
 - INode Interface, [342](#)
- IsCameraDescriptionFileDataReleased
 - Spinnaker::GenApi::CNodeMapFactory, [756](#)
- IsCompressed
 - Spinnaker::Image, [927](#)
 - Spinnaker::Image, [975](#)
- IsDeprecated
 - INode Interface, [342](#)
 - Spinnaker::GenApi::Node, [1094](#)
- IsDone
 - ICommand Interface, [316](#)
 - Spinnaker::GenApi::CommandNode, [768](#)
- IsEmpty
 - Spinnaker::GenApi::CNodeMapFactory, [756](#)
 - Spinnaker::GenApi::CSelectorSet, [790](#)
- IsFeature
 - INode Interface, [342](#)
 - Spinnaker::GenApi::Node, [1094](#)
- IsImageInUse
 - Spinnaker::IDataStream, [906](#)
- IsImplemented
 - INode Interface, [342](#), [343](#)
 - Pointer Class, [384](#)
- IsInUse
 - Spinnaker::Image, [928](#)
 - Spinnaker::IInterface, [940](#)
 - Spinnaker::ISystem, [1061](#)
- Spinnaker::Image, [975](#)
- Spinnaker::Interface, [1033](#)
- Spinnaker::System, [1148](#)
- IsIncomplete
 - Spinnaker::Image, [928](#)
 - Spinnaker::Image, [975](#)
- IsInitialized
 - Spinnaker::CameraBase, [662](#)
 - Spinnaker::ICameraBase, [885](#)
- IsLoaded
 - Spinnaker::GenApi::CNodeMapFactory, [756](#)
- IsOnSameSubnet
 - AdapterConfig, [416](#)
- isPixelFormatColor
 - Polarization.cpp, [1490](#)
- IsPreprocessed
 - Spinnaker::GenApi::CNodeMapFactory, [757](#)
- IsReadable
 - INode Interface, [343](#)
 - Pointer Class, [384](#)
- IsRelease
 - GigEVisionPerformance.cpp, [1464](#)
- IsSelector
 - Spinnaker::GenApi::Node, [1094](#)
- IsSelfClearing
 - IEnumEntry Interface, [322](#)
 - Spinnaker::GenApi::EnumEntryNode, [821](#)
- IsStreamable
 - INode Interface, [343](#)
 - Spinnaker::GenApi::Node, [1094](#)
- IsStreaming
 - Spinnaker::CameraBase, [662](#)
 - Spinnaker::ICameraBase, [886](#)
 - Spinnaker::IDataStream, [906](#)
 - SpinnakerDirectShow.h, [234](#)
- IsValid
 - Spinnaker::BasePtr, [492](#)
 - Spinnaker::CameraBase, [662](#)
 - Spinnaker::GenApi::CPointer, [775](#)
 - Spinnaker::GenICam::CGlobalLock, [724](#)
 - Spinnaker::ICameraBase, [886](#)
 - Spinnaker::IInterface, [940](#)
 - Spinnaker::Interface, [1033](#)
- IsValidIpAddress
 - AdapterConfig, [416](#)
- IsValidSubnetMask
 - AdapterConfig, [416](#)
- IsValueCacheValid
 - IValue Class, [369](#)
 - Spinnaker::GenApi::ValueNode, [1207](#)
- IsVisible
 - INode Interface, [343](#)
- IsWritable
 - INode Interface, [344](#)
 - Pointer Class, [384](#)
- IsZero
 - Spinnaker::GenApi::Counter, [772](#)
- IspEnable

- Spinnaker::Camera, [620](#)
- istream_type
 - Spinnaker::GenApi::IDevFileStreamBase, [909](#)
- Items
 - GVCP_EVENT_REQUEST_EXTENDED_ID, [874](#)
 - GVCP_EVENT_REQUEST, [873](#)
- JPEGOption, [1066](#)
 - Spinnaker::JPEGOption, [1066](#)
- JPG2Option, [1067](#)
 - Spinnaker::JPG2Option, [1068](#)
- jumboPacketValidValues
 - AdapterConfig::AdapterInfo, [485](#)
- jumboPackets
 - AdapterConfig::AdapterInfo, [485](#)
- jumboPacketsRegKey
 - AdapterConfig::AdapterInfo, [485](#)
- k_LoggingLevel
 - Logging.cpp, [1478](#)
- k_numImages
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1434](#)
- k_numLoops
 - BufferHandling.cpp, [1438](#)
- kDestinationDirectory
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1434](#)
- KillBufferEvent
 - Spinnaker::IDataStream, [906](#)
- LUTEnable
 - Spinnaker::Camera, [624](#)
- LUTIndex
 - Spinnaker::Camera, [624](#)
- LUTSelector
 - Spinnaker::Camera, [624](#)
- LUTSelectorEnums
 - CameraDefs Class, [134](#)
- LUTValue
 - Spinnaker::Camera, [624](#)
- LUTValueAll
 - Spinnaker::Camera, [625](#)
- labelClassification
 - Inference.cpp, [1474](#)
- labelDetection
 - Inference.cpp, [1474](#)
- Length
 - GVCP_REQUEST_HEADER, [877](#)
 - IPort Interface, [357](#)
 - U3V_COMMAND_HEADER, [1203](#)
- length
 - Spinnaker::GenICam::gcstring, [863](#)
- LibraryVersion, [1069](#)
- LineFilterWidth
 - Spinnaker::Camera, [620](#)
- LineFormat
 - Spinnaker::Camera, [621](#)
- LineFormatEnums
 - CameraDefs Class, [130](#)
- LineInputFilterSelector
 - Spinnaker::Camera, [621](#)
- LineInputFilterSelectorEnums
 - CameraDefs Class, [130](#)
- LineInverter
 - Spinnaker::Camera, [621](#)
- LineMode
 - Spinnaker::Camera, [621](#)
- LineModeEnums
 - CameraDefs Class, [131](#)
- LinePitch
 - Spinnaker::Camera, [621](#)
- LineSelector
 - Spinnaker::Camera, [621](#)
- LineSelectorEnums
 - CameraDefs Class, [131](#)
- LineSource
 - Spinnaker::Camera, [622](#)
- LineSourceEnums
 - CameraDefs Class, [131](#)
- LineStatus
 - Spinnaker::Camera, [622](#)
- LineStatusAll
 - Spinnaker::Camera, [622](#)
- Linear
 - Types Enums, [407](#)
- LinkErrorCount
 - Spinnaker::Camera, [622](#)
- LinkUptime
 - Spinnaker::Camera, [622](#)
- LoadAndInject
 - Spinnaker::GenApi::CNodeMapFactory, [757](#)
- LoadFileIntoMemory
 - Inference.cpp, [1474](#)
- LoadFromBag
 - Spinnaker::GenApi::CFeatureBag, [718](#)
- LoadXMLFromFile
 - INodeMapDyn Interface, [350](#)
 - Spinnaker::GenApi::NodeMap, [1103](#)
- LoadXMLFromFileInject
 - INodeMapDyn Interface, [350](#)
 - Spinnaker::GenApi::NodeMap, [1103](#)
- LoadXMLFromString
 - INodeMapDyn Interface, [351](#)
 - Spinnaker::GenApi::NodeMap, [1103](#)
- LoadXMLFromStringInject
 - INodeMapDyn Interface, [351](#)
 - Spinnaker::GenApi::NodeMap, [1103](#)
- LoadXMLFromZIPData
 - INodeMapDyn Interface, [351](#)
 - Spinnaker::GenApi::NodeMap, [1104](#)
- LoadXMLFromZIPFile
 - INodeMapDyn Interface, [351](#)
 - Spinnaker::GenApi::NodeMap, [1104](#)
- Lock
 - Spinnaker::GenApi::CLock, [743](#)
 - Spinnaker::GenICam::CGlobalLock, [724](#)
 - Spinnaker::GenICam::CLock, [741](#)
 - Spinnaker::GenICam::LockableObject, [1072](#)

- Spinnaker::GenICam::LockableObject::Lock, 1070
- LockEventHandlerMutex
 - SystemEventHandlerImpl, 1159
- LockableObject< Object >, 1071
- LockableObject< Object >::Lock, 1070
- Logarithmic
 - Types Enums, 407
- Logging EventHandler Class, 199
- Logging.cpp
 - k_LoggingLevel, 1478
 - main, 1477
- LoggingEventData, 1072
 - Spinnaker::LoggingEventData, 1073
- LoggingEventDataPtr, 1076
 - Spinnaker::LoggingEventDataPtr, 1077
- LoggingEventDataPtr Class, 200
- LoggingEventHandler, 1078
 - Spinnaker::LoggingEventHandler, 1079
- LoggingEventHandler Class, 201
- LoggingEventHandlerImpl, 1080
- LogicBlock.cpp
 - AcquireImages, 1478
 - ConfigureLogicBlock, 1478
 - ConfigureTrigger, 1479
 - GrabTwoImages, 1479
 - main, 1479
 - PrintDeviceInfo, 1479
 - ResetExposure, 1479
 - ResetTrigger, 1479
 - RunSingleCamera, 1479
- LogicBlockLUTInputActivation
 - Spinnaker::Camera, 622
- LogicBlockLUTInputActivationEnums
 - CameraDefs Class, 132
- LogicBlockLUTInputSelector
 - Spinnaker::Camera, 623
- LogicBlockLUTInputSelectorEnums
 - CameraDefs Class, 132
- LogicBlockLUTInputSource
 - Spinnaker::Camera, 623
- LogicBlockLUTInputSourceEnums
 - CameraDefs Class, 133
- LogicBlockLUTOutputValue
 - Spinnaker::Camera, 623
- LogicBlockLUTOutputValueAll
 - Spinnaker::Camera, 623
- LogicBlockLUTRowIndex
 - Spinnaker::Camera, 623
- LogicBlockLUTSelector
 - Spinnaker::Camera, 623
- LogicBlockLUTSelectorEnums
 - CameraDefs Class, 133
- LogicBlockSelector
 - Spinnaker::Camera, 624
- LogicBlockSelectorEnums
 - CameraDefs Class, 134
- LookupTable.cpp
 - AcquireImages, 1480
 - ConfigureLookupTables, 1480
 - main, 1480
 - PrintDeviceInfo, 1481
 - PrintRetrieveNodeFailure, 1481
 - ResetLookupTables, 1481
 - RunSingleCamera, 1481
- m_BaseAddress
 - Spinnaker::GenApi::CTestPortStruct, 794
- m_CallbackType
 - Spinnaker::GenApi::CNodeCallback, 749
- m_Callbacks
 - Spinnaker::GenApi::Node, 1096
- m_DebugCount
 - Spinnaker::GenICam::CGlobalLock, 725
- m_Lock
 - Spinnaker::GenICam::CGlobalLockUnlocker, 727
 - Spinnaker::GenICam::LockableObject, 1072
- m_NumReads
 - Spinnaker::GenApi::CTestPortStruct, 794
- m_NumWrites
 - Spinnaker::GenApi::CTestPortStruct, 795
- m_bOwnLock
 - Spinnaker::GenApi::CLock, 744
- m_enabled
 - Spinnaker::GenICam::CGlobalLockUnlocker, 727
- m_lock
 - Spinnaker::GenApi::CLock, 744
- m_lockEx
 - Spinnaker::GenApi::CLockEx, 747
- m_pCameraBaseData
 - Spinnaker::ICameraBase, 888
- m_pCameraListData
 - Spinnaker::ICameraList, 893
- m_pChunkAdapter
 - Spinnaker::GenApi::CChunkAdapter, 679
- m_pChunkPort
 - Spinnaker::GenApi::CChunkPort, 694
- m_pEnumeration
 - Spinnaker::GenApi::EnumNode, 827
- m_pEventAdapter
 - Spinnaker::GenApi::CEventAdapter, 703
- m_pEventData
 - Spinnaker::EventHandler, 835
- m_pEventPort
 - Spinnaker::GenApi::CEventPort, 716
- m_plInterfaceData
 - Spinnaker::IInterface, 941
- m_plInterfaceListData
 - Spinnaker::IInterfaceList, 949
- m_pNode
 - Spinnaker::GenApi::CEventPort, 717
 - Spinnaker::GenApi::CNodeCallback, 749
- m_pNodeData
 - Spinnaker::GenApi::Node, 1096
- m_pNodeMap
 - Spinnaker::GenApi::Node, 1096
- m_pPort
 - Spinnaker::GenApi::CChunkPort, 694

- m_pPortAdapter
 - Spinnaker::GenApi::CChunkPort, [694](#)
 - Spinnaker::GenApi::CEventPort, [717](#)
- m_pWriteList
 - Spinnaker::GenApi::CPortWriteList, [784](#)
- m_pT
 - Spinnaker::BasePtr, [494](#)
 - Spinnaker::GenApi::CPointer, [777](#)
- m_ptrPort
 - Spinnaker::GenApi::CPortImpl, [781](#)
- MAX_LENGTH
 - SpinnakerDirectShow.h, [1375](#)
- MILLISECOND
 - SerialRxTx.cpp, [1495](#)
- MJPGOption, [1083](#)
 - Spinnaker::Video::MJPGOption, [1084](#)
- Magic
 - GVCP_REQUEST_HEADER, [878](#)
- main
 - Acquisition.cpp, [1388](#)
 - AcquisitionMultipleCameraRecovery.cpp, [1430](#)
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1433](#)
 - AcquisitionMultipleThread.cpp, [1434](#)
 - ActionCommand.cpp, [1437](#)
 - BufferHandling.cpp, [1439](#)
 - ChunkData.cpp, [1441](#)
 - Compression.cpp, [1443](#)
 - CounterAndTimer.cpp, [1445](#)
 - DeviceEvents.cpp, [1448](#)
 - Enumeration.cpp, [1449](#)
 - Enumeration_QuickSpin.cpp, [1450](#)
 - EnumerationEvents.cpp, [1451](#)
 - ExceptionHandling.cpp, [1452](#)
 - Exposure.cpp, [1453](#)
 - Exposure_QuickSpin.cpp, [1454](#)
 - FileAccess_QuickSpin.cpp, [1457](#)
 - GigEVisionPerformance.cpp, [1461](#)
 - ImageEvents.cpp, [1466](#)
 - ImageFormatControl.cpp, [1468](#)
 - ImageFormatControl_QuickSpin.cpp, [1470](#)
 - Inference.cpp, [1474](#)
 - Logging.cpp, [1477](#)
 - LogicBlock.cpp, [1479](#)
 - LookupTable.cpp, [1480](#)
 - NodeMapCallback.cpp, [1482](#)
 - NodeMapInfo.cpp, [1484](#)
 - Polarization.cpp, [1489](#)
 - SaveToAvi.cpp, [1491](#)
 - Sequencer.cpp, [1493](#)
 - SerialRxTx.cpp, [1496](#)
 - Trigger.cpp, [1499](#)
 - Trigger_QuickSpin.cpp, [1501](#)
- Major
 - Spinnaker::GenICam::Version_t, [1209](#)
- major
 - Spinnaker::LibraryVersion, [1069](#)
- make_NodeCallback
 - NodeCallback Class, [372](#)
- max_size
 - Spinnaker::GenICam::gcstring, [863](#)
- maxChars
 - NodeMapInfo.cpp, [1487](#)
- MaxDeviceResetTime
 - Spinnaker::Camera, [625](#)
- MemSet
 - Spinnaker::GenApi::CTestPortStruct, [793](#)
- Member_NodeCallback
 - Spinnaker::GenApi::Member_NodeCallback, [1083](#)
- Member_NodeCallback< Client, Member >, [1081](#)
- MergeXMLFiles
 - INodeMapDyn Interface, [351](#)
- Minor
 - Spinnaker::GenICam::Version_t, [1209](#)
- minor
 - Spinnaker::LibraryVersion, [1069](#)
- NA
 - Types Enums, [403](#)
- networkFilePath
 - Inference.cpp, [1477](#)
- NI
 - Types Enums, [403](#)
- No
 - Types Enums, [409](#)
- Node, [1085](#)
 - Spinnaker::GenApi::Node, [1088](#)
- Node Class, [370](#)
- NodeCallback Class, [371](#)
 - Deregister, [372](#)
 - ECallbackType, [372](#)
 - make_NodeCallback, [372](#)
 - Register, [373](#)
- NodeGetDisplayName
 - Spinnaker Headers, [212](#)
- NodeGetType
 - Spinnaker Headers, [212](#)
- NodesAvailable
 - Spinnaker Headers, [212](#)
- NodesImplemented
 - Spinnaker Headers, [213](#)
- NodesReadable
 - Spinnaker Headers, [213](#)
- NodesWritable
 - Spinnaker Headers, [214](#)
- NodeList_t
 - Spinnaker GenApi Interfaces, [275](#)
- NodeMap, [1096](#)
 - Spinnaker::GenApi::CLOCK, [744](#)
 - Spinnaker::GenApi::NodeMap, [1098](#)
- NodeMap Class, [374](#)
- NodeMapCallback.cpp
 - ChangeHeightAndGain, [1482](#)
 - ConfigureCallbacks, [1482](#)
 - main, [1482](#)
 - OnGainNodeUpdate, [1482](#)
 - OnHeightNodeUpdate, [1482](#)
 - PrintDeviceInfo, [1483](#)

- ResetCallbacks, [1483](#)
- RunSingleCamera, [1483](#)
- NodeMapFactory Class, [375](#)
 - ECacheUsage_t, [375](#)
 - EContentType_t, [376](#)
- NodeMapGetNodeAtIndex
 - Spinnaker Headers, [214](#)
- NodeMapGetNumNodes
 - Spinnaker Headers, [214](#)
- NodeMapInfo.cpp
 - chosenRead, [1486](#)
 - Indent, [1484](#)
 - main, [1484](#)
 - maxChars, [1487](#)
 - PrintBooleanNode, [1485](#)
 - PrintCategoryNodeAndAllFeatures, [1485](#)
 - PrintCommandNode, [1485](#)
 - PrintEnumerationNodeAndCurrentEntry, [1485](#)
 - PrintEnumerationSelector, [1485](#)
 - PrintFloatNode, [1485](#)
 - PrintIntegerNode, [1486](#)
 - PrintNode, [1486](#)
 - PrintStringNode, [1486](#)
 - PrintValueNode, [1486](#)
 - readType, [1484](#)
 - RunSingleCamera, [1486](#)
- NodeMapRef Class, [377](#)
- NodeToString
 - Spinnaker Headers, [216](#)
- None
 - Types Enums, [408](#)
- npos
 - Spinnaker::GenlCam::gcstring, [867](#)
- NumAttachedChunks
 - AttachStatistics_t, [487](#)
- numBuffers
 - BufferHandling.cpp, [1438](#)
- NumChunkPorts
 - AttachStatistics_t, [487](#)
- NumChunks
 - AttachStatistics_t, [487](#)
- numImagesGrabbed
 - GrabInfo, [867](#)
- NumImagesToGrab
 - GigEVisionPerformance.cpp, [1464](#)
- numIncompleteImages
 - GrabInfo, [868](#)
- NumLinks
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, [1105](#)
- NumNodes
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, [1105](#)
- NumProperties
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, [1105](#)
- numRemovals
 - GrabInfo, [868](#)
- NumStrings
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, [1105](#)
- NumToCString
 - Conversion, [417](#)
- ODevFileStream
 - Spinnaker::GenApi, [475](#)
- ODevFileStreamBase< CharType, Traits >, [1106](#)
- ODevFileStreamBuf
 - Spinnaker::GenApi::ODevFileStreamBuf, [1109](#)
- ODevFileStreamBuf< CharType, Traits >, [1108](#)
- OffsetX
 - Spinnaker::Camera, [625](#)
- OffsetY
 - Spinnaker::Camera, [625](#)
- OnDeviceArrival
 - InterfaceEventHandlerImpl, [1044](#), [1045](#)
 - Spinnaker::DeviceArrivalEventHandler, [798](#)
 - Spinnaker::IDeviceArrivalEventHandler, [914](#)
 - Spinnaker::IInterfaceEventHandler, [946](#)
 - Spinnaker::InterfaceEventHandler, [1040](#)
- OnDeviceEvent
 - DeviceEventHandlerImpl, [804](#)
 - Spinnaker::DeviceEventHandler, [801](#)
 - Spinnaker::IDeviceEventHandler, [917](#)
- OnDeviceRemoval
 - InterfaceEventHandlerImpl, [1045](#)
 - Spinnaker::DeviceRemovalEventHandler, [808](#)
 - Spinnaker::IDeviceRemovalEventHandler, [919](#)
 - Spinnaker::IInterfaceEventHandler, [946](#)
 - Spinnaker::InterfaceEventHandler, [1041](#)
- OnGainNodeUpdate
 - NodeMapCallback.cpp, [1482](#)
- OnHeightNodeUpdate
 - NodeMapCallback.cpp, [1482](#)
- OnImageEvent
 - ImageEventHandlerImpl, [987](#), [988](#)
 - Spinnaker::IImageEventHandler, [933](#)
 - Spinnaker::ImageEventHandler, [984](#)
- OnInterfaceArrival
 - Spinnaker::IInterfaceArrivalEventHandler, [944](#)
 - Spinnaker::ISystemEventHandler, [1065](#)
 - Spinnaker::InterfaceArrivalEventHandler, [1038](#)
 - Spinnaker::SystemEventHandler, [1156](#)
 - SystemEventHandlerImpl, [1159](#)
- OnInterfaceRemoval
 - Spinnaker::IInterfaceRemovalEventHandler, [951](#)
 - Spinnaker::ISystemEventHandler, [1065](#)
 - Spinnaker::InterfaceRemovalEventHandler, [1053](#)
 - Spinnaker::SystemEventHandler, [1156](#)
 - SystemEventHandlerImpl, [1159](#)
- OnLogEvent
 - Spinnaker::ILoggingEventHandler, [953](#)
 - Spinnaker::LoggingEventHandler, [1079](#)
- Open
 - Spinnaker::Video::SpinVideo, [1134](#), [1135](#)
- open
 - Spinnaker::GenApi::IDevFileStreamBase, [910](#)

- Spinnaker::GenApi::IDevFileStreamBuf, [912](#)
- Spinnaker::GenApi::ODevFileStreamBase, [1107](#)
- Spinnaker::GenApi::ODevFileStreamBuf, [1109](#)
- openFile
 - Spinnaker::GenApi::FileProtocolAdapter, [845](#)
- OpenFileToRead
 - FileAccess_QuickSpin.cpp, [1457](#)
- OpenFileToWrite
 - FileAccess_QuickSpin.cpp, [1457](#)
- operator bool
 - Spinnaker::BasePtr, [492](#)
 - Spinnaker::GenApi::CPointer, [775](#)
- operator const char *
 - Spinnaker::GenICam::gcstring, [863](#)
- operator delete
 - Spinnaker::GenApi::double_autovector_t, [810](#)
 - Spinnaker::GenApi::int64_autovector_t, [1023](#)
 - Spinnaker::GenICam::gcstring, [863](#)
- operator new
 - Spinnaker::GenApi::double_autovector_t, [810](#)
 - Spinnaker::GenApi::int64_autovector_t, [1024](#)
 - Spinnaker::GenICam::gcstring, [863](#)
- operator T*
 - Spinnaker::BasePtr, [492](#)
 - Spinnaker::GenApi::CPointer, [775](#)
- operator unsigned int
 - Spinnaker::GenApi::Counter, [772](#)
- operator!=
 - INode Interface, [344](#)
 - Spinnaker::Exception, [840](#)
 - Spinnaker::GenApi::CPointer, [775](#), [776](#)
 - Spinnaker::GenApi::Node, [1094](#)
 - Spinnaker::GenICam::gcstring, [864](#)
- operator<
 - Spinnaker::GenICam::gcstring, [865](#)
- operator<<
 - GCString.h, [1306](#)
 - Spinnaker GenApi Classes, [271](#)
- operator>
 - Spinnaker::GenICam::gcstring, [865](#)
- operator>>
 - GCString.h, [1307](#)
 - Spinnaker GenApi Classes, [272](#)
- operator*
 - IEnumeration Interface, [325](#)
 - Spinnaker::GenApi::CPointer, [776](#)
 - Spinnaker::GenApi::EnumNode, [826](#)
 - Spinnaker::GenApi::FloatNode, [852](#)
 - Spinnaker::GenApi::IntegerNode, [1030](#)
 - Spinnaker::GenApi::StringNode, [1139](#)
- operator()
 - IBoolean Interface, [312](#)
 - Spinnaker::GenApi::CEnumerationTRef, [699](#)
 - Spinnaker::GenApi::CNodeCallback, [749](#)
 - Spinnaker::GenApi::CPointer, [776](#)
 - Spinnaker::GenApi::CommandNode, [769](#)
 - Spinnaker::GenApi::FloatNode, [852](#)
 - Spinnaker::GenApi::Function_NodeCallback, [857](#)
 - Spinnaker::GenApi::IntegerNode, [1029](#)
 - Spinnaker::GenApi::Member_NodeCallback, [1083](#)
 - Spinnaker::GenApi::StringNode, [1139](#)
- operator+
 - Spinnaker::GenICam::gcstring, [866](#)
- operator++
 - Spinnaker::GenApi::Counter, [772](#)
- operator+=
 - Spinnaker::GenICam::gcstring, [864](#)
- operator->
 - Spinnaker::BasePtr, [492](#)
 - Spinnaker::GenApi::CPointer, [776](#)
- operator--
 - Spinnaker::GenApi::Counter, [772](#)
- operator=
 - Chunk Data Inference Class, [174](#)
 - IBoolean Interface, [312](#)
 - IEnumerationT Interface, [327](#), [328](#)
 - IFloat Interface, [332](#)
 - IInteger Interface, [333](#)
 - Spinnaker GenApi Classes, [272](#)
 - Spinnaker::BasePtr, [493](#)
 - Spinnaker::CameraBase, [663](#)
 - Spinnaker::CameraList, [671](#)
 - Spinnaker::DeviceArrivalEventHandler, [798](#)
 - Spinnaker::DeviceEventHandler, [802](#)
 - Spinnaker::DeviceRemovalEventHandler, [808](#)
 - Spinnaker::EventHandler, [834](#)
 - Spinnaker::Exception, [840](#)
 - Spinnaker::GenApi::BooleanNode, [498](#)
 - Spinnaker::GenApi::CEnumerationTRef, [700](#)
 - Spinnaker::GenApi::CFloatPtr, [721](#)
 - Spinnaker::GenApi::CNodeMapFactory, [757](#)
 - Spinnaker::GenApi::CNodeMapRef, [760](#)
 - Spinnaker::GenApi::CPointer, [776](#)
 - Spinnaker::GenApi::EnumNode, [826](#)
 - Spinnaker::GenApi::FloatNode, [852](#)
 - Spinnaker::GenApi::IntegerNode, [1030](#)
 - Spinnaker::GenApi::StringNode, [1139](#)
 - Spinnaker::GenApi::double_autovector_t, [810](#)
 - Spinnaker::GenApi::int64_autovector_t, [1024](#)
 - Spinnaker::GenICam::gcstring, [865](#)
 - Spinnaker::ICameraBase, [886](#)
 - Spinnaker::ICameraList, [891](#)
 - Spinnaker::IDeviceArrivalEventHandler, [914](#)
 - Spinnaker::IDeviceEventHandler, [917](#)
 - Spinnaker::IDeviceRemovalEventHandler, [919](#)
 - Spinnaker::IImageEventHandler, [933](#)
 - Spinnaker::IInterface, [940](#)
 - Spinnaker::IInterfaceArrivalEventHandler, [944](#)
 - Spinnaker::IInterfaceEventHandler, [946](#)
 - Spinnaker::IInterfaceList, [949](#)
 - Spinnaker::IInterfaceRemovalEventHandler, [951](#)
 - Spinnaker::ILoggingEventHandler, [953](#)
 - Spinnaker::ISystem, [1061](#)
 - Spinnaker::ISystemEventHandler, [1066](#)
 - Spinnaker::ImageEventHandler, [985](#)
 - Spinnaker::ImagePtr, [992](#)

- Spinnaker::ImageStatistics, 998
- Spinnaker::InterfaceArrivalEventHandler, 1038
- Spinnaker::InterfaceEventHandler, 1041
- Spinnaker::InterfaceList, 1049
- Spinnaker::InterfaceRemovalEventHandler, 1054
- Spinnaker::LoggingEventHandler, 1080
- Spinnaker::SystemEventHandler, 1157
- operator==
 - BasePtr Class, 49
 - INode Interface, 344
 - Spinnaker::BasePtr, 493, 494
 - Spinnaker::Exception, 841
 - Spinnaker::GenApi::CFeatureBag, 719
 - Spinnaker::GenApi::CPointer, 777
 - Spinnaker::GenApi::Node, 1094
 - Spinnaker::GenICam::gcstring, 865
- operator[]
 - Spinnaker::CameraList, 671
 - Spinnaker::GenApi::double_autovector_t, 810
 - Spinnaker::GenApi::int64_autovector_t, 1024
 - Spinnaker::ICameraList, 891
 - Spinnaker::IInterfaceList, 949
 - Spinnaker::InterfaceList, 1049
- ostream_type
 - Spinnaker::GenApi::ODevFileStreamBase, 1107
- overflow
 - Spinnaker::GenApi::ODevFileStreamBuf, 1110
- PCFreq
 - PerformanceCounter, 419
- PGMOption, 1110
 - Spinnaker::PGMOption, 1111
- PMEMBERFUNC
 - Spinnaker::GenApi::Member_NodeCallback, 1082
- PNGOption, 1111
 - Spinnaker::PNGOption, 1112
- POEStatus
 - Spinnaker::TransportLayerInterface, 1186
- POEStatusEnum
 - TransportLayerDefs Class, 248
- PPMOption, 1125
 - Spinnaker::PPMOption, 1126
- PacketDelayToSet
 - GigEVisionPerformance.cpp, 1464
- PacketResendRequestCount
 - Spinnaker::Camera, 625
- PacketSizeToSet
 - GigEVisionPerformance.cpp, 1464
- ParseArguments
 - GigEVisionPerformance.cpp, 1462
- ParseDeviceEventExposureEnd
 - Spinnaker::DeviceEventUtility, 805
- ParseDeviceEventInference
 - Spinnaker::DeviceEventUtility, 806
- PayloadSize
 - Spinnaker::Camera, 626
- PayloadTypeInfoIDs
 - Spinnaker Definitions, 229
- pbackfail
 - Spinnaker::GenApi::IDevFileStreamBuf, 912
- PerformanceCounter, 418
 - CounterStart, 419
 - GetPerformanceCounter, 419
 - PCFreq, 419
 - StartPerformanceCounter, 419
- PersistFeature
 - Spinnaker::GenApi, 476
 - Spinnaker::GenApi::CFeatureBag, 719
- Persistence Class, 378
- PixelColorFilter
 - Spinnaker::Camera, 626
- PixelColorFilterEnums
 - CameraDefs Class, 134
- PixelDynamicRangeMax
 - Spinnaker::Camera, 626
- PixelDynamicRangeMin
 - Spinnaker::Camera, 626
- PixelFormat
 - Spinnaker::Camera, 626
- pixelFormat
 - ImageInfo, 989
- PixelFormatEnums
 - CameraDefs Class, 135
- PixelFormatInfoID
 - Spinnaker::Camera, 627
- PixelFormatInfoSelector
 - Spinnaker::Camera, 627
- PixelFormatInfoSelectorEnums
 - CameraDefs Class, 140
- PixelFormatIntType
 - Spinnaker Definitions, 229
- PixelFormatNamespaceID
 - Spinnaker Definitions, 230
- PixelFormatToSet
 - GigEVisionPerformance.cpp, 1465
- PixelSize
 - Spinnaker::Camera, 627
- PixelSizeEnums
 - CameraDefs Class, 146
- Pointer Class, 379
 - CBasePtr, 380
 - CBooleanPtr, 380
 - CCategoryPtr, 381
 - CChunkPortPtr, 381
 - CCommandPtr, 381
 - CDeviceInfoPtr, 381
 - CEnumEntryPtr, 381
 - CEnumerationPtr, 381
 - CIntegerPtr, 382
 - CNodeMapDynPtr, 382
 - CNodeMapPtr, 382
 - CNodePtr, 382
 - CPortConstructPtr, 382
 - CPortPtr, 382
 - CPortRecorderPtr, 383
 - CPortReplayPtr, 383
 - CPortWriteListPtr, 383

- CRegisterPtr, [383](#)
- CSelectorPtr, [383](#)
- CStringPtr, [383](#)
- CValuePtr, [384](#)
- GetInterfaceName, [384](#)
- IsAvailable, [384](#)
- IsImplemented, [384](#)
- IsReadable, [384](#)
- IsWritable, [384](#)
- Polarization.cpp
 - AcquireImages, [1488](#)
 - ConfigureStream, [1488](#)
 - CreateAndSaveAolpDolpImages, [1488](#)
 - CreateAndSaveGlareReducedImage, [1488](#)
 - CreateAndSaveStokesImages, [1488](#)
 - CreateHeatmapImages, [1488](#)
 - CreateNormalizedImage, [1488](#)
 - ExtractAndSavePolarQuadImages, [1489](#)
 - GetQuadFileNameAppendage, [1489](#)
 - isPixelFormatColor, [1490](#)
 - main, [1489](#)
 - PrintDeviceInfo, [1489](#)
 - RunSingleCamera, [1489](#)
 - SaveImage, [1489](#)
- PolarizationQuadrant
 - Spinnaker::ImageUtilityPolarization, [1011](#)
- Poll
 - INodeMap Interface, [348](#)
 - Spinnaker::GenApi::NodeMap, [1104](#)
- PopulateAdapterIpInfo
 - AdapterConfig, [416](#)
- PortImpl Class, [385](#)
- PortNode, [1113](#)
 - Spinnaker::GenApi::PortNode, [1115](#)
- PortNode Class, [386](#)
 - CPortRef, [386](#)
- PortRecorder, [1118](#)
 - Spinnaker::GenApi::PortRecorder, [1120](#)
- PortRecorder Class, [387](#)
 - CPortRecorderRef, [387](#)
- PortReplay, [1122](#)
 - Spinnaker::GenApi::PortReplay, [1123](#)
- PortReplay Class, [388](#)
- PortWriteList Class, [389](#)
- PowerSupplyCurrent
 - Spinnaker::Camera, [627](#)
- PowerSupplyVoltage
 - Spinnaker::Camera, [627](#)
- Prefix
 - U3V_COMMAND_HEADER, [1203](#)
- Preprocess
 - Spinnaker::GenApi::CNodeMapFactory, [757](#)
- PreprocessXMLFromFile
 - INodeMapDyn Interface, [352](#)
- PreprocessXMLFromZIPFile
 - INodeMapDyn Interface, [352](#)
- PrintAllNodes
 - GigEVisionPerformance.cpp, [1462](#)
- PrintBooleanNode
 - NodeMapInfo.cpp, [1485](#)
- PrintCPUUsage
 - GigEVisionPerformance.cpp, [1462](#)
- PrintCategoryNodeAndAllFeatures
 - NodeMapInfo.cpp, [1485](#)
- PrintCommandNode
 - NodeMapInfo.cpp, [1485](#)
- PrintDataStreamInfo
 - GigEVisionPerformance.cpp, [1462](#)
- PrintDebugMessage
 - FileAccess_QuickSpin.cpp, [1457](#)
- PrintDeviceInfo
 - Acquisition.cpp, [1388](#)
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1433](#)
 - AcquisitionMultipleThread.cpp, [1435](#)
 - ActionCommand.cpp, [1437](#)
 - BufferHandling.cpp, [1439](#)
 - ChunkData.cpp, [1441](#)
 - Compression.cpp, [1444](#)
 - CounterAndTimer.cpp, [1445](#)
 - DeviceEvents.cpp, [1448](#)
 - Exposure.cpp, [1453](#)
 - Exposure_QuickSpin.cpp, [1455](#)
 - FileAccess_QuickSpin.cpp, [1457](#)
 - GigEVisionPerformance.cpp, [1462](#)
 - ImageEvents.cpp, [1467](#)
 - ImageFormatControl.cpp, [1468](#)
 - ImageFormatControl_QuickSpin.cpp, [1470](#)
 - Inference.cpp, [1475](#)
 - LogicBlock.cpp, [1479](#)
 - LookupTable.cpp, [1481](#)
 - NodeMapCallback.cpp, [1483](#)
 - Polarization.cpp, [1489](#)
 - SaveToAvi.cpp, [1491](#)
 - Sequencer.cpp, [1493](#)
 - SerialRxTx.cpp, [1496](#)
 - Trigger.cpp, [1499](#)
 - Trigger_QuickSpin.cpp, [1502](#)
- PrintEnumerationNodeAndCurrentEntry
 - NodeMapInfo.cpp, [1485](#)
- PrintEnumerationSelector
 - NodeMapInfo.cpp, [1485](#)
- PrintExampleStatistics
 - AcquisitionMultipleCameraRecovery.cpp, [1430](#)
- PrintFloatNode
 - NodeMapInfo.cpp, [1485](#)
- PrintGenericHandlerMessage
 - InterfaceEventHandlerImpl, [1046](#)
- PrintIntegerNode
 - NodeMapInfo.cpp, [1486](#)
- PrintNode
 - NodeMapInfo.cpp, [1486](#)
- PrintResultMessage
 - FileAccess_QuickSpin.cpp, [1458](#)
- PrintRetrieveNodeFailure
 - LookupTable.cpp, [1481](#)
 - Sequencer.cpp, [1493](#)

- PrintStringNode
 - NodeMapInfo.cpp, [1486](#)
- PrintUsage
 - FileAccess_QuickSpin.cpp, [1458](#)
 - GigEVisionPerformance.cpp, [1462](#)
- PrintValueNode
 - NodeMapInfo.cpp, [1486](#)
- ProcessCompressedImagesFromFile
 - Compression.cpp, [1444](#)
- ProducerImpl
 - Spinnaker::IInterface, [941](#)
 - Spinnaker::InterfaceList, [1049](#)
- progressive
 - Spinnaker::JPEGOption, [1067](#)
- quality
 - Spinnaker::JPEGOption, [1067](#)
 - Spinnaker::JPG2Option, [1068](#)
 - Spinnaker::Video::MJPGOption, [1084](#)
- QueryInterface
 - Enumeration.cpp, [1449](#)
 - Enumeration_QuickSpin.cpp, [1450](#)
- radius
 - Chunk Data Inference Class, [176](#)
- rdbuf
 - Spinnaker::GenApi::IDevFileStreamBase, [910](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [1108](#)
- Read
 - Spinnaker::GenApi::CChunkPort, [693](#)
 - Spinnaker::GenApi::CEventPort, [716](#)
 - Spinnaker::GenApi::CPortImpl, [780](#)
 - Spinnaker::GenApi::CRegisterPortImpl, [787](#)
 - Spinnaker::GenApi::CTestPortStruct, [794](#)
 - Spinnaker::GenApi::PortNode, [1116](#)
 - Spinnaker::GenApi::PortRecorder, [1120](#)
 - Spinnaker::GenApi::PortReplay, [1124](#)
- read
 - Spinnaker::GenApi::FileProtocolAdapter, [845](#)
- ReadPort
 - Spinnaker::CameraBase, [663](#)
 - Spinnaker::ICameraBase, [886](#)
- ReadRegister
 - Spinnaker::GenApi::CRegisterPortImpl, [787](#)
- readType
 - NodeMapInfo.cpp, [1484](#)
- receiveBuffers
 - AdapterConfig::AdapterInfo, [485](#)
- receiveBuffersMax
 - AdapterConfig::AdapterInfo, [485](#)
- receiveBuffersMin
 - AdapterConfig::AdapterInfo, [486](#)
- receiveBuffersRegKey
 - AdapterConfig::AdapterInfo, [486](#)
- receiveBuffersStep
 - AdapterConfig::AdapterInfo, [486](#)
- rect
 - Chunk Data Inference Class, [176](#)
- Reference Interfaces, [390](#)
- SetNumEnums, [390](#)
- RefreshCameraList
 - AcquisitionMultipleCameraRecovery.cpp, [1431](#)
- RegionDestination
 - Spinnaker::Camera, [627](#)
- RegionDestinationEnums
 - CameraDefs Class, [147](#)
- RegionMode
 - Spinnaker::Camera, [628](#)
- RegionModeEnums
 - CameraDefs Class, [147](#)
- RegionSelector
 - Spinnaker::Camera, [628](#)
- RegionSelectorEnums
 - CameraDefs Class, [147](#)
- Register
 - NodeCallback Class, [373](#)
- RegisterAllInterfaceEvents
 - SystemEventHandlerImpl, [1159](#)
- RegisterCallback
 - INode Interface, [344](#)
 - Spinnaker::GenApi::Node, [1095](#)
- RegisterEventHandler
 - Spinnaker::CameraBase, [663](#), [664](#)
 - Spinnaker::ICameraBase, [886](#)
 - Spinnaker::IInterface, [940](#)
 - Spinnaker::ISystem, [1061](#)
 - Spinnaker::Interface, [1034](#)
 - Spinnaker::System, [1149](#)
- RegisterImageEventHandler
 - Spinnaker::IDataStream, [907](#)
- RegisterInterfaceEventHandler
 - Spinnaker::ISystem, [1061](#)
 - Spinnaker::System, [1149](#)
- RegisterInterfaceEventToSystem
 - SystemEventHandlerImpl, [1160](#)
- RegisterLoggingEventHandler
 - Spinnaker::ISystem, [1061](#)
 - Spinnaker::System, [1150](#)
- RegisterNode, [1127](#)
 - Spinnaker::GenApi::RegisterNode, [1128](#), [1129](#)
- RegisterNode Class, [391](#)
 - CRegisterRef, [391](#)
- RegisterPortImpl Class, [392](#)
- Release
 - Spinnaker::IImage, [928](#)
 - Spinnaker::Image, [976](#)
- ReleaseCameraDescriptionFileData
 - Spinnaker::GenApi::CNodeMapFactory, [757](#)
- ReleaseImage
 - Spinnaker::IDataStream, [907](#)
- ReleaseInstance
 - Spinnaker::ISystem, [1061](#)
 - Spinnaker::System, [1150](#)
- RemoveByDeviceID
 - Spinnaker::CameraList, [671](#)
 - Spinnaker::ICameraList, [892](#)
- RemoveByIndex

- Spinnaker::CameraList, [672](#)
- Spinnaker::ICameraList, [892](#)
- RemoveBySerial
 - Spinnaker::CameraList, [672](#)
 - Spinnaker::ICameraList, [892](#)
- ReplaceEnvironmentVariables
 - GCUtilities Utility, [308](#)
- Replay
 - IPortRecorder Interface, [359](#)
 - Spinnaker::GenApi::CPortImpl, [780](#)
 - Spinnaker::GenApi::CPortWriteList, [783](#)
 - Spinnaker::GenApi::PortNode, [1116](#)
 - Spinnaker::GenApi::PortRecorder, [1120](#)
 - Spinnaker::GenApi::PortReplay, [1124](#)
- ReqId
 - GVCP_REQUEST_HEADER, [878](#)
 - U3V_COMMAND_HEADER, [1203](#)
- Reserved
 - U3V_EVENT_DATA, [1204](#)
- reserved
 - Spinnaker::BMPOption, [495](#)
 - Spinnaker::JPEGOption, [1067](#)
 - Spinnaker::JPG2Option, [1068](#)
 - Spinnaker::PGMOption, [1111](#)
 - Spinnaker::PNGOption, [1112](#)
 - Spinnaker::PPMOption, [1126](#)
 - Spinnaker::TIFFOption, [1164](#)
 - Spinnaker::Video::AVIOption, [490](#)
 - Spinnaker::Video::H264Option, [879](#)
 - Spinnaker::Video::MJPGOption, [1084](#)
- ReservedOrEventSize
 - GVCP_EVENT_ITEM_BASIC, [871](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [872](#)
 - GVCP_EVENT_ITEM, [869](#)
- ResetCallbacks
 - NodeMapCallback.cpp, [1483](#)
- ResetCameraUserSetToDefault
 - AcquisitionMultipleCameraRecovery.cpp, [1431](#)
- ResetDeviceEvents
 - DeviceEvents.cpp, [1448](#)
- ResetExposure
 - Exposure.cpp, [1453](#)
 - Exposure_QuickSpin.cpp, [1455](#)
 - LogicBlock.cpp, [1479](#)
- ResetImage
 - Spinnaker::Image, [928](#), [929](#)
 - Spinnaker::Image, [976](#), [977](#)
- ResetImageEvents
 - ImageEvents.cpp, [1467](#)
- ResetLookupTables
 - LookupTable.cpp, [1481](#)
- ResetSequencer
 - Sequencer.cpp, [1493](#)
- ResetStatistics
 - Spinnaker::GenApi::CTestPortStruct, [794](#)
- ResetStreamHandle
 - Spinnaker::IDataStream, [907](#)
- ResetTrigger
 - BufferHandling.cpp, [1439](#)
 - CounterAndTimer.cpp, [1446](#)
 - LogicBlock.cpp, [1479](#)
 - Trigger.cpp, [1499](#)
 - Trigger_QuickSpin.cpp, [1502](#)
- resize
 - Spinnaker::GenICam::gcstring, [865](#)
- Restore
 - ISelectorDigit Interface, [365](#)
 - Spinnaker::GenApi::CSelectorSet, [790](#)
- result
 - Spinnaker::DeviceEventInferenceData, [805](#)
- RetrieveAllAdapters
 - AdapterConfig, [417](#)
- RetrieveImagesFromFiles
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1433](#)
- ReverseX
 - Spinnaker::Camera, [628](#)
- ReverseY
 - Spinnaker::Camera, [628](#)
- RevokeImages
 - Spinnaker::IDataStream, [907](#)
- RgbTransformLightSource
 - Spinnaker::Camera, [628](#)
- RgbTransformLightSourceEnums
 - CameraDefs Class, [147](#)
- RO
 - Types Enums, [403](#)
- rotatedRect
 - Chunk Data Inference Class, [176](#)
- rotationAngle
 - Chunk Data Inference Class, [176](#)
- RunCameras
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1433](#)
- RunMultipleCameras
 - AcquisitionMultipleThread.cpp, [1435](#)
 - ActionCommand.cpp, [1437](#)
- RunSingleCamera
 - Acquisition.cpp, [1388](#)
 - BufferHandling.cpp, [1439](#)
 - ChunkData, [1442](#)
 - Compression.cpp, [1444](#)
 - CounterAndTimer.cpp, [1446](#)
 - DeviceEvents.cpp, [1448](#)
 - Exposure.cpp, [1453](#)
 - Exposure_QuickSpin.cpp, [1455](#)
 - GigEVisionPerformance.cpp, [1462](#)
 - ImageEvents.cpp, [1467](#)
 - ImageFormatControl.cpp, [1469](#)
 - ImageFormatControl_QuickSpin.cpp, [1470](#)
 - Inference.cpp, [1475](#)
 - LogicBlock.cpp, [1479](#)
 - LookupTable.cpp, [1481](#)
 - NodeMapCallback.cpp, [1483](#)
 - NodeMapInfo.cpp, [1486](#)
 - Polarization.cpp, [1489](#)
 - SaveToAvi.cpp, [1491](#)
 - Sequencer.cpp, [1493](#)

- SerialRxTx.cpp, [1496](#)
- Trigger.cpp, [1500](#)
- Trigger_QuickSpin.cpp, [1502](#)
- RW
 - Types Enums, [403](#)
- SERIAL_PORT_BAUD_RATE
 - SerialRxTx.cpp, [1495](#)
- SERIAL_PORT_COMMUNICATION_TIMEOUT_MIL←LISECOND
 - SerialRxTx.cpp, [1495](#)
- SERIAL_PORT_DELAY
 - SerialRxTx.cpp, [1495](#)
- SERIAL_PORT_PARITY_BITS
 - SerialRxTx.cpp, [1495](#)
- SERIAL_PORT_STOP_BITS
 - SerialRxTx.cpp, [1495](#)
- SET_GUID
 - Spinnaker::GenApi, [476](#)
- SPINNAKER_API_ABSTRACT
 - Spinnaker Platform, [236](#)
- SPINNAKER_API
 - Spinnaker Platform, [236](#)
- SPINNAKER_LOCAL
 - Spinnaker Platform, [236](#)
- SPINUPDATE_API
 - SpinUpdate.h, [1376](#)
- Saturation
 - Spinnaker::Camera, [629](#)
- SaturationEnable
 - Spinnaker::Camera, [629](#)
- Save
 - Spinnaker::Image, [929](#), [930](#)
 - Spinnaker::Image, [978–980](#)
- SaveImage
 - Polarization.cpp, [1489](#)
- SaveToAvi.cpp
 - AcquireImages, [1491](#)
 - chosenVideoType, [1492](#)
 - main, [1491](#)
 - PrintDeviceInfo, [1491](#)
 - RunSingleCamera, [1491](#)
 - SaveVectorToVideo, [1491](#)
 - videoType, [1490](#)
- SaveVectorToVideo
 - SaveToAvi.cpp, [1491](#)
- Scan3dAxisMax
 - Spinnaker::Camera, [629](#)
- Scan3dAxisMin
 - Spinnaker::Camera, [629](#)
- Scan3dCoordinateOffset
 - Spinnaker::Camera, [629](#)
- Scan3dCoordinateReferenceSelector
 - Spinnaker::Camera, [630](#)
- Scan3dCoordinateReferenceSelectorEnums
 - CameraDefs Class, [148](#)
- Scan3dCoordinateReferenceValue
 - Spinnaker::Camera, [630](#)
- Scan3dCoordinateScale
 - Spinnaker::Camera, [630](#)
- Scan3dCoordinateSelector
 - Spinnaker::Camera, [630](#)
- Scan3dCoordinateSelectorEnums
 - CameraDefs Class, [148](#)
- Scan3dCoordinateSystem
 - Spinnaker::Camera, [630](#)
- Scan3dCoordinateSystemEnums
 - CameraDefs Class, [149](#)
- Scan3dCoordinateSystemReference
 - Spinnaker::Camera, [630](#)
- Scan3dCoordinateSystemReferenceEnums
 - CameraDefs Class, [149](#)
- Scan3dCoordinateTransformSelector
 - Spinnaker::Camera, [631](#)
- Scan3dCoordinateTransformSelectorEnums
 - CameraDefs Class, [149](#)
- Scan3dDistanceUnit
 - Spinnaker::Camera, [631](#)
- Scan3dDistanceUnitEnums
 - CameraDefs Class, [150](#)
- Scan3dInvalidDataFlag
 - Spinnaker::Camera, [631](#)
- Scan3dInvalidDataValue
 - Spinnaker::Camera, [631](#)
- Scan3dOutputMode
 - Spinnaker::Camera, [631](#)
- Scan3dOutputModeEnums
 - CameraDefs Class, [150](#)
- Scan3dTransformValue
 - Spinnaker::Camera, [631](#)
- SecondsCounter, [419](#)
 - endTime, [420](#)
 - GetSecondsCounter, [420](#)
 - StartSecondsCounter, [420](#)
 - startTime, [420](#)
 - timeDiff, [420](#)
- SelectorSet Class, [393](#)
- SendActionCommand
 - Spinnaker::Interface, [940](#)
 - Spinnaker::System, [1062](#)
 - Spinnaker::Interface, [1034](#)
 - Spinnaker::System, [1150](#)
- Sensor
 - Spinnaker::CCMSettings, [695](#)
- SensorDescription
 - Spinnaker::Camera, [632](#)
- SensorDigitizationTaps
 - Spinnaker::Camera, [632](#)
- SensorDigitizationTapsEnums
 - CameraDefs Class, [151](#)
- SensorHeight
 - Spinnaker::Camera, [632](#)
- SensorShutterMode
 - Spinnaker::Camera, [632](#)
- SensorShutterModeEnums
 - CameraDefs Class, [151](#)
- SensorTaps

- Spinnaker::Camera, [632](#)
- SensorTapsEnums
 - CameraDefs Class, [152](#)
- SensorWidth
 - Spinnaker::Camera, [632](#)
- Sequencer.cpp
 - AcquireImages, [1492](#)
 - ConfigureSequencerPartOne, [1492](#)
 - ConfigureSequencerPartTwo, [1493](#)
 - main, [1493](#)
 - PrintDeviceInfo, [1493](#)
 - PrintRetrieveNodeFailure, [1493](#)
 - ResetSequencer, [1493](#)
 - RunSingleCamera, [1493](#)
 - SetSingleState, [1493](#)
- SequencerConfigurationMode
 - Spinnaker::Camera, [633](#)
- SequencerConfigurationModeEnums
 - CameraDefs Class, [152](#)
- SequencerConfigurationValid
 - Spinnaker::Camera, [633](#)
- SequencerConfigurationValidEnums
 - CameraDefs Class, [153](#)
- SequencerFeatureEnable
 - Spinnaker::Camera, [633](#)
- SequencerMode
 - Spinnaker::Camera, [633](#)
- SequencerModeEnums
 - CameraDefs Class, [153](#)
- SequencerPathSelector
 - Spinnaker::Camera, [633](#)
- SequencerSetActive
 - Spinnaker::Camera, [634](#)
- SequencerSetLoad
 - Spinnaker::Camera, [634](#)
- SequencerSetNext
 - Spinnaker::Camera, [634](#)
- SequencerSetSave
 - Spinnaker::Camera, [634](#)
- SequencerSetSelector
 - Spinnaker::Camera, [634](#)
- SequencerSetStart
 - Spinnaker::Camera, [635](#)
- SequencerSetValid
 - Spinnaker::Camera, [635](#)
- SequencerSetValidEnums
 - CameraDefs Class, [153](#)
- SequencerTriggerActivation
 - Spinnaker::Camera, [635](#)
- SequencerTriggerActivationEnums
 - CameraDefs Class, [153](#)
- SequencerTriggerSource
 - Spinnaker::Camera, [635](#)
- SequencerTriggerSourceEnums
 - CameraDefs Class, [154](#)
- SerialPortBaudRate
 - Spinnaker::Camera, [635](#)
- SerialPortBaudRateEnums
 - CameraDefs Class, [154](#)
- SerialPortDataBits
 - Spinnaker::Camera, [636](#)
- SerialPortParity
 - Spinnaker::Camera, [636](#)
- SerialPortParityEnums
 - CameraDefs Class, [155](#)
- SerialPortSelector
 - Spinnaker::Camera, [636](#)
- SerialPortSelectorEnums
 - CameraDefs Class, [155](#)
- SerialPortSource
 - Spinnaker::Camera, [636](#)
- SerialPortSourceEnums
 - CameraDefs Class, [155](#)
- SerialPortStopBits
 - Spinnaker::Camera, [636](#)
- SerialPortStopBitsEnums
 - CameraDefs Class, [156](#)
- SerialReceiveFramingErrorCount
 - Spinnaker::Camera, [636](#)
- SerialReceiveParityErrorCount
 - Spinnaker::Camera, [637](#)
- SerialReceiveQueueClear
 - Spinnaker::Camera, [637](#)
- SerialReceiveQueueCurrentCharacterCount
 - Spinnaker::Camera, [637](#)
- SerialReceiveQueueMaxCharacterCount
 - Spinnaker::Camera, [637](#)
- SerialRx
 - SerialRxTx.cpp, [1496](#)
- SerialRxTx.cpp
 - COM_PORT_COUNT_MAX, [1494](#)
 - CleanUp, [1496](#)
 - ConfigureDevice, [1496](#)
 - DATA_BITS, [1495](#)
 - MILLISECOND, [1495](#)
 - main, [1496](#)
 - PrintDeviceInfo, [1496](#)
 - RunSingleCamera, [1496](#)
 - SERIAL_PORT_BAUD_RATE, [1495](#)
 - SERIAL_PORT_COMMUNICATION_TIMEOUT↔
_MILLISECOND, [1495](#)
 - SERIAL_PORT_DELAY, [1495](#)
 - SERIAL_PORT_PARITY_BITS, [1495](#)
 - SERIAL_PORT_STOP_BITS, [1495](#)
 - SerialRx, [1496](#)
 - SerialTx, [1497](#)
 - TWO_SECOND_DELAY, [1495](#)
- SerialTransmitQueueCurrentCharacterCount
 - Spinnaker::Camera, [637](#)
- SerialTransmitQueueMaxCharacterCount
 - Spinnaker::Camera, [637](#)
- SerialTx
 - SerialRxTx.cpp, [1497](#)
- Set
 - Spinnaker::GenApi::RegisterNode, [1130](#)
- SetBufferOwnership

- Spinnaker::CameraBase, [664](#)
- Spinnaker::ICameraBase, [887](#)
- SetChannelStatus
 - Spinnaker::ImageStatistics, [937](#)
 - Spinnaker::ImageStatistics, [998](#)
- SetChunkEnable
 - Inference.cpp, [1475](#)
- SetChunks
 - Spinnaker::ChunkData, [740](#)
 - Spinnaker::IChunkData, [901](#)
- SetCookie
 - IPortRecorder Interface, [360](#)
 - Spinnaker::GenApi::CPortWriteList, [783](#)
- SetDefaultColorProcessing
 - Spinnaker::Image, [980](#)
- SetEnumReference
 - Spinnaker::GenApi::CEnumerationTRef, [700](#)
- SetEventPayload
 - Spinnaker::EventHandler, [834](#)
- SetEventType
 - Spinnaker::EventHandler, [834](#)
- SetFirst
 - Spinnaker::GenApi::CSelectorSet, [790](#)
- SetFrameRate
 - GigEVisionPerformance.cpp, [1463](#)
- SetGenICamCLProtocolFolder
 - GCUtilities Utility, [309](#)
- SetGenICamCacheFolder
 - GCUtilities Utility, [308](#)
- SetGenICamLogConfig
 - GCUtilities Utility, [309](#)
- SetHeatmapColorGradient
 - Spinnaker::ImageUtilityHeatmap, [1009](#)
- SetHeatmapRange
 - Spinnaker::ImageUtilityHeatmap, [1010](#)
- SetInfo
 - Spinnaker::GenApi::CFeatureBag, [719](#)
- SetIntValue
 - IEnumeration Interface, [326](#)
 - Spinnaker::GenApi::EnumNode, [826](#)
- SetLoggingEventPriorityLevel
 - Spinnaker::ISystem, [1062](#)
 - Spinnaker::System, [1151](#)
- SetMaximumFileSize
 - Spinnaker::Video::SpinVideo, [1136](#)
- SetMessageCallback
 - SpinUpdate.h, [1377](#)
- SetNext
 - ISelectorDigit Interface, [365](#)
 - Spinnaker::GenApi::CSelectorSet, [790](#)
- SetNodeHandle
 - Spinnaker::GenApi::Node, [1095](#)
- SetNodeMap
 - Spinnaker::GenApi::Node, [1095](#)
- SetNumDecompressionThreads
 - Spinnaker::Image, [981](#)
- SetNumEnums
 - Reference Interfaces, [390](#)
- Spinnaker::GenApi::CEnumerationTRef, [700](#)
- SetPortImpl
 - Spinnaker::GenApi::CChunkPort, [693](#)
 - Spinnaker::GenApi::CEventPort, [716](#)
 - Spinnaker::GenApi::CPortImpl, [780](#)
 - Spinnaker::GenApi::CRegisterPortImpl, [787](#)
 - Spinnaker::GenApi::PortNode, [1116](#)
- SetProgressCallback
 - SpinUpdate.h, [1377](#)
- SetReference
 - Spinnaker::GenApi::BooleanNode, [498](#)
 - Spinnaker::GenApi::CEnumerationTRef, [700](#)
 - Spinnaker::GenApi::CategoryNode, [676](#)
 - Spinnaker::GenApi::CommandNode, [769](#)
 - Spinnaker::GenApi::EnumEntryNode, [821](#)
 - Spinnaker::GenApi::EnumNode, [826](#)
 - Spinnaker::GenApi::FloatNode, [852](#)
 - Spinnaker::GenApi::FloatRegNode, [855](#)
 - Spinnaker::GenApi::IntRegNode, [1056](#)
 - Spinnaker::GenApi::IntegerNode, [1030](#)
 - Spinnaker::GenApi::Node, [1095](#)
 - Spinnaker::GenApi::PortNode, [1117](#)
 - Spinnaker::GenApi::PortRecorder, [1121](#)
 - Spinnaker::GenApi::PortReplay, [1125](#)
 - Spinnaker::GenApi::RegisterNode, [1130](#)
 - Spinnaker::GenApi::StringNode, [1140](#)
 - Spinnaker::GenApi::StringRegNode, [1143](#)
 - Spinnaker::GenApi::ValueNode, [1208](#)
- SetSelectedCameraIndex
 - SpinnakerDirectShow.h, [234](#)
- SetSingleState
 - Sequencer.cpp, [1493](#)
- SetUserBuffers
 - Spinnaker::CameraBase, [664](#), [665](#)
 - Spinnaker::ICameraBase, [887](#)
- SetValue
 - Spinnaker::GenApi::BooleanNode, [498](#)
 - Spinnaker::GenApi::CEnumerationTRef, [701](#)
 - Spinnaker::GenApi::FloatNode, [852](#)
 - Spinnaker::GenApi::IntegerNode, [1030](#)
 - Spinnaker::GenApi::StringNode, [1140](#)
- SetupCounterAndTimer
 - CounterAndTimer.cpp, [1446](#)
- Sharpening
 - Spinnaker::Camera, [638](#)
- SharpeningAuto
 - Spinnaker::Camera, [638](#)
- SharpeningEnable
 - Spinnaker::Camera, [638](#)
- SharpeningThreshold
 - Spinnaker::Camera, [638](#)
- Signed
 - Types Enums, [408](#)
- SingleChunkData_t, [1130](#)
 - ChunkID, [1131](#)
 - ChunkLength, [1131](#)
 - ChunkOffset, [1131](#)
- SingleChunkDataStr_t, [1131](#)

- ChunkID, [1131](#)
- ChunkLength, [1131](#)
- ChunkOffset, [1132](#)
- size
 - Spinnaker::GenApi::double_autovector_t, [810](#)
 - Spinnaker::GenApi::int64_autovector_t, [1024](#)
 - Spinnaker::GenICam::gcstring, [865](#)
- SleepyWrapper
 - AcquisitionMultipleCameraRecovery.cpp, [1431](#)
 - ActionCommand.cpp, [1437](#)
 - BufferHandling.cpp, [1439](#)
 - ImageEvents.cpp, [1467](#)
- SoftwareSignalPulse
 - Spinnaker::Camera, [639](#)
- SoftwareSignalSelector
 - Spinnaker::Camera, [639](#)
- SoftwareSignalSelectorEnums
 - CameraDefs Class, [156](#)
- SourceCount
 - Spinnaker::Camera, [639](#)
- SourceDataRange
 - Spinnaker::ImageUtility, [1000](#)
- SourceSelector
 - Spinnaker::Camera, [639](#)
- SourceSelectorEnums
 - CameraDefs Class, [156](#)
- SpinTestCamera, [1132](#)
- SpinTestCamera Class, [394](#)
- SpinUpdate.h
 - GetErrorMessage, [1377](#)
 - SPINUPDATE_API, [1376](#)
 - SetMessageCallback, [1377](#)
 - SetProgressCallback, [1377](#)
 - UpdateFirmware, [1377](#)
 - UpdateFirmwareConsole, [1377](#)
 - UpdateFirmwareGUI, [1378](#)
 - UpdatorMessageCallback, [1378](#)
 - UpdatorProgressCallback, [1378](#)
- SpinVideo, [1133](#)
 - Spinnaker::Video::SpinVideo, [1133](#)
- Spinnaker, [421](#)
- Spinnaker Classes, [42](#)
 - InferenceBoxType, [45](#)
- Spinnaker Definitions, [220](#)
 - ActionCommandStatus, [224](#)
 - BufferOwnership, [224](#)
 - CCMColorTemperature, [224](#)
 - CCMSensor, [225](#)
 - CCMType, [225](#)
 - ColorProcessingAlgorithm, [225](#)
 - Error, [226](#)
 - EventType, [227](#)
 - ImageFileFormat, [227](#)
 - ImageStatus, [228](#)
 - PayloadTypeInfoIDs, [229](#)
 - PixelFormatIntType, [229](#)
 - PixelFormatNamespaceID, [230](#)
 - SpinnakerLogLevel, [230](#)
 - StatisticsChannel, [230](#)
 - Spinnaker EventHandler Classes, [178](#)
 - Spinnaker GenApi Classes, [262](#)
 - _ClearXMLCache, [268](#)
 - _Connect, [268](#), [269](#)
 - _Destroy, [269](#)
 - _GetDeviceName, [269](#)
 - _GetNode, [269](#)
 - _GetNodes, [269](#)
 - _GetSupportedSchemaVersions, [269](#)
 - _InvalidateNodes, [269](#)
 - _LoadXMLFromFile, [270](#)
 - _LoadXMLFromFileInject, [270](#)
 - _LoadXMLFromString, [270](#)
 - _LoadXMLFromStringInject, [270](#)
 - _LoadXMLFromZIPData, [270](#)
 - _LoadXMLFromZIPFile, [270](#)
 - _Poll, [270](#)
 - ~CNodeMapRefT, [272](#)
 - CNodeMapRef, [268](#)
 - CNodeMapRefT, [271](#)
 - CNodeRef, [268](#)
 - CSelectorRef, [268](#)
 - CastToIDestroy, [271](#)
 - EatComments, [271](#)
 - operator<<, [271](#)
 - operator>>, [272](#)
 - operator=, [272](#)
 - Spinnaker GenApi Enums, [399](#)
 - Spinnaker GenApi Interfaces, [274](#)
 - CallbackHandleType, [275](#)
 - NodeList_t, [275](#)
 - Spinnaker GenApi Utilities, [304](#)
 - Spinnaker Headers, [202](#)
 - BooleanGetValue, [205](#)
 - BooleanSetValue, [205](#)
 - bufferSize, [217](#)
 - CommandExecute, [206](#)
 - EVENT_TIMEOUT_INFINITE, [218](#)
 - EVENT_TIMEOUT_NONE, [218](#)
 - entryExist, [217](#)
 - entryIndex, [217](#)
 - enumerationEntryName, [217](#)
 - EnumerationGetEntry, [206](#)
 - EnumerationGetNumEntries, [206](#)
 - enumerationName, [218](#)
 - EnumerationSetEntry, [207](#)
 - FloatGetInc, [207](#)
 - FloatGetIncMode, [207](#)
 - FloatGetMax, [208](#)
 - FloatGetMin, [208](#)
 - FloatGetValue, [209](#)
 - FloatSetValue, [209](#)
 - IntegerGetInc, [209](#)
 - IntegerGetIncMode, [210](#)
 - IntegerGetMax, [210](#)
 - IntegerGetMin, [210](#)
 - IntegerGetValue, [211](#)

- IntegerSetValue, [211](#)
- NodeGetDisplayName, [212](#)
- NodeGetType, [212](#)
- NodesAvailable, [212](#)
- NodesImplemented, [213](#)
- NodesReadable, [213](#)
- NodesWritable, [214](#)
- NodeMapGetNodeAtIndex, [214](#)
- NodeMapGetNumNodes, [214](#)
- NodeToString, [216](#)
- StringGetValue, [216](#)
- StringSetValue, [217](#)
- Spinnaker Platform, [236](#)
 - SPINNAKER_API_ABSTRACT, [236](#)
 - SPINNAKER_API, [236](#)
 - SPINNAKER_LOCAL, [236](#)
- Spinnaker QuickSpin Classes, [242](#)
- Spinnaker Video Class, [237](#)
- Spinnaker Video Definitions, [238](#)
- Spinnaker.h, [219](#)
- Spinnaker::ActionCommandResult
 - DeviceAddress, [481](#)
 - Status, [481](#)
- Spinnaker::BMPOption
 - BMPOption, [495](#)
 - indexedColor_8bit, [495](#)
 - reserved, [495](#)
- Spinnaker::BasePtr
 - ~BasePtr, [491](#)
 - BasePtr, [491](#)
 - get, [492](#)
 - IsValid, [492](#)
 - m_pT, [494](#)
 - operator bool, [492](#)
 - operator T*, [492](#)
 - operator->, [492](#)
 - operator=, [493](#)
 - operator==, [493](#), [494](#)
- Spinnaker::CCMSettings
 - CCMSettings, [694](#)
 - ColorTemperature, [695](#)
 - CustomCCMCode, [695](#)
 - Sensor, [695](#)
 - Type, [695](#)
- Spinnaker::Camera
 - ~Camera, [529](#)
 - aPAUSEMACCtrlFramesReceived, [535](#)
 - aPAUSEMACCtrlFramesTransmitted, [535](#)
 - AasRoiEnable, [529](#)
 - AasRoiHeight, [530](#)
 - AasRoiOffsetX, [530](#)
 - AasRoiOffsetY, [530](#)
 - AasRoiWidth, [530](#)
 - AcquisitionAbort, [531](#)
 - AcquisitionArm, [531](#)
 - AcquisitionBurstFrameCount, [531](#)
 - AcquisitionFrameCount, [531](#)
 - AcquisitionFrameRate, [532](#)
 - AcquisitionFrameRateEnable, [532](#)
 - AcquisitionLineRate, [532](#)
 - AcquisitionMode, [532](#)
 - AcquisitionResultingFrameRate, [532](#)
 - AcquisitionStart, [532](#)
 - AcquisitionStatus, [533](#)
 - AcquisitionStatusSelector, [533](#)
 - AcquisitionStop, [533](#)
 - ActionDeviceKey, [533](#)
 - ActionGroupKey, [533](#)
 - ActionGroupMask, [533](#)
 - ActionQueueSize, [534](#)
 - ActionSelector, [534](#)
 - ActionUnconditionalMode, [534](#)
 - AdaptiveCompressionEnable, [534](#)
 - AdcBitDepth, [534](#)
 - AutoAlgorithmSelector, [535](#)
 - AutoExposureControlLoopDamping, [535](#)
 - AutoExposureControlPriority, [535](#)
 - AutoExposureEVCompensation, [536](#)
 - AutoExposureExposureTimeLowerLimit, [536](#)
 - AutoExposureExposureTimeUpperLimit, [536](#)
 - AutoExposureGainLowerLimit, [536](#)
 - AutoExposureGainUpperLimit, [537](#)
 - AutoExposureGreyValueLowerLimit, [537](#)
 - AutoExposureGreyValueUpperLimit, [537](#)
 - AutoExposureLightingMode, [537](#)
 - AutoExposureMeteringMode, [538](#)
 - AutoExposureTargetGreyValue, [538](#)
 - AutoExposureTargetGreyValueAuto, [538](#)
 - BalanceRatio, [539](#)
 - BalanceRatioSelector, [539](#)
 - BalanceWhiteAuto, [539](#)
 - BalanceWhiteAutoDamping, [539](#)
 - BalanceWhiteAutoLowerLimit, [540](#)
 - BalanceWhiteAutoProfile, [540](#)
 - BalanceWhiteAutoUpperLimit, [540](#)
 - BinningHorizontal, [540](#)
 - BinningHorizontalMode, [541](#)
 - BinningSelector, [541](#)
 - BinningVertical, [541](#)
 - BinningVerticalMode, [541](#)
 - BlackLevel, [542](#)
 - BlackLevelAuto, [542](#)
 - BlackLevelAutoBalance, [542](#)
 - BlackLevelClampingEnable, [542](#)
 - BlackLevelRaw, [542](#)
 - BlackLevelSelector, [543](#)
 - Camera, [529](#)
 - ChunkBlackLevel, [543](#)
 - ChunkBlackLevelSelector, [543](#)
 - ChunkCRC, [544](#)
 - ChunkCompressionMode, [543](#)
 - ChunkCompressionRatio, [543](#)
 - ChunkCounterSelector, [544](#)
 - ChunkCounterValue, [544](#)
 - ChunkEnable, [544](#)
 - ChunkEncoderSelector, [544](#)

- ChunkEncoderStatus, [544](#)
- ChunkEncoderValue, [545](#)
- ChunkExposureEndLineStatusAll, [545](#)
- ChunkExposureTime, [545](#)
- ChunkExposureTimeSelector, [545](#)
- ChunkFrameID, [545](#)
- ChunkGain, [545](#)
- ChunkGainSelector, [546](#)
- ChunkHeight, [546](#)
- ChunkImage, [546](#)
- ChunkImageComponent, [546](#)
- ChunkInferenceBoundingBoxResult, [546](#)
- ChunkInferenceConfidence, [546](#)
- ChunkInferenceFrameId, [547](#)
- ChunkInferenceResult, [547](#)
- ChunkLinePitch, [547](#)
- ChunkLineStatusAll, [547](#)
- ChunkModeActive, [547](#)
- ChunkOffsetX, [547](#)
- ChunkOffsetY, [548](#)
- ChunkPartSelector, [548](#)
- ChunkPixelDynamicRangeMax, [548](#)
- ChunkPixelDynamicRangeMin, [548](#)
- ChunkPixelFormat, [548](#)
- ChunkRegionID, [548](#)
- ChunkScan3dAxisMax, [549](#)
- ChunkScan3dAxisMin, [549](#)
- ChunkScan3dCoordinateOffset, [549](#)
- ChunkScan3dCoordinateReferenceSelector, [549](#)
- ChunkScan3dCoordinateReferenceValue, [549](#)
- ChunkScan3dCoordinateScale, [549](#)
- ChunkScan3dCoordinateSelector, [550](#)
- ChunkScan3dCoordinateSystem, [550](#)
- ChunkScan3dCoordinateSystemReference, [550](#)
- ChunkScan3dCoordinateTransformSelector, [550](#)
- ChunkScan3dDistanceUnit, [550](#)
- ChunkScan3dInvalidDataFlag, [550](#)
- ChunkScan3dInvalidDataValue, [551](#)
- ChunkScan3dOutputMode, [551](#)
- ChunkScan3dTransformValue, [551](#)
- ChunkScanLineSelector, [551](#)
- ChunkSelector, [551](#)
- ChunkSequencerSetActive, [551](#)
- ChunkSerialData, [552](#)
- ChunkSerialDataLength, [552](#)
- ChunkSerialReceiveOverflow, [552](#)
- ChunkSourceID, [552](#)
- ChunkStreamChannelID, [552](#)
- ChunkTimerSelector, [552](#)
- ChunkTimerValue, [553](#)
- ChunkTimestamp, [553](#)
- ChunkTimestampLatchValue, [553](#)
- ChunkTransferBlockID, [553](#)
- ChunkTransferQueueCurrentBlockCount, [553](#)
- ChunkTransferStreamID, [553](#)
- ChunkWidth, [554](#)
- CIConfiguration, [554](#)
- CITimeSlotsCount, [554](#)
- ColorTransformationEnable, [554](#)
- ColorTransformationSelector, [554](#)
- ColorTransformationValue, [555](#)
- ColorTransformationValueSelector, [555](#)
- CompressionRatio, [555](#)
- CompressionSaturationPriority, [555](#)
- CounterDelay, [555](#)
- CounterDuration, [556](#)
- CounterEventActivation, [556](#)
- CounterEventSource, [556](#)
- CounterReset, [556](#)
- CounterResetActivation, [556](#)
- CounterResetSource, [556](#)
- CounterSelector, [557](#)
- CounterStatus, [557](#)
- CounterTriggerActivation, [557](#)
- CounterTriggerSource, [557](#)
- CounterValue, [557](#)
- CounterValueAtReset, [557](#)
- CxpConnectionSelector, [558](#)
- CxpConnectionTestErrorCount, [558](#)
- CxpConnectionTestMode, [558](#)
- CxpConnectionTestPacketCount, [558](#)
- CxpLinkConfiguration, [558](#)
- CxpLinkConfigurationPreferred, [558](#)
- CxpLinkConfigurationStatus, [559](#)
- CxpPoCxpAuto, [559](#)
- CxpPoCxpStatus, [559](#)
- CxpPoCxpTripReset, [559](#)
- CxpPoCxpTurnOff, [559](#)
- DecimationHorizontal, [559](#)
- DecimationHorizontalMode, [560](#)
- DecimationSelector, [560](#)
- DecimationVertical, [560](#)
- DecimationVerticalMode, [560](#)
- DefectCorrectStaticEnable, [561](#)
- DefectCorrectionMode, [561](#)
- DefectTableApply, [561](#)
- DefectTableCoordinateX, [561](#)
- DefectTableCoordinateY, [561](#)
- DefectTableFactoryRestore, [562](#)
- DefectTableIndex, [562](#)
- DefectTablePixelCount, [562](#)
- DefectTableSave, [562](#)
- Deinterlacing, [563](#)
- DeviceCharacterSet, [563](#)
- DeviceClockFrequency, [563](#)
- DeviceClockSelector, [563](#)
- DeviceConnectionSelector, [563](#)
- DeviceConnectionSpeed, [564](#)
- DeviceConnectionStatus, [564](#)
- DeviceEventChannelCount, [564](#)
- DeviceFamilyName, [564](#)
- DeviceFeaturePersistenceEnd, [564](#)
- DeviceFeaturePersistenceStart, [564](#)
- DeviceFirmwareVersion, [565](#)
- DeviceGenCPVersionMajor, [565](#)
- DeviceGenCPVersionMinor, [565](#)

- DeviceID, [565](#)
- DeviceIndicatorMode, [565](#)
- DeviceLinkBandwidthReserve, [565](#)
- DeviceLinkCommandTimeout, [566](#)
- DeviceLinkConnectionCount, [566](#)
- DeviceLinkCurrentThroughput, [566](#)
- DeviceLinkHeartbeatMode, [566](#)
- DeviceLinkHeartbeatTimeout, [566](#)
- DeviceLinkSelector, [566](#)
- DeviceLinkSpeed, [567](#)
- DeviceLinkThroughputLimit, [567](#)
- DeviceLinkThroughputLimitMode, [567](#)
- DeviceManifestEntrySelector, [567](#)
- DeviceManifestPrimaryURL, [568](#)
- DeviceManifestSchemaMajorVersion, [568](#)
- DeviceManifestSchemaMinorVersion, [568](#)
- DeviceManifestSecondaryURL, [568](#)
- DeviceManifestXMLMajorVersion, [568](#)
- DeviceManifestXMLMinorVersion, [568](#)
- DeviceManifestXMLSubMinorVersion, [569](#)
- DeviceManufacturerInfo, [569](#)
- DeviceMaxThroughput, [569](#)
- DeviceModelName, [569](#)
- DevicePowerSupplySelector, [569](#)
- DeviceRegistersCheck, [570](#)
- DeviceRegistersEndianness, [570](#)
- DeviceRegistersStreamingEnd, [570](#)
- DeviceRegistersStreamingStart, [570](#)
- DeviceRegistersValid, [570](#)
- DeviceReset, [570](#)
- DeviceSFNCVersionMajor, [571](#)
- DeviceSFNCVersionMinor, [572](#)
- DeviceSFNCVersionSubMinor, [572](#)
- DeviceScanType, [571](#)
- DeviceSerialNumber, [571](#)
- DeviceSerialPortBaudRate, [571](#)
- DeviceSerialPortSelector, [571](#)
- DeviceStreamChannelCount, [572](#)
- DeviceStreamChannelEndianness, [572](#)
- DeviceStreamChannelLink, [572](#)
- DeviceStreamChannelPacketSize, [573](#)
- DeviceStreamChannelSelector, [573](#)
- DeviceStreamChannelType, [573](#)
- DeviceTLType, [574](#)
- DeviceTLVersionMajor, [574](#)
- DeviceTLVersionMinor, [574](#)
- DeviceTLVersionSubMinor, [574](#)
- DeviceTapGeometry, [573](#)
- DeviceTemperature, [573](#)
- DeviceTemperatureSelector, [573](#)
- DeviceType, [574](#)
- DeviceUptime, [575](#)
- DeviceUserID, [575](#)
- DeviceVendorName, [575](#)
- DeviceVersion, [575](#)
- EncoderDivider, [575](#)
- EncoderMode, [575](#)
- EncoderOutputMode, [576](#)
- EncoderReset, [576](#)
- EncoderResetActivation, [576](#)
- EncoderResetSource, [576](#)
- EncoderSelector, [576](#)
- EncoderSourceA, [576](#)
- EncoderSourceB, [577](#)
- EncoderStatus, [577](#)
- EncoderTimeout, [577](#)
- EncoderValue, [577](#)
- EncoderValueAtReset, [577](#)
- EnumerationCount, [577](#)
- EventAcquisitionEnd, [578](#)
- EventAcquisitionEndFrameID, [578](#)
- EventAcquisitionEndTimestamp, [578](#)
- EventAcquisitionError, [578](#)
- EventAcquisitionErrorFrameID, [578](#)
- EventAcquisitionErrorTimestamp, [578](#)
- EventAcquisitionStart, [579](#)
- EventAcquisitionStartFrameID, [579](#)
- EventAcquisitionStartTimestamp, [579](#)
- EventAcquisitionTransferEnd, [579](#)
- EventAcquisitionTransferEndFrameID, [579](#)
- EventAcquisitionTransferEndTimestamp, [579](#)
- EventAcquisitionTransferStart, [580](#)
- EventAcquisitionTransferStartFrameID, [580](#)
- EventAcquisitionTransferStartTimestamp, [580](#)
- EventAcquisitionTrigger, [580](#)
- EventAcquisitionTriggerFrameID, [580](#)
- EventAcquisitionTriggerTimestamp, [580](#)
- EventActionLate, [581](#)
- EventActionLateFrameID, [581](#)
- EventActionLateTimestamp, [581](#)
- EventCounter0End, [581](#)
- EventCounter0EndFrameID, [581](#)
- EventCounter0EndTimestamp, [581](#)
- EventCounter0Start, [582](#)
- EventCounter0StartFrameID, [582](#)
- EventCounter0StartTimestamp, [582](#)
- EventCounter1End, [582](#)
- EventCounter1EndFrameID, [582](#)
- EventCounter1EndTimestamp, [582](#)
- EventCounter1Start, [583](#)
- EventCounter1StartFrameID, [583](#)
- EventCounter1StartTimestamp, [583](#)
- EventEncoder0Restarted, [583](#)
- EventEncoder0RestartedFrameID, [583](#)
- EventEncoder0RestartedTimestamp, [583](#)
- EventEncoder0Stopped, [584](#)
- EventEncoder0StoppedFrameID, [584](#)
- EventEncoder0StoppedTimestamp, [584](#)
- EventEncoder1Restarted, [584](#)
- EventEncoder1RestartedFrameID, [584](#)
- EventEncoder1RestartedTimestamp, [584](#)
- EventEncoder1Stopped, [585](#)
- EventEncoder1StoppedFrameID, [585](#)
- EventEncoder1StoppedTimestamp, [585](#)
- EventError, [585](#)
- EventErrorCode, [585](#)

- EventErrorFrameID, [585](#)
- EventErrorTimestamp, [586](#)
- EventExposureEnd, [586](#)
- EventExposureEndFrameID, [586](#)
- EventExposureEndTimestamp, [586](#)
- EventExposureStart, [586](#)
- EventExposureStartFrameID, [586](#)
- EventExposureStartTimestamp, [587](#)
- EventFrameBurstEnd, [587](#)
- EventFrameBurstEndFrameID, [587](#)
- EventFrameBurstEndTimestamp, [587](#)
- EventFrameBurstStart, [587](#)
- EventFrameBurstStartFrameID, [587](#)
- EventFrameBurstStartTimestamp, [588](#)
- EventFrameEnd, [588](#)
- EventFrameEndFrameID, [588](#)
- EventFrameEndTimestamp, [588](#)
- EventFrameStart, [588](#)
- EventFrameStartFrameID, [588](#)
- EventFrameStartTimestamp, [589](#)
- EventFrameTransferEnd, [589](#)
- EventFrameTransferEndFrameID, [589](#)
- EventFrameTransferEndTimestamp, [589](#)
- EventFrameTransferStart, [589](#)
- EventFrameTransferStartFrameID, [589](#)
- EventFrameTransferStartTimestamp, [590](#)
- EventFrameTrigger, [590](#)
- EventFrameTriggerFrameID, [590](#)
- EventFrameTriggerTimestamp, [590](#)
- EventLine0AnyEdge, [590](#)
- EventLine0AnyEdgeFrameID, [590](#)
- EventLine0AnyEdgeTimestamp, [591](#)
- EventLine0FallingEdge, [591](#)
- EventLine0FallingEdgeFrameID, [591](#)
- EventLine0FallingEdgeTimestamp, [591](#)
- EventLine0RisingEdge, [591](#)
- EventLine0RisingEdgeFrameID, [591](#)
- EventLine0RisingEdgeTimestamp, [592](#)
- EventLine1AnyEdge, [592](#)
- EventLine1AnyEdgeFrameID, [592](#)
- EventLine1AnyEdgeTimestamp, [592](#)
- EventLine1FallingEdge, [592](#)
- EventLine1FallingEdgeFrameID, [592](#)
- EventLine1FallingEdgeTimestamp, [593](#)
- EventLine1RisingEdge, [593](#)
- EventLine1RisingEdgeFrameID, [593](#)
- EventLine1RisingEdgeTimestamp, [593](#)
- EventLinkSpeedChange, [593](#)
- EventLinkSpeedChangeFrameID, [593](#)
- EventLinkSpeedChangeTimestamp, [594](#)
- EventLinkTrigger0, [594](#)
- EventLinkTrigger0FrameID, [594](#)
- EventLinkTrigger0Timestamp, [594](#)
- EventLinkTrigger1, [594](#)
- EventLinkTrigger1FrameID, [594](#)
- EventLinkTrigger1Timestamp, [595](#)
- EventNotification, [595](#)
- EventSelector, [595](#)
- EventSequencerSetChange, [595](#)
- EventSequencerSetChangeFrameID, [595](#)
- EventSequencerSetChangeTimestamp, [595](#)
- EventSerialData, [596](#)
- EventSerialDataLength, [596](#)
- EventSerialPortReceive, [596](#)
- EventSerialPortReceiveTimestamp, [596](#)
- EventSerialReceiveOverflow, [596](#)
- EventStream0TransferBlockEnd, [596](#)
- EventStream0TransferBlockEndFrameID, [597](#)
- EventStream0TransferBlockEndTimestamp, [597](#)
- EventStream0TransferBlockStart, [597](#)
- EventStream0TransferBlockStartFrameID, [597](#)
- EventStream0TransferBlockStartTimestamp, [597](#)
- EventStream0TransferBlockTrigger, [597](#)
- EventStream0TransferBlockTriggerFrameID, [598](#)
- EventStream0TransferBlockTriggerTimestamp, [598](#)
- EventStream0TransferBurstEnd, [598](#)
- EventStream0TransferBurstEndFrameID, [598](#)
- EventStream0TransferBurstEndTimestamp, [598](#)
- EventStream0TransferBurstStart, [598](#)
- EventStream0TransferBurstStartFrameID, [599](#)
- EventStream0TransferBurstStartTimestamp, [599](#)
- EventStream0TransferEnd, [599](#)
- EventStream0TransferEndFrameID, [599](#)
- EventStream0TransferEndTimestamp, [599](#)
- EventStream0TransferOverflow, [599](#)
- EventStream0TransferOverflowFrameID, [600](#)
- EventStream0TransferOverflowTimestamp, [600](#)
- EventStream0TransferPause, [600](#)
- EventStream0TransferPauseFrameID, [600](#)
- EventStream0TransferPauseTimestamp, [600](#)
- EventStream0TransferResume, [600](#)
- EventStream0TransferResumeFrameID, [601](#)
- EventStream0TransferResumeTimestamp, [601](#)
- EventStream0TransferStart, [601](#)
- EventStream0TransferStartFrameID, [601](#)
- EventStream0TransferStartTimestamp, [601](#)
- EventTest, [601](#)
- EventTestTimestamp, [602](#)
- EventTimer0End, [602](#)
- EventTimer0EndFrameID, [602](#)
- EventTimer0EndTimestamp, [602](#)
- EventTimer0Start, [602](#)
- EventTimer0StartFrameID, [602](#)
- EventTimer0StartTimestamp, [603](#)
- EventTimer1End, [603](#)
- EventTimer1EndFrameID, [603](#)
- EventTimer1EndTimestamp, [603](#)
- EventTimer1Start, [603](#)
- EventTimer1StartFrameID, [603](#)
- EventTimer1StartTimestamp, [604](#)
- ExposureActiveMode, [604](#)
- ExposureAuto, [604](#)
- ExposureMode, [604](#)
- ExposureTime, [604](#)
- ExposureTimeMode, [604](#)
- ExposureTimeSelector, [605](#)

FactoryReset, 605
FileAccessBuffer, 605
FileAccessLength, 605
FileAccessOffset, 605
FileOpenMode, 605
FileOperationExecute, 606
FileOperationResult, 606
FileOperationSelector, 606
FileOperationStatus, 606
FileSelector, 606
FileSize, 607
Gain, 607
GainAuto, 607
GainAutoBalance, 607
GainSelector, 608
Gamma, 608
GammaEnable, 608
GevActiveLinkCount, 608
GevCCP, 608
GevCurrentDefaultGateway, 608
GevCurrentIPAddress, 609
GevCurrentIPConfigurationDHCP, 609
GevCurrentIPConfigurationLLA, 609
GevCurrentIPConfigurationPersistentIP, 609
GevCurrentPhysicalLinkConfiguration, 609
GevCurrentSubnetMask, 609
GevDiscoveryAckDelay, 610
GevFirstURL, 610
GevGVCPExtendedStatusCodes, 610
GevGVCPExtendedStatusCodesSelector, 610
GevGVCPHeartbeatDisable, 610
GevGVCPPendingAck, 610
GevGVCPPendingTimeout, 611
GevGVSPExtendedIDMode, 611
GevHeartbeatTimeout, 611
GevIEEE1588, 611
GevIEEE1588ClockAccuracy, 611
GevIEEE1588Mode, 611
GevIEEE1588Status, 612
GevIPConfigurationStatus, 612
GevInterfaceSelector, 612
GevMACAddress, 612
GevMCDA, 612
GevMCPHostPort, 612
GevMCRC, 613
GevMCSP, 613
GevMCTT, 613
GevNumberOfInterfaces, 613
GevPAUSEFrameReception, 613
GevPAUSEFrameTransmission, 613
GevPersistentDefaultGateway, 614
GevPersistentIPAddress, 614
GevPersistentSubnetMask, 614
GevPhysicalLinkConfiguration, 614
GevPrimaryApplicationIPAddress, 614
GevPrimaryApplicationSocket, 614
GevPrimaryApplicationSwitchoverKey, 615
GevSCCFGAllInTransmission, 615
GevSCCFGExtendedChunkData, 615
GevSCCFGPacketResendDestination, 615
GevSCCFGUnconditionalStreaming, 615
GevSCDA, 615
GevSCPDDirection, 616
GevSCPHostPort, 616
GevSCPInterfaceIndex, 616
GevSCPSBigEndian, 616
GevSCPSDoNotFragment, 616
GevSCPSFireTestPacket, 617
GevSCPSPacketSize, 617
GevSCPD, 616
GevSCSP, 617
GevSCZoneConfigurationLock, 617
GevSCZoneCount, 617
GevSCZoneDirectionAll, 617
GevSecondURL, 618
GevStreamChannelSelector, 618
GevSupportedOption, 618
GevSupportedOptionSelector, 618
GevTimestampTickFrequency, 618
GuiXmlManifestAddress, 618
Height, 619
HeightMax, 619
ImageComponentEnable, 619
ImageComponentSelector, 619
ImageCompressionBitrate, 619
ImageCompressionJPEGFormatOption, 619
ImageCompressionMode, 620
ImageCompressionQuality, 620
ImageCompressionRateOption, 620
Init, 529
IspEnable, 620
LUTEnable, 624
LUTIndex, 624
LUTSelector, 624
LUTValue, 624
LUTValueAll, 625
LineFilterWidth, 620
LineFormat, 621
LineInputFilterSelector, 621
LineInverter, 621
LineMode, 621
LinePitch, 621
LineSelector, 621
LineSource, 622
LineStatus, 622
LineStatusAll, 622
LinkErrorCount, 622
LinkUptime, 622
LogicBlockLUTInputActivation, 622
LogicBlockLUTInputSelector, 623
LogicBlockLUTInputSource, 623
LogicBlockLUTOutputValue, 623
LogicBlockLUTOutputValueAll, 623
LogicBlockLUTRowIndex, 623
LogicBlockLUTSelector, 623
LogicBlockSelector, 624

MaxDeviceResetTime, [625](#)
OffsetX, [625](#)
OffsetY, [625](#)
PacketResendRequestCount, [625](#)
PayloadSize, [626](#)
PixelColorFilter, [626](#)
PixelDynamicRangeMax, [626](#)
PixelDynamicRangeMin, [626](#)
PixelFormat, [626](#)
PixelFormatInfoID, [627](#)
PixelFormatInfoSelector, [627](#)
PixelSize, [627](#)
PowerSupplyCurrent, [627](#)
PowerSupplyVoltage, [627](#)
RegionDestination, [627](#)
RegionMode, [628](#)
RegionSelector, [628](#)
ReverseX, [628](#)
ReverseY, [628](#)
RgbTransformLightSource, [628](#)
Saturation, [629](#)
SaturationEnable, [629](#)
Scan3dAxisMax, [629](#)
Scan3dAxisMin, [629](#)
Scan3dCoordinateOffset, [629](#)
Scan3dCoordinateReferenceSelector, [630](#)
Scan3dCoordinateReferenceValue, [630](#)
Scan3dCoordinateScale, [630](#)
Scan3dCoordinateSelector, [630](#)
Scan3dCoordinateSystem, [630](#)
Scan3dCoordinateSystemReference, [630](#)
Scan3dCoordinateTransformSelector, [631](#)
Scan3dDistanceUnit, [631](#)
Scan3dInvalidDataFlag, [631](#)
Scan3dInvalidDataValue, [631](#)
Scan3dOutputMode, [631](#)
Scan3dTransformValue, [631](#)
SensorDescription, [632](#)
SensorDigitizationTaps, [632](#)
SensorHeight, [632](#)
SensorShutterMode, [632](#)
SensorTaps, [632](#)
SensorWidth, [632](#)
SequencerConfigurationMode, [633](#)
SequencerConfigurationValid, [633](#)
SequencerFeatureEnable, [633](#)
SequencerMode, [633](#)
SequencerPathSelector, [633](#)
SequencerSetActive, [634](#)
SequencerSetLoad, [634](#)
SequencerSetNext, [634](#)
SequencerSetSave, [634](#)
SequencerSetSelector, [634](#)
SequencerSetStart, [635](#)
SequencerSetValid, [635](#)
SequencerTriggerActivation, [635](#)
SequencerTriggerSource, [635](#)
SerialPortBaudRate, [635](#)
SerialPortDataBits, [636](#)
SerialPortParity, [636](#)
SerialPortSelector, [636](#)
SerialPortSource, [636](#)
SerialPortStopBits, [636](#)
SerialReceiveFramingErrorCount, [636](#)
SerialReceiveParityErrorCount, [637](#)
SerialReceiveQueueClear, [637](#)
SerialReceiveQueueCurrentCharacterCount, [637](#)
SerialReceiveQueueMaxCharacterCount, [637](#)
SerialTransmitQueueCurrentCharacterCount, [637](#)
SerialTransmitQueueMaxCharacterCount, [637](#)
Sharpening, [638](#)
SharpeningAuto, [638](#)
SharpeningEnable, [638](#)
SharpeningThreshold, [638](#)
SoftwareSignalPulse, [639](#)
SoftwareSignalSelector, [639](#)
SourceCount, [639](#)
SourceSelector, [639](#)
TLParamsLocked, [642](#)
Test0001, [639](#)
TestEventGenerate, [640](#)
TestPattern, [640](#)
TestPatternGeneratorSelector, [640](#)
TestPendingAck, [640](#)
TimerDelay, [640](#)
TimerDuration, [641](#)
TimerReset, [641](#)
TimerSelector, [641](#)
TimerStatus, [641](#)
TimerTriggerActivation, [641](#)
TimerTriggerSource, [641](#)
TimerValue, [642](#)
Timestamp, [642](#)
TimestampLatch, [642](#)
TimestampLatchValue, [642](#)
TimestampReset, [642](#)
TransferAbort, [643](#)
TransferBlockCount, [643](#)
TransferBurstCount, [643](#)
TransferComponentSelector, [643](#)
TransferControlMode, [643](#)
TransferOperationMode, [643](#)
TransferPause, [644](#)
TransferQueueCurrentBlockCount, [644](#)
TransferQueueMaxBlockCount, [644](#)
TransferQueueMode, [644](#)
TransferQueueOverflowCount, [644](#)
TransferResume, [644](#)
TransferSelector, [645](#)
TransferStart, [645](#)
TransferStatus, [645](#)
TransferStatusSelector, [645](#)
TransferStop, [645](#)
TransferStreamChannel, [645](#)
TransferTriggerActivation, [646](#)
TransferTriggerMode, [646](#)

- TransferTriggerSelector, [646](#)
- TransferTriggerSource, [646](#)
- TriggerActivation, [646](#)
- TriggerDelay, [646](#)
- TriggerDivider, [647](#)
- TriggerEventTest, [647](#)
- TriggerMode, [647](#)
- TriggerMultiplier, [647](#)
- TriggerOverlap, [647](#)
- TriggerSelector, [648](#)
- TriggerSoftware, [648](#)
- TriggerSource, [648](#)
- UserOutputSelector, [648](#)
- UserOutputValue, [648](#)
- UserOutputValueAll, [649](#)
- UserOutputValueAllMask, [649](#)
- UserSetDefault, [649](#)
- UserSetFeatureEnable, [649](#)
- UserSetLoad, [649](#)
- UserSetSave, [650](#)
- UserSetSelector, [650](#)
- V3_3Enable, [650](#)
- WhiteClip, [650](#)
- WhiteClipSelector, [650](#)
- Width, [651](#)
- WidthMax, [651](#)
- Spinnaker::CameraBase
 - ~CameraBase, [654](#)
 - BeginAcquisition, [655](#)
 - CameraBase, [654](#), [655](#)
 - DelInit, [655](#)
 - DiscoverMaxPacketSize, [655](#)
 - EndAcquisition, [656](#)
 - ForceIP, [656](#)
 - GetAccessMode, [656](#)
 - GetBufferOwnership, [657](#)
 - GetGuiXml, [657](#)
 - GetNextImage, [657](#)
 - GetNodeMap, [658](#)
 - GetNumDataStreams, [658](#)
 - GetNumImagesInUse, [659](#)
 - GetTLDeviceNodeMap, [659](#)
 - GetTLStreamNodeMap, [659](#)
 - GetUniqueID, [660](#)
 - GetUserBufferCount, [660](#)
 - GetUserBufferSize, [660](#)
 - GetUserBufferTotalSize, [661](#)
 - Init, [661](#)
 - InterfaceImpl, [666](#)
 - IsInitialized, [662](#)
 - IsStreaming, [662](#)
 - IsValid, [662](#)
 - operator=, [663](#)
 - ReadPort, [663](#)
 - RegisterEventHandler, [663](#), [664](#)
 - SetBufferOwnership, [664](#)
 - SetUserBuffers, [664](#), [665](#)
 - UnregisterEventHandler, [666](#)
 - WritePort, [666](#)
- Spinnaker::CameraList
 - ~CameraList, [668](#)
 - Append, [669](#)
 - CameraList, [668](#)
 - Clear, [669](#)
 - GetByDeviceID, [669](#)
 - GetByIndex, [670](#)
 - GetBySerial, [670](#)
 - GetSize, [671](#)
 - operator=, [671](#)
 - operator[], [671](#)
 - RemoveByDeviceID, [671](#)
 - RemoveByIndex, [672](#)
 - RemoveBySerial, [672](#)
- Spinnaker::ChunkData
 - ~ChunkData, [730](#)
 - ChunkData, [729](#), [730](#)
 - GetBlackLevel, [730](#)
 - GetCRC, [731](#)
 - GetCompressionMode, [730](#)
 - GetCompressionRatio, [730](#)
 - GetCounterValue, [731](#)
 - GetEncoderValue, [731](#)
 - GetExposureEndLineStatusAll, [731](#)
 - GetExposureTime, [732](#)
 - GetFrameID, [732](#)
 - GetGain, [732](#)
 - GetHeight, [732](#)
 - GetImage, [733](#)
 - GetInferenceBoundingBoxResult, [733](#)
 - GetInferenceConfidence, [733](#)
 - GetInferenceFrameId, [733](#)
 - GetInferenceResult, [734](#)
 - GetLinePitch, [734](#)
 - GetLineStatusAll, [734](#)
 - GetOffsetX, [734](#)
 - GetOffsetY, [735](#)
 - GetPartSelector, [735](#)
 - GetPixelDynamicRangeMax, [735](#)
 - GetPixelDynamicRangeMin, [735](#)
 - GetScan3dAxisMax, [736](#)
 - GetScan3dAxisMin, [736](#)
 - GetScan3dCoordinateOffset, [736](#)
 - GetScan3dCoordinateReferenceValue, [736](#)
 - GetScan3dCoordinateScale, [737](#)
 - GetScan3dInvalidDataValue, [737](#)
 - GetScan3dTransformValue, [737](#)
 - GetScanLineSelector, [737](#)
 - GetSequencerSetActive, [738](#)
 - GetSerialDataLength, [738](#)
 - GetStreamChannelID, [738](#)
 - GetTimerValue, [738](#)
 - GetTimestamp, [739](#)
 - GetTimestampLatchValue, [739](#)
 - GetTransferBlockID, [739](#)
 - GetTransferQueueCurrentBlockCount, [739](#)
 - GetWidth, [740](#)

- SetChunks, [740](#)
- Spinnaker::DeviceArrivalEventHandler
 - ~DeviceArrivalEventHandler, [798](#)
 - DeviceArrivalEventHandler, [797](#)
 - OnDeviceArrival, [798](#)
 - operator=, [798](#)
- Spinnaker::DeviceEventExposureEndData
 - frameID, [799](#)
- Spinnaker::DeviceEventHandler
 - ~DeviceEventHandler, [801](#)
 - DeviceEventHandler, [800](#)
 - GetDeviceEventId, [801](#)
 - GetDeviceEventName, [801](#)
 - OnDeviceEvent, [801](#)
 - operator=, [802](#)
- Spinnaker::DeviceEventInferenceData
 - confidence, [804](#)
 - frameID, [804](#)
 - result, [805](#)
- Spinnaker::DeviceEventUtility
 - ParseDeviceEventExposureEnd, [805](#)
 - ParseDeviceEventInference, [806](#)
- Spinnaker::DeviceRemovalEventHandler
 - ~DeviceRemovalEventHandler, [808](#)
 - DeviceRemovalEventHandler, [807](#)
 - OnDeviceRemoval, [808](#)
 - operator=, [808](#)
- Spinnaker::EventHandler
 - ~EventHandler, [833](#)
 - EventHandler, [833](#)
 - EventProcessor, [834](#)
 - GetEventPayloadData, [833](#)
 - GetEventPayloadDataSize, [833](#)
 - GetEventType, [833](#)
 - IDataStream, [834](#)
 - m_pEventData, [835](#)
 - operator=, [834](#)
 - SetEventPayload, [834](#)
 - SetEventType, [834](#)
 - Stream, [835](#)
- Spinnaker::Exception
 - ~Exception, [839](#)
 - Exception, [838](#), [839](#)
 - GetBuildDate, [839](#)
 - GetBuildTime, [839](#)
 - GetError, [839](#)
 - GetErrorMessage, [839](#)
 - GetFileName, [840](#)
 - GetFullErrorMessage, [840](#)
 - GetFunctionName, [840](#)
 - GetLineNumber, [840](#)
 - operator!=, [840](#)
 - operator=, [840](#)
 - operator==, [841](#)
 - what, [841](#)
- Spinnaker::GenApi, [460](#)
 - COMMAND_MAGIC, [476](#)
 - GENCP_COMMAND_HEADER_SIZE, [476](#)
 - GENCP_EVENT_BASIC_SIZE, [476](#)
 - GENCP_EVENT_CMD_ID, [477](#)
 - GVCP_MESSAGE_TAGS, [475](#)
 - IDevFileStream, [475](#)
 - IPersistScript, [477](#)
 - ODevFileStream, [475](#)
 - PersistFeature, [476](#)
 - SET_GUID, [476](#)
 - U3V_EVENT_PREFIX, [477](#)
- Spinnaker::GenApi::AutoLock
 - ~AutoLock, [489](#)
 - AutoLock, [489](#)
- Spinnaker::GenApi::BooleanNode
 - ~BooleanNode, [497](#)
 - BooleanNode, [497](#)
 - GetValue, [497](#)
 - operator=, [498](#)
 - SetReference, [498](#)
 - SetValue, [498](#)
- Spinnaker::GenApi::CChunkAdapter
 - ~CChunkAdapter, [677](#)
 - AttachBuffer, [677](#)
 - AttachNodeMap, [677](#)
 - CChunkAdapter, [677](#)
 - CheckBufferLayout, [678](#)
 - ClearCaches, [678](#)
 - DetachBuffer, [678](#)
 - DetachNodeMap, [678](#)
 - m_pChunkAdapter, [679](#)
 - UpdateBuffer, [678](#)
- Spinnaker::GenApi::CChunkAdapterDcam
 - ~CChunkAdapterDcam, [680](#)
 - AttachBuffer, [680](#)
 - CChunkAdapterDcam, [680](#)
 - CheckBufferLayout, [681](#)
 - CheckCRC, [681](#)
 - HasCRC, [681](#)
- Spinnaker::GenApi::CChunkAdapterGEV
 - ~CChunkAdapterGEV, [685](#)
 - AttachBuffer, [685](#)
 - CChunkAdapterGEV, [685](#)
 - CheckBufferLayout, [686](#)
- Spinnaker::GenApi::CChunkAdapterGeneric
 - ~CChunkAdapterGeneric, [683](#)
 - AttachBuffer, [683](#)
 - CChunkAdapterGeneric, [682](#)
 - CheckBufferLayout, [683](#)
- Spinnaker::GenApi::CChunkAdapterU3V
 - ~CChunkAdapterU3V, [687](#)
 - AttachBuffer, [688](#)
 - CChunkAdapterU3V, [687](#)
 - CheckBufferLayout, [688](#)
- Spinnaker::GenApi::CChunkPort
 - ~CChunkPort, [690](#)
 - AttachChunk, [691](#)
 - AttachPort, [691](#)
 - CChunkPort, [690](#)
 - CheckChunkID, [691](#)

- ClearCache, 691
- DetachChunk, 692
- DetachPort, 692
- GetAccessMode, 692
- GetChunkIDLength, 692
- GetPrincipalInterfaceType, 692
- GetSwapEndianness, 692
- InvalidateNode, 693
- m_pChunkPort, 694
- m_pPort, 694
- m_pPortAdapter, 694
- Read, 693
- SetPortImpl, 693
- UpdateBuffer, 693
- Write, 693
- Spinnaker::GenApi::CEnumerationTRef
 - ~CEnumerationTRef, 698
 - CEnumerationTRef, 698
 - GetCurrentEntry, 698
 - GetEntry, 699
 - GetValue, 699
 - operator(), 699
 - operator=, 700
 - SetEnumReference, 700
 - SetNumEnums, 700
 - SetReference, 700
 - SetValue, 701
- Spinnaker::GenApi::CEventAdapter
 - ~CEventAdapter, 702
 - AttachNodeMap, 702
 - CEventAdapter, 702
 - DeliverMessage, 702
 - DetachNodeMap, 703
 - m_pEventAdapter, 703
- Spinnaker::GenApi::CEventAdapter1394
 - ~CEventAdapter1394, 704
 - CEventAdapter1394, 704
 - DeliverEventMessage, 705
 - DeliverMessage, 705
- Spinnaker::GenApi::CEventAdapterGEV
 - ~CEventAdapterGEV, 709
 - CEventAdapterGEV, 709
 - DeliverEventMessage, 709
 - DeliverMessage, 709
- Spinnaker::GenApi::CEventAdapterGeneric
 - ~CEventAdapterGeneric, 706
 - CEventAdapterGeneric, 706
 - DeliverMessage, 707
- Spinnaker::GenApi::CEventAdapterU3V
 - ~CEventAdapterU3V, 711
 - CEventAdapterU3V, 711
 - DeliverEventMessage, 711
 - DeliverMessage, 711
- Spinnaker::GenApi::CEventPort
 - ~CEventPort, 714
 - AttachEvent, 714
 - AttachNode, 714
 - CEventPort, 713
 - CheckEventID, 714
 - DetachEvent, 715
 - DetachNode, 715
 - GetAccessMode, 715
 - GetEventIDLength, 715
 - GetPrincipalInterfaceType, 715
 - GetSwapEndianness, 715
 - InvalidateNode, 716
 - m_pEventPort, 716
 - m_pNode, 717
 - m_pPortAdapter, 717
 - Read, 716
 - SetPortImpl, 716
 - Write, 716
- Spinnaker::GenApi::CFeatureBag
 - ~CFeatureBag, 718
 - CFeatureBag, 718
 - GetFeatureBagHandle, 718
 - LoadFromBag, 718
 - operator==, 719
 - PersistFeature, 719
 - SetInfo, 719
 - StoreToBag, 719
- Spinnaker::GenApi::CFloatPtr
 - CFloatPtr, 721
 - GetEnumAlias, 721
 - GetIntAlias, 721
 - operator=, 721
- Spinnaker::GenApi::CGeneric_XMLLoaderParams
 - _Initialize, 722
- Spinnaker::GenApi::CLock
 - ~CLock, 743
 - CLock, 743
 - Lock, 743
 - m_bOwnLock, 744
 - m_lock, 744
 - NodeMap, 744
 - TryLock, 744
 - Unlock, 744
- Spinnaker::GenApi::CLockEx
 - m_lockEx, 747
- Spinnaker::GenApi::CNodeCallback
 - ~CNodeCallback, 748
 - CNodeCallback, 748
 - Destroy, 748
 - GetCallbackType, 748
 - GetNode, 749
 - m_CallbackType, 749
 - m_pNode, 749
 - operator(), 749
- Spinnaker::GenApi::CNodeMapFactory
 - ~CNodeMapFactory, 752
 - AddInjectionData, 754
 - ApplyStyleSheet, 754
 - CNodeMapFactory, 752, 753
 - ClearCache, 754
 - CreateEmptyNodeMap, 755
 - CreateNodeDataFromNodeMap, 755

- CreateNodeMap, 755
- ExtractSubtree, 755
- GetNodeStatistics, 756
- GetSupportedSchemaVersions, 756
- IsCameraDescriptionFileDataReleased, 756
- IsEmpty, 756
- IsLoaded, 756
- IsPreprocessed, 757
- LoadAndInject, 757
- operator=, 757
- Preprocess, 757
- ReleaseCameraDescriptionFileData, 757
- ToString, 758
- ToXml, 758
- Spinnaker::GenApi::CNodeMapFactory::NodeStatistics↔
 - _t
 - NumLinks, 1105
 - NumNodes, 1105
 - NumProperties, 1105
 - NumStrings, 1105
- Spinnaker::GenApi::CNodeMapRef
 - CNodeMapRef, 759, 760
 - operator=, 760
- Spinnaker::GenApi::CNodeMapRefT
 - _ClearXMLCache, 763
 - _Connect, 763
 - _GetDeviceName, 763
 - _GetNode, 763
 - _GetNodes, 764
 - _GetSupportedSchemaVersions, 764
 - _InvalidateNodes, 764
 - _LoadXMLFromFile, 764
 - _LoadXMLFromFileInject, 764
 - _LoadXMLFromString, 765
 - _LoadXMLFromStringInject, 765
 - _LoadXMLFromZIPData, 765
 - _LoadXMLFromZIPFile, 765
 - _Poll, 765
 - _Ptr, 766
- Spinnaker::GenApi::CPointer
 - ~CPointer, 774
 - CPointer, 774
 - IsValid, 775
 - m_pT, 777
 - operator bool, 775
 - operator T*, 775
 - operator!=, 775, 776
 - operator*, 776
 - operator(), 776
 - operator->, 776
 - operator=, 776
 - operator==, 777
- Spinnaker::GenApi::CPortImpl
 - ~CPortImpl, 779
 - CPortImpl, 779
 - GetAccessMode, 779
 - GetSwapEndianness, 779
 - InvalidateNode, 780
 - m_ptrPort, 781
 - Read, 780
 - Replay, 780
 - SetPortImpl, 780
 - Write, 780
- Spinnaker::GenApi::CPortWriteList
 - ~CPortWriteList, 782
 - CPortWriteList, 782
 - GetCookie, 783
 - GetPortWriteListHandle, 783
 - m_pWriteList, 784
 - Replay, 783
 - SetCookie, 783
 - Write, 783
- Spinnaker::GenApi::CRegisterPortImpl
 - ~CRegisterPortImpl, 786
 - CRegisterPortImpl, 786
 - GetAccessMode, 786
 - Read, 787
 - ReadRegister, 787
 - SetPortImpl, 787
 - Write, 787
 - WriteRegister, 788
- Spinnaker::GenApi::CSelectorSet
 - ~CSelectorSet, 790
 - CSelectorSet, 789
 - GetSelectorList, 790
 - IsEmpty, 790
 - Restore, 790
 - SetFirst, 790
 - SetNext, 790
 - ToString, 791
- Spinnaker::GenApi::CTestPortStruct
 - CTestPortStruct, 793
 - GetAccessMode, 793
 - GetNumReads, 793
 - GetNumWrites, 793
 - GetPrincipalInterfaceType, 793
 - m_BaseAddress, 794
 - m_NumReads, 794
 - m_NumWrites, 795
 - MemSet, 793
 - Read, 794
 - ResetStatistics, 794
 - Write, 794
- Spinnaker::GenApi::CategoryNode
 - ~CategoryNode, 675
 - CategoryNode, 675
 - GetFeatures, 675
 - SetReference, 676
- Spinnaker::GenApi::CommandNode
 - ~CommandNode, 768
 - CommandNode, 767, 768
 - Execute, 768
 - IsDone, 768
 - operator(), 769
 - SetReference, 769
- Spinnaker::GenApi::Counter

- Counter, [771](#)
- GetValue, [771](#)
- IsZero, [772](#)
- operator unsigned int, [772](#)
- operator++, [772](#)
- operator--, [772](#)
- Spinnaker::GenApi::EAccessModeClass
 - FromString, [811](#)
 - ToString, [812](#)
- Spinnaker::GenApi::ECachingModeClass
 - FromString, [813](#)
 - ToString, [813](#)
- Spinnaker::GenApi::EDisplayNotationClass
 - FromString, [814](#)
 - ToString, [814](#)
- Spinnaker::GenApi::EEndianessClass
 - FromString, [815](#)
 - ToString, [815](#)
- Spinnaker::GenApi::EGenApiSchemaVersionClass
 - FromString, [816](#)
 - ToString, [816](#)
- Spinnaker::GenApi::EInputDirectionClass
 - FromString, [817](#)
 - ToString, [817](#)
- Spinnaker::GenApi::ENameSpaceClass
 - FromString, [818](#)
 - ToString, [818](#)
- Spinnaker::GenApi::ERepresentationClass
 - FromString, [827](#)
 - ToString, [828](#)
- Spinnaker::GenApi::ESignClass
 - FromString, [829](#)
 - ToString, [829](#)
- Spinnaker::GenApi::ESlopeClass
 - FromString, [830](#)
 - ToString, [830](#)
- Spinnaker::GenApi::EStandardNameSpaceClass
 - FromString, [831](#)
 - ToString, [831](#)
- Spinnaker::GenApi::EVisibilityClass
 - FromString, [835](#)
 - ToString, [836](#)
- Spinnaker::GenApi::EYesNoClass
 - FromString, [842](#)
 - ToString, [842](#)
- Spinnaker::GenApi::EnumEntryNode
 - ~EnumEntryNode, [820](#)
 - EnumEntryNode, [820](#)
 - GetNumericValue, [820](#)
 - GetSymbolic, [821](#)
 - GetValue, [821](#)
 - IsSelfClearing, [821](#)
 - SetReference, [821](#)
- Spinnaker::GenApi::EnumNode
 - ~EnumNode, [824](#)
 - EnumNode, [824](#)
 - GetCurrentEntry, [824](#)
 - GetEntries, [824](#)
 - GetEntry, [825](#)
 - GetEntryByName, [825](#)
 - GetIntValue, [825](#)
 - GetSymbolics, [825](#)
 - m_pEnumeration, [827](#)
 - operator*, [826](#)
 - operator=, [826](#)
 - SetIntValue, [826](#)
 - SetReference, [826](#)
- Spinnaker::GenApi::FileProtocolAdapter
 - ~FileProtocolAdapter, [843](#)
 - attach, [843](#)
 - closeFile, [844](#)
 - deleteFile, [844](#)
 - FileProtocolAdapter, [843](#)
 - getBufSize, [844](#)
 - openFile, [845](#)
 - read, [845](#)
 - write, [846](#)
- Spinnaker::GenApi::FloatNode
 - ~FloatNode, [849](#)
 - FloatNode, [849](#)
 - GetDisplayNotation, [849](#)
 - GetDisplayPrecision, [849](#)
 - GetEnumAlias, [849](#)
 - GetInc, [849](#)
 - GetIncMode, [850](#)
 - GetIntAlias, [850](#)
 - GetListOfValidValues, [850](#)
 - GetMax, [850](#)
 - GetMin, [850](#)
 - GetRepresentation, [850](#)
 - GetUnit, [851](#)
 - GetValue, [851](#)
 - HasInc, [851](#)
 - ImposeMax, [851](#)
 - ImposeMin, [851](#)
 - operator*, [852](#)
 - operator(), [852](#)
 - operator=, [852](#)
 - SetReference, [852](#)
 - SetValue, [852](#)
- Spinnaker::GenApi::FloatRegNode
 - ~FloatRegNode, [855](#)
 - FloatRegNode, [854](#), [855](#)
 - SetReference, [855](#)
- Spinnaker::GenApi::Function_NodeCallback
 - Destroy, [857](#)
 - Function_NodeCallback, [857](#)
 - operator(), [857](#)
- Spinnaker::GenApi::IDevFileStreamBase
 - close, [910](#)
 - filebuf_type, [909](#)
 - ios_type, [909](#)
 - is_open, [910](#)
 - istream_type, [909](#)
 - open, [910](#)
 - rdbuf, [910](#)

- Spinnaker::GenApi::IDevFileStreamBuf
 - ~IDevFileStreamBuf, [912](#)
 - close, [912](#)
 - IDevFileStreamBuf, [912](#)
 - is_open, [912](#)
 - open, [912](#)
 - pbackfail, [912](#)
 - underflow, [912](#)
- Spinnaker::GenApi::IntRegNode
 - ~IntRegNode, [1056](#)
 - IntRegNode, [1055](#), [1056](#)
 - SetReference, [1056](#)
- Spinnaker::GenApi::IntegerNode
 - ~IntegerNode, [1027](#)
 - GetFloatAlias, [1027](#)
 - GetInc, [1028](#)
 - GetIncMode, [1028](#)
 - GetListOfValidValues, [1028](#)
 - GetMax, [1028](#)
 - GetMin, [1028](#)
 - GetRepresentation, [1028](#)
 - GetUnit, [1029](#)
 - GetValue, [1029](#)
 - ImposeMax, [1029](#)
 - ImposeMin, [1029](#)
 - IntegerNode, [1027](#)
 - operator*, [1030](#)
 - operator(), [1029](#)
 - operator=, [1030](#)
 - SetReference, [1030](#)
 - SetValue, [1030](#)
- Spinnaker::GenApi::Member_NodeCallback
 - Destroy, [1083](#)
 - Member_NodeCallback, [1083](#)
 - operator(), [1083](#)
 - PMEMBERFUNC, [1082](#)
- Spinnaker::GenApi::Node
 - ~Node, [1088](#)
 - DeregisterCallback, [1088](#)
 - GetAccessMode, [1088](#)
 - GetAlias, [1089](#)
 - GetCachingMode, [1089](#)
 - GetCastAlias, [1089](#)
 - GetChildren, [1089](#)
 - GetDescription, [1090](#)
 - GetDeviceName, [1090](#)
 - GetDisplayName, [1090](#)
 - GetDocuURL, [1090](#)
 - GetEventID, [1090](#)
 - GetName, [1090](#)
 - GetNamespace, [1090](#)
 - GetNodeHandle, [1091](#)
 - GetNodeMap, [1091](#)
 - GetParents, [1091](#)
 - GetPollingTime, [1091](#)
 - GetPrincipalInterfaceType, [1091](#)
 - GetProperty, [1092](#)
 - GetPropertyNames, [1092](#)
 - GetSelectedFeatures, [1092](#)
 - GetSelectingFeatures, [1092](#)
 - GetToolTip, [1092](#)
 - GetVisibility, [1093](#)
 - ImposeAccessMode, [1093](#)
 - ImposeVisibility, [1093](#)
 - InvalidateNode, [1093](#)
 - IsAccessModeCacheable, [1093](#)
 - IsCachable, [1093](#)
 - IsDeprecated, [1094](#)
 - IsFeature, [1094](#)
 - IsSelector, [1094](#)
 - IsStreamable, [1094](#)
 - m_Callbacks, [1096](#)
 - m_pNodeData, [1096](#)
 - m_pNodeMap, [1096](#)
 - Node, [1088](#)
 - operator!=, [1094](#)
 - operator==, [1094](#)
 - RegisterCallback, [1095](#)
 - SetNodeHandle, [1095](#)
 - SetNodeMap, [1095](#)
 - SetReference, [1095](#)
- Spinnaker::GenApi::NodeMap
 - _Ptr, [1104](#)
 - ~NodeMap, [1099](#)
 - ClearXMLCache, [1099](#)
 - Connect, [1099](#)
 - Destroy, [1099](#)
 - GetDeviceName, [1100](#)
 - GetDeviceVersion, [1100](#)
 - GetGenApiVersion, [1100](#)
 - GetLock, [1100](#)
 - GetModelName, [1100](#)
 - GetNode, [1100](#)
 - GetNodeMapHandle, [1101](#)
 - GetNodes, [1101](#)
 - GetNumNodes, [1101](#)
 - GetProductGuid, [1101](#)
 - GetSchemaVersion, [1101](#)
 - GetStandardNameSpace, [1101](#)
 - GetSupportedSchemaVersions, [1102](#)
 - GetToolTip, [1102](#)
 - GetVendorName, [1102](#)
 - GetVersionGuid, [1102](#)
 - InvalidateNodes, [1103](#)
 - LoadXMLFromFile, [1103](#)
 - LoadXMLFromFileInject, [1103](#)
 - LoadXMLFromString, [1103](#)
 - LoadXMLFromStringInject, [1103](#)
 - LoadXMLFromZIPData, [1104](#)
 - LoadXMLFromZIPFile, [1104](#)
 - NodeMap, [1098](#)
 - Poll, [1104](#)
- Spinnaker::GenApi::ODevFileStreamBase
 - close, [1107](#)
 - filebuf_type, [1107](#)
 - ios_type, [1107](#)

- is_open, 1107
- open, 1107
- ostream_type, 1107
- rdbuf, 1108
- Spinnaker::GenApi::ODevFileStreamBuf
 - ~ODevFileStreamBuf, 1109
 - close, 1109
 - is_open, 1109
 - ODevFileStreamBuf, 1109
 - open, 1109
 - overflow, 1110
 - sync, 1110
 - xspn, 1110
- Spinnaker::GenApi::PortNode
 - ~PortNode, 1115
 - CacheChunkData, 1115
 - GetChunkID, 1115
 - GetPortHandle, 1116
 - GetSwapEndianness, 1116
 - PortNode, 1115
 - Read, 1116
 - Replay, 1116
 - SetPortImpl, 1116
 - SetReference, 1117
 - StartRecording, 1117
 - StopRecording, 1117
 - Write, 1118
- Spinnaker::GenApi::PortRecorder
 - ~PortRecorder, 1120
 - GetAccessMode, 1120
 - PortRecorder, 1120
 - Read, 1120
 - Replay, 1120
 - SetReference, 1121
 - StartRecording, 1121
 - StopRecording, 1121
 - Write, 1121
- Spinnaker::GenApi::PortReplay
 - ~PortReplay, 1124
 - GetAccessMode, 1124
 - GetPortReplayHandle, 1124
 - PortReplay, 1123
 - Read, 1124
 - Replay, 1124
 - SetReference, 1125
 - Write, 1125
- Spinnaker::GenApi::RegisterNode
 - ~RegisterNode, 1129
 - Get, 1129
 - GetAddress, 1129
 - GetLength, 1130
 - RegisterNode, 1128, 1129
 - Set, 1130
 - SetReference, 1130
- Spinnaker::GenApi::StringNode
 - ~StringNode, 1138
 - GetMaxLength, 1139
 - GetValue, 1139
 - operator*, 1139
 - operator(), 1139
 - operator=, 1139
 - SetReference, 1140
 - SetValue, 1140
 - StringNode, 1138
- Spinnaker::GenApi::StringRegNode
 - ~StringRegNode, 1143
 - SetReference, 1143
 - StringRegNode, 1142, 1143
- Spinnaker::GenApi::ValueNode
 - ~ValueNode, 1207
 - FromString, 1207
 - GetNode, 1207
 - IsValueCacheValid, 1207
 - SetReference, 1208
 - ToString, 1208
 - ValueNode, 1206, 1207
- Spinnaker::GenApi::double_autovector_t
 - _pCount, 811
 - _pv, 811
 - ~double_autovector_t, 809
 - double_autovector_t, 809
 - operator delete, 810
 - operator new, 810
 - operator=, 810
 - operator[], 810
 - size, 810
- Spinnaker::GenApi::int64_autovector_t
 - _pCount, 1024
 - _pv, 1025
 - ~int64_autovector_t, 1023
 - int64_autovector_t, 1023
 - operator delete, 1023
 - operator new, 1024
 - operator=, 1024
 - operator[], 1024
 - size, 1024
- Spinnaker::GenICam, 477
 - getline, 478, 479
 - ThrowBadAlloc, 479
- Spinnaker::GenICam::AutoLock
 - ~AutoLock, 488
 - AutoLock, 488
- Spinnaker::GenICam::CGlobalLock
 - ~CGlobalLock, 724
 - CGlobalLock, 723, 724
 - IsValid, 724
 - Lock, 724
 - m_DebugCount, 725
 - TryLock, 724
 - Unlock, 725
- Spinnaker::GenICam::CGlobalLockUnlocker
 - ~CGlobalLockUnlocker, 726
 - CGlobalLockUnlocker, 726
 - m_Lock, 727
 - m_enabled, 727
 - UnlockEarly, 726

- Spinnaker::GenICam::CLock
 - ~CLock, [741](#)
 - CLock, [741](#)
 - Lock, [741](#)
 - TryLock, [741](#)
 - Unlock, [742](#)
- Spinnaker::GenICam::LockableObject
 - GetLock, [1072](#)
 - Lock, [1072](#)
 - m_Lock, [1072](#)
- Spinnaker::GenICam::LockableObject::Lock
 - ~Lock, [1070](#)
 - Lock, [1070](#)
- Spinnaker::GenICam::Version_t
 - Major, [1209](#)
 - Minor, [1209](#)
 - SubMinor, [1209](#)
- Spinnaker::GenICam::gcstring
 - _npos, [860](#)
 - ~gcstring, [860](#)
 - append, [860](#)
 - assign, [860](#), [861](#)
 - c_str, [861](#)
 - compare, [861](#)
 - empty, [861](#)
 - find, [861](#), [862](#)
 - find_first_not_of, [862](#)
 - find_first_of, [862](#)
 - gcstring, [859](#)
 - length, [863](#)
 - max_size, [863](#)
 - npos, [867](#)
 - operator const char *, [863](#)
 - operator delete, [863](#)
 - operator new, [863](#)
 - operator!=, [864](#)
 - operator<, [865](#)
 - operator>, [865](#)
 - operator+, [866](#)
 - operator+=, [864](#)
 - operator=, [865](#)
 - operator==, [865](#)
 - resize, [865](#)
 - size, [865](#)
 - substr, [866](#)
 - swap, [866](#)
- Spinnaker::ICameraBase
 - ~ICameraBase, [882](#)
 - BeginAcquisition, [882](#)
 - CameraInternal, [888](#)
 - DelInit, [882](#)
 - DiscoverMaxPacketSize, [883](#)
 - EndAcquisition, [883](#)
 - ForceIP, [883](#)
 - GetAccessMode, [883](#)
 - GetBufferOwnership, [883](#)
 - GetGuiXml, [883](#)
 - GetNextImage, [884](#)
 - GetNodeMap, [884](#)
 - GetNumDataStreams, [884](#)
 - GetNumImagesInUse, [884](#)
 - GetTLDeviceNodeMap, [884](#)
 - GetTLStreamNodeMap, [884](#)
 - GetUniqueID, [885](#)
 - GetUserBufferCount, [885](#)
 - GetUserBufferSize, [885](#)
 - GetUserBufferTotalSize, [885](#)
 - ICameraBase, [882](#)
 - Init, [885](#)
 - InterfaceImpl, [888](#)
 - IsInitialized, [885](#)
 - IsStreaming, [886](#)
 - IsValid, [886](#)
 - m_pCameraBaseData, [888](#)
 - operator=, [886](#)
 - ReadPort, [886](#)
 - RegisterEventHandler, [886](#)
 - SetBufferOwnership, [887](#)
 - SetUserBuffers, [887](#)
 - TLDevice, [888](#)
 - TLStream, [888](#)
 - UnregisterEventHandler, [887](#)
 - WritePort, [887](#)
- Spinnaker::ICameraList
 - ~ICameraList, [890](#)
 - Append, [890](#)
 - CameraListImpl, [892](#)
 - Clear, [890](#)
 - GetByDeviceID, [891](#)
 - GetByIndex, [891](#)
 - GetBySerial, [891](#)
 - GetSize, [891](#)
 - ICameraList, [890](#)
 - InterfaceImpl, [892](#)
 - m_pCameraListData, [893](#)
 - operator=, [891](#)
 - operator[], [891](#)
 - RemoveByDeviceID, [892](#)
 - RemoveByIndex, [892](#)
 - RemoveBySerial, [892](#)
- Spinnaker::IChunkData
 - ~IChunkData, [894](#)
 - GetBlackLevel, [895](#)
 - GetCRC, [895](#)
 - GetCompressionMode, [895](#)
 - GetCompressionRatio, [895](#)
 - GetCounterValue, [895](#)
 - GetEncoderValue, [895](#)
 - GetExposureEndLineStatusAll, [896](#)
 - GetExposureTime, [896](#)
 - GetFrameID, [896](#)
 - GetGain, [896](#)
 - GetHeight, [896](#)
 - GetImage, [896](#)
 - GetInferenceBoundingBoxResult, [897](#)
 - GetInferenceConfidence, [897](#)

- GetInferenceFrameId, [897](#)
- GetInferenceResult, [897](#)
- GetLinePitch, [897](#)
- GetLineStatusAll, [897](#)
- GetOffsetX, [898](#)
- GetOffsetY, [898](#)
- GetPartSelector, [898](#)
- GetPixelDynamicRangeMax, [898](#)
- GetPixelDynamicRangeMin, [898](#)
- GetScan3dAxisMax, [898](#)
- GetScan3dAxisMin, [899](#)
- GetScan3dCoordinateOffset, [899](#)
- GetScan3dCoordinateReferenceValue, [899](#)
- GetScan3dCoordinateScale, [899](#)
- GetScan3dInvalidDataValue, [899](#)
- GetScan3dTransformValue, [899](#)
- GetScanLineSelector, [900](#)
- GetSequencerSetActive, [900](#)
- GetSerialDataLength, [900](#)
- GetStreamChannelID, [900](#)
- GetTimerValue, [900](#)
- GetTimestamp, [900](#)
- GetTimestampLatchValue, [901](#)
- GetTransferBlockID, [901](#)
- GetTransferQueueCurrentBlockCount, [901](#)
- GetWidth, [901](#)
- IChunkData, [894](#)
- SetChunks, [901](#)
- Spinnaker::IDataStream
 - ~IDataStream, [902](#)
 - AnnounceImage, [903](#)
 - AttachBuffer, [903](#)
 - CleanupChunkAdapter, [903](#)
 - FlushQueueAllDiscard, [904](#)
 - GetBufferChunkData, [904](#)
 - GetBufferInfoBool8Type, [904](#)
 - GetBufferInfoPtrType, [904](#)
 - GetBufferInfoSizeType, [904](#)
 - GetBufferInfoUInt64Type, [904](#)
 - GetDeviceNodeMap, [905](#)
 - GetNextImage, [905](#)
 - GetNextImageInternal, [905](#)
 - GetNodeMap, [905](#)
 - GetNumImagesInUse, [905](#)
 - GetPort, [905](#)
 - GetStreamInfoBool8Type, [905](#)
 - GetStreamInfoSizeType, [906](#)
 - GetStreamType, [906](#)
 - IDataStream, [903](#)
 - InitChunkAdapter, [906](#)
 - IsCRCCheckEnabled, [906](#)
 - IsImageInUse, [906](#)
 - IsStreaming, [906](#)
 - KillBufferEvent, [906](#)
 - RegisterImageEventHandler, [907](#)
 - ReleaseImage, [907](#)
 - ResetStreamHandle, [907](#)
 - RevokeImages, [907](#)
 - StartStream, [907](#)
 - StopStream, [907](#)
 - TransportLayerStreamInfo, [907](#)
 - UnregisterImageEventHandler, [908](#)
 - WaitOnImageEvent, [908](#)
- Spinnaker::IDeviceArrivalEventHandler
 - ~IDeviceArrivalEventHandler, [914](#)
 - IDeviceArrivalEventHandler, [914](#)
 - OnDeviceArrival, [914](#)
 - operator=, [914](#)
- Spinnaker::IDeviceEventHandler
 - ~IDeviceEventHandler, [916](#)
 - GetDeviceEventId, [916](#)
 - GetDeviceEventName, [916](#)
 - IDeviceEventHandler, [916](#)
 - OnDeviceEvent, [917](#)
 - operator=, [917](#)
- Spinnaker::IDeviceRemovalEventHandler
 - ~IDeviceRemovalEventHandler, [918](#)
 - IDeviceRemovalEventHandler, [918](#)
 - OnDeviceRemoval, [919](#)
 - operator=, [919](#)
- Spinnaker::IImage
 - ~IImage, [921](#)
 - CalculateStatistics, [921](#)
 - CheckCRC, [921](#)
 - Convert, [921](#), [922](#)
 - DeepCopy, [922](#)
 - GetBitsPerPixel, [922](#)
 - GetBufferSize, [922](#)
 - GetChunkData, [922](#)
 - GetChunkLayoutId, [923](#)
 - GetColorProcessing, [923](#)
 - GetData, [923](#)
 - GetDataAbsoluteMax, [923](#)
 - GetDataAbsoluteMin, [923](#)
 - GetFrameID, [923](#)
 - GetHeight, [924](#)
 - GetID, [924](#)
 - GetImageData, [924](#)
 - GetImageSize, [924](#)
 - GetImageStatus, [924](#)
 - GetNumChannels, [924](#)
 - GetPayloadType, [925](#)
 - GetPixelFormat, [925](#)
 - GetPixelFormatIntType, [925](#)
 - GetPixelFormatName, [925](#)
 - GetPrivateData, [925](#)
 - GetStride, [925](#)
 - GetTLPayloadType, [926](#)
 - GetTLPixelFormat, [926](#)
 - GetTLPixelFormatNamespace, [926](#)
 - GetTimeStamp, [926](#)
 - GetValidPayloadSize, [926](#)
 - GetWidth, [926](#)
 - GetXOffset, [927](#)
 - GetXPadding, [927](#)
 - GetYOffset, [927](#)

- GetYPadding, 927
- HasCRC, 927
- Image, 921
- IsCompressed, 927
- IsInUse, 928
- IsIncomplete, 928
- Release, 928
- ResetImage, 928, 929
- Save, 929, 930
- Stream, 931
- Spinnaker::ImageEventHandler
 - ~ImageEventHandler, 932
 - ImageEventHandler, 932
 - OnImageEvent, 933
 - operator=, 933
- Spinnaker::ImageStatistics
 - ~ImageStatistics, 934
 - DisableAll, 935
 - EnableAll, 935
 - EnableGreyOnly, 935
 - EnableHSLOnly, 935
 - EnableRGBOnly, 935
 - GetChannelStatus, 935
 - GetHistogram, 936
 - GetMean, 936
 - GetNumPixelValues, 936
 - GetPixelValueRange, 936
 - GetRange, 936
 - GetStatistics, 937
 - ImageStatistics, 934
 - SetChannelStatus, 937
- Spinnaker::Interface
 - ~Interface, 939
 - GetCameras, 939
 - GetTLNodeMap, 940
 - Interface, 939
 - InterfaceInternal, 941
 - IsInUse, 940
 - IsValid, 940
 - m_pInterfaceData, 941
 - operator=, 940
 - ProducerImpl, 941
 - RegisterEventHandler, 940
 - SendActionCommand, 940
 - TLInterface, 942
 - UnregisterEventHandler, 941
 - UpdateCameras, 941
- Spinnaker::InterfaceArrivalEventHandler
 - ~InterfaceArrivalEventHandler, 943
 - InterfaceArrivalEventHandler, 943
 - OnInterfaceArrival, 944
 - operator=, 944
- Spinnaker::InterfaceEventHandler
 - ~InterfaceEventHandler, 945
 - InterfaceEventHandler, 945
 - OnDeviceArrival, 946
 - OnDeviceRemoval, 946
 - operator=, 946
- Spinnaker::InterfaceList
 - ~InterfaceList, 948
 - Append, 948
 - Clear, 948
 - GetByIndex, 948
 - GetSize, 949
 - InterfaceList, 948
 - InterfaceListImpl, 949
 - m_pInterfaceListData, 949
 - operator=, 949
 - operator[], 949
- Spinnaker::InterfaceRemovalEventHandler
 - ~InterfaceRemovalEventHandler, 951
 - InterfaceRemovalEventHandler, 951
 - OnInterfaceRemoval, 951
 - operator=, 951
- Spinnaker::ILoggingEventHandler
 - ~ILoggingEventHandler, 953
 - ILoggingEventHandler, 953
 - OnLogEvent, 953
 - operator=, 953
- Spinnaker::ISystem
 - ~ISystem, 1059
 - GetCameras, 1060
 - GetInterfaces, 1060
 - GetLibraryVersion, 1060
 - GetLoggingEventPriorityLevel, 1060
 - GetTLNodeMap, 1060
 - ISystem, 1059
 - IsInUse, 1061
 - operator=, 1061
 - RegisterEventHandler, 1061
 - RegisterInterfaceEventHandler, 1061
 - RegisterLoggingEventHandler, 1061
 - ReleaseInstance, 1061
 - SendActionCommand, 1062
 - SetLoggingEventPriorityLevel, 1062
 - SystemPtrInternal, 1063
 - TLSystem, 1063
 - UnregisterAllLoggingEventHandlers, 1062
 - UnregisterEventHandler, 1062
 - UnregisterInterfaceEventHandler, 1062
 - UnregisterLoggingEventHandler, 1063
 - UpdateCameras, 1063
 - UpdateInterfaceList, 1063
- Spinnaker::ISystemEventHandler
 - ~ISystemEventHandler, 1065
 - ISystemEventHandler, 1065
 - OnInterfaceArrival, 1065
 - OnInterfaceRemoval, 1065
 - operator=, 1066
- Spinnaker::Image
 - ~Image, 958
 - CalculateStatistics, 959
 - CheckCRC, 959
 - Convert, 959–961
 - Create, 961, 962
 - CreateShared, 963

- DeepCopy, 963
- GetBitsPerPixel, 963
- GetBufferSize, 963
- GetChunkData, 964
- GetChunkLayoutId, 964
- GetColorProcessing, 964
- GetData, 965
- GetDataAbsoluteMax, 965
- GetDataAbsoluteMin, 965
- GetDefaultColorProcessing, 966
- GetFrameID, 966
- GetHeight, 966
- GetID, 967
- GetImageData, 967
- GetImageSize, 967
- GetImageStatus, 967
- GetImageStatusDescription, 968
- GetNumChannels, 968
- GetNumDecompressionThreads, 968
- GetPayloadType, 969
- GetPixelFormat, 969
- GetPixelFormatIntType, 969
- GetPixelFormatName, 970
- GetPrivateData, 970
- GetStride, 970
- GetTLPayloadType, 971
- GetTLPixelFormat, 971
- GetTLPixelFormatNamespace, 972
- GetTimeStamp, 971
- GetValidPayloadSize, 972
- GetWidth, 973
- GetXOffset, 973
- GetXPadding, 973
- GetYOffset, 974
- GetYPadding, 974
- HasCRC, 974
- IDataStream, 981
- Image, 958
- ImageConverter, 981
- ImageConverterIpp, 981
- ImageFiler, 982
- ImageStatsCalculator, 982
- ImageUtilityImpl, 982
- ImageUtilityPolarizationImpl, 982
- IsCompressed, 975
- IsInUse, 975
- IsIncomplete, 975
- Release, 976
- ResetImage, 976, 977
- Save, 978–980
- SetDefaultColorProcessing, 980
- SetNumDecompressionThreads, 981
- Stream, 982
- Spinnaker::ImageEventHandler
 - ~ImageEventHandler, 984
 - ImageEventHandler, 984
 - OnImageEvent, 984
 - operator=, 985
- Spinnaker::ImagePtr
 - ~ImagePtr, 991
 - ImagePtr, 991
 - operator=, 992
- Spinnaker::ImageStatistics
 - ~ImageStatistics, 994
 - DisableAll, 994
 - EnableAll, 994
 - EnableGreyOnly, 994
 - EnableHSLOnly, 995
 - EnableRGBOnly, 995
 - GetChannelStatus, 995
 - GetHistogram, 995
 - GetMean, 996
 - GetNumPixelValues, 996
 - GetPixelValueRange, 996
 - GetRange, 997
 - GetStatistics, 997
 - ImageStatistics, 994
 - ImageStatsCalculator, 999
 - operator=, 998
 - SetChannelStatus, 998
- Spinnaker::ImageUtility
 - CreateNormalized, 1001–1003
 - CreateScaled, 1003
 - ImageScalingAlgorithm, 1000
 - SourceDataRange, 1000
- Spinnaker::ImageUtilityCCM
 - CreateColorCorrected, 1004, 1005
- Spinnaker::ImageUtilityHeatmap
 - CreateHeatmap, 1007, 1008
 - GetHeatmapColorGradient, 1008
 - GetHeatmapRange, 1009
 - HeatmapColor, 1007
 - SetHeatmapColorGradient, 1009
 - SetHeatmapRange, 1010
- Spinnaker::ImageUtilityPolarization
 - CreateAolp, 1012
 - CreateDolp, 1014
 - CreateGlareReduced, 1014, 1015
 - CreateStokesS0, 1015, 1016
 - CreateStokesS1, 1016, 1017
 - CreateStokesS2, 1017
 - ExtractPolarQuadrant, 1018
 - PolarizationQuadrant, 1011
- Spinnaker::Interface
 - ~Interface, 1032
 - GetCameras, 1032
 - GetTLNodeMap, 1033
 - InterfaceInternal, 1036
 - IsInUse, 1033
 - IsValid, 1033
 - RegisterEventHandler, 1034
 - SendActionCommand, 1034
 - UnregisterEventHandler, 1035
 - UpdateCameras, 1035
- Spinnaker::InterfaceArrivalEventHandler
 - ~InterfaceArrivalEventHandler, 1038

- InterfaceArrivalEventHandler, 1037
- OnInterfaceArrival, 1038
- operator=, 1038
- Spinnaker::InterfaceEventHandler
 - ~InterfaceEventHandler, 1040
 - InterfaceEventHandler, 1040
 - OnDeviceArrival, 1040
 - OnDeviceRemoval, 1041
 - operator=, 1041
- Spinnaker::InterfaceList
 - ~InterfaceList, 1047
 - Append, 1048
 - Clear, 1048
 - GetByIndex, 1048
 - GetSize, 1049
 - InterfaceList, 1047
 - operator=, 1049
 - operator[], 1049
 - ProducerImpl, 1049
 - SystemImpl, 1050
- Spinnaker::InterfacePtr
 - InterfacePtr, 1051
- Spinnaker::InterfaceRemovalEventHandler
 - ~InterfaceRemovalEventHandler, 1053
 - InterfaceRemovalEventHandler, 1053
 - OnInterfaceRemoval, 1053
 - operator=, 1054
- Spinnaker::JPEGOption
 - JPEGOption, 1066
 - progressive, 1067
 - quality, 1067
 - reserved, 1067
- Spinnaker::JPG2Option
 - JPG2Option, 1068
 - quality, 1068
 - reserved, 1068
- Spinnaker::LibraryVersion
 - build, 1069
 - major, 1069
 - minor, 1069
 - type, 1069
- Spinnaker::LoggingEventData
 - ~LoggingEventData, 1073
 - GetCategoryName, 1074
 - GetLogMessage, 1074
 - GetNDC, 1074
 - GetPriority, 1074
 - GetPriorityName, 1075
 - GetThreadName, 1075
 - GetTimestamp, 1075
 - LoggingEventData, 1073
 - SystemImpl, 1075
- Spinnaker::LoggingEventDataPtr
 - LoggingEventDataPtr, 1077
- Spinnaker::LoggingEventHandler
 - ~LoggingEventHandler, 1079
 - LoggingEventHandler, 1079
 - OnLogEvent, 1079
 - operator=, 1080
- Spinnaker::PGMOption
 - binaryFile, 1111
 - PGMOption, 1111
 - reserved, 1111
- Spinnaker::PNGOption
 - compressionLevel, 1112
 - interlaced, 1112
 - PNGOption, 1112
 - reserved, 1112
- Spinnaker::PPMOption
 - binaryFile, 1126
 - PPMOption, 1126
 - reserved, 1126
- Spinnaker::System
 - ~System, 1146
 - GetCameras, 1146
 - GetInstance, 1147
 - GetInterfaces, 1147
 - GetLibraryVersion, 1147
 - GetLoggingEventPriorityLevel, 1148
 - GetTLNodeMap, 1148
 - IsInUse, 1148
 - RegisterEventHandler, 1149
 - RegisterInterfaceEventHandler, 1149
 - RegisterLoggingEventHandler, 1150
 - ReleaseInstance, 1150
 - SendActionCommand, 1150
 - SetLoggingEventPriorityLevel, 1151
 - System, 1146
 - UnregisterAllLoggingEventHandlers, 1152
 - UnregisterEventHandler, 1152
 - UnregisterInterfaceEventHandler, 1152
 - UnregisterLoggingEventHandler, 1152
 - UpdateCameras, 1153
 - UpdateInterfaceList, 1153
- Spinnaker::SystemEventHandler
 - ~SystemEventHandler, 1156
 - OnInterfaceArrival, 1156
 - OnInterfaceRemoval, 1156
 - operator=, 1157
 - SystemEventHandler, 1155
- Spinnaker::SystemPtr
 - ~SystemPtr, 1162
 - SystemPtr, 1162
- Spinnaker::TIFFOption
 - compression, 1164
 - CompressionMethod, 1163
 - reserved, 1164
 - TIFFOption, 1164
- Spinnaker::TransportLayerDevice
 - ~TransportLayerDevice, 1167
 - CameraBase, 1167
 - CameraInternal, 1167
 - DeviceAccessStatus, 1168
 - DeviceCurrentSpeed, 1168
 - DeviceDisplayName, 1168
 - DeviceDriverVersion, 1168

- DeviceEndiannessMechanism, 1168
- DeviceID, 1169
- DeviceInstanceId, 1169
- DevicesUpdater, 1169
- DeviceLinkSpeed, 1169
- DeviceLocation, 1169
- DeviceModelName, 1169
- DeviceMulticastMonitorMode, 1170
- DevicePortId, 1170
- DeviceSerialNumber, 1170
- DeviceType, 1170
- DeviceU3VProtocol, 1170
- DeviceUserID, 1170
- DeviceVendorName, 1171
- DeviceVersion, 1171
- GUXMLLocation, 1174
- GUXMLPath, 1175
- GenICamXMLLocation, 1171
- GenICamXMLPath, 1171
- GevCCP, 1171
- GevDeviceAutoForceIP, 1171
- GevDeviceDiscoverMaximumPacketSize, 1172
- GevDeviceForceGateway, 1172
- GevDeviceForceIPAddress, 1172
- GevDeviceForceIP, 1172
- GevDeviceForceSubnetMask, 1172
- GevDeviceGateway, 1172
- GevDeviceIPAddress, 1173
- GevDevicesWrongSubnet, 1173
- GevDeviceMACAddress, 1173
- GevDeviceMaximumPacketSize, 1173
- GevDeviceMaximumRetryCount, 1173
- GevDeviceModelsBigEndian, 1173
- GevDevicePort, 1174
- GevDeviceReadAndWriteTimeout, 1174
- GevDeviceSubnetMask, 1174
- GevVersionMajor, 1174
- GevVersionMinor, 1174
- ICameraBase, 1168
- TransportLayerDevice, 1167
- Spinnaker::TransportLayerInterface
 - ~TransportLayerInterface, 1178
 - ActionCommand, 1178
 - DeviceAccessStatus, 1179
 - DeviceCount, 1179
 - DeviceID, 1179
 - DeviceModelName, 1179
 - DeviceSelector, 1179
 - DeviceSerialNumber, 1179
 - DeviceUnlock, 1180
 - DeviceUpdateList, 1180
 - DeviceVendorName, 1180
 - FilterDriverStatus, 1180
 - GevActionDeviceKey, 1180
 - GevActionGroupKey, 1180
 - GevActionGroupMask, 1181
 - GevActionTime, 1181
 - GevDeviceAutoForceIP, 1181
 - GevDeviceForceGateway, 1181
 - GevDeviceForceIPAddress, 1181
 - GevDeviceForceIP, 1181
 - GevDeviceForceSubnetMask, 1182
 - GevDeviceGateway, 1182
 - GevDeviceIPAddress, 1182
 - GevDeviceMACAddress, 1182
 - GevDeviceSubnetMask, 1182
 - GevInterfaceGateway, 1182
 - GevInterfaceGatewaySelector, 1183
 - GevInterfaceMACAddress, 1183
 - GevInterfaceMTU, 1183
 - GevInterfaceReceiveLinkSpeed, 1183
 - GevInterfaceSubnetIPAddress, 1183
 - GevInterfaceSubnetMask, 1183
 - GevInterfaceSubnetSelector, 1184
 - GevInterfaceTransmitLinkSpeed, 1184
 - HostAdapterDriverVersion, 1184
 - HostAdapterName, 1184
 - HostAdapterVendor, 1184
 - IInterface, 1178
 - IncompatibleDeviceCount, 1184
 - IncompatibleDeviceID, 1185
 - IncompatibleDeviceModelName, 1185
 - IncompatibleDeviceSelector, 1185
 - IncompatibleDeviceVendorName, 1185
 - IncompatibleGevDeviceIPAddress, 1185
 - IncompatibleGevDeviceMACAddress, 1185
 - IncompatibleGevDeviceSubnetMask, 1186
 - Interface, 1178
 - InterfaceDisplayName, 1186
 - InterfaceID, 1186
 - InterfaceInternal, 1178
 - InterfaceType, 1186
 - POEStatus, 1186
 - TransportLayerInterface, 1178
- Spinnaker::TransportLayerStream
 - ~TransportLayerStream, 1188
 - CameraBase, 1189
 - CameraInternal, 1189
 - GevFailedPacketCount, 1189
 - GevMaximumNumberResendRequests, 1189
 - GevPacketResendMode, 1190
 - GevPacketResendTimeout, 1190
 - GevResendPacketCount, 1190
 - GevResendRequestCount, 1190
 - GevTotalPacketCount, 1190
 - ICameraBase, 1189
 - StreamAnnounceBufferMinimum, 1190
 - StreamAnnouncedBufferCount, 1191
 - StreamBlockTransferSize, 1191
 - StreamBufferAlignment, 1191
 - StreamBufferCountManual, 1191
 - StreamBufferCountMax, 1191
 - StreamBufferCountMode, 1191
 - StreamBufferCountResult, 1192
 - StreamBufferHandlingMode, 1192
 - StreamCRCCheckEnable, 1192

- StreamChunkCountMaximum, 1192
- StreamDeliveredFrameCount, 1192
- StreamDroppedFrameCount, 1192
- StreamFailedBufferCount, 1193
- StreamID, 1193
- StreamInputBufferCount, 1193
- StreamIsGrabbing, 1193
- StreamLostFrameCount, 1193
- StreamOutputBufferCount, 1193
- StreamStartedFrameCount, 1194
- StreamType, 1194
- TransportLayerStream, 1188, 1189
- Spinnaker::TransportLayerSystem
 - ~TransportLayerSystem, 1196
 - EnumerateGEVInterfaces, 1197
 - EnumerateUSBInterfaces, 1197
 - GenTLSFNCVersionMajor, 1197
 - GenTLSFNCVersionMinor, 1197
 - GenTLSFNCVersionSubMinor, 1198
 - GenTLVersionMajor, 1198
 - GenTLVersionMinor, 1198
 - GevInterfaceDefaultGateway, 1198
 - GevInterfaceDefaultIPAddress, 1198
 - GevInterfaceDefaultSubnetMask, 1198
 - GevInterfaceMACAddress, 1199
 - GevVersionMajor, 1199
 - GevVersionMinor, 1199
 - ISystem, 1196
 - InterfaceDisplayName, 1199
 - InterfaceID, 1199
 - InterfaceSelector, 1199
 - InterfaceUpdateList, 1200
 - System, 1197
 - SystemPtrInternal, 1197
 - TLDisplayName, 1200
 - TLFileName, 1200
 - TLID, 1200
 - TLModelName, 1200
 - TLPath, 1200
 - TLType, 1201
 - TLVendorName, 1201
 - TLVersion, 1201
 - TransportLayerSystem, 1196
- Spinnaker::Video, 479
- Spinnaker::Video::AVIOption
 - AVIOption, 489
 - frameRate, 490
 - reserved, 490
- Spinnaker::Video::H264Option
 - bitrate, 879
 - frameRate, 879
 - H264Option, 879
 - height, 879
 - reserved, 879
 - width, 879
- Spinnaker::Video::MJPGOption
 - frameRate, 1084
 - MJPGOption, 1084
 - quality, 1084
 - reserved, 1084
- Spinnaker::Video::SpinVideo
 - ~SpinVideo, 1133
 - Append, 1134
 - Close, 1134
 - Open, 1134, 1135
 - SetMaximumFileSize, 1136
 - SpinVideo, 1133
- SpinnakerDirectShow.h, 232
 - BeginAcquisition, 232
 - EndAcquisition, 233
 - GetCameraInfo, 233
 - GetSelectedCameraIndex, 234
 - IID_ISpinnakerInterface, 235
 - IsStreaming, 234
 - MAX_LENGTH, 1375
 - SetSelectedCameraIndex, 234
- SpinnakerLogLevel
 - Spinnaker Definitions, 230
- src/Acquisition/Acquisition.cpp, 1387
- src/Acquisition/resource.h, 1388
- src/Acquisition/stdafx.cpp, 1390
- src/Acquisition/stdafx.h, 1399
- src/Acquisition/targetver.h, 1415
- src/AcquisitionMultipleCameraRecovery/Acquisition↔
 - MultipleCameraRecovery.cpp, 1429
- src/AcquisitionMultipleCameraRecovery/resource.↔
 - h, 1388
- src/AcquisitionMultipleCamerasWriteToFile/Acquisition↔
 - MultipleCamerasWriteToFile.cpp, 1432
- src/AcquisitionMultipleCamerasWriteToFile/resource.h, 1388
- src/AcquisitionMultipleThread/AcquisitionMultiple↔
 - Thread.cpp, 1434
- src/AcquisitionMultipleThread/resource.h, 1388
- src/ActionCommand/ActionCommand.cpp, 1435
- src/ActionCommand/resource.h, 1388
- src/ActionCommand/stdafx.cpp, 1391
- src/ActionCommand/stdafx.h, 1400
- src/ActionCommand/targetver.h, 1416
- src/BufferHandling/BufferHandling.cpp, 1437
- src/BufferHandling/resource.h, 1388
- src/BufferHandling/stdafx.cpp, 1391
- src/BufferHandling/stdafx.h, 1401
- src/BufferHandling/targetver.h, 1417
- src/ChunkData/ChunkData.cpp, 1440
- src/ChunkData/resource.h, 1388
- src/Compression/Compression.cpp, 1442
- src/Compression/resource.h, 1389
- src/CounterAndTimer/CounterAndTimer.cpp, 1444
- src/CounterAndTimer/resource.h, 1389
- src/CounterAndTimer/stdafx.cpp, 1392
- src/CounterAndTimer/stdafx.h, 1402
- src/CounterAndTimer/targetver.h, 1418
- src/DeviceEvents/DeviceEvents.cpp, 1446
- src/DeviceEvents/resource.h, 1389
- src/DeviceEvents/stdafx.cpp, 1392

- src/DeviceEvents/stdafx.h, 1403
- src/DeviceEvents/targetver.h, 1419
- src/Enumeration/Enumeration.cpp, 1449
- src/Enumeration/resource.h, 1389
- src/Enumeration/stdafx.cpp, 1393
- src/Enumeration/stdafx.h, 1404
- src/Enumeration/targetver.h, 1420
- src/Enumeration_QuickSpin/Enumeration_QuickSpin.↔
cpp, 1449
- src/Enumeration_QuickSpin/resource.h, 1389
- src/Enumeration_QuickSpin/stdafx.cpp, 1393
- src/Enumeration_QuickSpin/stdafx.h, 1405
- src/Enumeration_QuickSpin/targetver.h, 1421
- src/EnumerationEvents/EnumerationEvents.cpp, 1450
- src/EnumerationEvents/resource.h, 1389
- src/ExceptionHandling/ExceptionHandling.cpp, 1451
- src/ExceptionHandling/resource.h, 1389
- src/ExceptionHandling/stdafx.cpp, 1394
- src/ExceptionHandling/stdafx.h, 1406
- src/ExceptionHandling/targetver.h, 1422
- src/Exposure/Exposure.cpp, 1452
- src/Exposure/resource.h, 1389
- src/Exposure/stdafx.cpp, 1394
- src/Exposure/stdafx.h, 1407
- src/Exposure/targetver.h, 1423
- src/Exposure_QuickSpin/Exposure_QuickSpin.cpp,
1454
- src/Exposure_QuickSpin/resource.h, 1389
- src/Exposure_QuickSpin/stdafx.cpp, 1395
- src/Exposure_QuickSpin/stdafx.h, 1408
- src/Exposure_QuickSpin/targetver.h, 1424
- src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp,
1455
- src/FileAccess_QuickSpin/resource.h, 1389
- src/FileAccess_QuickSpin/stdafx.cpp, 1396
- src/FileAccess_QuickSpin/stdafx.h, 1409
- src/FileAccess_QuickSpin/targetver.h, 1425
- src/GigEVisionPerformance/CpuUtil.cpp, 1458
- src/GigEVisionPerformance/CpuUtil.h, 1459
- src/GigEVisionPerformance/GigEVisionPerformance.↔
cpp, 1460
- src/GigEVisionPerformance/GigEVisionPerformance.h,
1465
- src/GigEVisionPerformance/resource.h, 1389
- src/GigEVisionPerformance/stdafx.cpp, 1397
- src/GigEVisionPerformance/stdafx.h, 1410
- src/GigEVisionPerformance/targetver.h, 1426
- src/ImageEvents/ImageEvents.cpp, 1466
- src/ImageEvents/resource.h, 1389
- src/ImageFormatControl/ImageFormatControl.cpp,
1468
- src/ImageFormatControl/resource.h, 1389
- src/ImageFormatControl/stdafx.h, 1411
- src/ImageFormatControl_QuickSpin/ImageFormat↔
Control_QuickSpin.cpp, 1469
- src/ImageFormatControl_QuickSpin/resource.h, 1389
- src/ImageFormatControl_QuickSpin/stdafx.h, 1411
- src/Inference/Inference.cpp, 1470
- src/Inference/resource.h, 1389
- src/Logging/Logging.cpp, 1477
- src/Logging/resource.h, 1389
- src/LogicBlock/LogicBlock.cpp, 1478
- src/LogicBlock/resource.h, 1389
- src/LookupTable/LookupTable.cpp, 1480
- src/LookupTable/resource.h, 1390
- src/NodeMapCallback/NodeMapCallback.cpp, 1481
- src/NodeMapCallback/resource.h, 1390
- src/NodeMapInfo/NodeMapInfo.cpp, 1483
- src/NodeMapInfo/resource.h, 1390
- src/NodeMapInfo/stdafx.cpp, 1397
- src/NodeMapInfo/stdafx.h, 1412
- src/NodeMapInfo/targetver.h, 1427
- src/Polarization/Polarization.cpp, 1487
- src/Polarization/resource.h, 1390
- src/Polarization/stdafx.h, 1413
- src/SaveToAvi/SaveToAvi.cpp, 1490
- src/SaveToAvi/resource.h, 1390
- src/Sequencer/Sequencer.cpp, 1492
- src/Sequencer/resource.h, 1390
- src/Sequencer/stdafx.cpp, 1398
- src/Sequencer/stdafx.h, 1413
- src/Sequencer/targetver.h, 1428
- src/SerialRxTx/SerialRxTx.cpp, 1494
- src/SerialRxTx/resource.h, 1390
- src/SerialRxTx/stdafx.cpp, 1398
- src/SerialRxTx/stdafx.h, 1414
- src/SerialRxTx/targetver.h, 1429
- src/Trigger/Trigger.cpp, 1497
- src/Trigger/resource.h, 1390
- src/Trigger_QuickSpin/Trigger_QuickSpin.cpp, 1500
- src/Trigger_QuickSpin/resource.h, 1390
- Standard
 - Types Enums, 407
- StartCpuTracing
 - CpuUtil, 418
- StartPerformanceCounter
 - PerformanceCounter, 419
- StartRecording
 - Spinnaker::GenApi::PortNode, 1117
 - Spinnaker::GenApi::PortRecorder, 1121
- StartSecondsCounter
 - SecondsCounter, 420
- StartStream
 - Spinnaker::IDataStream, 907
- startTime
 - SecondsCounter, 420
- StatisticsChannel
 - Spinnaker Definitions, 230
- Status
 - Spinnaker::ActionCommandResult, 481
- StopCpuTracing
 - CpuUtil, 418
- StopRecording
 - IPortRecorder Interface, 360
 - Spinnaker::GenApi::PortNode, 1117
 - Spinnaker::GenApi::PortRecorder, 1121

- StopStream
 - Spinnaker::IDataStream, [907](#)
- StoreToBag
 - Spinnaker::GenApi::CFeatureBag, [719](#)
- Stream
 - Spinnaker::EventHandler, [835](#)
 - Spinnaker::Image, [931](#)
 - Spinnaker::Image, [982](#)
- StreamAnnounceBufferMinimum
 - Spinnaker::TransportLayerStream, [1190](#)
- StreamAnnouncedBufferCount
 - Spinnaker::TransportLayerStream, [1191](#)
- StreamBlockTransferSize
 - Spinnaker::TransportLayerStream, [1191](#)
- StreamBufferAlignment
 - Spinnaker::TransportLayerStream, [1191](#)
- StreamBufferCountManual
 - Spinnaker::TransportLayerStream, [1191](#)
- StreamBufferCountMax
 - Spinnaker::TransportLayerStream, [1191](#)
- StreamBufferCountMode
 - Spinnaker::TransportLayerStream, [1191](#)
- StreamBufferCountModeEnum
 - TransportLayerDefs Class, [248](#)
- StreamBufferCountResult
 - Spinnaker::TransportLayerStream, [1192](#)
- StreamBufferHandlingMode
 - Spinnaker::TransportLayerStream, [1192](#)
- StreamBufferHandlingModeEnum
 - TransportLayerDefs Class, [248](#)
- StreamCRCCheckEnable
 - Spinnaker::TransportLayerStream, [1192](#)
- StreamChannelId
 - GVCP_EVENT_ITEM_EXTENDED_ID, [872](#)
 - GVCP_EVENT_ITEM, [870](#)
- StreamChunkCountMaximum
 - Spinnaker::TransportLayerStream, [1192](#)
- StreamDeliveredFrameCount
 - Spinnaker::TransportLayerStream, [1192](#)
- StreamDroppedFrameCount
 - Spinnaker::TransportLayerStream, [1192](#)
- StreamFailedBufferCount
 - Spinnaker::TransportLayerStream, [1193](#)
- StreamID
 - Spinnaker::TransportLayerStream, [1193](#)
- StreamInputBufferCount
 - Spinnaker::TransportLayerStream, [1193](#)
- StreamIsGrabbing
 - Spinnaker::TransportLayerStream, [1193](#)
- StreamLostFrameCount
 - Spinnaker::TransportLayerStream, [1193](#)
- StreamOutputBufferCount
 - Spinnaker::TransportLayerStream, [1193](#)
- StreamStartedFrameCount
 - Spinnaker::TransportLayerStream, [1194](#)
- StreamType
 - Spinnaker::TransportLayerStream, [1194](#)
- StreamTypeEnum
 - TransportLayerDefs Class, [249](#)
- StringGetValue
 - Spinnaker Headers, [216](#)
- StringList_t
 - Types Enums, [402](#)
- StringNode, [1136](#)
 - Spinnaker::GenApi::StringNode, [1138](#)
- StringNode Class, [395](#)
 - CStringRef, [395](#)
- StringRegNode, [1140](#)
 - Spinnaker::GenApi::StringRegNode, [1142](#), [1143](#)
- StringRegNode Class, [396](#)
- StringSetValue
 - Spinnaker Headers, [217](#)
- StructPort Class, [397](#)
- SubMinor
 - Spinnaker::GenICam::Version_t, [1209](#)
- subnetLength
 - AdapterConfig::IpInfo, [1057](#)
- subnetMask
 - AdapterConfig::IpInfo, [1057](#)
- substr
 - Spinnaker::GenICam::gcstring, [866](#)
- swap
 - Spinnaker::GenICam::gcstring, [866](#)
- sync
 - Spinnaker::GenApi::ODevFileStreamBuf, [1110](#)
- Synch Class, [398](#)
- System, [1144](#)
 - Spinnaker::System, [1146](#)
 - Spinnaker::TransportLayerSystem, [1197](#)
- System Class, [239](#)
- System.h
 - FLIR_SPINNAKER_VERSION_BUILD, [1381](#)
 - FLIR_SPINNAKER_VERSION_MAJOR, [1381](#)
 - FLIR_SPINNAKER_VERSION_MINOR, [1381](#)
 - FLIR_SPINNAKER_VERSION_TYPE, [1381](#)
- SystemEventHandler, [1154](#)
 - Spinnaker::SystemEventHandler, [1155](#)
- SystemEventHandler Class, [240](#)
- SystemEventHandlerImpl, [1157](#)
 - ~SystemEventHandlerImpl, [1158](#)
 - LockEventHandlerMutex, [1159](#)
 - OnInterfaceArrival, [1159](#)
 - OnInterfaceRemoval, [1159](#)
 - RegisterAllInterfaceEvents, [1159](#)
 - RegisterInterfaceEventToSystem, [1160](#)
 - SystemEventHandlerImpl, [1158](#)
 - UnlockEventHandlerMutex, [1160](#)
 - UnregisterAllInterfaceEvents, [1160](#)
 - UnregisterInterfaceEventFromSystem, [1160](#)
- SystemImpl
 - Spinnaker::InterfaceList, [1050](#)
 - Spinnaker::LoggingEventData, [1075](#)
- SystemPtr, [1161](#)
 - Spinnaker::SystemPtr, [1162](#)
- SystemPtr Class, [241](#)
- SystemPtrInternal

- Spinnaker::ISystem, [1063](#)
- Spinnaker::TransportLayerSystem, [1197](#)
- TIFFOption, [1163](#)
 - Spinnaker::TIFFOption, [1164](#)
- TLDevice
 - Spinnaker::ICameraBase, [888](#)
- TLDisplayName
 - Spinnaker::TransportLayerSystem, [1200](#)
- TLFileName
 - Spinnaker::TransportLayerSystem, [1200](#)
- TLID
 - Spinnaker::TransportLayerSystem, [1200](#)
- TLInterface
 - Spinnaker::IInterface, [942](#)
- TLModelName
 - Spinnaker::TransportLayerSystem, [1200](#)
- TLParamsLocked
 - Spinnaker::Camera, [642](#)
- TLPath
 - Spinnaker::TransportLayerSystem, [1200](#)
- TLStream
 - Spinnaker::ICameraBase, [888](#)
- TLSystem
 - Spinnaker::ISystem, [1063](#)
- TLType
 - Spinnaker::TransportLayerSystem, [1201](#)
- TLTypeEnum
 - TransportLayerDefs Class, [249](#)
- TLVendorName
 - Spinnaker::TransportLayerSystem, [1201](#)
- TLVersion
 - Spinnaker::TransportLayerSystem, [1201](#)
- TWO_SECOND_DELAY
 - SerialRxTx.cpp, [1495](#)
- Test0001
 - Spinnaker::Camera, [639](#)
- TestDuration
 - GigEVisionPerformance.cpp, [1465](#)
- TestEventGenerate
 - Spinnaker::Camera, [640](#)
- TestPattern
 - Spinnaker::Camera, [640](#)
- TestPatternEnums
 - CameraDefs Class, [156](#)
- TestPatternGeneratorSelector
 - Spinnaker::Camera, [640](#)
- TestPatternGeneratorSelectorEnums
 - CameraDefs Class, [157](#)
- TestPendingAck
 - Spinnaker::Camera, [640](#)
- ThrowBadAlloc
 - Spinnaker::GenICam, [479](#)
- timeDiff
 - SecondsCounter, [420](#)
- TimerDelay
 - Spinnaker::Camera, [640](#)
- TimerDuration
 - Spinnaker::Camera, [641](#)
- TimerReset
 - Spinnaker::Camera, [641](#)
- TimerSelector
 - Spinnaker::Camera, [641](#)
- TimerSelectorEnums
 - CameraDefs Class, [157](#)
- TimerStatus
 - Spinnaker::Camera, [641](#)
- TimerStatusEnums
 - CameraDefs Class, [157](#)
- TimerTriggerActivation
 - Spinnaker::Camera, [641](#)
- TimerTriggerActivationEnums
 - CameraDefs Class, [158](#)
- TimerTriggerSource
 - Spinnaker::Camera, [641](#)
- TimerTriggerSourceEnums
 - CameraDefs Class, [158](#)
- TimerValue
 - Spinnaker::Camera, [642](#)
- Timestamp
 - Spinnaker::Camera, [642](#)
 - U3V_EVENT_DATA, [1204](#)
- TimestampHigh
 - GVCP_EVENT_ITEM_EXTENDED_ID, [872](#)
 - GVCP_EVENT_ITEM, [870](#)
- TimestampLatch
 - Spinnaker::Camera, [642](#)
- TimestampLatchValue
 - Spinnaker::Camera, [642](#)
- TimestampLow
 - GVCP_EVENT_ITEM_EXTENDED_ID, [872](#)
 - GVCP_EVENT_ITEM, [870](#)
- TimestampReset
 - Spinnaker::Camera, [642](#)
- ToString
 - ISelectorDigit Interface, [365](#)
 - IValue Class, [369](#)
 - Spinnaker::GenApi::CNodeMapFactory, [758](#)
 - Spinnaker::GenApi::CSelectorSet, [791](#)
 - Spinnaker::GenApi::EAccessModeClass, [812](#)
 - Spinnaker::GenApi::ECachingModeClass, [813](#)
 - Spinnaker::GenApi::EDisplayNotationClass, [814](#)
 - Spinnaker::GenApi::EEndianessClass, [815](#)
 - Spinnaker::GenApi::EGenApiSchemaVersion↔ Class, [816](#)
 - Spinnaker::GenApi::EInputDirectionClass, [817](#)
 - Spinnaker::GenApi::ENamespaceClass, [818](#)
 - Spinnaker::GenApi::ERepresentationClass, [828](#)
 - Spinnaker::GenApi::ESignClass, [829](#)
 - Spinnaker::GenApi::ESlopeClass, [830](#)
 - Spinnaker::GenApi::EStandardNameSpaceClass, [831](#)
 - Spinnaker::GenApi::EVisibilityClass, [836](#)
 - Spinnaker::GenApi::EYesNoClass, [842](#)
 - Spinnaker::GenApi::ValueNode, [1208](#)
- ToXml
 - Spinnaker::GenApi::CNodeMapFactory, [758](#)

- Tokenize
 - GCUtilities Utility, [309](#)
- topLeftXCoord
 - Chunk Data Inference Class, [176](#), [177](#)
- topLeftYCoord
 - Chunk Data Inference Class, [177](#)
- TransferAbort
 - Spinnaker::Camera, [643](#)
- TransferBlockCount
 - Spinnaker::Camera, [643](#)
- TransferBurstCount
 - Spinnaker::Camera, [643](#)
- TransferComponentSelector
 - Spinnaker::Camera, [643](#)
- TransferComponentSelectorEnums
 - CameraDefs Class, [160](#)
- TransferControlMode
 - Spinnaker::Camera, [643](#)
- TransferControlModeEnums
 - CameraDefs Class, [160](#)
- TransferOperationMode
 - Spinnaker::Camera, [643](#)
- TransferOperationModeEnums
 - CameraDefs Class, [160](#)
- TransferPause
 - Spinnaker::Camera, [644](#)
- TransferQueueCurrentBlockCount
 - Spinnaker::Camera, [644](#)
- TransferQueueMaxBlockCount
 - Spinnaker::Camera, [644](#)
- TransferQueueMode
 - Spinnaker::Camera, [644](#)
- TransferQueueModeEnums
 - CameraDefs Class, [161](#)
- TransferQueueOverflowCount
 - Spinnaker::Camera, [644](#)
- TransferResume
 - Spinnaker::Camera, [644](#)
- TransferSelector
 - Spinnaker::Camera, [645](#)
- TransferSelectorEnums
 - CameraDefs Class, [161](#)
- TransferStart
 - Spinnaker::Camera, [645](#)
- TransferStatus
 - Spinnaker::Camera, [645](#)
- TransferStatusSelector
 - Spinnaker::Camera, [645](#)
- TransferStatusSelectorEnums
 - CameraDefs Class, [161](#)
- TransferStop
 - Spinnaker::Camera, [645](#)
- TransferStreamChannel
 - Spinnaker::Camera, [645](#)
- TransferTriggerActivation
 - Spinnaker::Camera, [646](#)
- TransferTriggerActivationEnums
 - CameraDefs Class, [162](#)
- TransferTriggerMode
 - Spinnaker::Camera, [646](#)
- TransferTriggerModeEnums
 - CameraDefs Class, [162](#)
- TransferTriggerSelector
 - Spinnaker::Camera, [646](#)
- TransferTriggerSelectorEnums
 - CameraDefs Class, [162](#)
- TransferTriggerSource
 - Spinnaker::Camera, [646](#)
- TransferTriggerSourceEnums
 - CameraDefs Class, [163](#)
- transmitBuffers
 - AdapterConfig::AdapterInfo, [486](#)
- transmitBuffersMax
 - AdapterConfig::AdapterInfo, [486](#)
- transmitBuffersMin
 - AdapterConfig::AdapterInfo, [486](#)
- transmitBuffersRegKey
 - AdapterConfig::AdapterInfo, [486](#)
- transmitBuffersStep
 - AdapterConfig::AdapterInfo, [486](#)
- TransportLayerDefs Class, [243](#)
 - DeviceAccessStatusEnum, [244](#)
 - DeviceCurrentSpeedEnum, [245](#)
 - DeviceEndiannessMechanismEnum, [245](#)
 - DeviceTypeEnum, [246](#)
 - FilterDriverStatusEnum, [246](#)
 - GUIXMLLocationEnum, [247](#)
 - GenICamXMLLocationEnum, [246](#)
 - GevCCPEnum, [247](#)
 - InterfaceTypeEnum, [247](#)
 - POEStatusEnum, [248](#)
 - StreamBufferCountModeEnum, [248](#)
 - StreamBufferHandlingModeEnum, [248](#)
 - StreamTypeEnum, [249](#)
 - TLTypeEnum, [249](#)
- TransportLayerDevice, [1165](#)
 - Spinnaker::TransportLayerDevice, [1167](#)
- TransportLayerDevice Class, [251](#)
- TransportLayerInterface, [1175](#)
 - Spinnaker::TransportLayerInterface, [1178](#)
- TransportLayerInterface Class, [252](#)
- TransportLayerStream, [1187](#)
 - Spinnaker::TransportLayerStream, [1188](#), [1189](#)
- TransportLayerStream Class, [253](#)
- TransportLayerStreamInfo
 - Spinnaker::IDataStream, [907](#)
- TransportLayerSystem, [1194](#)
 - Spinnaker::TransportLayerSystem, [1196](#)
- TransportLayerSystem Class, [254](#)
- Trigger.cpp
 - AcquireImages, [1499](#)
 - chosenTrigger, [1500](#)
 - ConfigureTrigger, [1499](#)
 - GrabNextImageByTrigger, [1499](#)
 - main, [1499](#)
 - PrintDeviceInfo, [1499](#)

- ResetTrigger, [1499](#)
- RunSingleCamera, [1500](#)
- triggerType, [1498](#)
- Trigger_QuickSpin.cpp
 - AcquireImages, [1501](#)
 - chosenTrigger, [1502](#)
 - ConfigureTrigger, [1501](#)
 - GrabNextImageByTrigger, [1501](#)
 - main, [1501](#)
 - PrintDeviceInfo, [1502](#)
 - ResetTrigger, [1502](#)
 - RunSingleCamera, [1502](#)
 - triggerType, [1501](#)
- TriggerActivation
 - Spinnaker::Camera, [646](#)
- TriggerActivationEnums
 - CameraDefs Class, [164](#)
- TriggerDelay
 - Spinnaker::Camera, [646](#)
- TriggerDivider
 - Spinnaker::Camera, [647](#)
- TriggerEventTest
 - Spinnaker::Camera, [647](#)
- TriggerMode
 - Spinnaker::Camera, [647](#)
- TriggerModeEnums
 - CameraDefs Class, [164](#)
- TriggerMultiplier
 - Spinnaker::Camera, [647](#)
- TriggerOverlap
 - Spinnaker::Camera, [647](#)
- TriggerOverlapEnums
 - CameraDefs Class, [165](#)
- TriggerSelector
 - Spinnaker::Camera, [648](#)
- TriggerSelectorEnums
 - CameraDefs Class, [165](#)
- TriggerSoftware
 - Spinnaker::Camera, [648](#)
- TriggerSource
 - Spinnaker::Camera, [648](#)
- TriggerSourceEnums
 - CameraDefs Class, [165](#)
- triggerType
 - Trigger.cpp, [1498](#)
 - Trigger_QuickSpin.cpp, [1501](#)
- TryLock
 - Spinnaker::GenApi::CLock, [744](#)
 - Spinnaker::GenICam::CGlobalLock, [724](#)
 - Spinnaker::GenICam::CLock, [741](#)
- Type
 - Spinnaker::CCMSettings, [695](#)
- type
 - Spinnaker::LibraryVersion, [1069](#)
- Types Enums, [400](#)
 - _UndefinedRepresentation, [402](#)
 - Automatic, [408](#)
 - Beginner, [409](#)
 - Boolean, [407](#)
 - CL, [408](#)
 - Custom, [407](#)
 - Decreasing, [408](#)
 - EAccessMode, [403](#)
 - ECachingMode, [403](#)
 - EDisplayNotation, [403](#)
 - EEndianess, [405](#)
 - EGenApiSchemaVersion, [405](#)
 - ELncMode, [405](#)
 - EInputDirection, [406](#)
 - EInterfaceType, [406](#)
 - ELinkType, [406](#)
 - ENamespace, [407](#)
 - ERepresentation, [407](#)
 - ESign, [407](#)
 - ESlope, [408](#)
 - EStandardNameSpace, [408](#)
 - EVisibility, [408](#)
 - EXMLValidation, [409](#)
 - EYesNo, [409](#)
 - Expert, [409](#)
 - Guru, [409](#)
 - Increasing, [408](#)
 - Invisible, [409](#)
 - Linear, [407](#)
 - Logarithmic, [407](#)
 - NA, [403](#)
 - NI, [403](#)
 - No, [409](#)
 - None, [408](#)
 - RO, [403](#)
 - RW, [403](#)
 - Signed, [408](#)
 - Standard, [407](#)
 - StringList_t, [402](#)
 - Unsigned, [408](#)
 - Varying, [408](#)
 - WO, [403](#)
 - Yes, [409](#)
- Types.h
 - interface, [1367](#)
- U3V_CHUNK_TRAILER, [1201](#)
 - ChunkID, [1202](#)
 - ChunkLength, [1202](#)
- U3V_COMMAND_HEADER, [1202](#)
 - CommandId, [1202](#)
 - Flags, [1203](#)
 - Length, [1203](#)
 - Prefix, [1203](#)
 - ReqId, [1203](#)
- U3V_EVENT_DATA, [1203](#)
 - EventId, [1204](#)
 - Reserved, [1204](#)
 - Timestamp, [1204](#)
- U3V_EVENT_MESSAGE, [1204](#)
 - CommandHeader, [1205](#)
 - EventData, [1205](#)

- U3V_EVENT_PREFIX
 - Spinnaker::GenApi, [477](#)
- USE_TEMP_CACHE_FILE
 - GCUtilities.h, [1315](#)
- underflow
 - Spinnaker::GenApi::IDevFileStreamBuf, [912](#)
- Unlock
 - Spinnaker::GenApi::CLock, [744](#)
 - Spinnaker::GenICam::CGlobalLock, [725](#)
 - Spinnaker::GenICam::CLock, [742](#)
- UnlockEarly
 - Spinnaker::GenICam::CGlobalLockUnlocker, [726](#)
- UnlockEventHandlerMutex
 - SystemEventHandlerImpl, [1160](#)
- UnregisterAllInterfaceEvents
 - SystemEventHandlerImpl, [1160](#)
- UnregisterAllLoggingEventHandlers
 - Spinnaker::ISystem, [1062](#)
 - Spinnaker::System, [1152](#)
- UnregisterEventHandler
 - Spinnaker::CameraBase, [666](#)
 - Spinnaker::ICameraBase, [887](#)
 - Spinnaker::IInterface, [941](#)
 - Spinnaker::ISystem, [1062](#)
 - Spinnaker::Interface, [1035](#)
 - Spinnaker::System, [1152](#)
- UnregisterImageEventHandler
 - Spinnaker::IDataStream, [908](#)
- UnregisterInterfaceEventFromSystem
 - SystemEventHandlerImpl, [1160](#)
- UnregisterInterfaceEventHandler
 - Spinnaker::ISystem, [1062](#)
 - Spinnaker::System, [1152](#)
- UnregisterLoggingEventHandler
 - Spinnaker::ISystem, [1063](#)
 - Spinnaker::System, [1152](#)
- Unsigned
 - Types Enums, [408](#)
- UpdateBuffer
 - Spinnaker::GenApi::CChunkAdapter, [678](#)
 - Spinnaker::GenApi::CChunkPort, [693](#)
- UpdateCameras
 - Spinnaker::IInterface, [941](#)
 - Spinnaker::ISystem, [1063](#)
 - Spinnaker::Interface, [1035](#)
 - Spinnaker::System, [1153](#)
- UpdateFirmware
 - SpinUpdate.h, [1377](#)
- UpdateFirmwareConsole
 - SpinUpdate.h, [1377](#)
- UpdateFirmwareGUI
 - SpinUpdate.h, [1378](#)
- UpdateInterfaceList
 - Spinnaker::ISystem, [1063](#)
 - Spinnaker::System, [1153](#)
- UpdatorMessageCallback
 - SpinUpdate.h, [1378](#)
- UpdatorProgressCallback
 - SpinUpdate.h, [1378](#)
- UploadFileToCamera
 - Inference.cpp, [1475](#)
- UploadImage
 - FileAccess_QuickSpin.cpp, [1458](#)
- UrlDecode
 - GCUtilities Utility, [309](#)
- UrlEncode
 - GCUtilities Utility, [310](#)
- UseDuration
 - GigEVisionPerformance.cpp, [1465](#)
- UseMaxFramerate
 - GigEVisionPerformance.cpp, [1465](#)
- UserOutputSelector
 - Spinnaker::Camera, [648](#)
- UserOutputSelectorEnums
 - CameraDefs Class, [166](#)
- UserOutputValue
 - Spinnaker::Camera, [648](#)
- UserOutputValueAll
 - Spinnaker::Camera, [649](#)
- UserOutputValueAllMask
 - Spinnaker::Camera, [649](#)
- UserSetDefault
 - Spinnaker::Camera, [649](#)
- UserSetDefaultEnums
 - CameraDefs Class, [166](#)
- UserSetFeatureEnable
 - Spinnaker::Camera, [649](#)
- UserSetFramerate
 - GigEVisionPerformance.cpp, [1465](#)
- UserSetLoad
 - Spinnaker::Camera, [649](#)
- UserSetSave
 - Spinnaker::Camera, [650](#)
- UserSetSelector
 - Spinnaker::Camera, [650](#)
- UserSetSelectorEnums
 - CameraDefs Class, [167](#)
- V3_3Enable
 - Spinnaker::Camera, [650](#)
- ValidateIpAddress
 - AdapterConfig, [417](#)
- ValueNode, [1205](#)
 - Spinnaker::GenApi::ValueNode, [1206](#), [1207](#)
- ValueNode Class, [410](#)
 - CValueRef, [410](#)
- Varying
 - Types Enums, [408](#)
- Verify
 - IBoolean Interface, [312](#)
- Version_t, [1208](#)
- videoType
 - SaveToAvi.cpp, [1490](#)
- WaitForImages
 - ImageEvents.cpp, [1467](#)
- WaitOnImageEvent

- Spinnaker::IDataStream, [908](#)
- what
 - Spinnaker::Exception, [841](#)
- WhiteClip
 - Spinnaker::Camera, [650](#)
- WhiteClipSelector
 - Spinnaker::Camera, [650](#)
- WhiteClipSelectorEnums
 - CameraDefs Class, [167](#)
- Width
 - Spinnaker::Camera, [651](#)
- width
 - Spinnaker::Video::H264Option, [879](#)
- WidthMax
 - Spinnaker::Camera, [651](#)
- WO
 - Types Enums, [403](#)
- Write
 - IPort Interface, [356](#)
 - Spinnaker::GenApi::CChunkPort, [693](#)
 - Spinnaker::GenApi::CEventPort, [716](#)
 - Spinnaker::GenApi::CPortImpl, [780](#)
 - Spinnaker::GenApi::CPortWriteList, [783](#)
 - Spinnaker::GenApi::CRegisterPortImpl, [787](#)
 - Spinnaker::GenApi::CTestPortStruct, [794](#)
 - Spinnaker::GenApi::PortNode, [1118](#)
 - Spinnaker::GenApi::PortRecorder, [1121](#)
 - Spinnaker::GenApi::PortReplay, [1125](#)
- write
 - Spinnaker::GenApi::FileProtocolAdapter, [846](#)
- WritePort
 - Spinnaker::CameraBase, [666](#)
 - Spinnaker::ICameraBase, [887](#)
- WriteRegister
 - Spinnaker::GenApi::CRegisterPortImpl, [788](#)
- xspuIn
 - Spinnaker::GenApi::ODevFileStreamBuf, [1110](#)
- Yes
 - Types Enums, [409](#)
- z_numTriggers
 - BufferHandling.cpp, [1438](#)