

Spinnaker C++

1.8.0.0

Generated by Doxygen 1.8.13

Contents

1	Introduction	1
2	Software Licensing Information	3
3	Module Index	5
3.1	Modules	5
4	Namespace Index	9
4.1	Namespace List	9
5	Hierarchical Index	11
5.1	Class Hierarchy	11
6	Class Index	17
6.1	Class List	17
7	File Index	23
7.1	File List	23
8	Module Documentation	27
8.1	Spinnaker Event Classes	28
8.1.1	Detailed Description	29
8.2	ArrivalEvent Class	30
8.2.1	Detailed Description	30
8.3	Spinnaker Classes	31
8.3.1	Detailed Description	34
8.3.2	Enumeration Type Documentation	34

8.3.2.1	InferenceBoxType	34
8.4	AVI Recorder Class	35
8.4.1	Detailed Description	35
8.4.2	Function Documentation	35
8.4.2.1	DEPRECATED_CLASS()	35
8.5	BasePtr Class	38
8.5.1	Detailed Description	38
8.6	Camera Class	39
8.6.1	Detailed Description	39
8.7	Camera Base Class	40
8.7.1	Detailed Description	40
8.8	CameraDefs Class	41
8.8.1	Detailed Description	73
8.8.2	Enumeration Type Documentation	73
8.8.2.1	AcquisitionModeEnums	73
8.8.2.2	AcquisitionStatusSelectorEnums	73
8.8.2.3	ActionUnconditionalModeEnums	74
8.8.2.4	AdcBitDepthEnums	74
8.8.2.5	AutoAlgorithmSelectorEnums	74
8.8.2.6	AutoExposureControlPriorityEnums	75
8.8.2.7	AutoExposureLightingModeEnums	75
8.8.2.8	AutoExposureMeteringModeEnums	75
8.8.2.9	AutoExposureTargetGreyValueAutoEnums	76
8.8.2.10	BalanceRatioSelectorEnums	76
8.8.2.11	BalanceWhiteAutoEnums	77
8.8.2.12	BalanceWhiteAutoProfileEnums	77
8.8.2.13	BinningHorizontalModeEnums	77
8.8.2.14	BinningSelectorEnums	78
8.8.2.15	BinningVerticalModeEnums	78
8.8.2.16	BlackLevelAutoBalanceEnums	78

8.8.2.17	BlackLevelAutoEnums	79
8.8.2.18	BlackLevelSelectorEnums	79
8.8.2.19	ChunkBlackLevelSelectorEnums	79
8.8.2.20	ChunkCounterSelectorEnums	80
8.8.2.21	ChunkEncoderSelectorEnums	80
8.8.2.22	ChunkEncoderStatusEnums	80
8.8.2.23	ChunkExposureTimeSelectorEnums	80
8.8.2.24	ChunkGainSelectorEnums	81
8.8.2.25	ChunkImageComponentEnums	81
8.8.2.26	ChunkPixelFormatEnums	82
8.8.2.27	ChunkRegionIDEnums	82
8.8.2.28	ChunkScan3dCoordinateReferenceSelectorEnums	83
8.8.2.29	ChunkScan3dCoordinateSelectorEnums	83
8.8.2.30	ChunkScan3dCoordinateSystemEnums	83
8.8.2.31	ChunkScan3dCoordinateSystemReferenceEnums	84
8.8.2.32	ChunkScan3dCoordinateTransformSelectorEnums	84
8.8.2.33	ChunkScan3dDistanceUnitEnums	84
8.8.2.34	ChunkScan3dOutputModeEnums	85
8.8.2.35	ChunkSelectorEnums	86
8.8.2.36	ChunkSourceIDEnums	86
8.8.2.37	ChunkTimerSelectorEnums	86
8.8.2.38	ChunkTransferStreamIDEnums	87
8.8.2.39	CIConfigurationEnums	87
8.8.2.40	CITimeSlotsCountEnums	88
8.8.2.41	ColorTransformationSelectorEnums	88
8.8.2.42	ColorTransformationValueSelectorEnums	88
8.8.2.43	CounterEventActivationEnums	89
8.8.2.44	CounterEventSourceEnums	89
8.8.2.45	CounterResetActivationEnums	90
8.8.2.46	CounterResetSourceEnums	90

8.8.2.47	CounterSelectorEnums	91
8.8.2.48	CounterStatusEnums	91
8.8.2.49	CounterTriggerActivationEnums	91
8.8.2.50	CounterTriggerSourceEnums	92
8.8.2.51	CxpConnectionTestModeEnums	92
8.8.2.52	CxpLinkConfigurationEnums	92
8.8.2.53	CxpLinkConfigurationPreferredEnums	93
8.8.2.54	CxpLinkConfigurationStatusEnums	94
8.8.2.55	CxpPoCxpStatusEnums	95
8.8.2.56	DecimationHorizontalModeEnums	96
8.8.2.57	DecimationSelectorEnums	96
8.8.2.58	DecimationVerticalModeEnums	96
8.8.2.59	DefectCorrectionModeEnums	97
8.8.2.60	DeinterlacingEnums	97
8.8.2.61	DeviceCharacterSetEnums	97
8.8.2.62	DeviceClockSelectorEnums	98
8.8.2.63	DeviceConnectionStatusEnums	98
8.8.2.64	DeviceIndicatorModeEnums	98
8.8.2.65	DeviceLinkHeartbeatModeEnums	98
8.8.2.66	DeviceLinkThroughputLimitModeEnums	100
8.8.2.67	DevicePowerSupplySelectorEnums	100
8.8.2.68	DeviceRegistersEndiannessEnums	100
8.8.2.69	DeviceScanTypeEnums	101
8.8.2.70	DeviceSerialPortBaudRateEnums	101
8.8.2.71	DeviceSerialPortSelectorEnums	101
8.8.2.72	DeviceStreamChannelEndiannessEnums	102
8.8.2.73	DeviceStreamChannelTypeEnums	102
8.8.2.74	DeviceTapGeometryEnums	102
8.8.2.75	DeviceTemperatureSelectorEnums	103
8.8.2.76	DeviceTLTypeEnums	104

8.8.2.77	DeviceTypeEnums	104
8.8.2.78	EncoderModeEnums	104
8.8.2.79	EncoderOutputModeEnums	105
8.8.2.80	EncoderResetActivationEnums	105
8.8.2.81	EncoderResetSourceEnums	106
8.8.2.82	EncoderSelectorEnums	107
8.8.2.83	EncoderSourceAEnums	107
8.8.2.84	EncoderSourceBEnums	107
8.8.2.85	EncoderStatusEnums	108
8.8.2.86	EventNotificationEnums	108
8.8.2.87	EventSelectorEnums	108
8.8.2.88	ExposureActiveModeEnums	109
8.8.2.89	ExposureAutoEnums	109
8.8.2.90	ExposureModeEnums	109
8.8.2.91	ExposureTimeModeEnums	110
8.8.2.92	ExposureTimeSelectorEnums	110
8.8.2.93	FileOpenModeEnums	111
8.8.2.94	FileOperationSelectorEnums	111
8.8.2.95	FileOperationStatusEnums	111
8.8.2.96	FileSelectorEnums	112
8.8.2.97	GainAutoBalanceEnums	112
8.8.2.98	GainAutoEnums	112
8.8.2.99	GainSelectorEnums	113
8.8.2.100	GevCCPEnums	113
8.8.2.101	GevCurrentPhysicalLinkConfigurationEnums	113
8.8.2.102	GevGVCPExtendedStatusCodesSelectorEnums	114
8.8.2.103	GevGVSPExtendedIDModeEnums	114
8.8.2.104	GevIEEE1588ClockAccuracyEnums	114
8.8.2.105	GevIEEE1588ModeEnums	114
8.8.2.106	GevIEEE1588StatusEnums	115

8.8.2.107 GevIPConfigurationStatusEnums	115
8.8.2.108 GevPhysicalLinkConfigurationEnums	116
8.8.2.109 GevSupportedOptionSelectorEnums	116
8.8.2.110 ImageComponentSelectorEnums	117
8.8.2.111 ImageCompressionJPEGFormatOptionEnums	117
8.8.2.112 ImageCompressionModeEnums	118
8.8.2.113 ImageCompressionRateOptionEnums	118
8.8.2.114 LineFormatEnums	118
8.8.2.115 LineInputFilterSelectorEnums	119
8.8.2.116 LineModeEnums	119
8.8.2.117 LineSelectorEnums	119
8.8.2.118 LineSourceEnums	120
8.8.2.119 LogicBlockLUTInputActivationEnums	120
8.8.2.120 LogicBlockLUTInputSelectorEnums	121
8.8.2.121 LogicBlockLUTInputSourceEnums	121
8.8.2.122 LogicBlockLUTSelectorEnums	122
8.8.2.123 LogicBlockSelectorEnums	122
8.8.2.124 LUTSelectorEnums	122
8.8.2.125 PixelColorFilterEnums	123
8.8.2.126 PixelFormatEnums	123
8.8.2.127 PixelFormatInfoSelectorEnums	129
8.8.2.128 PixelSizeEnums	134
8.8.2.129 RegionDestinationEnums	135
8.8.2.130 RegionModeEnums	135
8.8.2.131 RegionSelectorEnums	136
8.8.2.132 RgbTransformLightSourceEnums	136
8.8.2.133 Scan3dCoordinateReferenceSelectorEnums	137
8.8.2.134 Scan3dCoordinateSelectorEnums	137
8.8.2.135 Scan3dCoordinateSystemEnums	137
8.8.2.136 Scan3dCoordinateSystemReferenceEnums	138

8.8.2.137 Scan3dCoordinateTransformSelectorEnums	138
8.8.2.138 Scan3dDistanceUnitEnums	138
8.8.2.139 Scan3dOutputModeEnums	139
8.8.2.140 SensorDigitizationTapsEnums	139
8.8.2.141 SensorShutterModeEnums	140
8.8.2.142 SensorTapsEnums	140
8.8.2.143 SequencerConfigurationModeEnums	141
8.8.2.144 SequencerConfigurationValidEnums	141
8.8.2.145 SequencerModeEnums	141
8.8.2.146 SequencerSetValidEnums	141
8.8.2.147 SequencerTriggerActivationEnums	142
8.8.2.148 SequencerTriggerSourceEnums	142
8.8.2.149 SerialPortBaudRateEnums	142
8.8.2.150 SerialPortParityEnums	143
8.8.2.151 SerialPortSelectorEnums	143
8.8.2.152 SerialPortSourceEnums	144
8.8.2.153 SerialPortStopBitsEnums	144
8.8.2.154 SoftwareSignalSelectorEnums	144
8.8.2.155 SourceSelectorEnums	145
8.8.2.156 TestPatternEnums	145
8.8.2.157 TestPatternGeneratorSelectorEnums	145
8.8.2.158 TimerSelectorEnums	146
8.8.2.159 TimerStatusEnums	146
8.8.2.160 TimerTriggerActivationEnums	146
8.8.2.161 TimerTriggerSourceEnums	147
8.8.2.162 TransferComponentSelectorEnums	148
8.8.2.163 TransferControlModeEnums	148
8.8.2.164 TransferOperationModeEnums	149
8.8.2.165 TransferQueueModeEnums	149
8.8.2.166 TransferSelectorEnums	149

8.8.2.167 TransferStatusSelectorEnums	150
8.8.2.168 TransferTriggerActivationEnums	150
8.8.2.169 TransferTriggerModeEnums	150
8.8.2.170 TransferTriggerSelectorEnums	151
8.8.2.171 TransferTriggerSourceEnums	151
8.8.2.172 TriggerActivationEnums	152
8.8.2.173 TriggerModeEnums	153
8.8.2.174 TriggerOverlapEnums	153
8.8.2.175 TriggerSelectorEnums	153
8.8.2.176 TriggerSourceEnums	154
8.8.2.177 UserOutputSelectorEnums	154
8.8.2.178 UserSetDefaultEnums	154
8.8.2.179 UserSetSelectorEnums	155
8.8.2.180 WhiteClipSelectorEnums	155
8.9 Camera List Class	156
8.9.1 Detailed Description	156
8.10 CameraPtr Class	157
8.10.1 Detailed Description	157
8.10.2 Function Documentation	157
8.10.2.1 CameraPtr() [1/4]	157
8.10.2.2 CameraPtr() [2/4]	157
8.10.2.3 CameraPtr() [3/4]	158
8.10.2.4 CameraPtr() [4/4]	158
8.11 ChunkData Class	159
8.11.1 Detailed Description	159
8.12 Chunk Data Inference Class	160
8.12.1 Detailed Description	161
8.12.2 Function Documentation	161
8.12.2.1 GetBoxAt()	161
8.12.2.2 GetBoxCount()	161

8.12.2.3	GetBoxSize()	161
8.12.2.4	GetVersion()	162
8.12.2.5	InferenceBoundingBoxResult() [1/3]	162
8.12.2.6	InferenceBoundingBoxResult() [2/3]	162
8.12.2.7	InferenceBoundingBoxResult() [3/3]	162
8.12.2.8	operator=()	162
8.12.2.9	~InferenceBoundingBoxResult()	163
8.12.3	Variable Documentation	163
8.12.3.1	bottomRightXCoord [1/2]	163
8.12.3.2	bottomRightXCoord [2/2]	163
8.12.3.3	bottomRightYCoord [1/2]	163
8.12.3.4	bottomRightYCoord [2/2]	163
8.12.3.5	boxType	163
8.12.3.6	centerXCoord	163
8.12.3.7	centerYCoord	164
8.12.3.8	circle	164
8.12.3.9	classId	164
8.12.3.10	confidence	164
8.12.3.11	radius	164
8.12.3.12	rect	164
8.12.3.13	rotatedRect	164
8.12.3.14	rotationAngle	164
8.12.3.15	topLeftXCoord [1/2]	165
8.12.3.16	topLeftXCoord [2/2]	165
8.12.3.17	topLeftYCoord [1/2]	165
8.12.3.18	topLeftYCoord [2/2]	165
8.13	DeviceEvent Class	166
8.13.1	Detailed Description	166
8.14	Event Class	167
8.14.1	Detailed Description	167

8.15 Exception Class	168
8.15.1 Detailed Description	168
8.16 Image Class	169
8.16.1 Detailed Description	169
8.17 ImageEvent Class	170
8.17.1 Detailed Description	170
8.18 ImagePtr Class	171
8.18.1 Detailed Description	171
8.19 ImageStatistics Class	172
8.19.1 Detailed Description	172
8.20 Image Utility Class	173
8.20.1 Detailed Description	173
8.21 Image Utility Heatmap Class	174
8.21.1 Detailed Description	174
8.22 Image Utility Polarization Class	175
8.22.1 Detailed Description	175
8.23 Interface Class	176
8.23.1 Detailed Description	176
8.24 InterfaceArrivalEvent Class	177
8.24.1 Detailed Description	177
8.25 InterfaceEvent Class	178
8.25.1 Detailed Description	178
8.26 InterfaceList Class	179
8.26.1 Detailed Description	179
8.27 InterfacePtr Class	180
8.27.1 Detailed Description	180
8.28 InterfaceRemovalEvent Class	181
8.28.1 Detailed Description	181
8.29 LoggingEvent Class	182
8.29.1 Detailed Description	182

8.30 Logging Event Class	183
8.30.1 Detailed Description	183
8.31 LoggingEventDataPtr Class	184
8.31.1 Detailed Description	184
8.32 RemovalEvent Class	185
8.32.1 Detailed Description	185
8.33 Spinnaker Headers	186
8.33.1 Detailed Description	187
8.33.2 Variable Documentation	187
8.33.2.1 EVENT_TIMEOUT_INFINITE	187
8.33.2.2 EVENT_TIMEOUT_NONE	187
8.34 Spinnaker.h	188
8.35 Spinnaker Definitions	189
8.35.1 Detailed Description	192
8.35.2 Enumeration Type Documentation	193
8.35.2.1 ActionCommandStatus	193
8.35.2.2 BufferOwnership	193
8.35.2.3 ColorProcessingAlgorithm	193
8.35.2.4 Error	194
8.35.2.5 EventType	195
8.35.2.6 ImageFileFormat	195
8.35.2.7 ImageStatus	197
8.35.2.8 PayloadTypeInfoIDs	197
8.35.2.9 PixelFormatIntType	198
8.35.2.10 PixelFormatNamespaceID	198
8.35.2.11 SpinnakerLogLevel	199
8.35.2.12 StatisticsChannel	199
8.35.3 Function Documentation	200
8.35.3.1 DEPRECATED_ENUM() [1/2]	200
8.35.3.2 DEPRECATED_ENUM() [2/2]	201

8.35.4 Variable Documentation	201
8.35.4.1 HeatMapColor	201
8.36 Spinnaker Platform	202
8.36.1 Detailed Description	202
8.36.2 Macro Definition Documentation	202
8.36.2.1 SPINNAKER_API	202
8.36.2.2 SPINNAKER_API_ABSTRACT	202
8.36.2.3 SPINNAKER_LOCAL	202
8.37 Spinnaker Video Class	203
8.37.1 Detailed Description	203
8.38 Spinnaker Video Definitions	204
8.39 System Class	205
8.39.1 Detailed Description	205
8.40 SystemEvent Class	206
8.40.1 Detailed Description	206
8.41 SystemPtr Class	207
8.41.1 Detailed Description	207
8.42 Spinnaker QuickSpin Classes	208
8.42.1 Detailed Description	208
8.43 TransportLayerDefs Class	209
8.43.1 Detailed Description	210
8.43.2 Enumeration Type Documentation	210
8.43.2.1 DeviceAccessStatusEnum	211
8.43.2.2 DeviceCurrentSpeedEnum	211
8.43.2.3 DeviceEndianessMechanismEnum	211
8.43.2.4 DeviceTypeEnum	212
8.43.2.5 FilterDriverStatusEnum	212
8.43.2.6 GenICamXMLLocationEnum	212
8.43.2.7 GevCCPEnum	213
8.43.2.8 GUIXMLLocationEnum	213

8.43.2.9 POEStatusEnum	213
8.43.2.10 StreamBufferCountModeEnum	214
8.43.2.11 StreamBufferHandlingModeEnum	214
8.43.2.12 StreamDefaultBufferCountModeEnum	215
8.43.2.13 StreamTypeEnum	215
8.44 TransportLayerDevice Class	216
8.44.1 Detailed Description	216
8.45 TransportLayerInterface Class	217
8.45.1 Detailed Description	217
8.46 TransportLayerStream Class	218
8.46.1 Detailed Description	218
8.47 TransportLayerSystem Class	219
8.47.1 Detailed Description	219
8.48 Camera Base Interface Class	220
8.48.1 Detailed Description	220
8.49 IChunkData Class	221
8.49.1 Detailed Description	221
8.50 IImage Class	222
8.50.1 Detailed Description	222
8.51 IImageStatistics Class	223
8.51.1 Detailed Description	223
8.52 IInterface Class	224
8.52.1 Detailed Description	224
8.53 IInterfaceList Class	225
8.53.1 Detailed Description	225
8.54 ISystem Class	226
8.54.1 Detailed Description	226
8.55 Spinnaker GenApi Classes	227
8.55.1 Detailed Description	233
8.55.2 Typedef Documentation	233

8.55.2.1 CNodeMapRef	233
8.55.2.2 CNodeRef	233
8.55.2.3 CSelectorRef	233
8.55.3 Function Documentation	233
8.55.3.1 _ClearXMLCache()	233
8.55.3.2 _Connect() [1/2]	234
8.55.3.3 _Connect() [2/2]	234
8.55.3.4 _Destroy()	234
8.55.3.5 _GetDeviceName()	234
8.55.3.6 _GetNode()	234
8.55.3.7 _GetNodes()	234
8.55.3.8 _GetSupportedSchemaVersions()	234
8.55.3.9 _InvalidateNodes()	235
8.55.3.10 _LoadXMLFromFile()	235
8.55.3.11 _LoadXMLFromFileInject()	235
8.55.3.12 _LoadXMLFromString()	235
8.55.3.13 _LoadXMLFromStringInject()	235
8.55.3.14 _LoadXMLFromZIPData()	235
8.55.3.15 _LoadXMLFromZIPFile()	235
8.55.3.16 _Poll()	236
8.55.3.17 CastToIDestroy()	236
8.55.3.18 CNodeMapRefT() [1/3]	236
8.55.3.19 CNodeMapRefT() [2/3]	236
8.55.3.20 CNodeMapRefT() [3/3]	236
8.55.3.21 EatComments()	236
8.55.3.22 operator<<()	237
8.55.3.23 operator=() [1/2]	237
8.55.3.24 operator=() [2/2]	237
8.55.3.25 operator>>()	237
8.55.3.26 ~CNodeMapRefT()	237

8.56 AutoVector Class	238
8.56.1 Detailed Description	238
8.57 Spinnaker GenApi Interfaces	239
8.57.1 Detailed Description	240
8.57.2 Typedef Documentation	240
8.57.2.1 CallbackHandleType	240
8.57.2.2 NodeList_t	240
8.58 IBase Interface	241
8.58.1 Detailed Description	241
8.58.2 Variable Documentation	241
8.58.2.1 IBase	241
8.59 BooleanNode Class	242
8.59.1 Detailed Description	242
8.59.2 Typedef Documentation	242
8.59.2.1 CBooleanRef	242
8.60 CategoryNode Class	243
8.60.1 Detailed Description	243
8.60.2 Typedef Documentation	243
8.60.2.1 CCategoryRef	243
8.61 ChunkAdapter Class	244
8.61.1 Detailed Description	244
8.62 ChunkAdapterDcam Class	245
8.62.1 Detailed Description	245
8.63 ChunkAdapterGeneric Class	246
8.63.1 Detailed Description	246
8.64 ChunkAdapterGEV Class	247
8.64.1 Detailed Description	247
8.65 ChunkPort Class	248
8.65.1 Detailed Description	248
8.66 CommandNode Class	249

8.66.1 Detailed Description	249
8.66.2 Typedef Documentation	249
8.66.2.1 CCommandRef	249
8.67 Container Class	250
8.68 Counter Class	251
8.68.1 Detailed Description	251
8.69 EnumClasses Class	252
8.69.1 Detailed Description	253
8.70 EnumEntryNode Class	254
8.70.1 Detailed Description	254
8.70.2 Typedef Documentation	254
8.70.2.1 CEnumEntryRef	254
8.71 EnumNode Class	255
8.71.1 Detailed Description	255
8.71.2 Typedef Documentation	255
8.71.2.1 CEnumerationRef	255
8.72 EnumNodeT Class	256
8.72.1 Detailed Description	256
8.73 EventAdapter Class	257
8.73.1 Detailed Description	257
8.74 EventAdapter1394 Class	258
8.74.1 Detailed Description	258
8.75 EventAdapterGeneric Class	259
8.75.1 Detailed Description	259
8.76 EventAdapterGEV Class	260
8.76.1 Detailed Description	260
8.77 EventAdapterU3V Class	261
8.77.1 Detailed Description	261
8.78 EventPort Class	262
8.78.1 Detailed Description	262

8.79 Filestream Class	263
8.79.1 Detailed Description	263
8.80 FloatNode Class	264
8.80.1 Detailed Description	264
8.80.2 Typedef Documentation	264
8.80.2.1 CFloatRef	264
8.81 FloatRegNode Class	265
8.81.1 Detailed Description	265
8.82 GCString Class	266
8.82.1 Detailed Description	266
8.83 GCSynch Class	267
8.83.1 Detailed Description	267
8.84 GCTypes Class	268
8.84.1 Detailed Description	268
8.84.2 Typedef Documentation	268
8.84.2.1 float32_t	268
8.84.2.2 float64_t	268
8.85 Spinnaker GenApi Utilities	269
8.85.1 Detailed Description	269
8.86 GCUtility Utility	270
8.86.1 Detailed Description	271
8.86.2 Function Documentation	271
8.86.2.1 DoesEnvironmentVariableExist()	271
8.86.2.2 GetFiles()	271
8.86.2.3 GetGenICamCacheFolder()	271
8.86.2.4 GetGenICamCLProtocolFolder()	272
8.86.2.5 GetGenICamLogConfig()	272
8.86.2.6 GetModulePathFromFunction()	272
8.86.2.7 GetValueOfEnvironmentVariable() [1/2]	272
8.86.2.8 GetValueOfEnvironmentVariable() [2/2]	273

8.86.2.9 INTEGRAL_CAST()	273
8.86.2.10 INTEGRAL_CAST2()	273
8.86.2.11 ReplaceEnvironmentVariables()	273
8.86.2.12 SetGenICamCacheFolder()	274
8.86.2.13 SetGenICamCLProtocolFolder()	274
8.86.2.14 SetGenICamLogConfig()	274
8.86.2.15 Tokenize()	274
8.86.2.16 UrlDecode()	274
8.86.2.17 UrlEncode()	275
8.87 IBoolean Interface	276
8.87.1 Detailed Description	276
8.87.2 Function Documentation	276
8.87.2.1 GetValue()	276
8.87.2.2 operator()()	277
8.87.2.3 operator=()	277
8.87.3 Variable Documentation	277
8.87.3.1 IBoolean	277
8.87.3.2 Verify	277
8.88 ICategory Interfaces	278
8.88.1 Detailed Description	278
8.88.2 Variable Documentation	278
8.88.2.1 ICategory	278
8.89 IChunkPort Interface	279
8.89.1 Detailed Description	279
8.89.2 Macro Definition Documentation	279
8.89.2.1 CHUNK_BASE_ADDRESS_REGISTER	279
8.89.2.2 CHUNK_BASE_ADDRESS_REGISTER_LEN	280
8.89.2.3 CHUNK_LENGTH_REGISTER	280
8.89.2.4 CHUNK_LENGTH_REGISTER_LEN	280
8.89.3 Function Documentation	280

8.89.3.1 CacheChunkData()	280
8.89.4 Variable Documentation	280
8.89.4.1 IChunkPort	280
8.90 ICommand Interface	281
8.90.1 Detailed Description	281
8.90.2 Function Documentation	281
8.90.2.1 IsDone()	281
8.90.3 Variable Documentation	281
8.90.3.1 ICommand	282
8.91 IDestroy Interface	283
8.91.1 Detailed Description	283
8.91.2 Variable Documentation	283
8.91.2.1 IDestroy	283
8.92 IDeviceInfo Interface	284
8.92.1 Detailed Description	284
8.92.2 Function Documentation	284
8.92.2.1 GetDeviceVersion()	284
8.92.2.2 GetGenApiVersion()	285
8.92.2.3 GetProductGuid()	285
8.92.2.4 GetSchemaVersion()	285
8.92.2.5 GetStandardNameSpace()	285
8.92.2.6 GetToolTip()	285
8.92.2.7 GetVendorName()	285
8.92.2.8 GetVersionGuid()	286
8.92.3 Variable Documentation	286
8.92.3.1 IDeviceInfo	286
8.93 IEnumEntry Interface	287
8.93.1 Detailed Description	287
8.93.2 Function Documentation	287
8.93.2.1 GetNumericValue()	287

8.93.2.2 GetSymbolic()	287
8.93.2.3 IsSelfClearing()	288
8.93.3 Variable Documentation	288
8.93.3.1 IEnumEntry	288
8.94 IEnumeration Interface	289
8.94.1 Detailed Description	289
8.94.2 Function Documentation	289
8.94.2.1 GetCurrentEntry()	289
8.94.2.2 GetEntries()	290
8.94.2.3 GetEntry()	290
8.94.2.4 GetEntryByName()	290
8.94.2.5 GetIntValue()	290
8.94.2.6 operator*()	290
8.94.2.7 SetIntValue()	291
8.94.3 Variable Documentation	291
8.94.3.1 IEnumeration	291
8.95 IEnumerationT Interface	292
8.95.1 Detailed Description	292
8.95.2 Function Documentation	292
8.95.2.1 GetEntry()	292
8.95.2.2 operator=() [1/2]	293
8.95.2.3 operator=() [2/2]	293
8.95.3 Variable Documentation	293
8.95.3.1 IEnumerationT	293
8.95.3.2 IEnumReference	293
8.96 IFloat Interface	294
8.96.1 Detailed Description	295
8.96.2 Function Documentation	295
8.96.2.1 GetDisplayNotation()	295
8.96.2.2 GetDisplayPrecision()	295

8.96.2.3 GetInc()	295
8.96.2.4 GetIncMode()	295
8.96.2.5 GetListOfValidValues()	295
8.96.2.6 GetMax()	296
8.96.2.7 GetMin()	296
8.96.2.8 GetRepresentation()	296
8.96.2.9 GetUnit()	296
8.96.2.10 HasInc()	296
8.96.2.11 ImposeMax()	296
8.96.2.12 ImposeMin()	297
8.96.2.13 operator=(())	297
8.96.3 Variable Documentation	297
8.96.3.1 IFloat	297
8.97 IInteger Interface	298
8.97.1 Detailed Description	298
8.97.2 Function Documentation	298
8.97.2.1 ImposeMax()	298
8.97.2.2 ImposeMin()	298
8.97.2.3 operator=(())	299
8.97.3 Variable Documentation	299
8.97.3.1 IInteger	299
8.98 INode Interface	300
8.98.1 Detailed Description	302
8.98.2 Function Documentation	302
8.98.2.1 Combine() [1/3]	302
8.98.2.2 Combine() [2/3]	302
8.98.2.3 Combine() [3/3]	302
8.98.2.4 DeregisterCallback()	303
8.98.2.5 GetAlias()	303
8.98.2.6 GetCachingMode()	303

8.98.2.7 GetCastAlias()	303
8.98.2.8 GetChildren()	303
8.98.2.9 GetDescription()	304
8.98.2.10 GetDisplayName()	304
8.98.2.11 GetDocuURL()	304
8.98.2.12 GetEventID()	304
8.98.2.13 GetNameSpace()	304
8.98.2.14 GetNodeMap()	304
8.98.2.15 GetParents()	304
8.98.2.16 GetPollingTime()	305
8.98.2.17 GetPrincipalInterfaceType()	305
8.98.2.18 GetProperty()	305
8.98.2.19 GetPropertyNames()	305
8.98.2.20 GetVisibility()	305
8.98.2.21 ImposeAccessMode()	306
8.98.2.22 ImposeVisibility()	306
8.98.2.23 InvalidateNode()	306
8.98.2.24 IsAccessModeCacheable()	306
8.98.2.25 IsAvailable() [1/3]	306
8.98.2.26 IsAvailable() [2/3]	306
8.98.2.27 IsAvailable() [3/3]	307
8.98.2.28 IsCachable()	307
8.98.2.29 IsCacheable()	307
8.98.2.30 IsDeprecated()	307
8.98.2.31 IsFeature()	307
8.98.2.32 IsImplemented() [1/3]	307
8.98.2.33 IsImplemented() [2/3]	308
8.98.2.34 IsImplemented() [3/3]	308
8.98.2.35 IsReadable() [1/3]	308
8.98.2.36 IsReadable() [2/3]	308

8.98.2.37 IsReadable() [3/3]	308
8.98.2.38 IsStreamable()	308
8.98.2.39 IsVisible()	309
8.98.2.40 IsWritable() [1/3]	309
8.98.2.41 IsWritable() [2/3]	309
8.98.2.42 IsWritable() [3/3]	309
8.98.2.43 operator"!="	309
8.98.2.44 operator==()	309
8.98.2.45 RegisterCallback()	310
8.98.3 Variable Documentation	310
8.98.3.1 INode	310
8.98.3.2 IReference	310
8.99 INodeMap Interface	311
8.99.1 Detailed Description	311
8.99.2 Function Documentation	311
8.99.2.1 Connect() [1/2]	312
8.99.2.2 Connect() [2/2]	312
8.99.2.3 GetDeviceName()	312
8.99.2.4 GetLock()	312
8.99.2.5 GetNode()	312
8.99.2.6 GetNumNodes()	313
8.99.2.7 InvalidateNodes()	313
8.99.2.8 Poll()	313
8.99.3 Variable Documentation	313
8.99.3.1 INodeMap	313
8.100 INodeMapDyn Interface	314
8.100.1 Detailed Description	315
8.100.2 Function Documentation	315
8.100.2.1 ExtractIndependentSubtree()	315
8.100.2.2 GetSupportedSchemaVersions()	315

8.100.2.3 LoadXMLFromFile()	315
8.100.2.4 LoadXMLFromFileInject()	316
8.100.2.5 LoadXMLFromString()	316
8.100.2.6 LoadXMLFromStringInject()	316
8.100.2.7 LoadXMLFromZIPData()	316
8.100.2.8 LoadXMLFromZIPFile()	316
8.100.2.9 MergeXMLFiles()	316
8.100.2.10 PreprocessXMLFromFile()	317
8.100.2.11 PreprocessXMLFromZIPFile()	317
8.100.3 Variable Documentation	318
8.100.3.1 INodeMapDyn	318
8.101 IntegerNode Class	319
8.101.1 Detailed Description	319
8.101.2 Typedef Documentation	319
8.101.2.1 CIntegerRef	319
8.102 IntRegNode Class	320
8.102.1 Detailed Description	320
8.103 IPort Interface	321
8.103.1 Detailed Description	321
8.103.2 Function Documentation	321
8.103.2.1 Write()	321
8.103.3 Variable Documentation	321
8.103.3.1 Address	322
8.103.3.2 IPort	322
8.103.3.3 Length	322
8.104 IPortConstruct Interface	323
8.104.1 Detailed Description	323
8.104.2 Function Documentation	323
8.104.2.1 GetSwapEndianess()	323
8.104.3 Variable Documentation	323

8.104.3.1 IPortConstruct	323
8.105 IPortRecorder Interface	324
8.105.1 Detailed Description	324
8.105.2 Function Documentation	324
8.105.2.1 GetCookie()	324
8.105.2.2 Replay()	325
8.105.2.3 SetCookie()	325
8.105.2.4 StopRecording()	325
8.105.3 Variable Documentation	325
8.105.3.1 Invalidate	325
8.105.3.2 IPortRecorder	325
8.105.3.3 IPortReplay	325
8.105.3.4 IPortWriteList	325
8.106 IRegister Interfaces	326
8.106.1 Detailed Description	326
8.106.2 Function Documentation	326
8.106.2.1 Get()	326
8.106.2.2 GetAddress()	327
8.106.2.3 GetLength()	327
8.106.3 Variable Documentation	327
8.106.3.1 IRegister	327
8.107 ISelector Interface	328
8.107.1 Detailed Description	328
8.107.2 Function Documentation	328
8.107.2.1 GetSelectedFeatures()	328
8.107.2.2 GetSelectingFeatures()	328
8.107.3 Variable Documentation	328
8.107.3.1 ISelector	328
8.108 ISelectorDigit Interface	329
8.108.1 Detailed Description	329

8.108.2 Function Documentation	329
8.108.2.1 GetSelectorList()	329
8.108.2.2 Restore()	330
8.108.2.3 SetNext()	330
8.108.2.4 ToString()	330
8.108.3 Variable Documentation	330
8.108.3.1 ISelectorDigit	331
8.109 IString Class	332
8.109.1 Detailed Description	332
8.109.2 Function Documentation	332
8.109.2.1 GetMaxLength()	332
8.109.3 Variable Documentation	332
8.109.3.1 IString	332
8.110 IValue Class	333
8.110.1 Detailed Description	333
8.110.2 Function Documentation	333
8.110.2.1 FromString()	333
8.110.2.2 IsValueCacheValid()	334
8.110.2.3 ToString()	334
8.110.3 Variable Documentation	334
8.110.3.1 IValue	334
8.111 Node Class	335
8.111.1 Detailed Description	335
8.112 NodeCallback Class	336
8.112.1 Detailed Description	337
8.112.2 Enumeration Type Documentation	337
8.112.2.1 ECallbackType	337
8.112.3 Function Documentation	337
8.112.3.1 Deregister()	337
8.112.3.2 make_NodeCallback() [1/2]	337

8.112.3.3 make_NodeCallback() [2/2]	338
8.112.3.4 Register() [1/2]	338
8.112.3.5 Register() [2/2]	338
8.113NodeMap Class	339
8.113.1 Detailed Description	339
8.114NodeMapFactory Class	340
8.114.1 Detailed Description	340
8.114.2 Enumeration Type Documentation	340
8.114.2.1 ECacheUsage_t	340
8.114.2.2 EContentType_t	341
8.115NodeMapRef Class	342
8.115.1 Detailed Description	342
8.116Persistence Class	343
8.116.1 Detailed Description	343
8.117Pointer Class	344
8.117.1 Detailed Description	345
8.117.2 Typedef Documentation	345
8.117.2.1 CBasePtr	345
8.117.2.2 CBooleanPtr	346
8.117.2.3 CCategoryPtr	346
8.117.2.4 CChunkPortPtr	346
8.117.2.5 CCommandPtr	346
8.117.2.6 CDeviceInfoPtr	346
8.117.2.7 CEnumEntryPtr	346
8.117.2.8 CEnumerationPtr	347
8.117.2.9 CIntegerPtr	347
8.117.2.10CNodeMapDynPtr	347
8.117.2.11CNodeMapPtr	347
8.117.2.12CNodePtr	347
8.117.2.13CPortConstructPtr	347

8.117.2.14CPortPtr	348
8.117.2.15CPortRecorderPtr	348
8.117.2.16CPortReplayPtr	348
8.117.2.17CPortWriteListPtr	348
8.117.2.18CRegisterPtr	348
8.117.2.19CSelectorPtr	348
8.117.2.20CStringPtr	349
8.117.2.21CValuePtr	349
8.117.3 Function Documentation	349
8.117.3.1 GetInterfaceName()	349
8.117.3.2 IsAvailable()	349
8.117.3.3 IsImplemented()	349
8.117.3.4 IsReadable()	349
8.117.3.5 IsWritable()	349
8.118PortImpl Class	350
8.118.1 Detailed Description	350
8.119PortNode Class	351
8.119.1 Detailed Description	351
8.119.2 Typedef Documentation	351
8.119.2.1 CPortRef	351
8.120PortRecorder Class	352
8.120.1 Detailed Description	352
8.120.2 Typedef Documentation	352
8.120.2.1 CPortRecorderRef	352
8.121PortReplay Class	353
8.121.1 Detailed Description	353
8.122PortWriteList Class	354
8.122.1 Detailed Description	354
8.123Reference Interfaces	355
8.123.1 Detailed Description	355

8.123.2 Function Documentation	355
8.123.2.1 SetNumEnums()	355
8.124 RegisterNode Class	356
8.124.1 Detailed Description	356
8.124.2 Typedef Documentation	356
8.124.2.1 CRegisterRef	356
8.125 RegisterPortImpl Class	357
8.125.1 Detailed Description	357
8.126 SelectorSet Class	358
8.126.1 Detailed Description	358
8.127 SpinTestCamera Class	359
8.127.1 Detailed Description	359
8.128 StringNode Class	360
8.128.1 Detailed Description	360
8.128.2 Typedef Documentation	360
8.128.2.1 CStringRef	360
8.129 StringRegNode Class	361
8.129.1 Detailed Description	361
8.130 StructPort Class	362
8.130.1 Detailed Description	362
8.131 Synch Class	363
8.131.1 Detailed Description	363
8.132 Spinnaker GenApi Enums	364
8.132.1 Detailed Description	364
8.133 Types Enums	365
8.133.1 Detailed Description	367
8.133.2 Macro Definition Documentation	367
8.133.2.1 _UndefinedRepresentation	367
8.133.3 Typedef Documentation	367
8.133.3.1 StringList_t	368

8.133.4 Enumeration Type Documentation	368
8.133.4.1 EAccessMode	368
8.133.4.2 ECachingMode	368
8.133.4.3 EDisplayNotation	368
8.133.4.4 EEndianess	370
8.133.4.5 EGenApiSchemaVersion	370
8.133.4.6 EIncMode	370
8.133.4.7 EInputDirection	371
8.133.4.8 EInterfaceType	371
8.133.4.9 ELinkType	371
8.133.4.10 ENamespace	372
8.133.4.11 ERepresentation	372
8.133.4.12 ESign	372
8.133.4.13 ESlope	373
8.133.4.14 EStandardNameSpace	373
8.133.4.15 EVisibility	373
8.133.4.16 EXMLValidation	374
8.133.4.17 EYesNo	374
8.134 ValueNode Class	375
8.134.1 Detailed Description	375
8.134.2 Typedef Documentation	375
8.134.2.1 CValueRef	375
8.135 ChunkAdapterU3V Class	376
8.135.1 Detailed Description	376

9 Namespace Documentation	377
9.1 AdapterConfig Namespace Reference	377
9.1.1 Enumeration Type Documentation	378
9.1.1.1 AdapterConfigErr	378
9.1.2 Function Documentation	378
9.1.2.1 AutoPopulateAdapterInfo()	378
9.1.2.2 AutoPopulateAdvancedProperties()	379
9.1.2.3 ConfigureAdapter()	379
9.1.2.4 GetAuto10GDesc()	379
9.1.2.5 GetAutoGigabitDesc()	379
9.1.2.6 GetAutoStartGateway()	379
9.1.2.7 GetAutoStartIp()	379
9.1.2.8 GetAutoSubnetMask()	379
9.1.2.9 GetAutoSubnetMaskLength()	380
9.1.2.10 GetConfigLogFileName()	380
9.1.2.11 GetEnumerationLogFileName()	380
9.1.2.12 GetGatewayAddress()	380
9.1.2.13 GetMaxIpAddress()	380
9.1.2.14 GetMinIpAddress()	380
9.1.2.15 GetSubnetMaskLength()	380
9.1.2.16 IsOnSameSubnet()	381
9.1.2.17 IsValidIpAddress()	381
9.1.2.18 IsValidSubnetMask()	381
9.1.2.19 PopulateAdapterIpInfo()	381
9.1.2.20 RetrieveAllAdapters()	381
9.1.2.21 ValidateIpAddress()	381
9.2 Spinnaker Namespace Reference	382
9.3 Spinnaker::GenApi Namespace Reference	421
9.3.1 Typedef Documentation	435
9.3.1.1 IDevFileStream	435

9.3.1.2	ODevFileStream	436
9.3.2	Enumeration Type Documentation	436
9.3.2.1	GVCP_MESSAGE_TAGS	436
9.3.3	Function Documentation	436
9.3.3.1	PersistFeature()	436
9.3.3.2	SET_GUID()	436
9.3.4	Variable Documentation	437
9.3.4.1	COMMAND_MAGIC	437
9.3.4.2	GENCP_COMMAND_HEADER_SIZE	437
9.3.4.3	GENCP_EVENT_BASIC_SIZE	437
9.3.4.4	GENCP_EVENT_CMD_ID	437
9.3.4.5	IPersistScript	437
9.3.4.6	U3V_EVENT_PREFIX	437
9.4	Spinnaker::GenICam Namespace Reference	438
9.4.1	Function Documentation	439
9.4.1.1	getline() [1/2]	439
9.4.1.2	getline() [2/2]	439
9.4.1.3	ThrowBadAlloc()	439
9.5	Spinnaker::Video Namespace Reference	440
10	Class Documentation	441
10.1	ActionCommandResult Struct Reference	441
10.1.1	Detailed Description	441
10.1.2	Member Data Documentation	441
10.1.2.1	DeviceAddress	441
10.1.2.2	Status	441
10.2	AdapterConfigException Class Reference	442
10.2.1	Constructor & Destructor Documentation	442
10.2.1.1	AdapterConfigException() [1/2]	442
10.2.1.2	AdapterConfigException() [2/2]	443
10.2.2	Member Function Documentation	443

10.2.2.1 ErrCode()	443
10.2.2.2 GetParamStr()	443
10.3 AdapterInfo Struct Reference	443
10.3.1 Constructor & Destructor Documentation	444
10.3.1.1 AdapterInfo()	444
10.3.2 Member Data Documentation	444
10.3.2.1 adapterDescription	444
10.3.2.2 adapterGUID	444
10.3.2.3 adapterName	445
10.3.2.4 dhcpEnabled	445
10.3.2.5 ipInfo	445
10.3.2.6 jumboPackets	445
10.3.2.7 jumboPacketsRegKey	445
10.3.2.8 jumboPacketValidValues	445
10.3.2.9 receiveBuffers	445
10.3.2.10 receiveBuffersMax	445
10.3.2.11 receiveBuffersMin	446
10.3.2.12 receiveBuffersRegKey	446
10.3.2.13 receiveBuffersStep	446
10.3.2.14 transmitBuffers	446
10.3.2.15 transmitBuffersMax	446
10.3.2.16 transmitBuffersMin	446
10.3.2.17 transmitBuffersRegKey	446
10.3.2.18 transmitBuffersStep	447
10.4 ArrivalEvent Class Reference	447
10.4.1 Detailed Description	448
10.4.2 Constructor & Destructor Documentation	448
10.4.2.1 ArrivalEvent()	448
10.4.2.2 ~ArrivalEvent()	448
10.4.3 Member Function Documentation	448

10.4.3.1	OnDeviceArrival()	449
10.4.3.2	operator=()	449
10.5	AttachStatistics_t Struct Reference	449
10.5.1	Detailed Description	449
10.5.2	Member Data Documentation	449
10.5.2.1	NumAttachedChunks	450
10.5.2.2	NumChunkPorts	450
10.5.2.3	NumChunks	450
10.6	AutoLock Class Reference	450
10.6.1	Constructor & Destructor Documentation	450
10.6.1.1	AutoLock()	450
10.6.1.2	~AutoLock()	451
10.7	AutoLock Class Reference	451
10.7.1	Constructor & Destructor Documentation	451
10.7.1.1	AutoLock()	451
10.7.1.2	~AutoLock()	451
10.8	AVIOption Struct Reference	451
10.8.1	Detailed Description	452
10.8.2	Constructor & Destructor Documentation	452
10.8.2.1	AVIOption()	452
10.8.3	Member Data Documentation	452
10.8.3.1	frameRate	452
10.8.3.2	reserved	452
10.9	BasePtr< T, B > Class Template Reference	452
10.9.1	Detailed Description	453
10.9.2	Constructor & Destructor Documentation	453
10.9.2.1	BasePtr() [1/2]	454
10.9.2.2	~BasePtr()	454
10.9.2.3	BasePtr() [2/2]	454
10.9.3	Member Function Documentation	454

10.9.3.1 get()	454
10.9.3.2 IsValid()	454
10.9.3.3 operator bool()	455
10.9.3.4 operator T*()	455
10.9.3.5 operator->()	455
10.9.3.6 operator=() [1/4]	455
10.9.3.7 operator=() [2/4]	455
10.9.3.8 operator=() [3/4]	455
10.9.3.9 operator=() [4/4]	456
10.9.3.10 operator==() [1/4]	456
10.9.3.11 operator==() [2/4]	456
10.9.3.12 operator==() [3/4]	456
10.9.3.13 operator==() [4/4]	456
10.9.4 Member Data Documentation	456
10.9.4.1 m_pT	457
10.10 BMPOption Struct Reference	457
10.10.1 Detailed Description	457
10.10.2 Constructor & Destructor Documentation	457
10.10.2.1 BMPOption()	457
10.10.3 Member Data Documentation	457
10.10.3.1 indexedColor_8bit	458
10.10.3.2 reserved	458
10.11 BooleanNode Class Reference	458
10.11.1 Detailed Description	459
10.11.2 Constructor & Destructor Documentation	459
10.11.2.1 BooleanNode() [1/2]	460
10.11.2.2 BooleanNode() [2/2]	460
10.11.2.3 ~BooleanNode()	460
10.11.3 Member Function Documentation	460
10.11.3.1 GetValue()	460

10.11.3.2 operator=()	460
10.11.3.3 SetReference()	461
10.11.3.4 SetValue()	461
10.12 Camera Class Reference	461
10.12.1 Detailed Description	491
10.12.2 Constructor & Destructor Documentation	491
10.12.2.1 ~Camera()	491
10.12.2.2 Camera()	492
10.12.3 Member Function Documentation	492
10.12.3.1 Init()	492
10.12.4 Member Data Documentation	492
10.12.4.1 AasRoiEnable	492
10.12.4.2 AasRoiHeight	492
10.12.4.3 AasRoiOffsetX	493
10.12.4.4 AasRoiOffsetY	493
10.12.4.5 AasRoiWidth	493
10.12.4.6 AcquisitionAbort	493
10.12.4.7 AcquisitionArm	493
10.12.4.8 AcquisitionBurstFrameCount	494
10.12.4.9 AcquisitionFrameCount	494
10.12.4.10 AcquisitionFrameRate	494
10.12.4.11 AcquisitionFrameRateEnable	494
10.12.4.12 AcquisitionLineRate	494
10.12.4.13 AcquisitionMode	495
10.12.4.14 AcquisitionResultingFrameRate	495
10.12.4.15 AcquisitionStart	495
10.12.4.16 AcquisitionStatus	495
10.12.4.17 AcquisitionStatusSelector	495
10.12.4.18 AcquisitionStop	495
10.12.4.19 ActionDeviceKey	496

10.12.4.20ActionGroupKey	496
10.12.4.21ActionGroupMask	496
10.12.4.22ActionQueueSize	496
10.12.4.23ActionSelector	496
10.12.4.24ActionUnconditionalMode	496
10.12.4.25AdaptiveCompressionEnable	497
10.12.4.26AdcBitDepth	497
10.12.4.27aPAUSEMACCtrlFramesReceived	497
10.12.4.28aPAUSEMACCtrlFramesTransmitted	497
10.12.4.29AutoAlgorithmSelector	497
10.12.4.30AutoExposureControlLoopDamping	498
10.12.4.31AutoExposureControlPriority	498
10.12.4.32AutoExposureEVCompensation	498
10.12.4.33AutoExposureExposureTimeLowerLimit	499
10.12.4.34AutoExposureExposureTimeUpperLimit	499
10.12.4.35AutoExposureGainLowerLimit	499
10.12.4.36AutoExposureGainUpperLimit	499
10.12.4.37AutoExposureGreyValueLowerLimit	499
10.12.4.38AutoExposureGreyValueUpperLimit	500
10.12.4.39AutoExposureLightingMode	500
10.12.4.40AutoExposureMeteringMode	500
10.12.4.41AutoExposureTargetGreyValue	501
10.12.4.42AutoExposureTargetGreyValueAuto	501
10.12.4.43BalanceRatio	501
10.12.4.44BalanceRatioSelector	502
10.12.4.45BalanceWhiteAuto	502
10.12.4.46BalanceWhiteAutoDamping	502
10.12.4.47BalanceWhiteAutoLowerLimit	502
10.12.4.48BalanceWhiteAutoProfile	503
10.12.4.49BalanceWhiteAutoUpperLimit	503

10.12.4.50BinningHorizontal	503
10.12.4.51BinningHorizontalMode	503
10.12.4.52BinningSelector	503
10.12.4.53BinningVertical	504
10.12.4.54BinningVerticalMode	504
10.12.4.55BlackLevel	504
10.12.4.56BlackLevelAuto	504
10.12.4.57BlackLevelAutoBalance	504
10.12.4.58BlackLevelClampingEnable	505
10.12.4.59BlackLevelRaw	505
10.12.4.60BlackLevelSelector	505
10.12.4.61ChunkBlackLevel	505
10.12.4.62ChunkBlackLevelSelector	505
10.12.4.63ChunkCounterSelector	506
10.12.4.64ChunkCounterValue	506
10.12.4.65ChunkCRC	506
10.12.4.66ChunkEnable	506
10.12.4.67ChunkEncoderSelector	506
10.12.4.68ChunkEncoderStatus	506
10.12.4.69ChunkEncoderValue	507
10.12.4.70ChunkExposureEndLineStatusAll	507
10.12.4.71ChunkExposureTime	507
10.12.4.72ChunkExposureTimeSelector	507
10.12.4.73ChunkFrameID	507
10.12.4.74ChunkGain	507
10.12.4.75ChunkGainSelector	508
10.12.4.76ChunkHeight	508
10.12.4.77ChunkImage	508
10.12.4.78ChunkImageComponent	508
10.12.4.79ChunkInferenceBoundingBoxResult	508

10.12.4.80	ChunkInferenceConfidence	508
10.12.4.81	ChunkInferenceResult	509
10.12.4.82	ChunkLinePitch	509
10.12.4.83	ChunkLineStatusAll	509
10.12.4.84	ChunkModeActive	509
10.12.4.85	ChunkOffsetX	509
10.12.4.86	ChunkOffsetY	509
10.12.4.87	ChunkPartSelector	510
10.12.4.88	ChunkPixelDynamicRangeMax	510
10.12.4.89	ChunkPixelDynamicRangeMin	510
10.12.4.90	ChunkPixelFormat	510
10.12.4.91	ChunkRegionID	510
10.12.4.92	ChunkScan3dAxisMax	510
10.12.4.93	ChunkScan3dAxisMin	511
10.12.4.94	ChunkScan3dCoordinateOffset	511
10.12.4.95	ChunkScan3dCoordinateReferenceSelector	511
10.12.4.96	ChunkScan3dCoordinateReferenceValue	511
10.12.4.97	ChunkScan3dCoordinateScale	511
10.12.4.98	ChunkScan3dCoordinateSelector	511
10.12.4.99	ChunkScan3dCoordinateSystem	512
10.12.4.100	hunkScan3dCoordinateSystemReference	512
10.12.4.101	hunkScan3dCoordinateTransformSelector	512
10.12.4.102	hunkScan3dDistanceUnit	512
10.12.4.103	hunkScan3dInvalidDataFlag	512
10.12.4.104	hunkScan3dInvalidDataValue	512
10.12.4.105	hunkScan3dOutputMode	513
10.12.4.106	hunkScan3dTransformValue	513
10.12.4.107	hunkScanLineSelector	513
10.12.4.108	hunkSelector	513
10.12.4.109	hunkSequencerSetActive	513

10.12.4.110hunkSerialData	513
10.12.4.110hunkSerialDataLength	514
10.12.4.110hunkSerialReceiveOverflow	514
10.12.4.110hunkSourceID	514
10.12.4.110hunkStreamChannelID	514
10.12.4.110hunkTimerSelector	514
10.12.4.110hunkTimerValue	514
10.12.4.110hunkTimestamp	515
10.12.4.110hunkTimestampLatchValue	515
10.12.4.110hunkTransferBlockID	515
10.12.4.120hunkTransferQueueCurrentBlockCount	515
10.12.4.120hunkTransferStreamID	515
10.12.4.120hunkWidth	515
10.12.4.120lConfiguration	516
10.12.4.120lTimeSlotsCount	516
10.12.4.120lColorTransformationEnable	516
10.12.4.120lColorTransformationSelector	516
10.12.4.120lColorTransformationValue	516
10.12.4.120lColorTransformationValueSelector	517
10.12.4.120lCompressionRatio	517
10.12.4.130lCounterDelay	517
10.12.4.130lCounterDuration	517
10.12.4.130lCounterEventActivation	517
10.12.4.130lCounterEventSource	517
10.12.4.130lCounterReset	518
10.12.4.130lCounterResetActivation	518
10.12.4.130lCounterResetSource	518
10.12.4.130lCounterSelector	518
10.12.4.130lCounterStatus	518
10.12.4.130lCounterTriggerActivation	518

10.12.4.140CounterTriggerSource	519
10.12.4.140CounterValue	519
10.12.4.140CounterValueAtReset	519
10.12.4.140xpConnectionSelector	519
10.12.4.140xpConnectionTestErrorCount	519
10.12.4.140xpConnectionTestMode	519
10.12.4.140xpConnectionTestPacketCount	520
10.12.4.140xpLinkConfiguration	520
10.12.4.140xpLinkConfigurationPreferred	520
10.12.4.140xpLinkConfigurationStatus	520
10.12.4.150xpPoCxpAuto	520
10.12.4.150xpPoCxpStatus	520
10.12.4.150xpPoCxpTripReset	521
10.12.4.150xpPoCxpTurnOff	521
10.12.4.150DecimationHorizontal	521
10.12.4.150DecimationHorizontalMode	521
10.12.4.150DecimationSelector	522
10.12.4.150DecimationVertical	522
10.12.4.150DecimationVerticalMode	522
10.12.4.150DefectCorrectionMode	522
10.12.4.160DefectCorrectStaticEnable	523
10.12.4.160DefectTableApply	523
10.12.4.160DefectTableCoordinateX	523
10.12.4.160DefectTableCoordinateY	523
10.12.4.160DefectTableFactoryRestore	524
10.12.4.160DefectTableIndex	524
10.12.4.160DefectTablePixelCount	524
10.12.4.160DefectTableSave	524
10.12.4.160Beinterlacing	524
10.12.4.160DeviceCharacterSet	525

10.12.4.17DeviceClockFrequency	525
10.12.4.17DeviceClockSelector	525
10.12.4.17DeviceConnectionSelector	525
10.12.4.17DeviceConnectionSpeed	525
10.12.4.17DeviceConnectionStatus	525
10.12.4.17DeviceEventChannelCount	526
10.12.4.17DeviceFamilyName	526
10.12.4.17DeviceFeaturePersistenceEnd	526
10.12.4.17DeviceFeaturePersistenceStart	526
10.12.4.17DeviceFirmwareVersion	526
10.12.4.18DeviceGenCPVersionMajor	526
10.12.4.18DeviceGenCPVersionMinor	527
10.12.4.18DeviceID	527
10.12.4.18DeviceIndicatorMode	527
10.12.4.18DeviceLinkBandwidthReserve	527
10.12.4.18DeviceLinkCommandTimeout	527
10.12.4.18DeviceLinkConnectionCount	527
10.12.4.18DeviceLinkCurrentThroughput	528
10.12.4.18DeviceLinkHeartbeatMode	528
10.12.4.18DeviceLinkHeartbeatTimeout	528
10.12.4.19DeviceLinkSelector	528
10.12.4.19DeviceLinkSpeed	528
10.12.4.19DeviceLinkThroughputLimit	529
10.12.4.19DeviceLinkThroughputLimitMode	529
10.12.4.19DeviceManifestEntrySelector	529
10.12.4.19DeviceManifestPrimaryURL	529
10.12.4.19DeviceManifestSchemaMajorVersion	529
10.12.4.19DeviceManifestSchemaMinorVersion	530
10.12.4.19DeviceManifestSecondaryURL	530
10.12.4.19DeviceManifestXMLMajorVersion	530

10.12.4.20DeviceManifestXMLMinorVersion	530
10.12.4.20DeviceManifestXMLSubMinorVersion	530
10.12.4.20DeviceManufacturerInfo	530
10.12.4.20DeviceMaxThroughput	531
10.12.4.20DeviceModelName	531
10.12.4.20DevicePowerSupplySelector	531
10.12.4.20DeviceRegistersCheck	531
10.12.4.20DeviceRegistersEndianness	531
10.12.4.20DeviceRegistersStreamingEnd	532
10.12.4.20DeviceRegistersStreamingStart	532
10.12.4.21DeviceRegistersValid	532
10.12.4.21DeviceReset	532
10.12.4.21DeviceScanType	532
10.12.4.21DeviceSerialNumber	532
10.12.4.21DeviceSerialPortBaudRate	533
10.12.4.21DeviceSerialPortSelector	533
10.12.4.21DeviceSFNCVersionMajor	533
10.12.4.21DeviceSFNCVersionMinor	533
10.12.4.21DeviceStreamChannelCount	533
10.12.4.22DeviceStreamChannelEndianness	534
10.12.4.22DeviceStreamChannelLink	534
10.12.4.22DeviceStreamChannelPacketSize	534
10.12.4.22DeviceStreamChannelSelector	534
10.12.4.22DeviceStreamChannelType	534
10.12.4.22DeviceTapGeometry	534
10.12.4.22DeviceTemperature	535
10.12.4.22DeviceTemperatureSelector	535
10.12.4.22DeviceTLType	535
10.12.4.22DeviceTLVersionMajor	535

10.12.4.23 DeviceTLVersionMinor	535
10.12.4.23 DeviceTLVersionSubMinor	536
10.12.4.23 DeviceType	536
10.12.4.23 DeviceUptime	536
10.12.4.23 DeviceUserID	536
10.12.4.23 DeviceVendorName	536
10.12.4.23 DeviceVersion	536
10.12.4.23 EncoderDivider	537
10.12.4.23 EncoderMode	537
10.12.4.23 EncoderOutputMode	537
10.12.4.24 EncoderReset	537
10.12.4.24 EncoderResetActivation	537
10.12.4.24 EncoderResetSource	537
10.12.4.24 EncoderSelector	538
10.12.4.24 EncoderSourceA	538
10.12.4.24 EncoderSourceB	538
10.12.4.24 EncoderStatus	538
10.12.4.24 EncoderTimeout	538
10.12.4.24 EncoderValue	538
10.12.4.24 EncoderValueAtReset	539
10.12.4.25 EnumerationCount	539
10.12.4.25 EventAcquisitionEnd	539
10.12.4.25 EventAcquisitionEndFrameID	539
10.12.4.25 EventAcquisitionEndTimestamp	539
10.12.4.25 EventAcquisitionError	539
10.12.4.25 EventAcquisitionErrorFrameID	540
10.12.4.25 EventAcquisitionErrorTimestamp	540
10.12.4.25 EventAcquisitionStart	540
10.12.4.25 EventAcquisitionStartFrameID	540
10.12.4.25 EventAcquisitionStartTimestamp	540

10.12.4.26EventAcquisitionTransferEnd	540
10.12.4.26EventAcquisitionTransferEndFrameID	541
10.12.4.26EventAcquisitionTransferEndTimestamp	541
10.12.4.26EventAcquisitionTransferStart	541
10.12.4.26EventAcquisitionTransferStartFrameID	541
10.12.4.26EventAcquisitionTransferStartTimestamp	541
10.12.4.26EventAcquisitionTrigger	541
10.12.4.26EventAcquisitionTriggerFrameID	542
10.12.4.26EventAcquisitionTriggerTimestamp	542
10.12.4.26EventActionLate	542
10.12.4.27EventActionLateFrameID	542
10.12.4.27EventActionLateTimestamp	542
10.12.4.27EventCounter0End	542
10.12.4.27EventCounter0EndFrameID	543
10.12.4.27EventCounter0EndTimestamp	543
10.12.4.27EventCounter0Start	543
10.12.4.27EventCounter0StartFrameID	543
10.12.4.27EventCounter0StartTimestamp	543
10.12.4.27EventCounter1End	543
10.12.4.27EventCounter1EndFrameID	544
10.12.4.27EventCounter1EndTimestamp	544
10.12.4.27EventCounter1Start	544
10.12.4.27EventCounter1StartFrameID	544
10.12.4.27EventCounter1StartTimestamp	544
10.12.4.28EventEncoder0Restarted	544
10.12.4.28EventEncoder0RestartedFrameID	545
10.12.4.28EventEncoder0RestartedTimestamp	545
10.12.4.28EventEncoder0Stopped	545
10.12.4.28EventEncoder0StoppedFrameID	545
10.12.4.28EventEncoder0StoppedTimestamp	545

10.12.4.29EventEncoder1Restarted	545
10.12.4.29EventEncoder1RestartedFrameID	546
10.12.4.29EventEncoder1RestartedTimestamp	546
10.12.4.29EventEncoder1Stopped	546
10.12.4.29EventEncoder1StoppedFrameID	546
10.12.4.29EventEncoder1StoppedTimestamp	546
10.12.4.29EventError	546
10.12.4.29EventErrorCode	547
10.12.4.29EventErrorFrameID	547
10.12.4.29EventErrorTimestamp	547
10.12.4.30EventExposureEnd	547
10.12.4.30EventExposureEndFrameID	547
10.12.4.30EventExposureEndTimestamp	547
10.12.4.30EventExposureStart	548
10.12.4.30EventExposureStartFrameID	548
10.12.4.30EventExposureStartTimestamp	548
10.12.4.30EventFrameBurstEnd	548
10.12.4.30EventFrameBurstEndFrameID	548
10.12.4.30EventFrameBurstEndTimestamp	548
10.12.4.30EventFrameBurstStart	549
10.12.4.31EventFrameBurstStartFrameID	549
10.12.4.31EventFrameBurstStartTimestamp	549
10.12.4.31EventFrameEnd	549
10.12.4.31EventFrameEndFrameID	549
10.12.4.31EventFrameEndTimestamp	549
10.12.4.31EventFrameStart	550
10.12.4.31EventFrameStartFrameID	550
10.12.4.31EventFrameStartTimestamp	550
10.12.4.31EventFrameTransferEnd	550
10.12.4.31EventFrameTransferEndFrameID	550

10.12.4.32EventFrameTransferEndTimestamp	550
10.12.4.32EventFrameTransferStart	551
10.12.4.32EventFrameTransferStartFrameID	551
10.12.4.32EventFrameTransferStartTimestamp	551
10.12.4.32EventFrameTrigger	551
10.12.4.32EventFrameTriggerFrameID	551
10.12.4.32EventFrameTriggerTimestamp	551
10.12.4.32EventLine0AnyEdge	552
10.12.4.32EventLine0AnyEdgeFrameID	552
10.12.4.32EventLine0AnyEdgeTimestamp	552
10.12.4.32EventLine0FallingEdge	552
10.12.4.32EventLine0FallingEdgeFrameID	552
10.12.4.32EventLine0FallingEdgeTimestamp	552
10.12.4.32EventLine0RisingEdge	553
10.12.4.32EventLine0RisingEdgeFrameID	553
10.12.4.32EventLine0RisingEdgeTimestamp	553
10.12.4.32EventLine1AnyEdge	553
10.12.4.32EventLine1AnyEdgeFrameID	553
10.12.4.32EventLine1AnyEdgeTimestamp	553
10.12.4.32EventLine1FallingEdge	554
10.12.4.32EventLine1FallingEdgeFrameID	554
10.12.4.32EventLine1FallingEdgeTimestamp	554
10.12.4.32EventLine1RisingEdge	554
10.12.4.32EventLine1RisingEdgeFrameID	554
10.12.4.32EventLine1RisingEdgeTimestamp	554
10.12.4.32EventLinkSpeedChange	555
10.12.4.32EventLinkSpeedChangeFrameID	555
10.12.4.32EventLinkSpeedChangeTimestamp	555
10.12.4.32EventLinkTrigger0	555
10.12.4.32EventLinkTrigger0FrameID	555

10.12.4.35EventLinkTrigger0Timestamp	555
10.12.4.35EventLinkTrigger1	556
10.12.4.35EventLinkTrigger1FrameID	556
10.12.4.35EventLinkTrigger1Timestamp	556
10.12.4.35EventNotification	556
10.12.4.35EventSelector	556
10.12.4.35EventSequencerSetChange	556
10.12.4.35EventSequencerSetChangeFrameID	557
10.12.4.35EventSequencerSetChangeTimestamp	557
10.12.4.35EventSerialData	557
10.12.4.36EventSerialDataLength	557
10.12.4.36EventSerialPortReceive	557
10.12.4.36EventSerialPortReceiveTimestamp	557
10.12.4.36EventSerialReceiveOverflow	558
10.12.4.36EventStream0TransferBlockEnd	558
10.12.4.36EventStream0TransferBlockEndFrameID	558
10.12.4.36EventStream0TransferBlockEndTimestamp	558
10.12.4.36EventStream0TransferBlockStart	558
10.12.4.36EventStream0TransferBlockStartFrameID	558
10.12.4.36EventStream0TransferBlockStartTimestamp	559
10.12.4.37EventStream0TransferBlockTrigger	559
10.12.4.37EventStream0TransferBlockTriggerFrameID	559
10.12.4.37EventStream0TransferBlockTriggerTimestamp	559
10.12.4.37EventStream0TransferBurstEnd	559
10.12.4.37EventStream0TransferBurstEndFrameID	559
10.12.4.37EventStream0TransferBurstEndTimestamp	560
10.12.4.37EventStream0TransferBurstStart	560
10.12.4.37EventStream0TransferBurstStartFrameID	560
10.12.4.37EventStream0TransferBurstStartTimestamp	560
10.12.4.37EventStream0TransferEnd	560

10.12.4.38EventStream0TransferEndFrameID	560
10.12.4.38EventStream0TransferEndTimestamp	561
10.12.4.38EventStream0TransferOverflow	561
10.12.4.38EventStream0TransferOverflowFrameID	561
10.12.4.38EventStream0TransferOverflowTimestamp	561
10.12.4.38EventStream0TransferPause	561
10.12.4.38EventStream0TransferPauseFrameID	561
10.12.4.38EventStream0TransferPauseTimestamp	562
10.12.4.38EventStream0TransferResume	562
10.12.4.38EventStream0TransferResumeFrameID	562
10.12.4.39EventStream0TransferResumeTimestamp	562
10.12.4.39EventStream0TransferStart	562
10.12.4.39EventStream0TransferStartFrameID	562
10.12.4.39EventStream0TransferStartTimestamp	563
10.12.4.39EventTest	563
10.12.4.39EventTestTimestamp	563
10.12.4.39EventTimer0End	563
10.12.4.39EventTimer0EndFrameID	563
10.12.4.39EventTimer0EndTimestamp	563
10.12.4.39EventTimer0Start	564
10.12.4.40EventTimer0StartFrameID	564
10.12.4.40EventTimer0StartTimestamp	564
10.12.4.40EventTimer1End	564
10.12.4.40EventTimer1EndFrameID	564
10.12.4.40EventTimer1EndTimestamp	564
10.12.4.40EventTimer1Start	565
10.12.4.40EventTimer1StartFrameID	565
10.12.4.40EventTimer1StartTimestamp	565
10.12.4.40ExposureActiveMode	565
10.12.4.40ExposureAuto	565

10.12.4.41ExposureMode	565
10.12.4.41ExposureTime	566
10.12.4.41ExposureTimeMode	566
10.12.4.41ExposureTimeSelector	566
10.12.4.41FactoryReset	566
10.12.4.41FileAccessBuffer	566
10.12.4.41FileAccessLength	566
10.12.4.41FileAccessOffset	567
10.12.4.41FileOpenMode	567
10.12.4.41FileOperationExecute	567
10.12.4.42FileOperationResult	567
10.12.4.42FileOperationSelector	567
10.12.4.42FileOperationStatus	568
10.12.4.42FileSelector	568
10.12.4.42FileSize	568
10.12.4.42Gain	568
10.12.4.42GainAuto	568
10.12.4.42GainAutoBalance	569
10.12.4.42GainSelector	569
10.12.4.42Gamma	569
10.12.4.43GammaEnable	569
10.12.4.43GetActiveLinkCount	569
10.12.4.43GetCCP	569
10.12.4.43GetCurrentDefaultGateway	570
10.12.4.43GetCurrentIPAddress	570
10.12.4.43GetCurrentIPConfigurationDHCP	570
10.12.4.43GetCurrentIPConfigurationLLA	570
10.12.4.43GetCurrentIPConfigurationPersistentIP	570
10.12.4.43GetCurrentPhysicalLinkConfiguration	570
10.12.4.43GetCurrentSubnetMask	571

10.12.4.440evDiscoveryAckDelay	571
10.12.4.441evFirstURL	571
10.12.4.442evGVCPExtendedStatusCodes	571
10.12.4.443evGVCPExtendedStatusCodesSelector	571
10.12.4.444evGVCPHeartbeatDisable	571
10.12.4.445evGVCPPendingAck	572
10.12.4.446evGVCPPendingTimeout	572
10.12.4.447evGVSPExtendedIDMode	572
10.12.4.448evHeartbeatTimeout	572
10.12.4.449evIEEE1588	572
10.12.4.450evIEEE1588ClockAccuracy	572
10.12.4.451evIEEE1588Mode	573
10.12.4.452evIEEE1588Status	573
10.12.4.453evInterfaceSelector	573
10.12.4.454evIPConfigurationStatus	573
10.12.4.455evMACAddress	573
10.12.4.456evMCDA	573
10.12.4.457evMCPhostPort	574
10.12.4.458evMCRC	574
10.12.4.459evMCSP	574
10.12.4.460evMCTT	574
10.12.4.461evNumberOfInterfaces	574
10.12.4.462evPAUSEFrameReception	574
10.12.4.463evPAUSEFrameTransmission	575
10.12.4.464evPersistentDefaultGateway	575
10.12.4.465evPersistentIPAddress	575
10.12.4.466evPersistentSubnetMask	575
10.12.4.467evPhysicalLinkConfiguration	575
10.12.4.468evPrimaryApplicationIPAddress	575
10.12.4.469evPrimaryApplicationSocket	576

10.12.4.47 <code>GevPrimaryApplicationSwitchoverKey</code>	576
10.12.4.47 <code>GevSCCFGAllInTransmission</code>	576
10.12.4.47 <code>GevSCCFGExtendedChunkData</code>	576
10.12.4.47 <code>GevSCCFGPacketResendDestination</code>	576
10.12.4.47 <code>GevSCCFGUnconditionalStreaming</code>	576
10.12.4.47 <code>GevSCDA</code>	577
10.12.4.47 <code>GevSCPD</code>	577
10.12.4.47 <code>GevSCPDirection</code>	577
10.12.4.47 <code>GevSCPHostPort</code>	577
10.12.4.47 <code>GevSCPInterfaceIndex</code>	577
10.12.4.48 <code>GevSCPSBigEndian</code>	577
10.12.4.48 <code>GevSCPSDoNotFragment</code>	578
10.12.4.48 <code>GevSCPSFireTestPacket</code>	578
10.12.4.48 <code>GevSCSPacketSize</code>	578
10.12.4.48 <code>GevSCSP</code>	578
10.12.4.48 <code>GevSCZoneConfigurationLock</code>	578
10.12.4.48 <code>GevSCZoneCount</code>	578
10.12.4.48 <code>GevSCZoneDirectionAll</code>	579
10.12.4.48 <code>GevSecondURL</code>	579
10.12.4.48 <code>GevStreamChannelSelector</code>	579
10.12.4.49 <code>GevSupportedOption</code>	579
10.12.4.49 <code>GevSupportedOptionSelector</code>	579
10.12.4.49 <code>GevTimestampTickFrequency</code>	579
10.12.4.49 <code>GuiXmlManifestAddress</code>	580
10.12.4.49 <code>eight</code>	580
10.12.4.49 <code>eightMax</code>	580
10.12.4.49 <code>ImageComponentEnable</code>	580
10.12.4.49 <code>ImageComponentSelector</code>	580
10.12.4.49 <code>ImageCompressionBitrate</code>	580
10.12.4.49 <code>ImageCompressionJPEGFormatOption</code>	581

10.12.4.500ImageCompressionMode	581
10.12.4.501ImageCompressionQuality	581
10.12.4.502ImageCompressionRateOption	581
10.12.4.503IpEnable	581
10.12.4.504LineFilterWidth	582
10.12.4.505LineFormat	582
10.12.4.506LineInputFilterSelector	582
10.12.4.507LineInverter	582
10.12.4.508LineMode	582
10.12.4.509LinePitch	582
10.12.4.510LineSelector	583
10.12.4.511LineSource	583
10.12.4.512LineStatus	583
10.12.4.513LineStatusAll	583
10.12.4.514LinkErrorCount	583
10.12.4.515LinkUptime	583
10.12.4.516LogicBlockLUTInputActivation	584
10.12.4.517LogicBlockLUTInputSelector	584
10.12.4.518LogicBlockLUTInputSource	584
10.12.4.519LogicBlockLUTOOutputValue	584
10.12.4.520LogicBlockLUTOOutputValueAll	584
10.12.4.521LogicBlockLUTRowIndex	584
10.12.4.522LogicBlockLUTSelector	585
10.12.4.523LogicBlockSelector	585
10.12.4.524UTEEnable	585
10.12.4.525UTIndex	585
10.12.4.526UTSelector	585
10.12.4.527UTValue	586
10.12.4.528UTValueAll	586
10.12.4.529MaxDeviceResetTime	586

10.12.4.530	OffsetX	586
10.12.4.530	OffsetY	586
10.12.4.532	PacketResendRequestCount	587
10.12.4.532	PayloadSize	587
10.12.4.532	PixelColorFilter	587
10.12.4.532	PixelDynamicRangeMax	587
10.12.4.532	PixelDynamicRangeMin	587
10.12.4.537	PixelFormat	588
10.12.4.538	PixelFormatInfoID	588
10.12.4.539	PixelFormatInfoSelector	588
10.12.4.540	PixelSize	588
10.12.4.541	PowerSupplyCurrent	588
10.12.4.542	PowerSupplyVoltage	588
10.12.4.543	RegionDestination	589
10.12.4.544	RegionMode	589
10.12.4.545	RegionSelector	589
10.12.4.546	ReverseX	589
10.12.4.547	ReverseY	589
10.12.4.548	RgbTransformLightSource	590
10.12.4.549	Saturation	590
10.12.4.550	SaturationEnable	590
10.12.4.553	Scan3dAxisMax	590
10.12.4.552	Scan3dAxisMin	590
10.12.4.553	Scan3dCoordinateOffset	591
10.12.4.553	Scan3dCoordinateReferenceSelector	591
10.12.4.555	Scan3dCoordinateReferenceValue	591
10.12.4.556	Scan3dCoordinateScale	591
10.12.4.557	Scan3dCoordinateSelector	591
10.12.4.558	Scan3dCoordinateSystem	591
10.12.4.559	Scan3dCoordinateSystemReference	592

10.12.4.56 Scan3dCoordinateTransformSelector	592
10.12.4.56 Scan3dDistanceUnit	592
10.12.4.56 Scan3dInvalidDataFlag	592
10.12.4.56 Scan3dInvalidDataValue	592
10.12.4.56 Scan3dOutputMode	592
10.12.4.56 Scan3dTransformValue	593
10.12.4.56 SensorDescription	593
10.12.4.56 SensorDigitizationTaps	593
10.12.4.56 SensorHeight	593
10.12.4.56 SensorShutterMode	593
10.12.4.56 SensorTaps	593
10.12.4.57 SensorWidth	594
10.12.4.57 SequencerConfigurationMode	594
10.12.4.57 SequencerConfigurationValid	594
10.12.4.57 SequencerFeatureEnable	594
10.12.4.57 SequencerMode	594
10.12.4.57 SequencerPathSelector	595
10.12.4.57 SequencerSetActive	595
10.12.4.57 SequencerSetLoad	595
10.12.4.57 SequencerSetNext	595
10.12.4.58 SequencerSetSave	595
10.12.4.58 SequencerSetSelector	596
10.12.4.58 SequencerSetStart	596
10.12.4.58 SequencerSetValid	596
10.12.4.58 SequencerTriggerActivation	596
10.12.4.58 SequencerTriggerSource	596
10.12.4.58 SerialPortBaudRate	597
10.12.4.58 SerialPortDataBits	597
10.12.4.58 SerialPortParity	597
10.12.4.58 SerialPortSelector	597

10.12.4.59SerialPortSource	597
10.12.4.59SerialPortStopBits	597
10.12.4.59SerialReceiveFramingErrorCount	598
10.12.4.59SerialReceiveParityErrorCount	598
10.12.4.59SerialReceiveQueueClear	598
10.12.4.59SerialReceiveQueueCurrentCharacterCount	598
10.12.4.59SerialReceiveQueueMaxCharacterCount	598
10.12.4.59SerialTransmitQueueCurrentCharacterCount	598
10.12.4.59SerialTransmitQueueMaxCharacterCount	599
10.12.4.59Sharpening	599
10.12.4.60SharpeningAuto	599
10.12.4.60SharpeningEnable	599
10.12.4.60SharpeningThreshold	600
10.12.4.60SoftwareSignalPulse	600
10.12.4.60SoftwareSignalSelector	600
10.12.4.60SourceCount	600
10.12.4.60SourceSelector	600
10.12.4.60Test0001	601
10.12.4.60TestEventGenerate	601
10.12.4.60TestPattern	601
10.12.4.61TestPatternGeneratorSelector	601
10.12.4.61TestPendingAck	601
10.12.4.61TimerDelay	602
10.12.4.61TimerDuration	602
10.12.4.61TimerReset	602
10.12.4.61TimerSelector	602
10.12.4.61TimerStatus	602
10.12.4.61TimerTriggerActivation	602
10.12.4.61TimerTriggerSource	603
10.12.4.61TimerValue	603

10.12.4.62Timestamp	603
10.12.4.62TimestampLatch	603
10.12.4.62TimestampLatchValue	603
10.12.4.62TimestampReset	603
10.12.4.62ParamsLocked	604
10.12.4.62TransferAbort	604
10.12.4.62TransferBlockCount	604
10.12.4.62TransferBurstCount	604
10.12.4.62TransferComponentSelector	604
10.12.4.62TransferControlMode	604
10.12.4.63TransferOperationMode	605
10.12.4.63TransferPause	605
10.12.4.63TransferQueueCurrentBlockCount	605
10.12.4.63TransferQueueMaxBlockCount	605
10.12.4.63TransferQueueMode	605
10.12.4.63TransferQueueOverflowCount	605
10.12.4.63TransferResume	606
10.12.4.63TransferSelector	606
10.12.4.63TransferStart	606
10.12.4.63TransferStatus	606
10.12.4.64TransferStatusSelector	606
10.12.4.64TransferStop	606
10.12.4.64TransferStreamChannel	607
10.12.4.64TransferTriggerActivation	607
10.12.4.64TransferTriggerMode	607
10.12.4.64TransferTriggerSelector	607
10.12.4.64TransferTriggerSource	607
10.12.4.64TriggerActivation	607
10.12.4.64TriggerDelay	608
10.12.4.64TriggerDivider	608

10.12.4.65TriggerEventTest	608
10.12.4.65TriggerMode	608
10.12.4.65TriggerMultiplier	608
10.12.4.65TriggerOverlap	609
10.12.4.65TriggerSelector	609
10.12.4.65TriggerSoftware	609
10.12.4.65TriggerSource	609
10.12.4.65UserOutputSelector	609
10.12.4.65UserOutputValue	610
10.12.4.65UserOutputValueAll	610
10.12.4.66UserOutputValueAllMask	610
10.12.4.66UserSetDefault	610
10.12.4.66UserSetFeatureEnable	610
10.12.4.66UserSetLoad	611
10.12.4.66UserSetSave	611
10.12.4.66UserSetSelector	611
10.12.4.66_3Enable	611
10.12.4.66WhiteClip	611
10.12.4.66WhiteClipSelector	612
10.12.4.66Width	612
10.12.4.67WidthMax	612
10.13 CameraBase Class Reference	613
10.13.1 Detailed Description	615
10.13.2 Constructor & Destructor Documentation	615
10.13.2.1 ~CameraBase()	615
10.13.2.2 CameraBase() [1/2]	616
10.13.2.3 CameraBase() [2/2]	616
10.13.3 Member Function Documentation	616
10.13.3.1 BeginAcquisition()	616
10.13.3.2 DeInit()	616

10.13.3.3 DiscoverMaxPacketSize()	617
10.13.3.4 EndAcquisition()	617
10.13.3.5 ForceIP()	617
10.13.3.6 GetAccessMode()	618
10.13.3.7 GetBufferOwnership()	618
10.13.3.8 GetGuiXml()	618
10.13.3.9 GetNextImage()	619
10.13.3.10 GetNodeMap()	619
10.13.3.11 GetNumDataStreams()	620
10.13.3.12 GetNumImagesInUse()	620
10.13.3.13 GetTLDeviceNodeMap()	620
10.13.3.14 GetTlStreamNodeMap()	621
10.13.3.15 GetUniqueId()	621
10.13.3.16 GetUserBufferCount()	621
10.13.3.17 GetUserBufferSize()	622
10.13.3.18 GetUserBufferTotalSize()	622
10.13.3.19 init()	623
10.13.3.20 IsInitialized()	623
10.13.3.21 IsStreaming()	623
10.13.3.22 IsValid()	624
10.13.3.23 operator=()	624
10.13.3.24 ReadPort()	624
10.13.3.25 RegisterEvent() [1/2]	624
10.13.3.26 RegisterEvent() [2/2]	625
10.13.3.27 SetBufferOwnership()	625
10.13.3.28 SetUserBuffers() [1/2]	626
10.13.3.29 SetUserBuffers() [2/2]	626
10.13.3.30 UnregisterEvent()	627
10.13.3.31 WritePort()	627
10.13.4 Friends And Related Function Documentation	627

10.13.4.1 InterfaceImpl	627
10.14 CameraList Class Reference	628
10.14.1 Detailed Description	629
10.14.2 Constructor & Destructor Documentation	629
10.14.2.1 CameraList() [1/2]	629
10.14.2.2 ~CameraList()	629
10.14.2.3 CameraList() [2/2]	629
10.14.3 Member Function Documentation	630
10.14.3.1 Append()	630
10.14.3.2 Clear()	631
10.14.3.3 GetByIndex()	631
10.14.3.4 GetBySerial()	632
10.14.3.5 GetSize()	632
10.14.3.6 operator=()	632
10.14.3.7 operator[]()	633
10.14.3.8 RemoveByIndex()	633
10.14.3.9 RemoveBySerial()	633
10.15 CameraPtr Class Reference	634
10.15.1 Detailed Description	635
10.16 CategoryNode Class Reference	635
10.16.1 Detailed Description	636
10.16.2 Constructor & Destructor Documentation	636
10.16.2.1 CategoryNode() [1/2]	636
10.16.2.2 CategoryNode() [2/2]	637
10.16.2.3 ~CategoryNode()	637
10.16.3 Member Function Documentation	637
10.16.3.1 GetFeatures()	637
10.16.3.2 SetReference()	637
10.17 CChunkAdapter Class Reference	637
10.17.1 Detailed Description	638

10.17.2 Constructor & Destructor Documentation	638
10.17.2.1 ~CChunkAdapter()	638
10.17.2.2 CChunkAdapter()	639
10.17.3 Member Function Documentation	639
10.17.3.1 AttachBuffer()	639
10.17.3.2 AttachNodeMap()	639
10.17.3.3 CheckBufferLayout()	639
10.17.3.4 ClearCaches()	640
10.17.3.5 DetachBuffer()	640
10.17.3.6 DetachNodeMap()	640
10.17.3.7 UpdateBuffer()	640
10.17.4 Member Data Documentation	640
10.17.4.1 m_pChunkAdapter	640
10.18 CChunkAdapterDcam Class Reference	641
10.18.1 Detailed Description	642
10.18.2 Constructor & Destructor Documentation	642
10.18.2.1 CChunkAdapterDcam()	642
10.18.2.2 ~CChunkAdapterDcam()	642
10.18.3 Member Function Documentation	642
10.18.3.1 AttachBuffer()	642
10.18.3.2 CheckBufferLayout()	643
10.18.3.3 CheckCRC()	643
10.18.3.4 HasCRC()	643
10.19 CChunkAdapterGeneric Class Reference	643
10.19.1 Constructor & Destructor Documentation	644
10.19.1.1 CChunkAdapterGeneric()	644
10.19.1.2 ~CChunkAdapterGeneric()	644
10.19.2 Member Function Documentation	645
10.19.2.1 AttachBuffer() [1/3]	645
10.19.2.2 AttachBuffer() [2/3]	645

10.19.2.3 AttachBuffer() [3/3]	645
10.19.2.4 CheckBufferLayout()	645
10.20 CChunkAdapterGEV Class Reference	646
10.20.1 Detailed Description	646
10.20.2 Constructor & Destructor Documentation	647
10.20.2.1 CChunkAdapterGEV()	647
10.20.2.2 ~CChunkAdapterGEV()	647
10.20.3 Member Function Documentation	647
10.20.3.1 AttachBuffer()	647
10.20.3.2 CheckBufferLayout()	647
10.21 CChunkAdapterU3V Class Reference	648
10.21.1 Detailed Description	648
10.21.2 Constructor & Destructor Documentation	649
10.21.2.1 CChunkAdapterU3V()	649
10.21.2.2 ~CChunkAdapterU3V()	649
10.21.3 Member Function Documentation	649
10.21.3.1 AttachBuffer()	649
10.21.3.2 CheckBufferLayout()	649
10.22 CChunkPort Class Reference	650
10.22.1 Detailed Description	651
10.22.2 Constructor & Destructor Documentation	651
10.22.2.1 CChunkPort()	651
10.22.2.2 ~CChunkPort()	652
10.22.3 Member Function Documentation	652
10.22.3.1 AttachChunk()	652
10.22.3.2 AttachPort()	652
10.22.3.3 CheckChunkID() [1/2]	652
10.22.3.4 CheckChunkID() [2/2]	652
10.22.3.5 ClearCache()	653
10.22.3.6 DetachChunk()	653

10.22.3.7 DetachPort()	653
10.22.3.8 GetAccessMode()	653
10.22.3.9 GetChunkIDLength()	653
10.22.3.10 GetPrincipalInterfaceType()	653
10.22.3.11 GetSwapEndianess()	654
10.22.3.12 InvalidateNode()	654
10.22.3.13 Read()	654
10.22.3.14 SetPortImpl()	654
10.22.3.15 UpdateBuffer()	654
10.22.3.16 Write()	654
10.22.4 Member Data Documentation	655
10.22.4.1 m_pChunkPort	655
10.22.4.2 m_pPort	655
10.22.4.3 m_pPortAdapter	655
10.23 CEnumerationTRef< EnumT > Class Template Reference	655
10.23.1 Detailed Description	657
10.23.2 Constructor & Destructor Documentation	657
10.23.2.1 CEnumerationTRef() [1/2]	657
10.23.2.2 CEnumerationTRef() [2/2]	657
10.23.2.3 ~CEnumerationTRef()	657
10.23.3 Member Function Documentation	657
10.23.3.1 GetCurrentEntry()	658
10.23.3.2 GetEntry() [1/2]	658
10.23.3.3 GetEntry() [2/2]	658
10.23.3.4 GetValue()	658
10.23.3.5 operator()()	659
10.23.3.6 operator=() [1/2]	659
10.23.3.7 operator=() [2/2]	659
10.23.3.8 SetEnumReference()	659
10.23.3.9 SetNumEnums()	659

10.23.3.10 SetReference()	660
10.23.3.11 SetValue()	660
10.24 CEventAdapter Class Reference	660
10.24.1 Detailed Description	661
10.24.2 Constructor & Destructor Documentation	661
10.24.2.1 CEventAdapter()	661
10.24.2.2 ~CEventAdapter()	661
10.24.3 Member Function Documentation	661
10.24.3.1 AttachNodeMap()	661
10.24.3.2 DeliverMessage()	662
10.24.3.3 DetachNodeMap()	662
10.24.4 Member Data Documentation	662
10.24.4.1 m_pEventAdapter	662
10.25 CEventAdapter1394 Class Reference	662
10.25.1 Detailed Description	663
10.25.2 Constructor & Destructor Documentation	663
10.25.2.1 CEventAdapter1394()	663
10.25.2.2 ~CEventAdapter1394()	663
10.25.3 Member Function Documentation	664
10.25.3.1 DeliverEventMessage()	664
10.25.3.2 DeliverMessage()	664
10.26 CEventAdapterGeneric Class Reference	664
10.26.1 Detailed Description	665
10.26.2 Constructor & Destructor Documentation	665
10.26.2.1 CEventAdapterGeneric()	665
10.26.2.2 ~CEventAdapterGeneric()	666
10.26.3 Member Function Documentation	666
10.26.3.1 DeliverMessage() [1/3]	666
10.26.3.2 DeliverMessage() [2/3]	666
10.26.3.3 DeliverMessage() [3/3]	666

10.27 CEventAdapterGEV Class Reference	667
10.27.1 Detailed Description	668
10.27.2 Constructor & Destructor Documentation	668
10.27.2.1 CEventAdapterGEV()	668
10.27.2.2 ~CEventAdapterGEV()	668
10.27.3 Member Function Documentation	668
10.27.3.1 DeliverEventMessage() [1/2]	668
10.27.3.2 DeliverEventMessage() [2/2]	668
10.27.3.3 DeliverMessage()	669
10.28 CEventAdapterU3V Class Reference	669
10.28.1 Detailed Description	670
10.28.2 Constructor & Destructor Documentation	670
10.28.2.1 CEventAdapterU3V()	670
10.28.2.2 ~CEventAdapterU3V()	670
10.28.3 Member Function Documentation	670
10.28.3.1 DeliverEventMessage()	670
10.28.3.2 DeliverMessage()	671
10.29 CEventPort Class Reference	671
10.29.1 Detailed Description	672
10.29.2 Constructor & Destructor Documentation	672
10.29.2.1 CEventPort()	673
10.29.2.2 ~CEventPort()	673
10.29.3 Member Function Documentation	673
10.29.3.1 AttachEvent()	673
10.29.3.2 AttachNode()	673
10.29.3.3 CheckEventID() [1/2]	673
10.29.3.4 CheckEventID() [2/2]	674
10.29.3.5 DetachEvent()	674
10.29.3.6 DetachNode()	674
10.29.3.7 GetAccessMode()	674

10.29.3.8 GetEventIDLength()	674
10.29.3.9 GetPrincipalInterfaceType()	674
10.29.3.10GetSwapEndianess()	675
10.29.3.11InvalidateNode()	675
10.29.3.12Read()	675
10.29.3.13SetPortImpl()	675
10.29.3.14Write()	675
10.29.4 Member Data Documentation	675
10.29.4.1 m_pEventPort	676
10.29.4.2 m_pNode	676
10.29.4.3 m_pPortAdapter	676
10.30 CFeatureBag Class Reference	676
10.30.1 Detailed Description	677
10.30.2 Constructor & Destructor Documentation	677
10.30.2.1 CFeatureBag()	677
10.30.2.2 ~CFeatureBag()	677
10.30.3 Member Function Documentation	677
10.30.3.1 GetFeatureBagHandle()	677
10.30.3.2 LoadFromBag()	677
10.30.3.3 operator==()	678
10.30.3.4 PersistFeature()	678
10.30.3.5 SetInfo()	678
10.30.3.6 StoreToBag()	678
10.31 CFloatPtr Class Reference	679
10.31.1 Detailed Description	680
10.31.2 Constructor & Destructor Documentation	680
10.31.2.1 CFloatPtr() [1/2]	680
10.31.2.2 CFloatPtr() [2/2]	680
10.31.3 Member Function Documentation	680
10.31.3.1 GetEnumAlias()	680

10.31.3.2 GetIntAlias()	680
10.31.3.3 operator=()	681
10.32 CGeneric_XMLLoaderParams Class Reference	681
10.32.1 Detailed Description	681
10.32.2 Member Function Documentation	681
10.32.2.1 _Initialize()	682
10.33 CGlobalLock Class Reference	682
10.33.1 Detailed Description	682
10.33.2 Constructor & Destructor Documentation	682
10.33.2.1 CGlobalLock() [1/2]	683
10.33.2.2 CGlobalLock() [2/2]	683
10.33.2.3 ~CGlobalLock()	683
10.33.3 Member Function Documentation	683
10.33.3.1 IsValid()	683
10.33.3.2 Lock()	683
10.33.3.3 TryLock()	684
10.33.3.4 Unlock()	684
10.33.4 Member Data Documentation	684
10.33.4.1 m_DebugCount	684
10.34 CGlobalLockUnlocker Class Reference	684
10.34.1 Detailed Description	685
10.34.2 Constructor & Destructor Documentation	685
10.34.2.1 CGlobalLockUnlocker()	685
10.34.2.2 ~CGlobalLockUnlocker()	685
10.34.3 Member Function Documentation	685
10.34.3.1 UnlockEarly()	685
10.34.4 Member Data Documentation	686
10.34.4.1 m_enabled	686
10.34.4.2 m_Lock	686
10.35 ChunkData Class Reference	686

10.35.1 Detailed Description	688
10.35.2 Constructor & Destructor Documentation	688
10.35.2.1 ChunkData() [1/2]	688
10.35.2.2 ChunkData() [2/2]	689
10.35.2.3 ~ChunkData()	689
10.35.3 Member Function Documentation	689
10.35.3.1 GetBlackLevel()	689
10.35.3.2 GetCounterValue()	689
10.35.3.3 GetCRC()	689
10.35.3.4 GetEncoderValue()	690
10.35.3.5 GetExposureEndLineStatusAll()	690
10.35.3.6 GetExposureTime()	690
10.35.3.7 GetFrameID()	690
10.35.3.8 GetGain()	691
10.35.3.9 GetHeight()	691
10.35.3.10 GetImage()	691
10.35.3.11 GetInferenceBoundingBoxResult()	691
10.35.3.12 GetInferenceConfidence()	692
10.35.3.13 GetInferenceResult()	692
10.35.3.14 GetLinePitch()	692
10.35.3.15 GetLineStatusAll()	692
10.35.3.16 GetOffsetX()	693
10.35.3.17 GetOffsetY()	693
10.35.3.18 GetPartSelector()	693
10.35.3.19 GetPixelDynamicRangeMax()	693
10.35.3.20 GetPixelDynamicRangeMin()	694
10.35.3.21 GetScan3dAxisMax()	694
10.35.3.22 GetScan3dAxisMin()	694
10.35.3.23 GetScan3dCoordinateOffset()	694
10.35.3.24 GetScan3dCoordinateReferenceValue()	695

10.35.3.25GetScan3dCoordinateScale()	695
10.35.3.26GetScan3dInvalidDataValue()	695
10.35.3.27GetScan3dTransformValue()	695
10.35.3.28GetScanLineSelector()	696
10.35.3.29GetSequencerSetActive()	696
10.35.3.30GetSerialDataLength()	696
10.35.3.31GetStreamChannelID()	696
10.35.3.32GetTimerValue()	697
10.35.3.33GetTimestamp()	697
10.35.3.34GetTimestampLatchValue()	697
10.35.3.35GetTransferBlockID()	697
10.35.3.36GetTransferQueueCurrentBlockCount()	698
10.35.3.37GetWidth()	698
10.35.3.38SetChunks()	698
10.36CLock Class Reference	698
10.36.1 Detailed Description	699
10.36.2 Constructor & Destructor Documentation	699
10.36.2.1 CLock()	699
10.36.2.2 ~CLock()	699
10.36.3 Member Function Documentation	699
10.36.3.1 Lock()	699
10.36.3.2 TryLock()	700
10.36.3.3 Unlock()	700
10.37CLock Class Reference	700
10.37.1 Detailed Description	701
10.37.2 Constructor & Destructor Documentation	701
10.37.2.1 CLock() [1/2]	701
10.37.2.2 CLock() [2/2]	701
10.37.2.3 ~CLock()	701
10.37.3 Member Function Documentation	701

10.37.3.1 Lock()	702
10.37.3.2 TryLock()	702
10.37.3.3 Unlock()	702
10.37.4 Friends And Related Function Documentation	702
10.37.4.1 NodeMap	702
10.37.5 Member Data Documentation	702
10.37.5.1 m_bOwnLock	702
10.37.5.2 m_lock	702
10.38 CLockEx Class Reference	703
10.38.1 Detailed Description	703
10.39 CLockEx Class Reference	704
10.39.1 Detailed Description	704
10.39.2 Member Data Documentation	705
10.39.2.1 m_lockEx	705
10.40 CNodeCallback Class Reference	705
10.40.1 Detailed Description	706
10.40.2 Constructor & Destructor Documentation	706
10.40.2.1 CNodeCallback()	706
10.40.2.2 ~CNodeCallback()	706
10.40.3 Member Function Documentation	706
10.40.3.1 Destroy()	706
10.40.3.2 GetCallbackType()	707
10.40.3.3 GetNode()	707
10.40.3.4 operator()()	707
10.40.4 Member Data Documentation	707
10.40.4.1 m_CallbackType	707
10.40.4.2 m_pNode	707
10.41 CNodeMapFactory Class Reference	708
10.41.1 Detailed Description	709
10.41.2 Constructor & Destructor Documentation	710

10.41.2.1 CNodeMapFactory() [1/5]	710
10.41.2.2 ~CNodeMapFactory()	710
10.41.2.3 CNodeMapFactory() [2/5]	710
10.41.2.4 CNodeMapFactory() [3/5]	710
10.41.2.5 CNodeMapFactory() [4/5]	711
10.41.2.6 CNodeMapFactory() [5/5]	712
10.41.3 Member Function Documentation	712
10.41.3.1 AddInjectionData()	712
10.41.3.2 ApplyStyleSheet()	712
10.41.3.3 ClearCache()	713
10.41.3.4 CreateEmptyNodeMap()	713
10.41.3.5 CreateNodeDataFromNodeMap()	713
10.41.3.6 CreateNodeMap() [1/2]	713
10.41.3.7 CreateNodeMap() [2/2]	713
10.41.3.8 ExtractSubtree()	714
10.41.3.9 GetNodeStatistics()	714
10.41.3.10GetSupportedSchemaVersions()	714
10.41.3.11IsCameraDescriptionFileDataReleased()	714
10.41.3.12IsEmpty()	714
10.41.3.13IsLoaded()	715
10.41.3.14IsPreprocessed()	715
10.41.3.15LoadAndInject()	715
10.41.3.16operator=()	715
10.41.3.17Preprocess()	715
10.41.3.18ReleaseCameraDescriptionFileData()	716
10.41.3.19ToString()	716
10.41.3.20ToXml()	716
10.42 CNodeMapRef Class Reference	716
10.42.1 Detailed Description	717
10.42.2 Constructor & Destructor Documentation	717

10.42.2.1 CNodeMapRef() [1/3]	718
10.42.2.2 CNodeMapRef() [2/3]	718
10.42.2.3 CNodeMapRef() [3/3]	718
10.42.3 Member Function Documentation	718
10.42.3.1 operator=() [1/2]	718
10.42.3.2 operator=() [2/2]	718
10.43 CNodeMapRefT< TCameraParams > Class Template Reference	719
10.43.1 Detailed Description	720
10.43.2 Member Function Documentation	721
10.43.2.1 _ClearXMLCache()	721
10.43.2.2 _Connect() [1/2]	721
10.43.2.3 _Connect() [2/2]	721
10.43.2.4 _GetDeviceName()	721
10.43.2.5 _GetNode()	722
10.43.2.6 _GetNodes()	722
10.43.2.7 _GetSupportedSchemaVersions()	722
10.43.2.8 _InvalidateNodes()	722
10.43.2.9 _LoadXMLFromFile()	722
10.43.2.10 _LoadXMLFromFileInject()	723
10.43.2.11 _LoadXMLFromString()	723
10.43.2.12 _LoadXMLFromStringInject()	723
10.43.2.13 _LoadXMLFromZIPData()	723
10.43.2.14 _LoadXMLFromZIPFile()	723
10.43.2.15 Poll()	724
10.43.3 Member Data Documentation	724
10.43.3.1 _Ptr	724
10.44 CommandNode Class Reference	724
10.44.1 Detailed Description	725
10.44.2 Constructor & Destructor Documentation	725
10.44.2.1 CommandNode() [1/2]	726

10.44.2.2 CommandNode() [2/2]	726
10.44.2.3 ~CommandNode()	726
10.44.3 Member Function Documentation	726
10.44.3.1 Execute()	726
10.44.3.2 IsDone()	726
10.44.3.3 operator()()	727
10.44.3.4 SetReference()	727
10.45 Counter Class Reference	727
10.45.1 Detailed Description	727
10.45.2 Constructor & Destructor Documentation	728
10.45.2.1 Counter()	728
10.45.3 Member Function Documentation	728
10.45.3.1 GetValue()	728
10.45.3.2 IsZero()	728
10.45.3.3 operator unsigned int()	728
10.45.3.4 operator++() [1/2]	728
10.45.3.5 operator++() [2/2]	728
10.45.3.6 operator--() [1/2]	729
10.45.3.7 operator--() [2/2]	729
10.46 CPointer< T, B > Class Template Reference	729
10.46.1 Detailed Description	730
10.46.2 Constructor & Destructor Documentation	730
10.46.2.1 CPointer() [1/2]	730
10.46.2.2 CPointer() [2/2]	731
10.46.2.3 ~CPointer()	731
10.46.3 Member Function Documentation	731
10.46.3.1 IsValid()	731
10.46.3.2 operator bool()	731
10.46.3.3 operator T*()	731
10.46.3.4 operator"!=() [1/5]	732

10.46.3.5 operator"!=() [2/5]	732
10.46.3.6 operator"!=() [3/5]	732
10.46.3.7 operator"!=() [4/5]	732
10.46.3.8 operator"!=() [5/5]	732
10.46.3.9 operator()()	732
10.46.3.10 operator*()	733
10.46.3.11 operator->()	733
10.46.3.12 operator=()	733
10.46.3.13 operator==([1/3]	733
10.46.3.14 operator==([2/3]	733
10.46.3.15 operator==([3/3]	733
10.46.4 Member Data Documentation	734
10.46.4.1 m_pT	734
10.47 CPortImpl Class Reference	734
10.47.1 Detailed Description	735
10.47.2 Constructor & Destructor Documentation	735
10.47.2.1 CPortImpl()	735
10.47.2.2 ~CPortImpl()	735
10.47.3 Member Function Documentation	736
10.47.3.1 GetAccessMode()	736
10.47.3.2 GetSwapEndianess()	736
10.47.3.3 InvalidateNode()	736
10.47.3.4 Read()	736
10.47.3.5 Replay()	736
10.47.3.6 SetPortImpl()	737
10.47.3.7 Write()	737
10.47.4 Member Data Documentation	737
10.47.4.1 m_ptrPort	737
10.48 CPortWriteList Class Reference	738
10.48.1 Detailed Description	739

10.48.2 Constructor & Destructor Documentation	739
10.48.2.1 CPortWriteList()	739
10.48.2.2 ~CPortWriteList()	739
10.48.3 Member Function Documentation	739
10.48.3.1 GetCookie()	739
10.48.3.2 GetPortWriteListHandle()	739
10.48.3.3 Replay()	740
10.48.3.4 SetCookie()	740
10.48.3.5 Write()	740
10.48.4 Member Data Documentation	740
10.48.4.1 m_pWriteList	740
10.49 CRegisterPortImpl Class Reference	741
10.49.1 Detailed Description	742
10.49.2 Constructor & Destructor Documentation	742
10.49.2.1 CRegisterPortImpl()	742
10.49.2.2 ~CRegisterPortImpl()	742
10.49.3 Member Function Documentation	742
10.49.3.1 GetAccessMode()	743
10.49.3.2 Read()	743
10.49.3.3 ReadRegister()	743
10.49.3.4 SetPortImpl()	743
10.49.3.5 Write()	744
10.49.3.6 WriteRegister()	744
10.50 CSelectorSet Class Reference	744
10.50.1 Detailed Description	745
10.50.2 Constructor & Destructor Documentation	745
10.50.2.1 CSelectorSet()	745
10.50.2.2 ~CSelectorSet()	746
10.50.3 Member Function Documentation	746
10.50.3.1 GetSelectorList()	746

10.50.3.2 IsEmpty()	746
10.50.3.3 Restore()	746
10.50.3.4 SetFirst()	746
10.50.3.5 SetNext()	747
10.50.3.6 ToString()	747
10.51 CTestPortStruct< CDataStruct > Class Template Reference	747
10.51.1 Detailed Description	748
10.51.2 Constructor & Destructor Documentation	749
10.51.2.1 CTestPortStruct()	749
10.51.3 Member Function Documentation	749
10.51.3.1 GetAccessMode()	749
10.51.3.2 GetNumReads()	749
10.51.3.3 GetNumWrites()	749
10.51.3.4 GetPrincipalInterfaceType()	749
10.51.3.5 MemSet()	750
10.51.3.6 Read()	750
10.51.3.7 ResetStatistics()	750
10.51.3.8 Write()	750
10.51.4 Member Data Documentation	750
10.51.4.1 m_BaseAddress	750
10.51.4.2 m_NumReads	751
10.51.4.3 m_NumWrites	751
10.52 DCAM_CHECKSUM Struct Reference	751
10.52.1 Member Data Documentation	751
10.52.1.1 CRCChecksum	751
10.53 DCAM_CHUNK_TRAILER Struct Reference	751
10.53.1 Member Data Documentation	752
10.53.1.1 ChunkID	752
10.53.1.2 ChunkLength	752
10.53.1.3 InverseChunkLength	752

10.54 DeviceEvent Class Reference	752
10.54.1 Detailed Description	753
10.54.2 Constructor & Destructor Documentation	753
10.54.2.1 DeviceEvent()	754
10.54.2.2 ~DeviceEvent()	754
10.54.3 Member Function Documentation	754
10.54.3.1 GetDeviceEventId()	754
10.54.3.2 GetDeviceEventName()	754
10.54.3.3 OnDeviceEvent()	754
10.54.3.4 operator=()	755
10.55 double_autovector_t Class Reference	755
10.55.1 Detailed Description	755
10.55.2 Constructor & Destructor Documentation	756
10.55.2.1 double_autovector_t() [1/3]	756
10.55.2.2 double_autovector_t() [2/3]	756
10.55.2.3 double_autovector_t() [3/3]	756
10.55.2.4 ~double_autovector_t()	756
10.55.3 Member Function Documentation	756
10.55.3.1 operator delete()	756
10.55.3.2 operator new()	756
10.55.3.3 operator=()	757
10.55.3.4 operator[]() [1/2]	757
10.55.3.5 operator[]() [2/2]	757
10.55.3.6 size()	757
10.55.4 Member Data Documentation	757
10.55.4.1 _pCount	757
10.55.4.2 _pv	757
10.56 EAccessModeClass Class Reference	758
10.56.1 Detailed Description	758
10.56.2 Member Function Documentation	758

10.56.2.1 FromString()	758
10.56.2.2 ToString() [1/2]	758
10.56.2.3 ToString() [2/2]	758
10.57 ECachingModeClass Class Reference	759
10.57.1 Detailed Description	759
10.57.2 Member Function Documentation	759
10.57.2.1 FromString()	759
10.57.2.2 ToString() [1/2]	759
10.57.2.3 ToString() [2/2]	759
10.58 EDisplayNotationClass Class Reference	760
10.58.1 Detailed Description	760
10.58.2 Member Function Documentation	760
10.58.2.1 FromString()	760
10.58.2.2 ToString() [1/2]	760
10.58.2.3 ToString() [2/2]	760
10.59 EEndianessClass Class Reference	761
10.59.1 Detailed Description	761
10.59.2 Member Function Documentation	761
10.59.2.1 FromString()	761
10.59.2.2 ToString() [1/2]	761
10.59.2.3 ToString() [2/2]	761
10.60 EGenApiSchemaVersionClass Class Reference	762
10.60.1 Detailed Description	762
10.60.2 Member Function Documentation	762
10.60.2.1 FromString()	762
10.60.2.2 ToString() [1/2]	762
10.60.2.3 ToString() [2/2]	762
10.61 EInputDirectionClass Class Reference	763
10.61.1 Detailed Description	763
10.61.2 Member Function Documentation	763

10.61.2.1 FromString()	763
10.61.2.2 ToString() [1/2]	763
10.61.2.3 ToString() [2/2]	763
10.62 ENameSpaceClass Class Reference	764
10.62.1 Detailed Description	764
10.62.2 Member Function Documentation	764
10.62.2.1 FromString()	764
10.62.2.2 ToString() [1/2]	764
10.62.2.3 ToString() [2/2]	764
10.63 EnumEntryNode Class Reference	765
10.63.1 Detailed Description	766
10.63.2 Constructor & Destructor Documentation	766
10.63.2.1 EnumEntryNode() [1/2]	766
10.63.2.2 EnumEntryNode() [2/2]	766
10.63.2.3 ~EnumEntryNode()	766
10.63.3 Member Function Documentation	766
10.63.3.1 GetNumericValue()	767
10.63.3.2 GetSymbolic()	767
10.63.3.3 GetValue()	767
10.63.3.4 IsSelfClearing()	767
10.63.3.5 SetReference()	767
10.64 EnumNode Class Reference	768
10.64.1 Detailed Description	770
10.64.2 Constructor & Destructor Documentation	770
10.64.2.1 EnumNode() [1/2]	770
10.64.2.2 EnumNode() [2/2]	770
10.64.2.3 ~EnumNode()	770
10.64.3 Member Function Documentation	770
10.64.3.1 GetCurrentEntry()	770
10.64.3.2 GetEntries()	771

10.64.3.3 GetEntry()	771
10.64.3.4 GetEntryByName()	771
10.64.3.5 GetIntValue()	771
10.64.3.6 GetSymbolics()	772
10.64.3.7 operator*()	772
10.64.3.8 operator=()	772
10.64.3.9 SetIntValue()	772
10.64.3.10 SetReference()	772
10.64.4 Member Data Documentation	773
10.64.4.1 m_pEnumeration	773
10.65 ERepresentationClass Class Reference	773
10.65.1 Detailed Description	773
10.65.2 Member Function Documentation	773
10.65.2.1 FromString()	774
10.65.2.2 ToString() [1/2]	774
10.65.2.3 ToString() [2/2]	774
10.66 ESignClass Class Reference	774
10.66.1 Detailed Description	774
10.66.2 Member Function Documentation	775
10.66.2.1 FromString()	775
10.66.2.2 ToString() [1/2]	775
10.66.2.3 ToString() [2/2]	775
10.67 ESlopeClass Class Reference	775
10.67.1 Detailed Description	776
10.67.2 Member Function Documentation	776
10.67.2.1 FromString()	776
10.67.2.2 ToString() [1/2]	776
10.67.2.3 ToString() [2/2]	776
10.68 EStandardNameSpaceClass Class Reference	776
10.68.1 Detailed Description	777

10.68.2 Member Function Documentation	777
10.68.2.1 FromString()	777
10.68.2.2 ToString() [1/2]	777
10.68.2.3 ToString() [2/2]	777
10.69 Event Class Reference	778
10.69.1 Detailed Description	779
10.69.2 Constructor & Destructor Documentation	779
10.69.2.1 ~Event()	779
10.69.2.2 Event()	779
10.69.3 Member Function Documentation	779
10.69.3.1 GetEventPayloadData()	779
10.69.3.2 GetEventPayloadDataSize()	780
10.69.3.3 GetEventType()	780
10.69.3.4 operator=()	780
10.69.3.5 SetEventPayload()	780
10.69.3.6 SetEventType()	780
10.69.4 Friends And Related Function Documentation	781
10.69.4.1 EventProcessor	781
10.69.4.2 IDataStream	781
10.69.4.3 Stream	781
10.69.5 Member Data Documentation	781
10.69.5.1 m_pEventData	781
10.70 EVisibilityClass Class Reference	781
10.70.1 Detailed Description	782
10.70.2 Member Function Documentation	782
10.70.2.1 FromString()	782
10.70.2.2 ToString() [1/2]	782
10.70.2.3 ToString() [2/2]	782
10.71 Exception Class Reference	783
10.71.1 Detailed Description	784

10.71.2 Constructor & Destructor Documentation	784
10.71.2.1 Exception() [1/4]	784
10.71.2.2 Exception() [2/4]	784
10.71.2.3 Exception() [3/4]	785
10.71.2.4 Exception() [4/4]	785
10.71.2.5 ~Exception()	785
10.71.3 Member Function Documentation	786
10.71.3.1 GetBuildDate()	786
10.71.3.2 GetBuildTime()	786
10.71.3.3 GetError()	786
10.71.3.4 GetErrorMessage()	786
10.71.3.5 GetFileName()	786
10.71.3.6 GetFullErrorMessage()	786
10.71.3.7 GetFunctionName()	787
10.71.3.8 GetLineNumber()	787
10.71.3.9 operator"!="()	787
10.71.3.10 operator=(())	787
10.71.3.11 operator==(())	787
10.71.3.12 what()	787
10.72 EYesNoClass Class Reference	788
10.72.1 Detailed Description	788
10.72.2 Member Function Documentation	788
10.72.2.1 FromString()	788
10.72.2.2 ToString() [1/2]	788
10.72.2.3 ToString() [2/2]	788
10.73 FileProtocolAdapter Class Reference	789
10.73.1 Detailed Description	789
10.73.2 Constructor & Destructor Documentation	789
10.73.2.1 FileProtocolAdapter()	789
10.73.2.2 ~FileProtocolAdapter()	789

10.73.3 Member Function Documentation	790
10.73.3.1 attach()	790
10.73.3.2 closeFile()	791
10.73.3.3 deleteFile()	791
10.73.3.4 getBufSize()	791
10.73.3.5 openFile()	792
10.73.3.6 read()	792
10.73.3.7 write()	793
10.74 FloatNode Class Reference	793
10.74.1 Detailed Description	795
10.74.2 Constructor & Destructor Documentation	796
10.74.2.1 FloatNode() [1/2]	796
10.74.2.2 FloatNode() [2/2]	796
10.74.2.3 ~FloatNode()	796
10.74.3 Member Function Documentation	796
10.74.3.1 GetDisplayNotation()	796
10.74.3.2 GetDisplayPrecision()	796
10.74.3.3 GetEnumAlias()	796
10.74.3.4 GetInc()	797
10.74.3.5 GetIncMode()	797
10.74.3.6 GetIntAlias()	797
10.74.3.7 GetListOfValidValues()	797
10.74.3.8 GetMax()	797
10.74.3.9 GetMin()	797
10.74.3.10GetRepresentation()	798
10.74.3.11GetUnit()	798
10.74.3.12GetValue()	798
10.74.3.13HasInc()	798
10.74.3.14ImposeMax()	798
10.74.3.15ImposeMin()	799

10.74.3.16operator()	799
10.74.3.17operator*()	799
10.74.3.18operator=()	799
10.74.3.19SetReference()	799
10.74.3.20SetValue()	799
10.75FloatRegNode Class Reference	800
10.75.1 Detailed Description	801
10.75.2 Constructor & Destructor Documentation	801
10.75.2.1 FloatRegNode() [1/2]	802
10.75.2.2 FloatRegNode() [2/2]	802
10.75.2.3 ~FloatRegNode()	802
10.75.3 Member Function Documentation	802
10.75.3.1 SetReference()	802
10.76Function_NodeCallback< Function > Class Template Reference	803
10.76.1 Detailed Description	803
10.76.2 Constructor & Destructor Documentation	804
10.76.2.1 Function_NodeCallback()	804
10.76.3 Member Function Documentation	804
10.76.3.1 Destroy()	804
10.76.3.2 operator()()	804
10.77gcstring Class Reference	805
10.77.1 Constructor & Destructor Documentation	806
10.77.1.1 gcstring() [1/5]	806
10.77.1.2 gcstring() [2/5]	806
10.77.1.3 gcstring() [3/5]	806
10.77.1.4 gcstring() [4/5]	806
10.77.1.5 gcstring() [5/5]	807
10.77.1.6 ~gcstring()	807
10.77.2 Member Function Documentation	807
10.77.2.1 _npos()	807

10.77.2.2 <code>append()</code> [1/2]	807
10.77.2.3 <code>append()</code> [2/2]	807
10.77.2.4 <code>assign()</code> [1/4]	807
10.77.2.5 <code>assign()</code> [2/4]	808
10.77.2.6 <code>assign()</code> [3/4]	808
10.77.2.7 <code>assign()</code> [4/4]	808
10.77.2.8 <code>c_str()</code>	808
10.77.2.9 <code>compare()</code>	808
10.77.2.10 <code>empty()</code>	808
10.77.2.11 <code>find()</code> [1/5]	809
10.77.2.12 <code>find()</code> [2/5]	809
10.77.2.13 <code>find()</code> [3/5]	809
10.77.2.14 <code>find()</code> [4/5]	809
10.77.2.15 <code>find()</code> [5/5]	809
10.77.2.16 <code>ind_first_not_of()</code>	809
10.77.2.17 <code>ind_first_of()</code>	810
10.77.2.18 <code>length()</code>	810
10.77.2.19 <code>max_size()</code>	810
10.77.2.20 <code>operator const char *</code>	810
10.77.2.21 <code>operator delete()</code> [1/2]	810
10.77.2.22 <code>operator delete()</code> [2/2]	810
10.77.2.23 <code>operator new()</code> [1/2]	810
10.77.2.24 <code>operator new()</code> [2/2]	811
10.77.2.25 <code>operator"!="()</code> [1/2]	811
10.77.2.26 <code>operator"!="()</code> [2/2]	811
10.77.2.27 <code>operator+=()</code> [1/5]	811
10.77.2.28 <code>operator+=()</code> [2/5]	811
10.77.2.29 <code>operator+=()</code> [3/5]	811
10.77.2.30 <code>operator+=()</code> [4/5]	811
10.77.2.31 <code>operator+=()</code> [5/5]	812

10.77.2.3 <code>operator<()</code>	812
10.77.2.3 <code>operator=()</code>	812
10.77.2.3 <code>operator==() [1/2]</code>	812
10.77.2.3 <code>operator==() [2/2]</code>	812
10.77.2.3 <code>operator>()</code>	812
10.77.2.3 <code>resize()</code>	812
10.77.2.3 <code>size()</code>	813
10.77.2.3 <code>substr()</code>	813
10.77.2.4 <code>swap()</code>	813
10.77.3 Friends And Related Function Documentation	813
10.77.3.1 <code>operator+ [1/3]</code>	813
10.77.3.2 <code>operator+ [2/3]</code>	813
10.77.3.3 <code>operator+ [3/3]</code>	813
10.77.4 Member Data Documentation	814
10.77.4.1 <code>npos</code>	814
10.78 GVCP_CHUNK_TRAILER Struct Reference	814
10.78.1 Detailed Description	814
10.78.2 Member Data Documentation	814
10.78.2.1 <code>ChunkID</code>	814
10.78.2.2 <code>ChunkLength</code>	814
10.79 GVCP_EVENT_ITEM Struct Reference	815
10.79.1 Detailed Description	815
10.79.2 Member Data Documentation	815
10.79.2.1 <code>BlockId</code>	815
10.79.2.2 <code>EventId</code>	815
10.79.2.3 <code>ReservedOrEventSize</code>	815
10.79.2.4 <code>StreamChannelId</code>	815
10.79.2.5 <code>TimestampHigh</code>	816
10.79.2.6 <code>TimestampLow</code>	816
10.80 GVCP_EVENT_ITEM_BASIC Struct Reference	816

10.80.1 Detailed Description	816
10.80.2 Member Data Documentation	816
10.80.2.1 EventId	816
10.80.2.2 ReservedOrEventSize	816
10.81GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference	817
10.81.1 Detailed Description	817
10.81.2 Member Data Documentation	817
10.81.2.1 BlockId	817
10.81.2.2 BlockId64High	817
10.81.2.3 BlockId64Low	817
10.81.2.4 EventId	817
10.81.2.5 ReservedOrEventSize	818
10.81.2.6 StreamChannelId	818
10.81.2.7 TimestampHigh	818
10.81.2.8 TimestampLow	818
10.82GVCP_EVENT_REQUEST Struct Reference	818
10.82.1 Detailed Description	819
10.82.2 Member Data Documentation	819
10.82.2.1 Header	819
10.82.2.2 Items	819
10.83GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference	819
10.83.1 Detailed Description	820
10.83.2 Member Data Documentation	820
10.83.2.1 Header	820
10.83.2.2 Items	820
10.84GVCP_EVENTDATA_REQUEST Struct Reference	820
10.84.1 Detailed Description	821
10.84.2 Member Data Documentation	821
10.84.2.1 Data	821
10.84.2.2 Event	821

10.84.2.3 Header	821
10.85GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference	821
10.85.1 Detailed Description	822
10.85.2 Member Data Documentation	822
10.85.2.1 Data	822
10.85.2.2 Event	822
10.85.2.3 Header	822
10.86GVCP_REQUEST_HEADER Struct Reference	822
10.86.1 Detailed Description	823
10.86.2 Member Data Documentation	823
10.86.2.1 Command	823
10.86.2.2 Flags	823
10.86.2.3 Length	823
10.86.2.4 Magic	823
10.86.2.5 ReqId	823
10.87H264Option Struct Reference	823
10.87.1 Detailed Description	824
10.87.2 Constructor & Destructor Documentation	824
10.87.2.1 H264Option()	824
10.87.3 Member Data Documentation	824
10.87.3.1 bitrate	824
10.87.3.2 frameRate	825
10.87.3.3 height	825
10.87.3.4 reserved	825
10.87.3.5 width	825
10.88IArrivalEvent Class Reference	826
10.88.1 Constructor & Destructor Documentation	827
10.88.1.1 ~IArrivalEvent()	827
10.88.1.2 IArrivalEvent() [1/2]	827
10.88.1.3 IArrivalEvent() [2/2]	827

10.88.2 Member Function Documentation	827
10.88.2.1 OnDeviceArrival()	827
10.88.2.2 operator=()	827
10.89 ICameraBase Class Reference	828
10.89.1 Detailed Description	829
10.89.2 Constructor & Destructor Documentation	830
10.89.2.1 ~ICameraBase()	830
10.89.2.2 ICameraBase() [1/2]	830
10.89.2.3 ICameraBase() [2/2]	830
10.89.3 Member Function Documentation	830
10.89.3.1 BeginAcquisition()	830
10.89.3.2 DelInit()	830
10.89.3.3 DiscoverMaxPacketSize()	831
10.89.3.4 EndAcquisition()	831
10.89.3.5 ForceIP()	831
10.89.3.6 GetAccessMode()	831
10.89.3.7 GetBufferOwnership()	831
10.89.3.8 GetGuiXml()	831
10.89.3.9 GetNextImage()	832
10.89.3.10 GetNodeMap()	832
10.89.3.11 GetNumDataStreams()	832
10.89.3.12 GetNumImagesInUse()	832
10.89.3.13 GetTLDeviceNodeMap()	832
10.89.3.14 GetTLEStreamNodeMap()	832
10.89.3.15 GetUniqueID()	833
10.89.3.16 GetUserBufferCount()	833
10.89.3.17 GetUserBufferSize()	833
10.89.3.18 GetUserBufferTotalSize()	833
10.89.3.19 Init()	833
10.89.3.20 IsInitialized()	833

10.89.3.21IsStreaming()	834
10.89.3.22IsValid()	834
10.89.3.23operator=()	834
10.89.3.24ReadPort()	834
10.89.3.25RegisterEvent() [1/2]	834
10.89.3.26RegisterEvent() [2/2]	834
10.89.3.27SetBufferOwnership()	835
10.89.3.28SetUserBuffers() [1/2]	835
10.89.3.29SetUserBuffers() [2/2]	835
10.89.3.30UnregisterEvent()	835
10.89.3.31WritePort()	835
10.89.4 Friends And Related Function Documentation	836
10.89.4.1 CameralInternal	836
10.89.4.2 InterfacImpl	836
10.89.5 Member Data Documentation	836
10.89.5.1 m_pCameraBaseData	836
10.89.5.2 TLDevice	836
10.89.5.3 TLStream	836
10.90 ICameraList Class Reference	837
10.90.1 Detailed Description	837
10.90.2 Constructor & Destructor Documentation	838
10.90.2.1 ~ICameraList()	838
10.90.2.2 ICameraList() [1/2]	838
10.90.2.3 ICameraList() [2/2]	838
10.90.3 Member Function Documentation	838
10.90.3.1 Append()	838
10.90.3.2 Clear()	838
10.90.3.3 GetByIndex()	839
10.90.3.4 GetBySerial()	839
10.90.3.5 GetSize()	839

10.90.3.6 operator=()	839
10.90.3.7 operator[]()	839
10.90.3.8 RemoveByIndex()	839
10.90.3.9 RemoveBySerial()	840
10.90.4 Friends And Related Function Documentation	840
10.90.4.1 CameraListImpl	840
10.90.4.2 InterfacelImpl	840
10.90.5 Member Data Documentation	840
10.90.5.1 m_pCameraListData	840
10.91 IChunkData Class Reference	840
10.91.1 Detailed Description	841
10.91.2 Constructor & Destructor Documentation	842
10.91.2.1 ~IChunkData()	842
10.91.2.2 IChunkData()	842
10.91.3 Member Function Documentation	842
10.91.3.1 GetBlackLevel()	842
10.91.3.2 GetCounterValue()	842
10.91.3.3 GetCRC()	842
10.91.3.4 GetEncoderValue()	843
10.91.3.5 GetExposureEndLineStatusAll()	843
10.91.3.6 GetExposureTime()	843
10.91.3.7 GetFrameID()	843
10.91.3.8 GetGain()	843
10.91.3.9 GetHeight()	843
10.91.3.10 GetImage()	844
10.91.3.11 GetInferenceBoundingBoxResult()	844
10.91.3.12 GetInferenceConfidence()	844
10.91.3.13 GetInferenceResult()	844
10.91.3.14 GetLinePitch()	844
10.91.3.15 GetLineStatusAll()	844

10.91.3.16GetOffsetX()	845
10.91.3.17GetOffsetY()	845
10.91.3.18GetPartSelector()	845
10.91.3.19GetPixelDynamicRangeMax()	845
10.91.3.20GetPixelDynamicRangeMin()	845
10.91.3.21GetScan3dAxisMax()	845
10.91.3.22GetScan3dAxisMin()	846
10.91.3.23GetScan3dCoordinateOffset()	846
10.91.3.24GetScan3dCoordinateReferenceValue()	846
10.91.3.25GetScan3dCoordinateScale()	846
10.91.3.26GetScan3dInvalidDataValue()	846
10.91.3.27GetScan3dTransformValue()	846
10.91.3.28GetScanLineSelector()	847
10.91.3.29GetSequencerSetActive()	847
10.91.3.30GetSerialDataLength()	847
10.91.3.31GetStreamChannelID()	847
10.91.3.32GetTimerValue()	847
10.91.3.33GetTimestamp()	847
10.91.3.34GetTimestampLatchValue()	848
10.91.3.35GetTransferBlockID()	848
10.91.3.36GetTransferQueueCurrentBlockCount()	848
10.91.3.37GetWidth()	848
10.91.3.38SetChunks()	848
10.92 IDatasream Class Reference	849
10.92.1 Constructor & Destructor Documentation	849
10.92.1.1 ~IDatasream()	850
10.92.1.2 IDatasream()	850
10.92.2 Member Function Documentation	850
10.92.2.1 AnnounceImage() [1/3]	850
10.92.2.2 AnnounceImage() [2/3]	850

10.92.2.3 AnnounceImage() [3/3]	850
10.92.2.4 AttachBuffer()	850
10.92.2.5 CleanupChunkAdapter()	851
10.92.2.6 FlushQueueAllDiscard()	851
10.92.2.7 GetBufferChunkData()	851
10.92.2.8 GetBufferInfoBool8Type()	851
10.92.2.9 GetBufferInfoPtrType()	851
10.92.2.10GetBufferInfoSizeType()	851
10.92.2.11GetBufferInfoUInt64Type()	852
10.92.2.12GetDeviceNodeMap()	852
10.92.2.13GetNextImage()	852
10.92.2.14GetNextImageInternal()	852
10.92.2.15GetNodeMap()	852
10.92.2.16GetNumImagesInUse()	852
10.92.2.17GetPort()	852
10.92.2.18GetStreamType()	853
10.92.2.19InitChunkAdapter()	853
10.92.2.20IsCRCCheckEnabled()	853
10.92.2.21IsImageInUse()	853
10.92.2.22IsStreaming()	853
10.92.2.23KillBufferEvent()	853
10.92.2.24RegisterImageEvent()	853
10.92.2.25ReleaseImage()	854
10.92.2.26RevokeImages()	854
10.92.2.27StartStream()	854
10.92.2.28StopStream()	854
10.92.2.29TransportLayerStreamInfo()	854
10.92.2.30UnregisterImageEvent()	854
10.92.2.31WaitOnImageEvent()	854
10.93IDevFileStreamBase< CharType, Traits > Class Template Reference	855

10.93.1 Member Typedef Documentation	856
10.93.1.1 <code>filebuf_type</code>	856
10.93.1.2 <code>ios_type</code>	856
10.93.1.3 <code>istream_type</code>	856
10.93.2 Member Function Documentation	856
10.93.2.1 <code>close()</code>	856
10.93.2.2 <code>is_open()</code>	856
10.93.2.3 <code>open()</code>	856
10.93.2.4 <code>rdbuf()</code>	857
10.94 <code>IDevFileStreamBuf< CharType, Traits ></code> Class Template Reference	857
10.94.1 Constructor & Destructor Documentation	858
10.94.1.1 <code>IDevFileStreamBuf()</code>	858
10.94.1.2 <code>~IDevFileStreamBuf()</code>	858
10.94.2 Member Function Documentation	858
10.94.2.1 <code>close()</code>	858
10.94.2.2 <code>is_open()</code>	858
10.94.2.3 <code>open()</code>	859
10.94.2.4 <code>pbackfail()</code>	859
10.94.2.5 <code>underflow()</code>	859
10.95 <code>IDeviceEvent</code> Class Reference	859
10.95.1 Constructor & Destructor Documentation	860
10.95.1.1 <code>~IDeviceEvent()</code>	860
10.95.1.2 <code>IDeviceEvent() [1/2]</code>	860
10.95.1.3 <code>IDeviceEvent() [2/2]</code>	861
10.95.2 Member Function Documentation	861
10.95.2.1 <code>GetDeviceEventId()</code>	861
10.95.2.2 <code>GetDeviceEventName()</code>	861
10.95.2.3 <code>OnDeviceEvent()</code>	861
10.95.2.4 <code>operator=()</code>	861
10.96 <code>IImage</code> Class Reference	862

10.96.1 Detailed Description	863
10.96.2 Constructor & Destructor Documentation	863
10.96.2.1 ~IImage()	864
10.96.2.2 IImage()	864
10.96.3 Member Function Documentation	864
10.96.3.1 CalculateStatistics()	864
10.96.3.2 CheckCRC()	864
10.96.3.3 Convert() [1/2]	864
10.96.3.4 Convert() [2/2]	864
10.96.3.5 DeepCopy()	865
10.96.3.6 DEPRECATED_FUNC() [1/3]	865
10.96.3.7 DEPRECATED_FUNC() [2/3]	865
10.96.3.8 DEPRECATED_FUNC() [3/3]	865
10.96.3.9 GetBitsPerPixel()	865
10.96.3.10 GetBufferSize()	866
10.96.3.11 GetChunkData()	866
10.96.3.12 GetChunkLayoutId()	866
10.96.3.13 GetColorProcessing()	866
10.96.3.14 GetData()	866
10.96.3.15 GetDataAbsoluteMax()	866
10.96.3.16 GetDataAbsoluteMin()	867
10.96.3.17 GetFrameID()	867
10.96.3.18 GetHeight()	867
10.96.3.19 GetID()	867
10.96.3.20 GetImageData()	867
10.96.3.21 GetImageSize()	867
10.96.3.22 GetImageStatus()	868
10.96.3.23 GetNumChannels()	868
10.96.3.24 GetPayloadType()	868
10.96.3.25 GetPixelFormat()	868

10.96.3.26GetPixelFormatIntType()	868
10.96.3.27GetPixelFormatName()	868
10.96.3.28GetPrivateData()	869
10.96.3.29GetStride()	869
10.96.3.30GetTimeStamp()	869
10.96.3.31GetTLPayloadType()	869
10.96.3.32GetTLPixelFormat()	869
10.96.3.33GetTLPixelFormatNamespace()	869
10.96.3.34GetValidPayloadSize()	870
10.96.3.35GetWidth()	870
10.96.3.36GetXOffset()	870
10.96.3.37GetXPadding()	870
10.96.3.38GetYOffset()	870
10.96.3.39GetYPadding()	870
10.96.3.40HasCRC()	871
10.96.3.41IsIncomplete()	871
10.96.3.42IsInUse()	871
10.96.3.43Release()	871
10.96.3.44ResetImage() [1/2]	871
10.96.3.45ResetImage() [2/2]	872
10.96.3.46Save() [1/8]	872
10.96.3.47Save() [2/8]	872
10.96.3.48Save() [3/8]	872
10.96.3.49Save() [4/8]	872
10.96.3.50Save() [5/8]	873
10.96.3.51Save() [6/8]	873
10.96.3.52Save() [7/8]	873
10.96.3.53Save() [8/8]	873
10.96.4 Friends And Related Function Documentation	873
10.96.4.1 Stream	873

10.97 IImageEvent Class Reference	874
10.97.1 Constructor & Destructor Documentation	875
10.97.1.1 ~IImageEvent()	875
10.97.1.2 IImageEvent() [1/2]	875
10.97.1.3 IImageEvent() [2/2]	875
10.97.2 Member Function Documentation	875
10.97.2.1 OnImageEvent()	875
10.97.2.2 operator=()	875
10.98 IImageStatistics Class Reference	876
10.98.1 Detailed Description	876
10.98.2 Constructor & Destructor Documentation	877
10.98.2.1 ~IImageStatistics()	877
10.98.2.2 IImageStatistics() [1/2]	877
10.98.2.3 IImageStatistics() [2/2]	877
10.98.3 Member Function Documentation	877
10.98.3.1 DisableAll()	877
10.98.3.2 EnableAll()	877
10.98.3.3 EnableGreyOnly()	877
10.98.3.4 EnableHSLOnly()	878
10.98.3.5 EnableRGBOnly()	878
10.98.3.6 GetChannelStatus()	878
10.98.3.7 GetHistogram()	878
10.98.3.8 GetMean()	878
10.98.3.9 GetNumPixelValues()	879
10.98.3.10 GetPixelValueRange()	879
10.98.3.11 GetRange()	879
10.98.3.12 GetStatistics()	879
10.98.3.13 SetChannelStatus()	880
10.99 IInterface Class Reference	880
10.99.1 Detailed Description	881

10.99.2 Constructor & Destructor Documentation	881
10.99.2.1 ~IInterface()	881
10.99.2.2 IInterface() [1/2]	882
10.99.2.3 IInterface() [2/2]	882
10.99.3 Member Function Documentation	882
10.99.3.1 GetCameras()	882
10.99.3.2 GetTLNodeMap()	882
10.99.3.3 IsInUse()	882
10.99.3.4 IsValid()	882
10.99.3.5 operator=()	883
10.99.3.6 RegisterEvent()	883
10.99.3.7 SendActionCommand()	883
10.99.3.8 UnregisterEvent()	883
10.99.3.9 UpdateCameras()	883
10.99.4 Friends And Related Function Documentation	883
10.99.4.1 InterfaceInternal	884
10.99.4.2 SystemImpl	884
10.99.5 Member Data Documentation	884
10.99.5.1 m_pInterfaceData	884
10.99.5.2 TLInterface	884
10.100 IInterfaceArrivalEvent Class Reference	884
10.100.1 Constructor & Destructor Documentation	885
10.100.1.1 ~IInterfaceArrivalEvent()	885
10.100.1.2 IInterfaceArrivalEvent() [1/2]	885
10.100.1.3 IInterfaceArrivalEvent() [2/2]	886
10.100.2 Member Function Documentation	886
10.100.2.1 OnInterfaceArrival()	886
10.100.2.2 operator=()	886
10.101 IInterfaceEvent Class Reference	886
10.101.1 Constructor & Destructor Documentation	887

10.101.1.1~IInterfaceEvent()	887
10.101.1.2IInterfaceEvent() [1/2]	887
10.101.1.3IInterfaceEvent() [2/2]	888
10.101.2Member Function Documentation	888
10.101.2.1OnDeviceArrival()	888
10.101.2.2OnDeviceRemoval()	888
10.101.2.3operator=(*)	888
10.101.3InterfaceList Class Reference	889
10.102.1Detailed Description	889
10.102.2Constructor & Destructor Documentation	889
10.102.2.1~IInterfaceList()	890
10.102.2.2IInterfaceList() [1/2]	890
10.102.2.3IInterfaceList() [2/2]	890
10.102.3Member Function Documentation	890
10.102.3.1Clear()	890
10.102.3.2GetByIndex()	890
10.102.3.3GetSize()	890
10.102.3.4operator=(*)	891
10.102.3.5operator[]()	891
10.102.4Member Data Documentation	891
10.102.4.1m_pInterfaceListData	891
10.101.4InterfaceRemovalEvent Class Reference	891
10.103.1Constructor & Destructor Documentation	892
10.103.1.1~IInterfaceRemovalEvent()	892
10.103.1.2IInterfaceRemovalEvent() [1/2]	892
10.103.1.3IInterfaceRemovalEvent() [2/2]	893
10.103.2Member Function Documentation	893
10.103.2.1OnInterfaceRemoval()	893
10.103.2.2operator=(*)	893
10.104LoggingEvent Class Reference	893

10.104.1 Constructor & Destructor Documentation	894
10.104.1.1~ILoggingEvent()	894
10.104.1.2LoggingEvent() [1/2]	894
10.104.1.3LoggingEvent() [2/2]	895
10.104.2 Member Function Documentation	895
10.104.2.1OnLogEvent()	895
10.104.2.2operator=()	895
10.105 Image Class Reference	895
10.105.1 Detailed Description	899
10.105.2 Constructor & Destructor Documentation	899
10.105.2.1~Image()	900
10.105.2.2Image() [1/3]	900
10.105.2.3Image() [2/3]	900
10.105.2.4Image() [3/3]	900
10.105.3 Member Function Documentation	900
10.105.3.1CalculateStatistics()	900
10.105.3.2CheckCRC()	901
10.105.3.3Convert() [1/3]	901
10.105.3.4Convert() [2/3]	901
10.105.3.5Convert() [3/3]	902
10.105.3.6Create() [1/3]	902
10.105.3.7Create() [2/3]	902
10.105.3.8Create() [3/3]	903
10.105.3.9CreateShared()	903
10.105.3.10DeepCopy() [1/2]	903
10.105.3.10DeepCopy() [2/2]	904
10.105.3.10DEPRECATED_FUNC() [1/6]	904
10.105.3.10DEPRECATED_FUNC() [2/6]	904
10.105.3.10DEPRECATED_FUNC() [3/6]	905
10.105.3.10DEPRECATED_FUNC() [4/6]	905

10.105.3.1 D EPRECATED_FUNC() [5/6]	906
10.105.3.1 D EPRECATED_FUNC() [6/6]	906
10.105.3.1 B stractPolarization()	906
10.105.3.1 G etBitsPerPixel()	907
10.105.3.2 G etBufferSize()	907
10.105.3.2 G etChunkData()	907
10.105.3.2 G etChunkLayoutId()	908
10.105.3.2 G etColorProcessing()	908
10.105.3.2 G etData()	908
10.105.3.2 S etDataAbsoluteMax()	909
10.105.3.2 S etDataAbsoluteMin()	909
10.105.3.2 G etDefaultColorProcessing()	909
10.105.3.2 S etFrameID()	910
10.105.3.2 G etHeight()	910
10.105.3.3 G etID()	910
10.105.3.3 G etImageData()	910
10.105.3.3 G etImageSize()	911
10.105.3.3 S etImageStatus()	911
10.105.3.3 G etImageStatusDescription()	911
10.105.3.3 S etNumChannels()	911
10.105.3.3 G etPayloadType()	912
10.105.3.3 G etPixelFormat()	912
10.105.3.3 S etPixelFormatIntType()	912
10.105.3.3 G etPixelFormatName()	913
10.105.3.4 G etPolarizationAlgorithm()	913
10.105.3.4 G etPolarizationValues()	913
10.105.3.4 G etPrivateData()	914
10.105.3.4 G etStride()	914
10.105.3.4 G etTimeStamp()	914
10.105.3.4 S etTLPayloadType()	915

10.105.3.46GetTLPixelFormat()	915
10.105.3.47GetTLPixelFormatNamespace()	915
10.105.3.48GetValidPayloadSize()	916
10.105.3.49GetWidth()	916
10.105.3.50GetXOffset()	916
10.105.3.51GetXPadding()	917
10.105.3.52GetYOffset()	917
10.105.3.53GetYPadding()	917
10.105.3.54asCRC()	918
10.105.3.55Compressed()	918
10.105.3.56Incomplete()	918
10.105.3.57InUse()	918
10.105.3.58Release()	919
10.105.3.59ResetImage() [1/2]	919
10.105.3.60ResetImage() [2/2]	919
10.105.3.61Save() [1/8]	920
10.105.3.62Save() [2/8]	920
10.105.3.63Save() [3/8]	920
10.105.3.64Save() [4/8]	921
10.105.3.65Save() [5/8]	921
10.105.3.66Save() [6/8]	921
10.105.3.67Save() [7/8]	922
10.105.3.68Save() [8/8]	922
10.105.3.69SetDefaultColorProcessing()	922
10.105.4Friends And Related Function Documentation	923
10.105.4.1IDataStream	923
10.105.4.2ImageConverter	923
10.105.4.3ImageFiler	923
10.105.4.4ImageStatsCalculator	923
10.105.4.5ImageUtilityImpl	923

10.105.4.6ImageUtilityPolarizationImpl	924
10.105.4.7Stream	924
10.106Image Event Class Reference	924
10.106.1Detailed Description	925
10.106.2Constructor & Destructor Documentation	925
10.106.2.1ImageEvent()	926
10.106.2.2~ImageEvent()	926
10.106.3Member Function Documentation	926
10.106.3.1OnImageEvent()	926
10.106.3.2operator=()	926
10.107ImagePtr Class Reference	927
10.107.1Detailed Description	928
10.107.2Constructor & Destructor Documentation	928
10.107.2.1ImagePtr() [1/4]	928
10.107.2.2ImagePtr() [2/4]	928
10.107.2.3ImagePtr() [3/4]	928
10.107.2.4ImagePtr() [4/4]	928
10.107.2.5~ImagePtr()	929
10.107.3Member Function Documentation	929
10.107.3.1operator=()	929
10.108ImageStatistics Class Reference	929
10.108.1Detailed Description	930
10.108.2Constructor & Destructor Documentation	931
10.108.2.1ImageStatistics() [1/2]	931
10.108.2.2~ImageStatistics()	931
10.108.2.3ImageStatistics() [2/2]	931
10.108.3Member Function Documentation	931
10.108.3.1DisableAll()	931
10.108.3.2EnableAll()	931
10.108.3.3EnableGreyOnly()	932

10.108.3.4EnableHSLOnly()	932
10.108.3.5EnableRGBOnly()	932
10.108.3.6GetChannelStatus()	932
10.108.3.7GetHistogram()	933
10.108.3.8GetMean()	933
10.108.3.9GetNumPixelValues()	933
10.108.3.10GetPixelValueRange()	934
10.108.3.11GetRange()	934
10.108.3.12GetStatistics()	934
10.108.3.13operator=()	935
10.108.3.14GetChannelStatus()	935
10.108.4Friends And Related Function Documentation	936
10.108.4.1ImageStatsCalculator	936
10.109ImageUtility Class Reference	936
10.109.1Detailed Description	937
10.109.2Member Enumeration Documentation	937
10.109.2.1ImageScalingAlgorithm	937
10.109.2.2SourceDataRange	937
10.109.3Member Function Documentation	938
10.109.3.1CreateNormalized() [1/5]	938
10.109.3.2CreateNormalized() [2/5]	938
10.109.3.3CreateNormalized() [3/5]	939
10.109.3.4CreateNormalized() [4/5]	939
10.109.3.5CreateNormalized() [5/5]	940
10.109.3.6CreateScaled() [1/2]	940
10.109.3.7CreateScaled() [2/2]	941
10.110ImageUtilityHeatmap Class Reference	941
10.110.1Detailed Description	942
10.110.2Member Enumeration Documentation	942
10.110.2.1HeatmapColor	942

10.110.3Member Function Documentation	942
10.110.3.1CreateHeatmap() [1/2]	942
10.110.3.2CreateHeatmap() [2/2]	943
10.110.3.3GetHeatmapColorGradient()	943
10.110.3.4GetHeatmapRange()	944
10.110.3.5SetHeatmapColorGradient()	944
10.110.3.6SetHeatmapRange()	945
10.11ImageUtilityPolarization Class Reference	945
10.111.Detailed Description	946
10.111.1Member Enumeration Documentation	946
10.111.2.1PolarizationQuadrant	946
10.111.3Member Function Documentation	947
10.111.3.1CreateAolp() [1/2]	947
10.111.3.2CreateAolp() [2/2]	947
10.111.3.3CreateDolp() [1/2]	948
10.111.3.4CreateDolp() [2/2]	948
10.111.3.5CreateGlareReduced() [1/2]	948
10.111.3.6CreateGlareReduced() [2/2]	949
10.111.3.7CreateStokesS0() [1/2]	949
10.111.3.8CreateStokesS0() [2/2]	950
10.111.3.9CreateStokesS1() [1/2]	950
10.111.3.10CreateStokesS1() [2/2]	950
10.111.3.11CreateStokesS2() [1/2]	951
10.111.3.12CreateStokesS2() [2/2]	951
10.111.3.13ExtractPolarQuadrant() [1/2]	952
10.111.3.14ExtractPolarQuadrant() [2/2]	952
10.11InferenceBoundingBox Struct Reference	953
10.112.Detailed Description	953
10.11InferenceBoundingBoxResult Class Reference	953
10.113.Detailed Description	954

10.114InferenceBoxCircle Struct Reference	954
10.115InferenceBoxRect Struct Reference	954
10.115.1Detailed Description	954
10.116InferenceBoxRotatedRect Struct Reference	955
10.117Int64_autovector_t Class Reference	955
10.117.1Detailed Description	955
10.117.2Constructor & Destructor Documentation	955
10.117.2.1int64_autovector_t() [1/3]	956
10.117.2.2nt64_autovector_t() [2/3]	956
10.117.2.3nt64_autovector_t() [3/3]	956
10.117.2.4~int64_autovector_t()	956
10.117.3Member Function Documentation	956
10.117.3.1operator delete()	956
10.117.3.2operator new()	956
10.117.3.3operator=()	957
10.117.3.4operator[]() [1/2]	957
10.117.3.5operator[]() [2/2]	957
10.117.3.6size()	957
10.117.4Member Data Documentation	957
10.117.4.1_pCount	957
10.117.4.2_pv	957
10.118IntegerNode Class Reference	958
10.118.1Detailed Description	960
10.118.2Constructor & Destructor Documentation	960
10.118.2.1IntegerNode() [1/2]	960
10.118.2.2IntegerNode() [2/2]	960
10.118.2.3~IntegerNode()	960
10.118.3Member Function Documentation	960
10.118.3.1GetFloatAlias()	961
10.118.3.2GetInc()	961

10.118.3.3GetIncMode()	961
10.118.3.4GetListOfValidValues()	961
10.118.3.5GetMax()	961
10.118.3.6GetMin()	961
10.118.3.7GetRepresentation()	962
10.118.3.8GetUnit()	962
10.118.3.9GetValue()	962
10.118.3.10imposeMax()	962
10.118.3.11imposeMin()	962
10.118.3.12operator()()	963
10.118.3.13operator*()	963
10.118.3.14operator=()	963
10.118.3.15SetReference()	963
10.118.3.16SetValue()	963
10.119Interface Class Reference	964
10.119.1Detailed Description	965
10.119.2Constructor & Destructor Documentation	965
10.119.2.1~Interface()	965
10.119.3Member Function Documentation	965
10.119.3.1GetCameras()	966
10.119.3.2GetTLNodeMap()	966
10.119.3.3IsInUse()	966
10.119.3.4IsValid()	967
10.119.3.5RegisterEvent()	967
10.119.3.6SendActionCommand()	967
10.119.3.7UnregisterEvent()	968
10.119.3.8UpdateCameras()	968
10.119.4Friends And Related Function Documentation	969
10.119.4.1InterfaceInternal	969
10.120InterfaceArrivalEvent Class Reference	969

10.120.1Detailed Description	970
10.120.2Constructor & Destructor Documentation	970
10.120.2.1InterfaceArrivalEvent()	970
10.120.2.2~InterfaceArrivalEvent()	970
10.120.3Member Function Documentation	970
10.120.3.1OnInterfaceArrival()	970
10.120.3.2operator=()	971
10.121InterfaceEvent Class Reference	971
10.121.1Detailed Description	972
10.121.2Constructor & Destructor Documentation	972
10.121.2.1InterfaceEvent()	973
10.121.2.2~InterfaceEvent()	973
10.121.3Member Function Documentation	973
10.121.3.1OnDeviceArrival()	973
10.121.3.2OnDeviceRemoval()	973
10.121.3.3operator=()	973
10.122InterfaceList Class Reference	974
10.122.1Detailed Description	975
10.122.2Constructor & Destructor Documentation	975
10.122.2.1InterfaceList() [1/2]	975
10.122.2.2~InterfaceList()	975
10.122.2.3InterfaceList() [2/2]	975
10.122.3Member Function Documentation	975
10.122.3.1Clear()	976
10.122.3.2GetByIndex()	976
10.122.3.3GetSize()	976
10.122.3.4operator=()	977
10.122.3.5operator[]()	977
10.122.4Friends And Related Function Documentation	977
10.122.4.1SystemImpl	977

10.128InterfacePtr Class Reference	977
10.123.Detailed Description	978
10.123.2Constructor & Destructor Documentation	978
10.123.2.1lInterfacePtr() [1/4]	978
10.123.2.2lInterfacePtr() [2/4]	978
10.123.2.3lInterfacePtr() [3/4]	979
10.123.2.4lInterfacePtr() [4/4]	979
10.124InterfaceRemovalEvent Class Reference	979
10.124.Detailed Description	980
10.124.2Constructor & Destructor Documentation	980
10.124.2.1lInterfaceRemovalEvent()	981
10.124.2.2~lInterfaceRemovalEvent()	981
10.124.3Member Function Documentation	981
10.124.3.1OnInterfaceRemoval()	981
10.124.3.2operator=()	981
10.125IntRegNode Class Reference	982
10.125.Detailed Description	983
10.125.2Constructor & Destructor Documentation	983
10.125.2.1lIntRegNode() [1/2]	984
10.125.2.2lIntRegNode() [2/2]	984
10.125.2.3~lIntRegNode()	984
10.125.3Member Function Documentation	984
10.125.3.1SetReference()	984
10.126IpInfo Struct Reference	984
10.126.Constructor & Destructor Documentation	985
10.126.1.1lIpInfo()	985
10.126.2Member Data Documentation	985
10.126.2.1gateway	985
10.126.2.2pAddress	985
10.126.2.3subnetLength	985

10.126.2.4	subnetMask	985
10.127	IRemovalEvent Class Reference	986
10.127.	Constructor & Destructor Documentation	987
10.127.1.1~	IRemovalEvent()	987
10.127.1.2	IRemovalEvent() [1/2]	987
10.127.1.3	IRemovalEvent() [2/2]	987
10.127.1	Member Function Documentation	987
10.127.2.1	OnDeviceRemoval()	987
10.127.2.2	operator=()	987
10.128	ISystem Class Reference	988
10.128.1	Detailed Description	989
10.128.2	Constructor & Destructor Documentation	989
10.128.2.1~	ISystem()	989
10.128.2.2	ISystem() [1/2]	989
10.128.2.3	ISystem() [2/2]	989
10.128.3	Member Function Documentation	990
10.128.3.1	GetCameras()	990
10.128.3.2	GetInterfaces()	990
10.128.3.3	GetLibraryVersion()	990
10.128.3.4	GetLoggingEventPriorityLevel()	990
10.128.3.5	GetTLNodeMap()	990
10.128.3.6	IsInUse()	991
10.128.3.7	operator=()	991
10.128.3.8	RegisterEvent()	991
10.128.3.9	RegisterInterfaceEvent()	991
10.128.3.10	RegisterLoggingEvent()	991
10.128.3.11	ReleaseInstance()	991
10.128.3.12	SendActionCommand()	992
10.128.3.13	SetLoggingEventPriorityLevel()	992
10.128.3.14	UnregisterAllLoggingEvent()	992

10.128.3.15UnregisterEvent()	992
10.128.3.16UnregisterInterfaceEvent()	992
10.128.3.17UnregisterLoggingEvent()	993
10.128.3.18updateCameras()	993
10.128.3.19updateInterfaceList()	993
10.128.4Friends And Related Function Documentation	993
10.128.4.1SystemPtrInternal	993
10.128.5Member Data Documentation	993
10.128.5.1TLSYSTEM	993
10.128.6SystemEvent Class Reference	994
10.129.Constructor & Destructor Documentation	995
10.129.1.1~ISystemEvent()	995
10.129.1.2SystemEvent() [1/2]	995
10.129.1.3SystemEvent() [2/2]	995
10.129.2Member Function Documentation	995
10.129.2.1OnInterfaceArrival()	995
10.129.2.2OnInterfaceRemoval()	996
10.129.2.3operator=(*)	996
10.130PEGOption Struct Reference	996
10.130.Detailed Description	996
10.130.2Constructor & Destructor Documentation	996
10.130.2.1JPEGOption()	997
10.130.3Member Data Documentation	997
10.130.3.1progressive	997
10.130.3.2quality	997
10.130.3.3reserved	997
10.131JPG2Option Struct Reference	997
10.131.Detailed Description	998
10.131.2Constructor & Destructor Documentation	998
10.131.2.1JPG2Option()	998

10.131.3Member Data Documentation	998
10.131.3.1quality	998
10.131.3.2reserved	998
10.132LibraryVersion Struct Reference	999
10.132.1Detailed Description	999
10.132.2Member Data Documentation	999
10.132.2.1build	999
10.132.2.2major	999
10.132.2.3minor	999
10.132.2.4type	1000
10.133LockableObject< Object >::Lock Class Reference	1000
10.133.1Detailed Description	1000
10.133.2Constructor & Destructor Documentation	1000
10.133.2.1Lock()	1000
10.133.2.2~Lock()	1000
10.134LockableObject< Object > Class Template Reference	1001
10.134.1Detailed Description	1001
10.134.2Member Function Documentation	1002
10.134.2.1GetLock()	1002
10.134.3Friends And Related Function Documentation	1002
10.134.3.1Lock	1002
10.134.4Member Data Documentation	1002
10.134.4.1m_Lock	1002
10.135LoggingEvent Class Reference	1003
10.135.1Detailed Description	1004
10.135.2Constructor & Destructor Documentation	1004
10.135.2.1LoggingEvent()	1004
10.135.2.2~LoggingEvent()	1004
10.135.3Member Function Documentation	1004
10.135.3.1OnLogEvent()	1004

10.135.3. <code>operator=()</code>	1005
10.136 <code>LoggingEventData</code> Class Reference	1005
10.136.1Detailed Description	1006
10.136.2Constructor & Destructor Documentation	1006
10.136.2.1 <code>~LoggingEventData()</code>	1006
10.136.2.2 <code>LoggingEventData()</code>	1006
10.136.3Member Function Documentation	1006
10.136.3.1 <code>GetCategoryName()</code>	1006
10.136.3.2 <code>GetLogMessage()</code>	1006
10.136.3.3 <code>GetNDC()</code>	1007
10.136.3.4 <code>GetPriority()</code>	1007
10.136.3.5 <code>GetPriorityName()</code>	1007
10.136.3.6 <code>GetThreadName()</code>	1007
10.136.3.7 <code>GetTimestamp()</code>	1008
10.136.4Friends And Related Function Documentation	1008
10.136.4.1 <code>SystemImpl</code>	1008
10.137 <code>LoggingEventDataPtr</code> Class Reference	1008
10.137.1Detailed Description	1009
10.137.2Constructor & Destructor Documentation	1009
10.137.2.1 <code>LoggingEventDataPtr() [1/4]</code>	1009
10.137.2.2 <code>LoggingEventDataPtr() [2/4]</code>	1009
10.137.2.3 <code>LoggingEventDataPtr() [3/4]</code>	1009
10.137.2.4 <code>LoggingEventDataPtr() [4/4]</code>	1010
10.138 <code>Member_NodeCallback< Client, Member ></code> Class Template Reference	1010
10.138.1Detailed Description	1011
10.138.2Member Typedef Documentation	1011
10.138.2.1 <code>PMEMBERFUNC</code>	1011
10.138.3Constructor & Destructor Documentation	1011
10.138.3.1 <code>Member_NodeCallback()</code>	1011
10.138.4Member Function Documentation	1011

10.138.4.1 <i>Destroy()</i>	1012
10.138.4.2 <i>operator()()</i>	1012
10.139 <i>MJPGOption Struct Reference</i>	1012
10.139.1 <i>Detailed Description</i>	1012
10.139.2 <i>Constructor & Destructor Documentation</i>	1012
10.139.2.1 <i>MJPGOption()</i>	1013
10.139.3 <i>Member Data Documentation</i>	1013
10.139.3.1 <i>frameRate</i>	1013
10.139.3.2 <i>quality</i>	1013
10.139.3.3 <i>reserved</i>	1013
10.140 <i>Node Class Reference</i>	1014
10.140.1 <i>Detailed Description</i>	1016
10.140.2 <i>Constructor & Destructor Documentation</i>	1016
10.140.2.1 <i>Node() [1/2]</i>	1016
10.140.2.2 <i>Node() [2/2]</i>	1017
10.140.2.3 <i>~Node()</i>	1017
10.140.3 <i>Member Function Documentation</i>	1017
10.140.3.1 <i>DeregisterCallback()</i>	1017
10.140.3.2 <i>GetAccessMode()</i>	1017
10.140.3.3 <i>GetAlias()</i>	1017
10.140.3.4 <i>GetCachingMode()</i>	1018
10.140.3.5 <i>GetCastAlias()</i>	1018
10.140.3.6 <i>GetChildren()</i>	1018
10.140.3.7 <i>GetDescription()</i>	1018
10.140.3.8 <i>GetDeviceName()</i>	1018
10.140.3.9 <i>GetDisplayName()</i>	1019
10.140.3.10 <i>GetDocuURL()</i>	1019
10.140.3.11 <i>GetEventID()</i>	1019
10.140.3.12 <i>GetName()</i>	1019
10.140.3.13 <i>GetNameSpace()</i>	1019

10.140.3.1 <code>GetNodeHandle()</code>	1019
10.140.3.1 <code>GetNodeMap()</code>	1019
10.140.3.1 <code>GetParents()</code>	1019
10.140.3.1 <code>GetPollingTime()</code>	1020
10.140.3.1 <code>GetPrincipalInterfaceType()</code>	1020
10.140.3.1 <code>GetProperty()</code>	1020
10.140.3.2 <code>GetPropertyNames()</code>	1020
10.140.3.2 <code>GetSelectedFeatures()</code>	1020
10.140.3.2 <code>GetSelectingFeatures()</code>	1021
10.140.3.2 <code>GetToolTip()</code>	1021
10.140.3.2 <code>GetVisibility()</code>	1021
10.140.3.2 <code>ImposeAccessMode()</code>	1021
10.140.3.2 <code>ImposeVisibility()</code>	1021
10.140.3.2 <code>ValidateNode()</code>	1021
10.140.3.2 <code>AccessModeCacheable()</code>	1022
10.140.3.2 <code>Cachable()</code>	1022
10.140.3.3 <code>Deprecated()</code>	1022
10.140.3.3 <code>Feature()</code>	1022
10.140.3.3 <code>Selector()</code>	1022
10.140.3.3 <code>Streamable()</code>	1022
10.140.3.3 <code>operator"!="()</code>	1023
10.140.3.3 <code>operator=="()</code>	1023
10.140.3.3 <code>RegisterCallback()</code>	1023
10.140.3.3 <code>SetNodeHandle()</code>	1023
10.140.3.3 <code>SetNodeMap()</code>	1023
10.140.3.3 <code>SetReference() [1/2]</code>	1023
10.140.3.4 <code>SetReference() [2/2]</code>	1024
10.140.4Member Data Documentation	1024
10.140.4.1 <code>m_Callbacks</code>	1024
10.140.4.2 <code>m_pNodeData</code>	1024

10.140.4.3m_pNodeMap	1024
10.14NodeMap Class Reference	1024
10.141. Detailed Description	1026
10.141.2Constructor & Destructor Documentation	1026
10.141.2.1NodeMap()	1027
10.141.2.2~NodeMap()	1027
10.141.3Member Function Documentation	1027
10.141.3.1ClearXMLCache()	1027
10.141.3.2Connect() [1/2]	1027
10.141.3.3Connect() [2/2]	1027
10.141.3.4Destroy()	1028
10.141.3.5GetDeviceName()	1028
10.141.3.6GetDeviceVersion()	1028
10.141.3.7GetGenApiVersion()	1028
10.141.3.8GetLock()	1028
10.141.3.9GetModelName()	1028
10.141.3.10GetNode()	1029
10.141.3.10GetNodeMapHandle()	1029
10.141.3.10GetNodes()	1029
10.141.3.10GetNumNodes()	1029
10.141.3.10GetProductGuid()	1029
10.141.3.10GetSchemaVersion()	1029
10.141.3.10GetStandardNameSpace()	1030
10.141.3.10GetSupportedSchemaVersions()	1030
10.141.3.10GetToolTip()	1030
10.141.3.10GetVendorName()	1030
10.141.3.20GetVersionGuid()	1031
10.141.3.21InvalidateNodes()	1031
10.141.3.22LoadXMLFromFile()	1031
10.141.3.23LoadXMLFromFileInject()	1031

10.141.3.24loadXMLFromString()	1031
10.141.3.25loadXMLFromStringInject()	1032
10.141.3.26loadXMLFromZIPData()	1032
10.141.3.27loadXMLFromZIPFile()	1032
10.141.3.28roll()	1032
10.141.4Member Data Documentation	1032
10.141.4.1_Ptr	1032
10.142NodeMapFactory::NodeStatistics_t Struct Reference	1033
10.142.1Member Data Documentation	1033
10.142.1.1NumLinks	1033
10.142.1.2NumNodes	1033
10.142.1.3NumProperties	1033
10.142.1.4NumStrings	1033
10.143DevFileStreamBase< CharType, Traits > Class Template Reference	1034
10.143.1Member Typedef Documentation	1035
10.143.1.1filebuf_type	1035
10.143.1.2os_type	1035
10.143.1.3ostream_type	1035
10.143.2Member Function Documentation	1035
10.143.2.1close()	1035
10.143.2.2s_open()	1035
10.143.2.3open()	1035
10.143.2.4dbuf()	1036
10.144DevFileStreamBuf< CharType, Traits > Class Template Reference	1036
10.144.1Constructor & Destructor Documentation	1037
10.144.1.1ODevFileStreamBuf()	1037
10.144.1.2~ODevFileStreamBuf()	1037
10.144.2Member Function Documentation	1037
10.144.2.1close()	1037
10.144.2.2s_open()	1037

10.144.2.3open()	1038
10.144.2.4overflow()	1038
10.144.2.5sync()	1038
10.144.2.6xputn()	1038
10.145 PGMOption Struct Reference	1038
10.145.1Detailed Description	1039
10.145.2Constructor & Destructor Documentation	1039
10.145.2.1PGMOption()	1039
10.145.3Member Data Documentation	1039
10.145.3.1binaryFile	1039
10.145.3.2reserved	1039
10.146 PNGOption Struct Reference	1039
10.146.1Detailed Description	1040
10.146.2Constructor & Destructor Documentation	1040
10.146.2.1PNGOption()	1040
10.146.3Member Data Documentation	1040
10.146.3.1compressionLevel	1040
10.146.3.2interlaced	1040
10.146.3.3reserved	1041
10.147 PortNode Class Reference	1041
10.147.1Detailed Description	1043
10.147.2Constructor & Destructor Documentation	1043
10.147.2.1PortNode() [1/2]	1043
10.147.2.2PortNode() [2/2]	1043
10.147.2.3~PortNode()	1043
10.147.3Member Function Documentation	1043
10.147.3.1CacheChunkData()	1043
10.147.3.2GetChunkID()	1044
10.147.3.3GetPortHandle()	1044
10.147.3.4GetSwapEndianess()	1044

10.147.3.5Read()	1044
10.147.3.6Replay()	1044
10.147.3.7SetPortImpl()	1045
10.147.3.8SetReference() [1/3]	1045
10.147.3.9SetReference() [2/3]	1045
10.147.3.10SetReference() [3/3]	1045
10.147.3.11StartRecording()	1045
10.147.3.12StopRecording()	1046
10.147.3.13Write()	1046
10.148PortRecorder Class Reference	1046
10.148.1Detailed Description	1047
10.148.2Constructor & Destructor Documentation	1047
10.148.2.1PortRecorder()	1048
10.148.2.2~PortRecorder()	1048
10.148.3Member Function Documentation	1048
10.148.3.1GetAccessMode()	1048
10.148.3.2SetReference()	1048
10.148.3.3StartRecording()	1048
10.148.3.4StopRecording()	1049
10.149PortReply Class Reference	1049
10.149.1Detailed Description	1050
10.149.2Constructor & Destructor Documentation	1050
10.149.2.1PortReply()	1050
10.149.2.2~PortReply()	1051
10.149.3Member Function Documentation	1051
10.149.3.1GetPortReplyHandle()	1051
10.149.3.2Replay()	1051
10.149.3.3SetReference()	1051
10.150PMOption Struct Reference	1051
10.150.1Detailed Description	1052

10.150.2Constructor & Destructor Documentation	1052
10.150.2.1PPMOption()	1052
10.150.3Member Data Documentation	1052
10.150.3.1binaryFile	1052
10.150.3.2reserved	1052
10.151RegisterNode Class Reference	1053
10.151.1Detailed Description	1054
10.151.2Constructor & Destructor Documentation	1054
10.151.2.1RegisterNode() [1/2]	1055
10.151.2.2RegisterNode() [2/2]	1055
10.151.2.3~RegisterNode()	1055
10.151.3Member Function Documentation	1055
10.151.3.1Get()	1055
10.151.3.2GetAddress()	1055
10.151.3.3GetLength()	1056
10.151.3.4Set()	1056
10.151.3.5SetReference()	1056
10.152RemovalEvent Class Reference	1057
10.152.1Detailed Description	1058
10.152.2Constructor & Destructor Documentation	1058
10.152.2.1RemovalEvent()	1058
10.152.2.2~RemovalEvent()	1058
10.152.3Member Function Documentation	1058
10.152.3.1OnDeviceRemoval()	1058
10.152.3.2operator=(())	1059
10.153SingleChunkData_t Struct Reference	1059
10.153.1Member Data Documentation	1059
10.153.1.1ChunkID	1059
10.153.1.2ChunkLength	1059
10.153.1.3ChunkOffset	1059

10.15 \$SingleChunkDataStr_t Struct Reference	1060
10.154.1 Member Data Documentation	1060
10.154.1.1 ChunkID	1060
10.154.1.2 ChunkLength	1060
10.154.1.3 ChunkOffset	1060
10.155 SpinTestCamera Class Reference	1061
10.156 SpinVideo Class Reference	1061
10.156.1 Detailed Description	1062
10.156.2 Constructor & Destructor Documentation	1062
10.156.2.1 SpinVideo()	1062
10.156.2.2 ~SpinVideo()	1062
10.156.3 Member Function Documentation	1062
10.156.3.1 Append()	1062
10.156.3.2 Close()	1063
10.156.3.3 Open() [1/3]	1063
10.156.3.4 Open() [2/3]	1063
10.156.3.5 Open() [3/3]	1065
10.156.3.6 SetMaximumFileSize()	1065
10.157 StringNode Class Reference	1066
10.157.1 Detailed Description	1067
10.157.2 Constructor & Destructor Documentation	1068
10.157.2.1 StringNode() [1/2]	1068
10.157.2.2 StringNode() [2/2]	1068
10.157.2.3 ~StringNode()	1068
10.157.3 Member Function Documentation	1068
10.157.3.1 GetMaxLength()	1068
10.157.3.2 GetValue()	1068
10.157.3.3 operator()()	1069
10.157.3.4 operator*()	1069
10.157.3.5 operator=(())	1069

10.157.3.6SetReference()	1069
10.157.3.7SetValue()	1069
10.158\$StringRegNode Class Reference	1070
10.158.1Detailed Description	1071
10.158.2Constructor & Destructor Documentation	1071
10.158.2.1StringRegNode() [1/2]	1072
10.158.2.2StringRegNode() [2/2]	1072
10.158.2.3~StringRegNode()	1072
10.158.3Member Function Documentation	1072
10.158.3.1SetReference()	1072
10.159\$System Class Reference	1073
10.159.1Detailed Description	1074
10.159.2Constructor & Destructor Documentation	1075
10.159.2.1~System()	1075
10.159.2.2System()	1075
10.159.3Member Function Documentation	1075
10.159.3.1GetCameras()	1075
10.159.3.2GetInstance()	1076
10.159.3.3GetInterfaces()	1076
10.159.3.4GetLibraryVersion()	1077
10.159.3.5GetLoggingEventPriorityLevel()	1077
10.159.3.6GetTLNodeMap()	1077
10.159.3.7IsInUse()	1078
10.159.3.8RegisterEvent()	1078
10.159.3.9RegisterInterfaceEvent()	1078
10.159.3.10RegisterLoggingEvent()	1079
10.159.3.11ReleaseInstance()	1079
10.159.3.12SendActionCommand()	1079
10.159.3.13SetLoggingEventPriorityLevel()	1080
10.159.3.14UnregisterAllLoggingEvent()	1081

10.159.3.15UnregisterEvent()	1081
10.159.3.16UnregisterInterfaceEvent()	1081
10.159.3.17UnregisterLoggingEvent()	1081
10.159.3.18updateCameras()	1082
10.159.3.19updateInterfaceList()	1082
10.16\$ystemEvent Class Reference	1083
10.160.1Detailed Description	1084
10.160.2Constructor & Destructor Documentation	1084
10.160.2.1SystemEvent()	1084
10.160.2.2~SystemEvent()	1084
10.160.3Member Function Documentation	1084
10.160.3.1OnInterfaceArrival()	1084
10.160.3.2OnInterfaceRemoval()	1085
10.160.3.3operator=(*)	1085
10.16\$ystemPtr Class Reference	1085
10.161.1Detailed Description	1086
10.161.2Constructor & Destructor Documentation	1086
10.161.2.1SystemPtr() [1/4]	1086
10.161.2.2SystemPtr() [2/4]	1087
10.161.2.3SystemPtr() [3/4]	1087
10.161.2.4SystemPtr() [4/4]	1087
10.161.2.5~SystemPtr()	1087
10.16TIFFOption Struct Reference	1087
10.162.1Detailed Description	1088
10.162.2Member Enumeration Documentation	1088
10.162.2.1CompressionMethod	1088
10.162.3Constructor & Destructor Documentation	1088
10.162.3.1TIFFOption()	1088
10.162.4Member Data Documentation	1089
10.162.4.1compression	1089

10.162.4.2reserved	1089
10.163TransportLayerDevice Class Reference	1089
10.163.1Detailed Description	1091
10.163.2Constructor & Destructor Documentation	1091
10.163.2.1TransportLayerDevice() [1/2]	1091
10.163.2.2~TransportLayerDevice()	1092
10.163.2.3TransportLayerDevice() [2/2]	1092
10.163.3Friends And Related Function Documentation	1092
10.163.3.1CameraBase	1092
10.163.3.2CameralInternal	1092
10.163.3.3ICameraBase	1092
10.163.4Member Data Documentation	1092
10.163.4.1DeviceAccessStatus	1092
10.163.4.2DeviceCurrentSpeed	1093
10.163.4.3DeviceDisplayName	1093
10.163.4.4DeviceDriverVersion	1093
10.163.4.5DeviceEndianessMechanism	1093
10.163.4.6DeviceID	1093
10.163.4.7DeviceInstanceId	1093
10.163.4.8DeviceIsUpdater	1094
10.163.4.9DeviceLinkSpeed	1094
10.163.4.10DeviceLocation	1094
10.163.4.11DeviceModelName	1094
10.163.4.12DeviceMulticastMonitorMode	1094
10.163.4.13DeviceSerialNumber	1094
10.163.4.14DeviceType	1095
10.163.4.15DeviceU3VProtocol	1095
10.163.4.16DeviceUserID	1095
10.163.4.17DeviceVendorName	1095
10.163.4.18DeviceVersion	1095

10.163.4.19GenICamXMLLocation	1095
10.163.4.20GenICamXMLPath	1096
10.163.4.21evCCP	1096
10.163.4.22evDeviceDiscoverMaximumPacketSize	1096
10.163.4.23evDeviceForceGateway	1096
10.163.4.24evDeviceForceIP	1096
10.163.4.25evDeviceForceIPAddress	1096
10.163.4.26evDeviceForceIPEx	1097
10.163.4.27evDeviceForceSubnetMask	1097
10.163.4.28evDeviceGateway	1097
10.163.4.29evDeviceIPAddress	1097
10.163.4.30evDeviceIsWrongSubnet	1097
10.163.4.31evDeviceMACAddress	1097
10.163.4.32evDeviceMaximumPacketSize	1098
10.163.4.33evDeviceMaximumRetryCount	1098
10.163.4.34evDeviceModelsBigEndian	1098
10.163.4.35evDevicePort	1098
10.163.4.36evDeviceReadAndWriteTimeout	1098
10.163.4.37evDeviceSubnetMask	1098
10.163.4.38evVersionMajor	1099
10.163.4.39evVersionMinor	1099
10.163.4.40GUIMLLocation	1099
10.163.4.41GUIMLPath	1099
10.164TransportLayerInterface Class Reference	1099
10.164.1Detailed Description	1101
10.164.2Constructor & Destructor Documentation	1102
10.164.2.1TransportLayerInterface() [1/2]	1102
10.164.2.2~TransportLayerInterface()	1102
10.164.2.3TransportLayerInterface() [2/2]	1102
10.164.3Friends And Related Function Documentation	1102

10.164.3.1Interface	1102
10.164.3.2Interface	1102
10.164.3.3InterfaceInternal	1102
10.164.4Member Data Documentation	1102
10.164.4.1ActionCommand	1103
10.164.4.2AutoForceIP	1103
10.164.4.3DeviceAccessStatus	1103
10.164.4.4DeviceCount	1103
10.164.4.5DeviceID	1103
10.164.4.6DeviceModelName	1103
10.164.4.7DeviceSelector	1104
10.164.4.8DeviceUnlock	1104
10.164.4.9DeviceUpdateList	1104
10.164.4.10DeviceVendorName	1104
10.164.4.11FilterDriverStatus	1104
10.164.4.12evActionDeviceKey	1104
10.164.4.13evActionGroupKey	1105
10.164.4.14evActionGroupMask	1105
10.164.4.15evActionTime	1105
10.164.4.16evDeviceIPAddress	1105
10.164.4.17evDeviceMACAddress	1105
10.164.4.18evDeviceSubnetMask	1105
10.164.4.19evInterfaceGateway	1106
10.164.4.20evInterfaceIPAddress	1106
10.164.4.21evInterfaceMACAddress	1106
10.164.4.22evInterfaceMTU	1106
10.164.4.23evInterfaceReceiveLinkSpeed	1106
10.164.4.24evInterfaceSubnetMask	1106
10.164.4.25evInterfaceTransmitLinkSpeed	1107
10.164.4.26hostAdapterDriverVersion	1107

10.164.4.27hostAdapterName	1107
10.164.4.28hostAdapterVendor	1107
10.164.4.29compatibleDeviceCount	1107
10.164.4.30compatibleDeviceID	1107
10.164.4.31compatibleDeviceModelName	1108
10.164.4.32compatibleDeviceSelector	1108
10.164.4.33compatibleDeviceVendorName	1108
10.164.4.34compatibleGevDeviceIPAddress	1108
10.164.4.35compatibleGevDeviceMACAddress	1108
10.164.4.36compatibleGevDeviceSubnetMask	1108
10.164.4.37InterfaceDisplayName	1109
10.164.4.38InterfaceID	1109
10.164.4.39InterfaceType	1109
10.164.4.40OEStatus	1109
10.165TransportLayerStream Class Reference	1109
10.165.Detailed Description	1111
10.165.2Constructor & Destructor Documentation	1111
10.165.2.1TransportLayerStream() [1/2]	1111
10.165.2.2~TransportLayerStream()	1111
10.165.2.3TransportLayerStream() [2/2]	1111
10.165.Friends And Related Function Documentation	1111
10.165.3.1CameraBase	1111
10.165.3.2CameralInternal	1112
10.165.3.3CameraBase	1112
10.165.4Member Data Documentation	1112
10.165.4.1GevFailedPacketCount	1112
10.165.4.2GevMaximumNumberResendBuffers	1112
10.165.4.3GevMaximumNumberResendRequests	1112
10.165.4.4GevPacketResendMode	1112
10.165.4.5GevPacketResendTimeout	1113

10.165.4.6GevResendPacketCount	1113
10.165.4.7GevResendRequestCount	1113
10.165.4.8GevTotalPacketCount	1113
10.165.4.9StreamBlockTransferSize	1113
10.165.4.10StreamBufferCountManual	1113
10.165.4.11StreamBufferCountMax	1114
10.165.4.12StreamBufferCountMode	1114
10.165.4.13StreamBufferCountResult	1114
10.165.4.14StreamBufferHandlingMode	1114
10.165.4.15StreamBufferUnderrunCount	1114
10.165.4.16StreamCRCCheckEnable	1114
10.165.4.17StreamDefaultBufferCount	1115
10.165.4.18StreamDefaultBufferCountMax	1115
10.165.4.19StreamDefaultBufferCountMode	1115
10.165.4.20StreamFailedBufferCount	1115
10.165.4.21StreamID	1115
10.165.4.22StreamTotalBufferCount	1115
10.165.4.23StreamType	1116
10.166TransportLayerSystem Class Reference	1116
10.166.1Detailed Description	1116
10.166.2Constructor & Destructor Documentation	1116
10.166.2.1TransportLayerSystem() [1/2]	1117
10.166.2.2~TransportLayerSystem()	1117
10.166.2.3TransportLayerSystem() [2/2]	1117
10.166.3Friends And Related Function Documentation	1117
10.166.3.1ISystem	1117
10.166.3.2System	1117
10.166.3.3SystemPtrInternal	1117
10.166.4Member Data Documentation	1117
10.166.4.1AutoForceIP	1118

10.166.4.2 E numerateGEVInterfaces	1118
10.167 V _CHUNK_TRAILER Struct Reference	1118
10.167.1 D etailed Description	1118
10.167.2 M ember Data Documentation	1118
10.167.2.1 C hunkID	1118
10.167.2.2 C hunkLength	1119
10.168 V _COMMAND_HEADER Struct Reference	1119
10.168.1 D etailed Description	1119
10.168.2 M ember Data Documentation	1119
10.168.2.1 C ommandId	1119
10.168.2.2 F lags	1119
10.168.2.3 L ength	1119
10.168.2.4 P refix	1120
10.168.2.5 R eqId	1120
10.169 V _EVENT_DATA Struct Reference	1120
10.169.1 D etailed Description	1120
10.169.2 M ember Data Documentation	1120
10.169.2.1 E ventId	1120
10.169.2.2 R eserved	1120
10.169.2.3 T imestamp	1121
10.170 V _EVENT_MESSAGE Struct Reference	1121
10.170.1 D etailed Description	1121
10.170.2 M ember Data Documentation	1121
10.170.2.1 C ommandHeader	1121
10.170.2.2 E ventData	1122
10.171 V alueNode Class Reference	1122
10.171.1 D etailed Description	1123
10.171.2 C onstructor & Destructor Documentation	1123
10.171.2.1 V alueNode() [1/2]	1123
10.171.2.2 V alueNode() [2/2]	1123
10.171.2.3~ValueNode()	1124
10.171.3 M ember Function Documentation	1124
10.171.3.1 F romString()	1124
10.171.3.2 G etNode()	1124
10.171.3.3 I sValueCacheValid()	1124
10.171.3.4 S etReference()	1124
10.171.3.5 T oString()	1125
10.172 V ersion_t Struct Reference	1125
10.172.1 D etailed Description	1125
10.172.2 M ember Data Documentation	1125
10.172.2.1 M ajor	1125
10.172.2.2 M inor	1126
10.172.2.3 S ubMinor	1126

11 File Documentation	1127
11.1 doc/Doxygen/spindocs/Licensing.dox File Reference	1127
11.2 doc/Doxygen/spindocs/MainPage.dox File Reference	1127
11.3 include/AdapterConfig.h File Reference	1127
11.3.1 Macro Definition Documentation	1128
11.3.1.1 ADAPTERCONFIG_API	1129
11.4 include/ArrivalEvent.h File Reference	1129
11.5 include/AVIRecorder.h File Reference	1131
11.6 include/BasePtr.h File Reference	1131
11.7 include/Camera.h File Reference	1133
11.8 include/CameraBase.h File Reference	1135
11.9 include/CameraDefs.h File Reference	1137
11.10 include/CameraList.h File Reference	1170
11.11 include/CameraPtr.h File Reference	1172
11.12 include/ChunkData.h File Reference	1174
11.13 include/ChunkDataInference.h File Reference	1176
11.14 include/DeviceEvent.h File Reference	1178
11.15 include/Event.h File Reference	1180
11.16 include/Exception.h File Reference	1181
11.17 include/Image.h File Reference	1183
11.18 include/ImageEvent.h File Reference	1185
11.19 include/ImagePtr.h File Reference	1186
11.20 include/ImageStatistics.h File Reference	1188
11.21 include/ImageUtility.h File Reference	1190
11.22 include/ImageUtilityHeatmap.h File Reference	1190
11.23 include/ImageUtilityPolarization.h File Reference	1191
11.24 include/Interface.h File Reference	1191
11.25 include/Interface/IArrivalEvent.h File Reference	1193
11.26 include/Interface/ICameraBase.h File Reference	1195
11.27 include/Interface/ICameraList.h File Reference	1197

11.28include/Interface/IChunkData.h File Reference	1199
11.29include/Interface/IDeviceEvent.h File Reference	1201
11.30include/Interface/ILImage.h File Reference	1203
11.31include/Interface/ILImageEvent.h File Reference	1205
11.32include/Interface/ILImageStatistics.h File Reference	1206
11.33include/Interface/ILInterface.h File Reference	1208
11.34include/Interface/ILInterfaceArrivalEvent.h File Reference	1210
11.35include/Interface/ILInterfaceEvent.h File Reference	1211
11.36include/Interface/ILInterfaceList.h File Reference	1213
11.37include/Interface/ILInterfaceRemovalEvent.h File Reference	1214
11.38include/Interface/ILoggingEvent.h File Reference	1216
11.39include/Interface/IRemovalEvent.h File Reference	1217
11.40include/Interface/IStream.h File Reference	1219
11.41include/Interface/ISystem.h File Reference	1219
11.42include/Interface/ISystemEvent.h File Reference	1221
11.43include/InterfaceArrivalEvent.h File Reference	1222
11.44include/InterfaceEvent.h File Reference	1223
11.45include/InterfaceList.h File Reference	1225
11.46include/InterfacePtr.h File Reference	1226
11.47include/InterfaceRemovalEvent.h File Reference	1228
11.48include/LoggingEvent.h File Reference	1229
11.49include/LoggingEventData.h File Reference	1231
11.50include/LoggingEventDataPtr.h File Reference	1233
11.51include/RemovalEvent.h File Reference	1235
11.52include/SpinGenApi/Autovector.h File Reference	1237
11.53include/SpinGenApi/Base.h File Reference	1238
11.54include/SpinGenApi/BooleanNode.h File Reference	1239
11.55include/SpinGenApi/CategoryNode.h File Reference	1241
11.56include/SpinGenApi/ChunkAdapter.h File Reference	1243
11.57include/SpinGenApi/ChunkAdapterDcam.h File Reference	1245

11.58include/SpinGenApi/ChunkAdapterGeneric.h File Reference	1247
11.59include/SpinGenApi/ChunkAdapterGEV.h File Reference	1249
11.60include/SpinGenApi/ChunkAdapterU3V.h File Reference	1251
11.61include/SpinGenApi/ChunkPort.h File Reference	1253
11.62include/SpinGenApi/CommandNode.h File Reference	1255
11.63include/SpinGenApi/Compatibility.h File Reference	1258
11.63.1 Macro Definition Documentation	1259
11.63.1.1 FMT_I64	1259
11.64include/SpinGenApi/Container.h File Reference	1259
11.65include/SpinGenApi/Counter.h File Reference	1259
11.66include/SpinGenApi/EnumClasses.h File Reference	1260
11.67include/SpinGenApi/EnumEntryNode.h File Reference	1262
11.68include/SpinGenApi/EnumNode.h File Reference	1264
11.69include/SpinGenApi/EnumNodeT.h File Reference	1266
11.70include/SpinGenApi/EventAdapter.h File Reference	1268
11.71include/SpinGenApi/EventAdapter1394.h File Reference	1270
11.72include/SpinGenApi/EventAdapterGeneric.h File Reference	1272
11.73include/SpinGenApi/EventAdapterGEV.h File Reference	1274
11.74include/SpinGenApi/EventAdapterU3V.h File Reference	1276
11.75include/SpinGenApi/EventPort.h File Reference	1278
11.76include/SpinGenApi/Filestream.h File Reference	1280
11.77include/SpinGenApi/FloatNode.h File Reference	1282
11.78include/SpinGenApi/FloatRegNode.h File Reference	1284
11.79include/SpinGenApi/GCBase.h File Reference	1286
11.80include/SpinGenApi/GCString.h File Reference	1287
11.80.1 Macro Definition Documentation	1288
11.80.1.1 GCSTRING_NPOS	1288
11.80.2 Function Documentation	1288
11.80.2.1 operator<<()	1289
11.80.2.2 operator>>()	1289

11.81 <code>include/SpinGenApi/GCStringVector.h</code> File Reference	1289
11.82 <code>include/SpinGenApi/GCSynch.h</code> File Reference	1290
11.83 <code>include/SpinGenApi/GCTypes.h</code> File Reference	1291
11.83.1 Macro Definition Documentation	1292
11.83.1.1 <code>__STDC_CONSTANT_MACROS</code>	1292
11.83.1.2 <code>__STDC_LIMIT_MACROS</code>	1292
11.83.1.3 <code>GC_INT32_MAX</code>	1292
11.83.1.4 <code>GC_INT32_MIN</code>	1292
11.83.1.5 <code>GC_INT64_MAX</code>	1292
11.83.1.6 <code>GC_INT64_MIN</code>	1293
11.83.1.7 <code>GC_INT8_MAX</code>	1293
11.83.1.8 <code>GC_INT8_MIN</code>	1293
11.83.1.9 <code>GC_UINT32_MAX</code>	1293
11.83.1.10 <code>GC_UINT64_MAX</code>	1293
11.83.1.11 <code>GC_UINT8_MAX</code>	1293
11.84 <code>include/SpinGenApi/GCUtilities.h</code> File Reference	1294
11.84.1 Macro Definition Documentation	1297
11.84.1.1 <code>__ERR_</code>	1297
11.84.1.2 <code>__LINE_STR_</code>	1297
11.84.1.3 <code>__LOCATION_</code>	1297
11.84.1.4 <code>__OUTPUT_FORMATER_</code>	1297
11.84.1.5 <code>__TODO_</code>	1297
11.84.1.6 <code>__WARN_</code>	1297
11.84.1.7 <code>_TO_STRING</code>	1298
11.84.1.8 <code>EXPAND_TO_STRINGISE</code>	1298
11.84.1.9 <code>GC_COUNTOF</code>	1298
11.84.1.10 <code>GENICAM_DEPRECATED</code>	1298
11.84.1.11 <code>GENICAM_UNUSED</code>	1298
11.84.1.12 <code>USE_TEMP_CACHE_FILE [1/2]</code>	1298
11.84.1.13 <code>USE_TEMP_CACHE_FILE [2/2]</code>	1298

11.85#include/SpinGenApi/IBoolean.h File Reference	1299
11.86#include/SpinGenApi/ICategory.h File Reference	1301
11.87#include/SpinGenApi/IChunkPort.h File Reference	1303
11.88#include/SpinGenApi/ICommand.h File Reference	1305
11.89#include/SpinGenApi/IDestroy.h File Reference	1307
11.90#include/SpinGenApi/IDeviceInfo.h File Reference	1308
11.91#include/SpinGenApi/IEnumEntry.h File Reference	1310
11.92#include/SpinGenApi/IEnumeration.h File Reference	1312
11.93#include/SpinGenApi/IEnumerationT.h File Reference	1313
11.94#include/SpinGenApi/IFloat.h File Reference	1315
11.95#include/SpinGenApi/IInteger.h File Reference	1317
11.96#include/SpinGenApi/INode.h File Reference	1319
11.97#include/SpinGenApi/INodeMap.h File Reference	1322
11.98#include/SpinGenApi/INodeMapDyn.h File Reference	1323
11.99#include/SpinGenApi/IntegerNode.h File Reference	1325
11.100#include/SpinGenApi/IntRegNode.h File Reference	1327
11.101#include/SpinGenApi/IPort.h File Reference	1329
11.102#include/SpinGenApi/IPortConstruct.h File Reference	1330
11.103#include/SpinGenApi/IPortRecorder.h File Reference	1332
11.104#include/SpinGenApi/IRegister.h File Reference	1334
11.105#include/SpinGenApi/ISelector.h File Reference	1335
11.106#include/SpinGenApi/ISelectorDigit.h File Reference	1336
11.107#include/SpinGenApi/IString.h File Reference	1338
11.108#include/SpinGenApi/IValue.h File Reference	1340
11.109#include/SpinGenApi/Node.h File Reference	1341
11.110#include/SpinGenApi/NodeCallback.h File Reference	1343
11.111#include/SpinGenApi/NodeCallbackImpl.h File Reference	1345
11.112#include/SpinGenApi/NodeMap.h File Reference	1346
11.113#include/SpinGenApi/NodeMapFactory.h File Reference	1348
11.114#include/SpinGenApi/NodeMapRef.h File Reference	1349

11.115	include/SpinGenApi/Persistence.h File Reference	1350
11.116	include/SpinGenApi/Pointer.h File Reference	1352
11.117	include/SpinGenApi/PortImpl.h File Reference	1355
11.118	include/SpinGenApi/PortNode.h File Reference	1356
11.119	include/SpinGenApi/PortRecorder.h File Reference	1358
11.120	include/SpinGenApi/PortReplay.h File Reference	1359
11.121	include/SpinGenApi/PortWriteList.h File Reference	1360
11.122	include/SpinGenApi/Reference.h File Reference	1362
11.123	include/SpinGenApi/RegisterNode.h File Reference	1363
11.124	include/SpinGenApi/RegisterPortImpl.h File Reference	1365
11.125	include/SpinGenApi/SelectorSet.h File Reference	1365
11.126	include/SpinGenApi/SpinnakerGenApi.h File Reference	1366
11.127	include/SpinGenApi/SpinTestCamera.h File Reference	1368
11.128	include/SpinGenApi/StringNode.h File Reference	1368
11.129	include/SpinGenApi/StringRegNode.h File Reference	1370
11.130	include/SpinGenApi/StructPort.h File Reference	1372
11.131	include/SpinGenApi/Synch.h File Reference	1372
11.132	include/SpinGenApi/Types.h File Reference	1373
11.132.1	Macro Definition Documentation	1376
11.132.1.1	interface	1376
11.132.8	include/SpinGenApi/ValueNode.h File Reference	1377
11.134	include/Spinnaker.h File Reference	1379
11.135	include/SpinnakerDefs.h File Reference	1380
11.136	include/SpinnakerPlatform.h File Reference	1384
11.137	include/SpinUpdate.h File Reference	1384
11.137.1	Macro Definition Documentation	1385
11.137.1.1	SPINUPDATE_API	1385
11.137.2	Function Documentation	1385
11.137.2.1	GetErrorMessage()	1385
11.137.2.2	SetMessageCallback()	1385

11.137.2.3SetProgressCallback()	1385
11.137.2.4UpdateFirmware()	1386
11.137.2.5UpdateFirmwareConsole()	1386
11.137.2.6UpdateFirmwareGUI()	1386
11.137.3Variable Documentation	1386
11.137.3.1UpdatorMessageCallback	1386
11.137.3.2UpdatorProgressCallback	1387
11.138#include/SpinVideo.h File Reference	1387
11.139#include/SpinVideoDefs.h File Reference	1387
11.140#include/System.h File Reference	1388
11.140.1Macro Definition Documentation	1389
11.140.1.1FLIR_SPINNAKER_VERSION_BUILD	1390
11.140.1.2FLIR_SPINNAKER_VERSION_MAJOR	1390
11.140.1.3FLIR_SPINNAKER_VERSION_MINOR	1390
11.140.1.4FLIR_SPINNAKER_VERSION_TYPE	1390
11.140.2Include/SystemEvent.h File Reference	1390
11.140.3Include/SystemPtr.h File Reference	1391
11.141#include/TransportLayerDefs.h File Reference	1393
11.142#include/TransportLayerDevice.h File Reference	1395
11.143#include/TransportLayerInterface.h File Reference	1397
11.144#include/TransportLayerStream.h File Reference	1399
11.145#include/TransportLayerSystem.h File Reference	1401
Index	1403

Chapter 1

Introduction

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

Chapter 2

Software Licensing Information

Table 2.1 License table

Component	License
Spinnaker	Copyright (c) 2001-2019 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
AdapterList	The Code Project Open License (CPOL)
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 (CPOL) 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 3

Module Index

3.1 Modules

Here is a list of all modules:

Spinnaker Event Classes	28
ArrivalEvent Class	30
DeviceEvent Class	166
Event Class	167
ImageEvent Class	170
InterfaceArrivalEvent Class	177
InterfaceEvent Class	178
InterfaceRemovalEvent Class	181
LoggingEvent Class	182
Logging Event Class	183
LoggingEventDataPtr Class	184
RemovalEvent Class	185
SystemEvent Class	206
Spinnaker Classes	31
AVI Recorder Class	35
BasePtr Class	38
Camera Class	39
Camera Base Class	40
CameraDefs Class	41
Camera List Class	156
CameraPtr Class	157
ChunkData Class	159
Chunk Data Inference Class	160
Exception Class	168
Image Class	169
ImagePtr Class	171
ImageStatistics Class	172
Image Utility Class	173
Image Utility Heatmap Class	174
Image Utility Polarization Class	175
Interface Class	176
InterfaceList Class	179
InterfacePtr Class	180
Spinnaker Video Class	203
System Class	205

SystemPtr Class	207
Camera Base Interface Class	220
IChunkData Class	221
IIImage Class	222
IIImageStatistics Class	223
IIInterface Class	224
IIInterfaceList Class	225
ISystem Class	226
Spinnaker Headers	186
Spinnaker.h	188
Spinnaker Definitions	189
Spinnaker Platform	202
Spinnaker Video Definitions	204
Spinnaker QuickSpin Classes	208
TransportLayerDefs Class	209
TransportLayerDevice Class	216
TransportLayerInterface Class	217
TransportLayerStream Class	218
TransportLayerSystem Class	219
Spinnaker GenApi Classes	227
AutoVector Class	238
BooleanNode Class	242
CategoryNode Class	243
ChunkAdapter Class	244
ChunkAdapterDcam Class	245
ChunkAdapterGeneric Class	246
ChunkAdapterGEV Class	247
ChunkPort Class	248
CommandNode Class	249
Container Class	250
Counter Class	251
EnumClasses Class	252
EnumEntryNode Class	254
EnumNode Class	255
EnumNodeT Class	256
EventAdapter Class	257
EventAdapter1394 Class	258
EventAdapterGeneric Class	259
EventAdapterGEV Class	260
EventAdapterU3V Class	261
EventPort Class	262
Filestream Class	263
FloatNode Class	264
FloatRegNode Class	265
GCString Class	266
GCSynch Class	267
GCTypes Class	268
IntegerNode Class	319
IntRegNode Class	320
IString Class	332
IValue Class	333
Node Class	335
NodeCallback Class	336
NodeMap Class	339
NodeMapFactory Class	340
NodeMapRef Class	342
Persistence Class	343

Pointer Class	344
PortImpl Class	350
PortNode Class	351
PortRecorder Class	352
PortReplay Class	353
PortWriteList Class	354
RegisterNode Class	356
RegisterPortImpl Class	357
SelectorSet Class	358
SpinTestCamera Class	359
StringNode Class	360
StringRegNode Class	361
StructPort Class	362
Synch Class	363
ValueNode Class	375
ChunkAdapterU3V Class	376
IPortRecorder Interface	324
Spinnaker GenApi Interfaces	239
IBase Interface	241
IBoolean Interface	276
ICategory Interfaces	278
IChunkPort Interface	279
ICommand Interface	281
IDestroy Interface	283
IDeviceInfo Interface	284
IEnumEntry Interface	287
IEnumerator Interface	289
IEnumeratorT Interface	292
IFloat Interface	294
IInteger Interface	298
INode Interface	300
INodeMap Interface	311
INodeMapDyn Interface	314
IPort Interface	321
IPortConstruct Interface	323
IPortRecorder Interface	324
IRegister Interfaces	326
ISelector Interface	328
ISelectorDigit Interface	329
Reference Interfaces	355
Spinnaker GenApi Utilities	269
GCUtilities Utility	270
Spinnaker GenApi Enums	364
Types Enums	365

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

AdapterConfig	377
Spinnaker	382
Spinnaker::GenApi	421
Spinnaker::GenICam	438
Spinnaker::Video	440

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

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

ActionCommandResult	441
AdapterInfo	443
AttachStatistics_t	449
AutoLock	450
AutoLock	451
AVIOption	451
BasePtr< T, B >	452
BasePtr< Camera, ICameraBase >	452
CameraPtr	634
BasePtr< IImage >	452
ImagePtr	927
BasePtr< IInterface >	452
InterfacePtr	977
BasePtr< ISystem >	452
SystemPtr	1085
BasePtr< LoggingEventData >	452
LoggingEventDataPtr	1008
basic_istream	
IDevFileStreamBase< CharType, Traits >	855
basic_ostream	
ODevFileStreamBase< CharType, Traits >	1034
basic_streampbuf	
IDevFileStreamBuf< CharType, Traits >	857
ODevFileStreamBuf< CharType, Traits >	1036
BMPOption	457
CChunkAdapter	637
CChunkAdapterDcam	641
CChunkAdapterGeneric	643
CChunkAdapterGEV	646
CChunkAdapterU3V	648
CDataStruct	
CTestPortStruct< CDataStruct >	747
CEventAdapter	660

CEventAdapter1394	662
CEventAdapterGeneric	664
CEventAdapterGEV	667
CEventAdapterU3V	669
CGeneric_XMLLoaderParams	681
CNodeMapRefT< GenApi::CGeneric_XMLLoaderParams >	719
CNodeMapRef	716
CGlobalLock	682
CGlobalLockUnlocker	684
CLock	698
CLockEx	703
CLock	700
CLockEx	704
CNodeCallback	705
Function_NodeCallback< Function >	803
Member_NodeCallback< Client, Member >	1010
CNodeMapFactory	708
Counter	727
CPointer< T, B >	729
CPointer< IFloat, IBase >	729
CFloatPtr	679
CPointer< INode, IBase >	729
DCAM_CHECKSUM	751
DCAM_CHUNK_TRAILER	751
double_vector_t	755
EAccessModeClass	758
ECachingModeClass	759
EDisplayNotationClass	760
EEndianessClass	761
EGenApiSchemaVersionClass	762
EInputDirectionClass	763
ENameSpaceClass	764
ERepresentationClass	773
ESignClass	774
ESlopeClass	775
EStandardNameSpaceClass	776
Event	778
IArrivalEvent	826
ArrivalEvent	447
IInterfaceEvent	886
InterfaceEvent	971
IDeviceEvent	859
DeviceEvent	752
IImageEvent	874
ImageEvent	924
IInterfaceArrivalEvent	884
InterfaceArrivalEvent	969
ISystemEvent	994
SystemEvent	1083
IInterfaceRemovalEvent	891
InterfaceRemovalEvent	979
ISystemEvent	994
ILoggingEvent	893
LoggingEvent	1003
IRemovalEvent	986
IInterfaceEvent	886

RemovalEvent	1057
EVISIBILITYCLASS	781
exception	
AdapterConfigException	442
Exception	783
EYESNOCCLASS	788
FILEPROTOCOLADAPTER	789
GCSTRING	805
GVCP_CHUNK_TRAILER	814
GVCP_EVENT_ITEM	815
GVCP_EVENT_ITEM_BASIC	816
GVCP_EVENT_ITEM_EXTENDED_ID	817
GVCP_EVENT_REQUEST	818
GVCP_EVENT_REQUEST_EXTENDED_ID	819
GVCP_EVENTDATA_REQUEST	820
GVCP_EVENTDATA_REQUEST_EXTENDED_ID	821
GVCP_REQUEST_HEADER	822
H264OPTION	823
IBOOLEAN	
BooleanNode	458
ICAMERABASE	828
CameraBase	613
Camera	461
ICAMERALIST	837
CameraList	628
ICATEGORY	
CategoryNode	635
ICUNKDATA	840
ChunkData	686
ICUNKPORT	
PortNode	1041
PortReplay	1049
PortRecorder	1046
ICOMMAND	
CommandNode	724
IDATASTREAM	849
IDeviceInfo	
NodeMap	1024
SpinTestCamera	1061
IEnumEntry	
EnumEntryNode	765
IEnumeration	
EnumNode	768
CEnumerationTRef< EnumT >	655
IEnumerationT	
CEnumerationTRef< EnumT >	655
IFloat	
FloatNode	793
FloatRegNode	800
Image	
Image	862
Image	895
ImageStatistics	
ImageStatistics	876
ImageStatistics	929
IInteger	
IntegerNode	958
IntRegNode	982

IInterface	880
Interface	964
IIInterfaceList	889
InterfaceList	974
ImageUtility	936
ImageUtilityHeatmap	941
ImageUtilityPolarization	945
InferenceBoundingBox	953
InferenceBoundingBoxResult	953
InferenceBoxCircle	954
InferenceBoxRect	954
InferenceBoxRotatedRect	955
INode	
Node	1014
CSelectorSet	744
PortNode	1041
ValueNode	1122
BooleanNode	458
CategoryNode	635
CommandNode	724
EnumEntryNode	765
EnumNode	768
FloatNode	793
IntegerNode	958
RegisterNode	1053
FloatRegNode	800
IntRegNode	982
StringRegNode	1070
StringNode	1066
StringRegNode	1070
INodeMap	
NodeMap	1024
int64_avector_t	955
IPersistScript	
CFeatureBag	676
IpInfo	984
IPortConstruct	
CChunkPort	650
CEventPort	671
CPortImpl	734
CRegisterPortImpl	741
CTestPortStruct< CDataStruct >	747
PortNode	1041
IPortRecorder	
PortNode	1041
PortRecorder	1046
IPortReplay	
CPortImpl	734
PortReplay	1049
IPortWriteList	
CPortWriteList	738
IRegister	
RegisterNode	1053
IString	
StringNode	1066
ISystem	
System	1073

IValue	
ValueNode	1122
JPEGOption	996
JPG2Option	997
LibraryVersion	999
LockableObject< Object >::Lock	1000
LockableObject< Object >	1001
LoggingEventData	1005
MJPGOption	1012
CNodeMapFactory::NodeStatistics_t	1033
PGMOption	1038
PNGOption	1039
PPMOption	1051
SingleChunkData_t	1059
SingleChunkDataStr_t	1060
SpinVideo	1061
TIFFOption	1087
TransportLayerDevice	1089
TransportLayerInterface	1099
TransportLayerStream	1109
TransportLayerSystem	1116
U3V_CHUNK_TRAILER	1118
U3V_COMMAND_HEADER	1119
U3V_EVENT_DATA	1120
U3V_EVENT_MESSAGE	1121
Version_t	1125
TCameraParams	
CNodeMapRefT< TCameraParams >	719

Chapter 6

Class Index

6.1 Class List

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

ActionCommandResult	
Action Command Result	441
AdapterConfigException	
.	442
AdapterInfo	
.	443
ArrivalEvent	
An event handler for capturing the device arrival event	447
AttachStatistics_t	
Delivers information about the attached chunks and nodes	449
AutoLock	
.	450
AutoLock	
.	451
AVIOption	
Options for saving AVI files	451
BasePtr< T, B >	
The base class of the SystemPtr , CameraPtr , InterfacePtr , ImagePtr and LoggingEventDataPtr objects	452
BMPOption	
Options for saving Bitmap image	457
BooleanNode	
Interface for string properties	458
Camera	
The camera object class	461
CameraBase	
The base class for the camera object	613
CameraList	
Used to hold a list of camera objects	628
CameraPtr	
A reference tracked pointer to a camera object	634
CategoryNode	
Interface for string properties	635
CChunkAdapter	
Connects a chunked buffer to a node map	637
CChunkAdapterDcam	
Connects a chunked DCAM buffer to a node map	641
CChunkAdapterGeneric	
.	643
CChunkAdapterGEV	
Connects a chunked DCAM buffer to a node map	646

CChunkAdapterU3V	Connects a chunked U3V buffer to a node map	648
CChunkPort	Port attachable to a chunk in a buffer	650
CEnumerationTRef< EnumT >	Interface for string properties	655
CEventAdapter	Delivers Events to ports	660
CEventAdapter1394	Distribute the events to the node map	662
CEventAdapterGeneric	Connects a generic event to a node map	664
CEventAdapterGEV	Connects a GigE Event to a node map	667
CEventAdapterU3V	Connects a U3V Event to a node map	669
CEventPort	Port attachable to an event	671
CFeatureBag	Bag holding streamable features of a nodetree	676
CFloatPtr	SmartPointer for IFloat interface pointer	679
CGeneric_XMLLoaderParams	Empty base class used by class CNodeMapRef as generic template argument	681
CGlobalLock	Named global lock which can be used over process boundaries	682
CGlobalLockUnlocker	Unlocks the global lock object on destruction	684
ChunkData	The chunk data which contains additional information about an image	686
CLock	A lock class	698
CLock	A lock class	700
CLockEx	This class is for testing purposes only	703
CLockEx	This class is for testing purposes only	704
CNodeCallback	Callback body instance for INode pointers	705
CNodeMapFactory	The node map factory is used for creating node maps from camera description files	708
CNodeMapRef	Smartpointer for NodeMaps with create function	716
CNodeMapRefT< TCameraParams >	Smartpointer template for NodeMaps with create function	719
CommandNode	Interface for string properties	724
Counter	Definition of a simple Counter class	727
CPointer< T, B >	Encapsulates a GenApi pointer dealing with the dynamic_cast automatically	729
CPortImpl	Standard implementation for a port	734
CPortWriteList	Container holding a list of port write commands	738
CRegisterPortImpl	Standard implementation for a port using a register based transport layer	741

CSelectorSet	
The set of selectors selecting a given node	744
CTestPortStruct< CDataStruct >	
Implements a register spaces based on a C++ struct	747
DCAM_CHECKSUM	
DCAM_CHUNK_TRAILER	
DeviceEvent	
A handler to device events	752
double_vector_t	
Vector of doubles with reference counting	755
EAccessModeClass	
Holds conversion methods for the access mode enumeration	758
ECachingModeClass	
Holds conversion methods for the caching mode enumeration	759
EDisplayNotationClass	
Holds conversion methods for the notation type of floats	760
EEndianessClass	
Holds conversion methods for the endianess enumeration	761
EGenApiSchemaVersionClass	
Helper class converting EGenApiSchemaVersion from and to string	762
EInputDirectionClass	
Holds conversion methods for the notation type of floats	763
ENamespaceClass	
Holds conversion methods for the namespace enumeration	764
EnumEntryNode	
Interface for string properties	765
EnumNode	
Interface for string properties	768
ERepresentationClass	
Holds conversion methods for the representation enumeration	773
ESignClass	
Holds conversion methods for the sign enumeration	774
ESlopeClass	
Holds conversion methods for the converter formulas	775
EStandardNameSpaceClass	
Holds conversion methods for the standard namespace enumeration	776
Event	
The base class for all event types	778
EVisibilityClass	
Holds conversion methods for the visibility enumeration	781
Exception	
The Exception object represents an error that is returned from the library	783
EYesNoClass	
Holds conversion methods for the standard namespace enumeration	788
FileProtocolAdapter	
Adapter between the std::iostreambuf and the SFNC Features representing the device file system	789
FloatNode	
Interface for string properties	793
FloatRegNode	
Interface for string properties	800
Function_NodeCallback< Function >	
Container for a function pointer	803
gcstring	
GVCP_CHUNK_TRAILER	
Header of a GVCP request packet	814
GVCP_EVENT_ITEM	
Layout of a GVCP event item (Extended ID flag not set)	815

GVCP_EVENT_ITEM_BASIC	Layout of a GVCP event item (common to all types)	816
GVCP_EVENT_ITEM_EXTENDED_ID	Layout of a GVCP event item (Extended ID flag set)	817
GVCP_EVENT_REQUEST	Layout of a GVCP event request packet (Extended ID flag not set)	818
GVCP_EVENT_REQUEST_EXTENDED_ID	Layout of a GVCP event request packet (Extended ID flag set)	819
GVCP_EVENTDATA_REQUEST	Layout of a GVCP event data request packet (Extended ID flag not set)	820
GVCP_EVENTDATA_REQUEST_EXTENDED_ID	Layout of a GVCP event data request packet (Extended ID flag set)	821
GVCP_REQUEST_HEADER	Header of a GVCP request packet	822
H264Option	Options for saving H264 files	823
IArrivalEvent	826
ICameraBase	The interface file for base class for the camera object	828
ICameraList	Used to hold a list of camera objects	837
IChunkData	The Interface file for ChunkData	840
IDataStream	849
IDevFileStreamBase< CharType, Traits >	855
IDevFileStreamBuf< CharType, Traits >	857
IDeviceEvent	859
IImage	The interface file for Image	862
IImageEvent	874
IImageStatistics	The interface file for image statistics	876
IInterface	The interface file for Interface	880
IInterfaceArrivalEvent	884
IInterfaceEvent	886
IInterfaceList	The interface file for InterfaceList class	889
IInterfaceRemovalEvent	891
ILoggingEvent	893
Image	The image object class	895
ImageEvent	A handler for capturing image arrival events	924
ImagePtr	A reference tracked pointer to an image object	927
ImageStatistics	Represents image statistics for an image	929
ImageUtility	Static helper functions for the image object class	936
ImageUtilityHeatmap	Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16	941
ImageUtilityPolarization	Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8	945
InferenceBoundingBox	Inference Bounding Boxes data structure	953

InferenceBoundingBoxResult	An inference bounding boxes object which holds information about the detected bounding boxes	953
InferenceBoxCircle		954
InferenceBoxRect	Inference Bounding Box Type Data Structures	954
InferenceBoxRotatedRect		955
int64 __ autovector __ t	Vector of integers with reference counting	955
IntegerNode	Interface for string properties	958
Interface	An interface object which holds a list of cameras	964
InterfaceArrivalEvent	An event handler for capturing the interface arrival event	969
InterfaceEvent	A handler to device arrival and removal events on all interfaces	971
InterfaceList	A list of the available interfaces on the system	974
InterfacePtr	A reference tracked pointer to the interface object	977
InterfaceRemovalEvent	An event handler for capturing the interface removal event	979
IntRegNode	Interface for string properties	982
IPlInfo		984
IRemovalEvent		986
ISystem	The interface file for System	988
ISystemEvent		994
JPEGOption	Options for saving JPEG image	996
JPG2Option	Options for saving JPEG2000 image	997
LibraryVersion	Provides easier access to the current version of Spinnaker	999
LockableObject< Object >::Lock	A scopelevel Lock class	1000
LockableObject< Object >	Instance-Lock for an object	1001
LoggingEvent	An event handler for capturing the device logging event	1003
LoggingEventData	The LoggingEventData object	1005
LoggingEventDataPtr	A reference tracked pointer to the LoggingEvent object	1008
Member_NodeCallback< Client, Member >	Container for a member function pointer	1010
MJPGOption	Options for saving MJPG files	1012
Node	Class common to all nodes	1014
NodeMap	Smart pointer template for NodeMaps with create function	1024
CNodeMapFactory::NodeStatistics_t		1033
ODevFileStreamBase< CharType, Traits >		1034
ODevFileStreamBuf< CharType, Traits >		1036
PGMOption	Options for saving PGM images	1038

PNGOption	Options for saving PNG images	1039
PortNode	Interface for value properties	1041
PortRecorder	Interface for recording write commands on a port	1046
PortReplay	Interface for replaying write commands on a port	1049
PPMOption	Options for saving PPM images	1051
RegisterNode	Interface for string properties	1053
RemovalEvent	An event handler for capturing the device removal event	1057
SingleChunkData_t	1059
SingleChunkDataStr_t	1060
SpinTestCamera	1061
SpinVideo	Provides the functionality for the user to record images to an AVI/MP4 file	1061
StringNode	Interface for string properties	1066
StringRegNode	Interface for string properties	1070
System	The system object is used to retrieve the list of interfaces and cameras available	1073
SystemEvent	A handler to interface arrival and removal events on the system	1083
SystemPtr	A reference tracked pointer to a system object	1085
TIFFOption	Options for saving TIFF images	1087
TransportLayerDevice	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1089
TransportLayerInterface	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1099
TransportLayerStream	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1109
TransportLayerSystem	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1116
U3V_CHUNK_TRAILER	Header of a GVCP request packet	1118
U3V_COMMAND_HEADER	U3V/GenCP command header	1119
U3V_EVENT_DATA	U3V/GenCP EVENT_CMD specific command data	1120
U3V_EVENT_MESSAGE	Entire event data message (without the variable-sized data field)	1121
ValueNode	Interface for value properties	1122
Version_t	Version	1125

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

include/AdapterConfig.h	1127
include/ArrivalEvent.h	1129
include/AVIRecorder.h	1131
include/BasePtr.h	1131
include/Camera.h	1133
include/CameraBase.h	1135
include/CameraDefs.h	1137
include/CameraList.h	1170
include/CameraPtr.h	1172
include/ChunkData.h	1174
include/ChunkDataInference.h	1176
include/DeviceEvent.h	1178
include/Event.h	1180
include/Exception.h	1181
include/IImage.h	1183
include/IImageEvent.h	1185
include/IImagePtr.h	1186
include/IImageStatistics.h	1188
include/IImageUtility.h	1190
include/IImageUtilityHeatmap.h	1190
include/IImageUtilityPolarization.h	1191
include/Interface.h	1191
include/InterfaceArrivalEvent.h	1222
include/InterfaceEvent.h	1223
include/InterfaceList.h	1225
include/InterfacePtr.h	1226
include/InterfaceRemovalEvent.h	1228
include/LoggingEvent.h	1229
include/LoggingEventData.h	1231
include/LoggingEventDataPtr.h	1233
include/RemovalEvent.h	1235
include/Spinnaker.h	1379
include/SpinnakerDefs.h	1380
include/SpinnakerPlatform.h	1384
include/SpinUpdate.h	1384

include/SpinVideo.h	1387
include/SpinVideoDefs.h	1387
include/System.h	1388
include/SystemEvent.h	1390
include/SystemPtr.h	1391
include/TransportLayerDefs.h	1393
include/TransportLayerDevice.h	1395
include/TransportLayerInterface.h	1397
include/TransportLayerStream.h	1399
include/TransportLayerSystem.h	1401
include/Interface/IArrivalEvent.h	1193
include/Interface/ICameraBase.h	1195
include/Interface/ICameraList.h	1197
include/Interface/IChunkData.h	1199
include/Interface/IDeviceEvent.h	1201
include/Interface/IIImage.h	1203
include/Interface/IIImageEvent.h	1205
include/Interface/IIImageStatistics.h	1206
include/Interface/IInterface.h	1208
include/Interface/IInterfaceArrivalEvent.h	1210
include/Interface/IInterfaceEvent.h	1211
include/Interface/IInterfaceList.h	1213
include/Interface/IInterfaceRemovalEvent.h	1214
include/Interface/ILoggingEvent.h	1216
include/Interface/IRemovalEvent.h	1217
include/Interface/IStream.h	1219
include/Interface/ISystem.h	1219
include/Interface/ISystemEvent.h	1221
include/SpinGenApi/Autovector.h	1237
include/SpinGenApi/Base.h	1238
include/SpinGenApi/BooleanNode.h	1239
include/SpinGenApi/CategoryNode.h	1241
include/SpinGenApi/ChunkAdapter.h	1243
include/SpinGenApi/ChunkAdapterDcam.h	1245
include/SpinGenApi/ChunkAdapterGeneric.h	1247
include/SpinGenApi/ChunkAdapterGEV.h	1249
include/SpinGenApi/ChunkAdapterU3V.h	1251
include/SpinGenApi/ChunkPort.h	1253
include/SpinGenApi/CommandNode.h	1255
include/SpinGenApi/Compatibility.h	1258
include/SpinGenApi/Container.h	1259
include/SpinGenApi/Counter.h	1259
include/SpinGenApi/EnumClasses.h	1260
include/SpinGenApi/EnumEntryNode.h	1262
include/SpinGenApi/EnumNode.h	1264
include/SpinGenApi/EnumNodeT.h	1266
include/SpinGenApi/EventAdapter.h	1268
include/SpinGenApi/EventAdapter1394.h	1270
include/SpinGenApi/EventAdapterGeneric.h	1272
include/SpinGenApi/EventAdapterGEV.h	1274
include/SpinGenApi/EventAdapterU3V.h	1276
include/SpinGenApi/EventPort.h	1278
include/SpinGenApi/Filestream.h	1280
include/SpinGenApi/FloatNode.h	1282
include/SpinGenApi/FloatRegNode.h	1284
include/SpinGenApi/GCBase.h	1286
include/SpinGenApi/GCString.h	1287
include/SpinGenApi/GCStringVector.h	1289

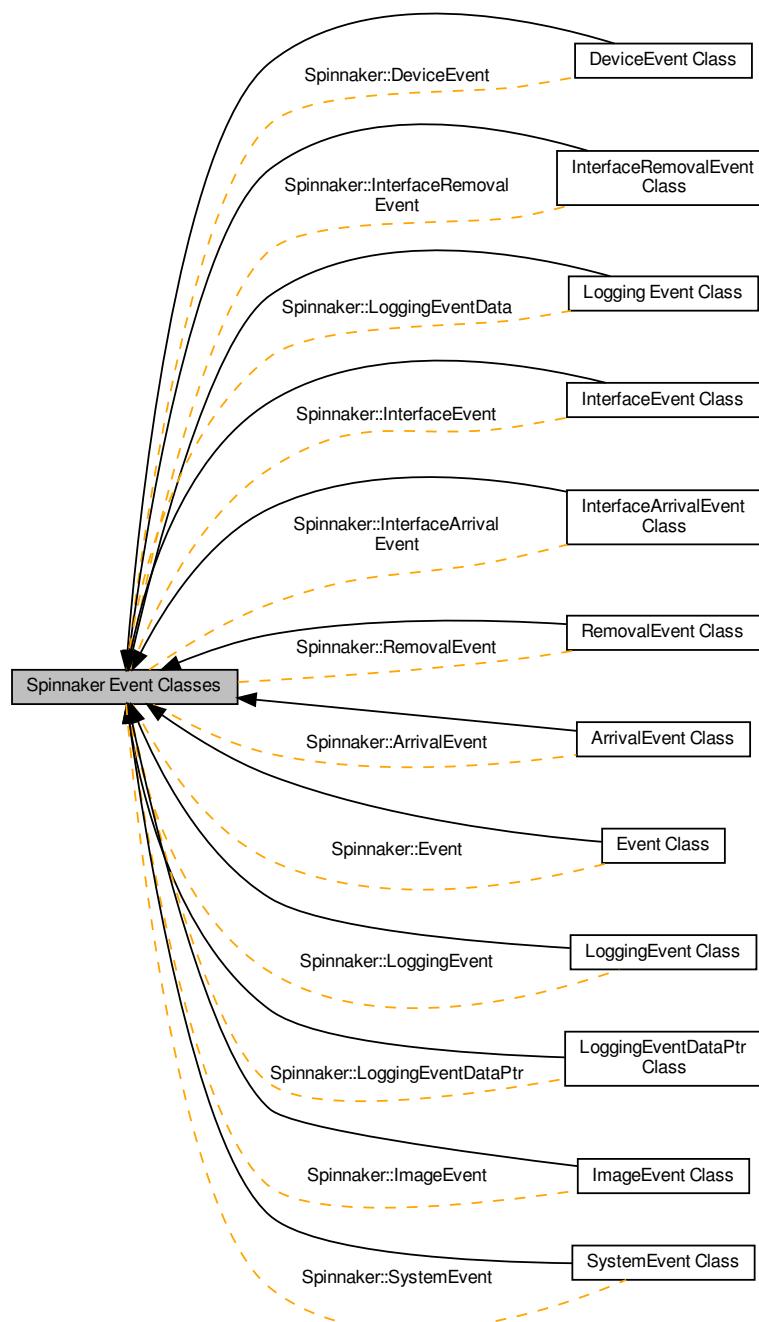
include/SpinGenApi/GCSynch.h	1290
include/SpinGenApi/GCTypes.h	1291
include/SpinGenApi/GCUtilities.h	1294
include/SpinGenApi/IBoolean.h	1299
include/SpinGenApi/ICategory.h	1301
include/SpinGenApi/IChunkPort.h	1303
include/SpinGenApi/ICommand.h	1305
include/SpinGenApi/IDestroy.h	1307
include/SpinGenApi/IDeviceInfo.h	1308
include/SpinGenApi/IEnumEntry.h	1310
include/SpinGenApi/IEnumeration.h	1312
include/SpinGenApi/IEnumerationT.h	1313
include/SpinGenApi/IFloat.h	1315
include/SpinGenApi/IInteger.h	1317
include/SpinGenApi/INode.h	1319
include/SpinGenApi/INodeMap.h	1322
include/SpinGenApi/INodeMapDyn.h	1323
include/SpinGenApi/IntegerNode.h	1325
include/SpinGenApi/IntRegNode.h	1327
include/SpinGenApi/IPort.h	1329
include/SpinGenApi/IPortConstruct.h	1330
include/SpinGenApi/IPortRecorder.h	1332
include/SpinGenApi/IRegister.h	1334
include/SpinGenApi/ISelector.h	1335
include/SpinGenApi/ISelectorDigit.h	1336
include/SpinGenApi/IString.h	1338
include/SpinGenApi/IValue.h	1340
include/SpinGenApi/Node.h	1341
include/SpinGenApi/NodeCallback.h	1343
include/SpinGenApi/NodeCallbackImpl.h	1345
include/SpinGenApi/NodeMap.h	1346
include/SpinGenApi/NodeMapFactory.h	1348
include/SpinGenApi/NodeMapRef.h	1349
include/SpinGenApi/Persistence.h	1350
include/SpinGenApi/Pointer.h	1352
include/SpinGenApi/PortImpl.h	1355
include/SpinGenApi/PortNode.h	1356
include/SpinGenApi/PortRecorder.h	1358
include/SpinGenApi/PortReplay.h	1359
include/SpinGenApi/PortWriteList.h	1360
include/SpinGenApi/Reference.h	1362
include/SpinGenApi/RegisterNode.h	1363
include/SpinGenApi/RegisterPortImpl.h	1365
include/SpinGenApi/SelectorSet.h	1365
include/SpinGenApi/SpinnakerGenApi.h	1366
include/SpinGenApi/SpinTestCamera.h	1368
include/SpinGenApi/StringNode.h	1368
include/SpinGenApi/StringRegNode.h	1370
include/SpinGenApi/StructPort.h	1372
include/SpinGenApi/Synch.h	1372
include/SpinGenApi/Types.h	1373
include/SpinGenApi/ValueNode.h	1377

Chapter 8

Module Documentation

8.1 Spinnaker Event Classes

Collaboration diagram for Spinnaker Event Classes:



Modules

- [ArrivalEvent Class](#)
- [DeviceEvent Class](#)

- [Event Class](#)
- [ImageEvent Class](#)
- [InterfaceArrivalEvent Class](#)
- [InterfaceEvent Class](#)
- [InterfaceRemovalEvent Class](#)
- [LoggingEvent Class](#)
- [Logging Event Class](#)
- [LoggingEventDataPtr Class](#)
- [RemovalEvent Class](#)
- [SystemEvent Class](#)

Classes

- class [ArrivalEvent](#)
An event handler for capturing the device arrival event.
- class [DeviceEvent](#)
A handler to device events.
- class [Event](#)
The base class for all event types.
- class [ImageEvent](#)
A handler for capturing image arrival events.
- class [InterfaceArrivalEvent](#)
An event handler for capturing the interface arrival event.
- class [InterfaceEvent](#)
A handler to device arrival and removal events on all interfaces.
- class [InterfaceRemovalEvent](#)
An event handler for capturing the interface removal event.
- class [LoggingEvent](#)
An event handler for capturing the device logging event.
- class [LoggingEventData](#)
The [LoggingEventData](#) object.
- class [LoggingEventDataPtr](#)
A reference tracked pointer to the [LoggingEvent](#) object.
- class [RemovalEvent](#)
An event handler for capturing the device removal event.
- class [SystemEvent](#)
A handler to interface arrival and removal events on the system.

8.1.1 Detailed Description

8.2 ArrivalEvent Class

Collaboration diagram for ArrivalEvent Class:



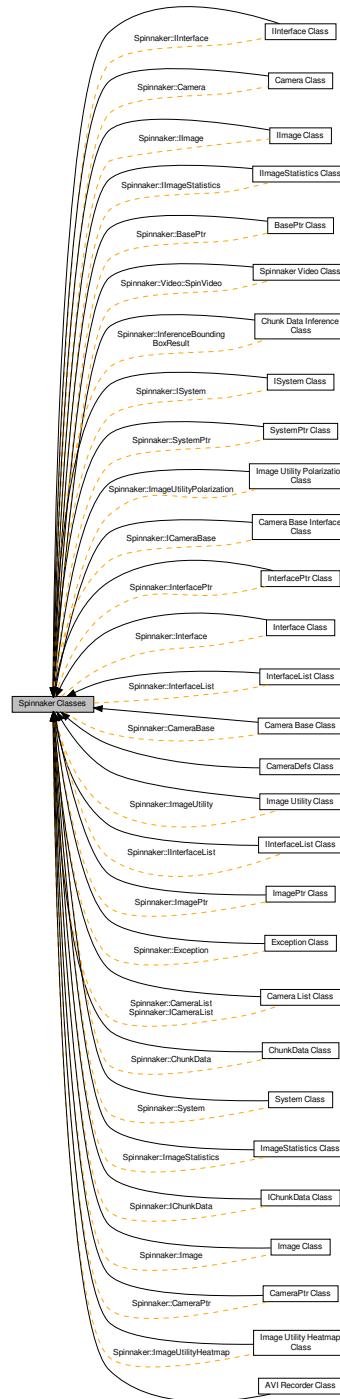
Classes

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

8.2.1 Detailed Description

8.3 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 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 [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](#) {
 [INERENCE_BOX_TYPE_RECTANGLE](#) = 0,
 [INERENCE_BOX_TYPE_CIRCLE](#) = 1,
 [INERENCE_BOX_TYPE_ROTATED_RECTANGLE](#) = 2 }
- Inference Bounding Box Type.*

8.3.1 Detailed Description

8.3.2 Enumeration Type Documentation

8.3.2.1 InferenceBoxType

```
enum InferenceBoxType
```

Inference Bounding Box Type.

Enumerator

INFERENCE_BOX_TYPE_RECTANGLE	
INFERENCE_BOX_TYPE_CIRCLE	
INFERENCE_BOX_TYPE_ROTATED_RECTANGLE	

8.4 AVI Recorder Class

Collaboration diagram for AVI Recorder Class:



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.

8.4.1 Detailed Description

8.4.2 Function Documentation

8.4.2.1 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 plImage)`

8.5 BasePtr Class

Collaboration diagram for BasePtr Class:



Classes

- class [BasePtr< T, B >](#)

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

8.5.1 Detailed Description

8.6 Camera Class

Collaboration diagram for Camera Class:



Classes

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

8.6.1 Detailed Description

8.7 Camera Base Class

Collaboration diagram for Camera Base Class:



Classes

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

8.7.1 Detailed Description

8.8 CameraDefs Class

Collaboration diagram for CameraDefs Class:



Enumerations

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

• 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_DEVICETLTYPENumber }
```

- 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_GEVSUPPORTEOPTIONSELECTOR }
- 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 `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_EXPOSUREACTIVE MODE }

• 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_DEVICESTREAMCHANNELENDIANCESS }
- 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_CXP CONNECTIONTESTMODE }

• enum CxpPoCxpStatusEnums {
    CxpPoCxpStatus_Auto,
    CxpPoCxpStatus_Off,
    CxpPoCxpStatus_Tripped,
    NUM_CXPOCXPSTATUS }

```

8.8.1 Detailed Description

8.8.2 Enumeration Type Documentation

8.8.2.1 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

<code>AcquisitionMode_Continuous</code>	
<code>AcquisitionMode_SingleFrame</code>	
<code>AcquisitionMode_MultiFrame</code>	
<code>NUM_ACQUISITIONMODE</code>	

8.8.2.2 AcquisitionStatusSelectorEnums

enum `AcquisitionStatusSelectorEnums`

< Selects the internal acquisition signal to read using `AcquisitionStatus`.

Enumerator

<code>AcquisitionStatusSelector_AcquisitionTriggerWait</code>	Device is currently waiting for a trigger for the capture of one or many frames.
<code>AcquisitionStatusSelector_AcquisitionActive</code>	Device is currently doing an acquisition of one or many frames.

Enumerator

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_ACQUISITIONSTATUSSELECTOR	

8.8.2.3 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	

8.8.2.4 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	

8.8.2.5 AutoAlgorithmSelectorEnums

```
enum AutoAlgorithmSelectorEnums
```

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

Enumerator

AutoAlgorithmSelector_Awb	Selects the Auto White Balance algorithm.
AutoAlgorithmSelector_Ae	Selects the Auto Exposure algorithm.
NUM_AUTOALGORITHMSELECTOR	

8.8.2.6 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

AutoExposureControlPriority_Gain	
AutoExposureControlPriority_ExposureTime	
NUM_AUTOEXPOSURECONTROLPRIORITY	

8.8.2.7 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

AutoExposureLightingMode_AutoDetect	
AutoExposureLightingMode_Backlight	
AutoExposureLightingMode_Frontlight	
AutoExposureLightingMode_Normal	
NUM_AUTOEXPOSURELIGHTINGMODE	

8.8.2.8 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

AutoExposureMeteringMode_Average	
AutoExposureMeteringMode_Spot	
AutoExposureMeteringMode_Partial	
AutoExposureMeteringMode_CenterWeighted	
AutoExposureMeteringMode_HistogramPeak	
NUM_AUTOEXPOSUREMETERINGMODE	

8.8.2.9 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

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

8.8.2.10 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	

8.8.2.11 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

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

8.8.2.12 BalanceWhiteAutoProfileEnums

enum `BalanceWhiteAutoProfileEnums`

< Selects the profile used by BalanceWhiteAuto.

Enumerator

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

8.8.2.13 BinningHorizontalModeEnums

enum `BinningHorizontalModeEnums`

<

Enumerator

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

8.8.2.14 BinningSelectorEnums

enum `BinningSelectorEnums`

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

Enumerator

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

8.8.2.15 BinningVerticalModeEnums

enum `BinningVerticalModeEnums`

<

Enumerator

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

8.8.2.16 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

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

8.8.2.17 BlackLevelAutoEnums

enum [BlackLevelAutoEnums](#)

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

Enumerator

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

8.8.2.18 BlackLevelSelectorEnums

enum [BlackLevelSelectorEnums](#)

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

Enumerator

<code>BlackLevelSelector_All</code>	
<code>BlackLevelSelector_Analog</code>	
<code>BlackLevelSelector_Digital</code>	
<code>NUM_BLACKLEVELSELECTOR</code>	

8.8.2.19 ChunkBlackLevelSelectorEnums

enum [ChunkBlackLevelSelectorEnums](#)

< Selects which black level to retrieve

Enumerator

<code>ChunkBlackLevelSelector_All</code>	
<code>NUM_CHUNKBLACKLEVELSELECTOR</code>	

8.8.2.20 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	

8.8.2.21 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	

8.8.2.22 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	

8.8.2.23 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	

8.8.2.24 ChunkGainSelectorEnums

```
enum ChunkGainSelectorEnums
```

< Selects which gain to retrieve

Enumerator

ChunkGainSelector_All	
ChunkGainSelector_Red	
ChunkGainSelector_Green	
ChunkGainSelector_Blue	
NUM_CHUNKGAINSELECTOR	

8.8.2.25 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.

Enumerator

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	

8.8.2.26 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	

8.8.2.27 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	

8.8.2.28 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	

8.8.2.29 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	

8.8.2.30 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	

8.8.2.31 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_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE	

8.8.2.32 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	

8.8.2.33 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	

8.8.2.34 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_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.
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	

8.8.2.35 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	

8.8.2.36 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	

8.8.2.37 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	

8.8.2.38 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	

8.8.2.39 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	

8.8.2.40 CITimeSlotsCountEnums

enum `CITimeSlotsCountEnums`

< 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

<code>CITimeSlotsCount_One</code>	One
<code>CITimeSlotsCount_Two</code>	Two
<code>CITimeSlotsCount_Three</code>	Three
<code>NUM_CLTIMESLOTSCOUNT</code>	

8.8.2.41 ColorTransformationSelectorEnums

enum `ColorTransformationSelectorEnums`

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

Enumerator

<code>ColorTransformationSelector_RGBtoRGB</code>	
<code>ColorTransformationSelector_RGBtoYUV</code>	
<code>NUM_COLORTRANSFORMATIONSELECTOR</code>	

8.8.2.42 ColorTransformationValueSelectorEnums

enum `ColorTransformationValueSelectorEnums`

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

Enumerator

<code>ColorTransformationValueSelector_Gain00</code>	
<code>ColorTransformationValueSelector_Gain01</code>	
<code>ColorTransformationValueSelector_Gain02</code>	
<code>ColorTransformationValueSelector_Gain10</code>	
<code>ColorTransformationValueSelector_Gain11</code>	
<code>ColorTransformationValueSelector_Gain12</code>	
<code>ColorTransformationValueSelector_Gain20</code>	
<code>ColorTransformationValueSelector_Gain21</code>	
<code>ColorTransformationValueSelector_Gain22</code>	
<code>ColorTransformationValueSelector_Offset0</code>	
<code>ColorTransformationValueSelector_Offset1</code>	
<code>ColorTransformationValueSelector_Offset2</code>	
<code>NUM_COLORTRANSFORMATIONVALUESELECTOR</code>	

8.8.2.43 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	

8.8.2.44 CounterEventSourceEnums

enum `CounterEventSourceEnums`

< Selects the event that will increment the counter

Enumerator

CounterEventSource_Off	Off
CounterEventSource_MHzTick	MHzTick
CounterEventSource_Line0	Line0
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	

8.8.2.45 CounterResetActivationEnums

enum `CounterResetActivationEnums`

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

Enumerator

<code>CounterResetActivation_LevelLow</code>	
<code>CounterResetActivation_LevelHigh</code>	
<code>CounterResetActivation_FallingEdge</code>	
<code>CounterResetActivation_RisingEdge</code>	
<code>CounterResetActivation_AnyEdge</code>	
<code>NUM_COUNTERRESETACTIVATION</code>	

8.8.2.46 CounterResetSourceEnums

enum `CounterResetSourceEnums`

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

Enumerator

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

8.8.2.47 CounterSelectorEnums

```
enum CounterSelectorEnums
```

< Selects which counter to configure

Enumerator

CounterSelector_Counter0	
CounterSelector_Counter1	
NUM_COUNTERSELECTOR	

8.8.2.48 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	

8.8.2.49 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	

8.8.2.50 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	

8.8.2.51 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 CONNECTIONTESTMODE	

8.8.2.52 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).
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	

8.8.2.53 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).
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	

8.8.2.54 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.

Enumerator

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).
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	

8.8.2.55 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_CXPOCXPSTATUS	

8.8.2.56 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	

8.8.2.57 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	

8.8.2.58 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	

8.8.2.59 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	

8.8.2.60 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	

8.8.2.61 DeviceCharacterSetEnums

```
enum DeviceCharacterSetEnums
```

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

Enumerator

DeviceCharacterSet_UTF8	
DeviceCharacterSet_ASCII	
NUM_DEVICECHARACTERSET	

8.8.2.62 DeviceClockSelectorEnums

enum `DeviceClockSelectorEnums`

< Selects the clock frequency to access from the device.

Enumerator

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

8.8.2.63 DeviceConnectionStatusEnums

enum `DeviceConnectionStatusEnums`

< Indicates the status of the specified Connection.

Enumerator

<code>DeviceConnectionStatus_Active</code>	Connection is in use.
<code>DeviceConnectionStatus_Inactive</code>	Connection is not in use.
<code>NUM_DEVICECONNECTIONSTATUS</code>	

8.8.2.64 DeviceIndicatorModeEnums

enum `DeviceIndicatorModeEnums`

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

Enumerator

<code>DeviceIndicatorMode_Inactive</code>	
<code>DeviceIndicatorMode_Active</code>	
<code>DeviceIndicatorMode_ErrorStatus</code>	
<code>NUM_DEVICEINDICATORMODE</code>	

8.8.2.65 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	

8.8.2.66 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	

8.8.2.67 DevicePowerSupplySelectorEnums

```
enum DevicePowerSupplySelectorEnums
```

< Selects the power supply source to control or read.

Enumerator

DevicePowerSupplySelector_External	
NUM_DEVICEPOWERSUPPLYSELECTOR	

8.8.2.68 DeviceRegistersEndiannessEnums

```
enum DeviceRegistersEndiannessEnums
```

< Endianess of the registers of the device.

Enumerator

DeviceRegistersEndianness_Little	
DeviceRegistersEndianness_Big	
NUM_DEVICEREGISTERSENDIANNESS	

8.8.2.69 DeviceScanTypeEnums

enum [DeviceScanTypeEnums](#)

< Scan type of the sensor of the device.

Enumerator

DeviceScanType_Areascan	
NUM_DEVICESCANTYPE	

8.8.2.70 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	

8.8.2.71 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	

8.8.2.72 DeviceStreamChannelEndiannessEnums

enum `DeviceStreamChannelEndiannessEnums`

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

Enumerator

<code>DeviceStreamChannelEndianness_Big</code>	Stream channel data is big Endian.
<code>DeviceStreamChannelEndianness_Little</code>	Stream channel data is little Endian.
<code>NUM_DEVICESTREAMCHANNELENDIANCESS</code>	

8.8.2.73 DeviceStreamChannelTypeEnums

enum `DeviceStreamChannelTypeEnums`

< Reports the type of the stream channel.

Enumerator

<code>DeviceStreamChannelType_Transmitter</code>	Data stream transmitter channel.
<code>DeviceStreamChannelType_Receiver</code>	Data stream receiver channel.
<code>NUM_DEVICESTREAMCHANNELTYPE</code>	

8.8.2.74 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

<code>DeviceTapGeometry_Geometry_1X_1Y</code>	<code>Geometry_1X_1Y</code>
<code>DeviceTapGeometry_Geometry_1X2_1Y</code>	<code>Geometry_1X2_1Y</code>
<code>DeviceTapGeometry_Geometry_1X2_1Y2</code>	<code>Geometry_1X2_1Y2</code>
<code>DeviceTapGeometry_Geometry_2X_1Y</code>	<code>Geometry_2X_1Y</code>
<code>DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y</code>	<code>Geometry_2X_1Y2Geometry_2XE_1Y</code>
<code>DeviceTapGeometry_Geometry_2XE_1Y2</code>	<code>Geometry_2XE_1Y2</code>
<code>DeviceTapGeometry_Geometry_2XM_1Y</code>	<code>Geometry_2XM_1Y</code>
<code>DeviceTapGeometry_Geometry_2XM_1Y2</code>	<code>Geometry_2XM_1Y2</code>
<code>DeviceTapGeometry_Geometry_1X_1Y2</code>	<code>Geometry_1X_1Y2</code>
<code>DeviceTapGeometry_Geometry_1X_2YE</code>	<code>Geometry_1X_2YE</code>
<code>DeviceTapGeometry_Geometry_1X3_1Y</code>	<code>Geometry_1X3_1Y</code>

Enumerator

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
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	

8.8.2.75 DeviceTemperatureSelectorEnums

```
enum DeviceTemperatureSelectorEnums
```

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

Enumerator

DeviceTemperatureSelector_Sensor	
NUM_DEVICETEMPERATURESELECTOR	

8.8.2.76 DeviceTLTypeEnums

```
enum DeviceTLTypeEnums
```

< Transport Layer type of the device.

Enumerator

DeviceTLType_GigEVision	
DeviceTLType_CameraLink	
DeviceTLType_CameraLinkHS	
DeviceTLType_CoaXPress	
DeviceTLType_USB3Vision	
DeviceTLType_Custom	
NUM_DEVICETLTYPE	

8.8.2.77 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	

8.8.2.78 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	

8.8.2.79 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.
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	

8.8.2.80 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	

8.8.2.81 EncoderResetSourceEnums

enum `EncoderResetSourceEnums`

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

Enumerator

<code>EncoderResetSource_Off</code>	Disable the Encoder Reset trigger.
<code>EncoderResetSource_AcquisitionTrigger</code>	Resets with the reception of the Acquisition Trigger.
<code>EncoderResetSource_AcquisitionStart</code>	Resets with the reception of the Acquisition Start.
<code>EncoderResetSource_AcquisitionEnd</code>	Resets with the reception of the Acquisition End.
<code>EncoderResetSource_FrameTrigger</code>	Resets with the reception of the Frame Start Trigger.
<code>EncoderResetSource_FrameStart</code>	Resets with the reception of the Frame Start.
<code>EncoderResetSource_FrameEnd</code>	Resets with the reception of the Frame End.
<code>EncoderResetSource_ExposureStart</code>	Resets with the reception of the Exposure Start.
<code>EncoderResetSource_ExposureEnd</code>	Resets with the reception of the Exposure End.
<code>EncoderResetSource_Line0</code>	Resets by the chosen I/O Line.
<code>EncoderResetSource_Line1</code>	Resets by the chosen I/O Line.
<code>EncoderResetSource_Line2</code>	Resets by the chosen I/O Line.
<code>EncoderResetSource_Counter0Start</code>	Resets with the reception of the Counter Start.
<code>EncoderResetSource_Counter1Start</code>	Resets with the reception of the Counter Start.
<code>EncoderResetSource_Counter2Start</code>	Resets with the reception of the Counter Start.
<code>EncoderResetSource_Counter0End</code>	Resets with the reception of the Counter End.
<code>EncoderResetSource_Counter1End</code>	Resets with the reception of the Counter End.
<code>EncoderResetSource_Counter2End</code>	Resets with the reception of the Counter End.
<code>EncoderResetSource_Timer0Start</code>	Resets with the reception of the Timer Start.
<code>EncoderResetSource_Timer1Start</code>	Resets with the reception of the Timer Start.
<code>EncoderResetSource_Timer2Start</code>	Resets with the reception of the Timer Start.
<code>EncoderResetSource_Timer0End</code>	Resets with the reception of the Timer End.
<code>EncoderResetSource_Timer1End</code>	Resets with the reception of the Timer End.
<code>EncoderResetSource_Timer2End</code>	Resets with the reception of the Timer End.
<code>EncoderResetSource_UserOutput0</code>	Resets by the chosen User Output bit.
<code>EncoderResetSource_UserOutput1</code>	Resets by the chosen User Output bit.
<code>EncoderResetSource_UserOutput2</code>	Resets by the chosen User Output bit.
<code>EncoderResetSource_SoftwareSignal0</code>	Resets on the reception of the Software Signal.
<code>EncoderResetSource_SoftwareSignal1</code>	Resets on the reception of the Software Signal.
<code>EncoderResetSource_SoftwareSignal2</code>	Resets on the reception of the Software Signal.
<code>EncoderResetSource_Action0</code>	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
<code>EncoderResetSource_Action1</code>	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
<code>EncoderResetSource_Action2</code>	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
<code>EncoderResetSource_LinkTrigger0</code>	Resets on the reception of the chosen Link Trigger (received from the transport layer).
<code>EncoderResetSource_LinkTrigger1</code>	Resets on the reception of the chosen Link Trigger (received from the transport layer).

Enumerator

EncoderResetSource_LinkTrigger2	Resets on the reception of the chosen Link Trigger (received from the transport layer).
NUM_ENCODERRESETSOURCE	

8.8.2.82 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	

8.8.2.83 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	

8.8.2.84 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	

8.8.2.85 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	

8.8.2.86 EventNotificationEnums

```
enum EventNotificationEnums
```

< Enables/Disables the selected event.

Enumerator

EventNotification_On	
EventNotification_Off	
NUM_EVENTNOTIFICATION	

8.8.2.87 EventSelectorEnums

```
enum EventSelectorEnums
```

< Selects which [Event](#) to enable or disable.

Enumerator

EventSelector_Error	
EventSelector_ExposureEnd	
EventSelector_SerialPortReceive	
NUM_EVENTSELECTOR	

8.8.2.88 ExposureActiveModeEnums

```
enum ExposureActiveModeEnums
```

< Control sensor active exposure mode.

Enumerator

ExposureActiveMode_Line1	
ExposureActiveMode_AnyPixels	
ExposureActiveMode_AllPixels	
NUM_EXPOSUREACTIVEMODE	

8.8.2.89 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	

8.8.2.90 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	

8.8.2.91 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	

8.8.2.92 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	

8.8.2.93 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	

8.8.2.94 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	

8.8.2.95 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	

8.8.2.96 FileSelectorEnums

enum `FileSelectorEnums`

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

Enumerator

<code>FileSelector_UserSetDefault</code>	
<code>FileSelector_UserSet0</code>	
<code>FileSelector_UserSet1</code>	
<code>FileSelector_UserFile1</code>	
<code>FileSelector_SerialPort0</code>	
<code>NUM_FILESELECTOR</code>	

8.8.2.97 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

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

8.8.2.98 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

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

8.8.2.99 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	

8.8.2.100 GevCCPEnums

enum [GevCCPEnums](#)

< Controls the device access privilege of an application.

Enumerator

GevCCP_OpenAccess	
GevCCP_ExclusiveAccess	
GevCCP_ControlAccess	
NUM_GEVCCP	

8.8.2.101 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	

8.8.2.102 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	

8.8.2.103 GevGVSPExtendedIDModeEnums

enum [GevGVSPExtendedIDModeEnums](#)

< Enables the extended IDs mode.

Enumerator

GevGVSPExtendedIDMode_Off	Off
GevGVSPExtendedIDMode_On	On
NUM_GEVGVSPEXTENDEDIDMODE	

8.8.2.104 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	

8.8.2.105 GevIEEE1588ModeEnums

enum [GevIEEE1588ModeEnums](#)

< Provides the mode of the IEEE 1588 clock.

Enumerator

GevIEEE1588Mode_Auto	Automatic
GevIEEE1588Mode_SlaveOnly	Slave Only
NUM_GEVIEEE1588MODE	

8.8.2.106 GevIEEE1588StatusEnums

```
enum GevIEEE1588StatusEnums
```

< Provides the status of the IEEE 1588 clock.

Enumerator

GevIEEE1588Status_Initializing	Initializing
GevIEEE1588Status_Faulty	Faulty
GevIEEE1588Status_Disabled	Disabled
GevIEEE1588Status_Listening	Listening
GevIEEE1588Status_PreMaster	Pre Master
GevIEEE1588Status_Master	Master
GevIEEE1588Status_Passive	Passive
GevIEEE1588Status_Uncalibrated	Uncalibrated
GevIEEE1588Status_Slave	Slave
NUM_GEVIEEE1588STATUS	

8.8.2.107 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	

8.8.2.108 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	

8.8.2.109 GevSupportedOptionSelectorEnums

enum [GevSupportedOptionSelectorEnums](#)

< Selects the GEV option to interrogate for existing support.

Enumerator

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	

8.8.2.110 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.
<code>ImageComponentSelector_Disparity</code>	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.
<code>ImageComponentSelector_Confidence</code>	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.
<code>ImageComponentSelector_Scatter</code>	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.
<code>NUM_IMAGECOMPONENTSELECTOR</code>	

8.8.2.111 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

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

Enumerator

ImageCompressionJPEGFormatOption_Progressive	Indicates this is a progressive (multi-scan) DCT-based JPEG.
NUM_IMAGECOMPRESSIONJPEGFORMATOPTION	

8.8.2.112 ImageCompressionModeEnums

```
enum ImageCompressionModeEnums
```

<

Enumerator

ImageCompressionMode_Off	
ImageCompressionMode_Lossless	
NUM_IMAGECOMPRESSIONMODE	

8.8.2.113 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	

8.8.2.114 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	

8.8.2.115 LineInputFilterSelectorEnums

```
enum LineInputFilterSelectorEnums
```

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

Enumerator

LineInputFilterSelector_Deglitch	
LineInputFilterSelector_Debounce	
NUM_LINEINPUTFILTERSELECTION	

8.8.2.116 LineModeEnums

```
enum LineModeEnums
```

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

Enumerator

LineMode_Input	
LineMode_Output	
NUM_LINEMODE	

8.8.2.117 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	

8.8.2.118 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	

8.8.2.119 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	

8.8.2.120 LogicBlockLUTInputSelectorEnums

```
enum LogicBlockLUTInputSelectorEnums
```

< Controls which LogicBlockLUT Input Source & Activation to access.

Enumerator

LogicBlockLUTInputSelector_Input0	
LogicBlockLUTInputSelector_Input1	
LogicBlockLUTInputSelector_Input2	
LogicBlockLUTInputSelector_Input3	
NUM_LOGICBLOCKLUTINPUTSELECTOR	

8.8.2.121 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

Enumerator

LogicBlockLUTInputSource_Counter1End	Counter1End
LogicBlockLUTInputSource_LogicBlock0	LogicBlock0
LogicBlockLUTInputSource_LogicBlock1	LogicBlock1
LogicBlockLUTInputSource_ExposureStart	ExposureStart
LogicBlockLUTInputSource_ExposureEnd	ExposureEnd
LogicBlockLUTInputSource_FrameTriggerWait	FrameTriggerWait
LogicBlockLUTInputSource_AcquisitionActive	AcquisitionActive
NUM_LOGICBLOCKLUTINPUTSOURCE	

8.8.2.122 LogicBlockLUTSelectorEnums

```
enum LogicBlockLUTSelectorEnums
```

< Selects which LogicBlock LUT to configure

Enumerator

LogicBlockLUTSelector_Value	
LogicBlockLUTSelector_Enable	
NUM_LOGICBLOCKLUTSELECTOR	

8.8.2.123 LogicBlockSelectorEnums

```
enum LogicBlockSelectorEnums
```

< Selects which LogicBlock to configure

Enumerator

LogicBlockSelector_LogicBlock0	
LogicBlockSelector_LogicBlock1	
NUM_LOGICBLOCKSELECTOR	

8.8.2.124 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	

8.8.2.125 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.
PixelColorFilter_BayerGB	Bayer Green Blue filter.
PixelColorFilter_BayerGR	Bayer Green Red filter.
PixelColorFilter_BayerBG	Bayer Blue Green filter.
NUM_PIXELCOLORFILTER	

8.8.2.126 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	

Enumerator

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	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

Enumerator

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
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

Enumerator

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
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

Enumerator

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
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_CbYCrY	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_CbYCrY	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

Enumerator

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
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	

Enumerator

NUM_PIXELFORMAT	
-----------------	--

8.8.2.127 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
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

Enumerator

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
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

Enumerator

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
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

Enumerator

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
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

Enumerator

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
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_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

Enumerator

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
PixelFormatInfoSelector_YCbCr709_422_12p	YCbCr 4:2:2 12-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_422_12p_Cb←YCrY	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_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_LLkBayerRG8	Lossless Compression Bayer Red Green filter 8-bit
PixelFormatInfoSelector_JPEGMono8	JPEG Monochrome 8-bit
PixelFormatInfoSelector_JPEGColor8	JPEG Color 8-bit
NUM_PIXELFORMATINFOSELECTOR	

8.8.2.128 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	

8.8.2.129 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	

8.8.2.130 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	

8.8.2.131 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

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

8.8.2.132 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

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

8.8.2.133 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	

8.8.2.134 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	

8.8.2.135 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	

8.8.2.136 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	

8.8.2.137 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	

8.8.2.138 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	

8.8.2.139 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.
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	

8.8.2.140 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	

8.8.2.141 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	

8.8.2.142 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	

8.8.2.143 SequencerConfigurationModeEnums

```
enum SequencerConfigurationModeEnums
```

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

Enumerator

SequencerConfigurationMode_Off	
SequencerConfigurationMode_On	
NUM_SEQUENCERCONFIGURATIONMODE	

8.8.2.144 SequencerConfigurationValidEnums

```
enum SequencerConfigurationValidEnums
```

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

Enumerator

SequencerConfigurationValid_No	
SequencerConfigurationValid_Yes	
NUM_SEQUENCERCONFIGURATIONVALID	

8.8.2.145 SequencerModeEnums

```
enum SequencerModeEnums
```

< Controls whether or not a sequencer is active.

Enumerator

SequencerMode_Off	
SequencerMode_On	
NUM_SEQUENCERMODE	

8.8.2.146 SequencerSetValidEnums

```
enum SequencerSetValidEnums
```

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

Enumerator

SequencerSetValid_No	
SequencerSetValid_Yes	
NUM_SEQUENCERSETVALID	

8.8.2.147 SequencerTriggerActivationEnums

enum [SequencerTriggerActivationEnums](#)

< Specifies the activation mode of the sequencer trigger.

Enumerator

SequencerTriggerActivation_RisingEdge	
SequencerTriggerActivation_FallingEdge	
SequencerTriggerActivation_AnyEdge	
SequencerTriggerActivation_LevelHigh	
SequencerTriggerActivation_LevelLow	
NUM_SEQUENCERTRIGGERACTIVATION	

8.8.2.148 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	

8.8.2.149 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	

8.8.2.150 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	

8.8.2.151 SerialPortSelectorEnums

```
enum SerialPortSelectorEnums
```

< Selects which serial port of the device to control.

Enumerator

SerialPortSelector_SerialPort0	
NUM_SERIALPORTSELECTOR	

8.8.2.152 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	

8.8.2.153 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	

8.8.2.154 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	

8.8.2.155 SourceSelectorEnums

enum `SourceSelectorEnums`

< Selects the source to control.

Enumerator

<code>SourceSelector_Source0</code>	Selects the data source 0.
<code>SourceSelector_Source1</code>	Selects the data source 1.
<code>SourceSelector_Source2</code>	Selects the data source 2.
<code>SourceSelector_All</code>	Selects all the data sources.
<code>NUM_SOURCESELECTOR</code>	

8.8.2.156 TestPatternEnums

enum `TestPatternEnums`

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

Enumerator

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

8.8.2.157 TestPatternGeneratorSelectorEnums

enum `TestPatternGeneratorSelectorEnums`

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

Enumerator

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

8.8.2.158 TimerSelectorEnums

enum `TimerSelectorEnums`

< Selects which Timer to configure.

Enumerator

<code>TimerSelector_Timer0</code>	Selects the Timer 0.
<code>TimerSelector_Timer1</code>	Selects the Timer 1.
<code>TimerSelector_Timer2</code>	Selects the Timer 2.
<code>NUM_TIMERSELECTOR</code>	

8.8.2.159 TimerStatusEnums

enum `TimerStatusEnums`

< Returns the current status of the Timer.

Enumerator

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

8.8.2.160 TimerTriggerActivationEnums

enum `TimerTriggerActivationEnums`

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

Enumerator

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

8.8.2.161 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.
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.

Enumerator

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	

8.8.2.162 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	

8.8.2.163 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	

8.8.2.164 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	

8.8.2.165 TransferQueueModeEnums

```
enum TransferQueueModeEnums
```

< Specifies the operation mode of the transfer queue.

Enumerator

TransferQueueMode_FirstInFirstOut	Blocks first In are transferred Out first.
NUM_TRANSFERQUEuemode	

8.8.2.166 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 simultaneously.
NUM_TRANSFERSELECTOR	

8.8.2.167 TransferStatusSelectorEnums

enum `TransferStatusSelectorEnums`

< Selects which status of the transfer module to read.

Enumerator

<code>TransferStatusSelector_Streaming</code>	Data blocks are transmitted when enough data is available.
<code>TransferStatusSelector_Paused</code>	Data blocks transmission is suspended immediately.
<code>TransferStatusSelector_Stopping</code>	Data blocks transmission is stopping. The current block transmission will be completed and the transfer state will stop.
<code>TransferStatusSelector_Stopped</code>	Data blocks transmission is stopped.
<code>TransferStatusSelector_QueueOverflow</code>	Data blocks queue is in overflow state.
<code>NUM_TRANSFERSTATUSSELECTOR</code>	

8.8.2.168 TransferTriggerActivationEnums

enum `TransferTriggerActivationEnums`

< Specifies the activation mode of the transfer control trigger.

Enumerator

<code>TransferTriggerActivation_RisingEdge</code>	Specifies that the trigger is considered valid on the rising edge of the source signal.
<code>TransferTriggerActivation_FallingEdge</code>	Specifies that the trigger is considered valid on the falling edge of the source signal.
<code>TransferTriggerActivation_AnyEdge</code>	Specifies that the trigger is considered valid on the falling or rising edge of the source signal.
<code>TransferTriggerActivation_LevelHigh</code>	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.
<code>TransferTriggerActivation_LevelLow</code>	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.
<code>NUM_TRANSFERTRIGGERACTIVATION</code>	

8.8.2.169 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	

8.8.2.170 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	

8.8.2.171 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.

Enumerator

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.
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	

8.8.2.172 TriggerActivationEnums

```
enum TriggerActivationEnums
```

< Specifies the activation mode of the trigger.

Enumerator

TriggerActivation_LevelLow	
TriggerActivation_LevelHigh	
TriggerActivation_FallingEdge	
TriggerActivation_RisingEdge	
TriggerActivation_AnyEdge	
NUM_TRIGGERACTIVATION	

8.8.2.173 TriggerModeEnums

```
enum TriggerModeEnums
```

< Controls whether or not trigger is active.

Enumerator

TriggerMode_Off	
TriggerMode_On	
NUM_TRIGGERMODE	

8.8.2.174 TriggerOverlapEnums

```
enum TriggerOverlapEnums
```

< Specifies the overlap mode of the trigger.

Enumerator

TriggerOverlap_Off	
TriggerOverlap_ReadOut	
TriggerOverlap_PreviousFrame	
NUM_TRIGGEROVERLAP	

8.8.2.175 TriggerSelectorEnums

```
enum TriggerSelectorEnums
```

< Selects the type of trigger to configure.

Enumerator

TriggerSelector_AcquisitionStart	
TriggerSelector_FrameStart	
TriggerSelector_FrameBurstStart	
NUM_TRIGGERSELECTOR	

8.8.2.176 TriggerSourceEnums

enum [TriggerSourceEnums](#)

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

Enumerator

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	

8.8.2.177 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	

8.8.2.178 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	

8.8.2.179 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	

8.8.2.180 WhiteClipSelectorEnums

```
enum WhiteClipSelectorEnums
```

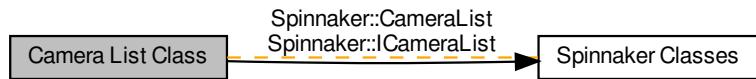
< Selects which White Clip to control.

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	

8.9 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.

8.9.1 Detailed Description

8.10 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 \(\)](#)

8.10.1 Detailed Description

8.10.2 Function Documentation

8.10.2.1 CameraPtr() [1/4]

[CameraPtr \(\) throw \(\)](#) [inline]

Default constructor.

8.10.2.2 CameraPtr() [2/4]

[CameraPtr \(](#)
 [const int \) throw \(\)](#) [inline]

Default constructor.

8.10.2.3 CameraPtr() [3/4]

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

Default constructor with argument.

8.10.2.4 CameraPtr() [4/4]

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

8.11 ChunkData Class

Collaboration diagram for ChunkData Class:



Classes

- class [ChunkData](#)
The chunk data which contains additional information about an image.

8.11.1 Detailed Description

8.12 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`

8.12.1 Detailed Description

8.12.2 Function Documentation

8.12.2.1 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.
--------------	--------------------------------------

8.12.2.2 GetBoxCount()

```
int16_t GetBoxCount ( ) const
```

Returns the number of bounding boxes.

8.12.2.3 GetBoxSize()

```
int8_t GetBoxSize ( ) const
```

Returns the number of bytes allocated for one bounding box.

8.12.2.4 GetVersion()

```
int8_t GetVersion ( ) const
```

Returns the bounding box format version number.

8.12.2.5 InferenceBoundingBoxResult() [1/3]

```
InferenceBoundingBoxResult ( )
```

Default Constructor.

8.12.2.6 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.

8.12.2.7 InferenceBoundingBoxResult() [3/3]

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

Copy Constructor.

8.12.2.8 operator=()

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

Assignment Operator.

8.12.2.9 ~InferenceBoundingBoxResult()

```
~InferenceBoundingBoxResult ()
```

Destructor.

8.12.3 Variable Documentation**8.12.3.1 bottomRightXCoord [1/2]**

```
int16_t bottomRightXCoord
```

8.12.3.2 bottomRightXCoord [2/2]

```
int16_t bottomRightXCoord
```

8.12.3.3 bottomRightYCoord [1/2]

```
int16_t bottomRightYCoord
```

8.12.3.4 bottomRightYCoord [2/2]

```
int16_t bottomRightYCoord
```

8.12.3.5 boxType

```
InferenceBoxType boxType
```

8.12.3.6 centerXCoord

```
int16_t centerXCoord
```

8.12.3.7 centerYCoord

```
int16_t centerYCoord
```

8.12.3.8 circle

```
InferenceBoxCircle circle
```

8.12.3.9 classId

```
int16_t classId
```

8.12.3.10 confidence

```
float32_t confidence
```

8.12.3.11 radius

```
int16_t radius
```

8.12.3.12 rect

```
InferenceBoxRect rect
```

8.12.3.13 rotatedRect

```
InferenceBoxRotatedRect rotatedRect
```

8.12.3.14 rotationAngle

```
short rotationAngle
```

8.12.3.15 topLeftXCoord [1/2]

```
int16_t topLeftXCoord
```

8.12.3.16 topLeftXCoord [2/2]

```
int16_t topLeftXCoord
```

8.12.3.17 topLeftYCoord [1/2]

```
int16_t topLeftYCoord
```

8.12.3.18 topLeftYCoord [2/2]

```
int16_t topLeftYCoord
```

8.13 DeviceEvent Class

Collaboration diagram for DeviceEvent Class:



Classes

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

8.13.1 Detailed Description

8.14 Event Class

Collaboration diagram for Event Class:



Classes

- class [Event](#)

The base class for all event types.

8.14.1 Detailed Description

8.15 Exception Class

Collaboration diagram for Exception Class:



Classes

- class [Exception](#)

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

8.15.1 Detailed Description

8.16 Image Class

Collaboration diagram for Image Class:



Classes

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

8.16.1 Detailed Description

8.17 ImageEvent Class

Collaboration diagram for ImageEvent Class:



Classes

- class [ImageEvent](#)

A handler for capturing image arrival events.

8.17.1 Detailed Description

8.18 ImagePtr Class

Collaboration diagram for ImagePtr Class:



Classes

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

8.18.1 Detailed Description

8.19 ImageStatistics Class

Collaboration diagram for ImageStatistics Class:



Classes

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

8.19.1 Detailed Description

8.20 Image Utility Class

Collaboration diagram for Image Utility Class:



Classes

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

8.20.1 Detailed Description

8.21 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.

8.21.1 Detailed Description

8.22 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.

8.22.1 Detailed Description

8.23 Interface Class

Collaboration diagram for Interface Class:



Classes

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

8.23.1 Detailed Description

8.24 InterfaceArrivalEvent Class

Collaboration diagram for InterfaceArrivalEvent Class:



Classes

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

8.24.1 Detailed Description

8.25 InterfaceEvent Class

Collaboration diagram for InterfaceEvent Class:



Classes

- class [InterfaceEvent](#)
A handler to device arrival and removal events on all interfaces.

8.25.1 Detailed Description

8.26 InterfaceList Class

Collaboration diagram for InterfaceList Class:



Classes

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

8.26.1 Detailed Description

8.27 InterfacePtr Class

Collaboration diagram for InterfacePtr Class:



Classes

- class [InterfacePtr](#)

A reference tracked pointer to the interface object.

8.27.1 Detailed Description

8.28 InterfaceRemovalEvent Class

Collaboration diagram for InterfaceRemovalEvent Class:



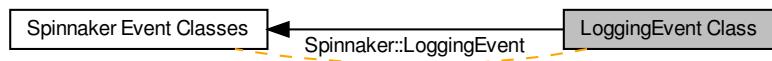
Classes

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

8.28.1 Detailed Description

8.29 LoggingEvent Class

Collaboration diagram for LoggingEvent Class:



Classes

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

8.29.1 Detailed Description

8.30 Logging Event Class

Collaboration diagram for Logging Event Class:



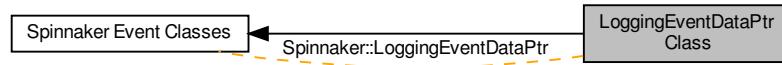
Classes

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

8.30.1 Detailed Description

8.31 LoggingEventDataPtr Class

Collaboration diagram for LoggingEventDataPtr Class:



Classes

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

8.31.1 Detailed Description

8.32 RemovalEvent Class

Collaboration diagram for RemovalEvent Class:



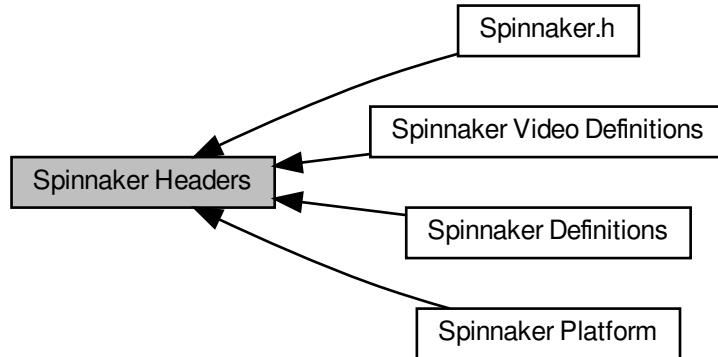
Classes

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

8.32.1 Detailed Description

8.33 Spinnaker Headers

Collaboration diagram for Spinnaker Headers:



Modules

- [Spinnaker.h](#)
Global header file for Spinnaker.
- [Spinnaker Definitions](#)
Definitions file for Spinnaker.
- [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](#) = 0xFFFFFFFFFFFFFF

8.33.1 Detailed Description

8.33.2 Variable Documentation

8.33.2.1 EVENT_TIMEOUT_INFINITE

```
const uint64_t EVENT_TIMEOUT_INFINITE = 0xFFFFFFFFFFFFFF
```

8.33.2.2 EVENT_TIMEOUT_NONE

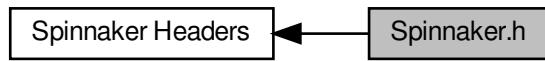
```
const uint64_t EVENT_TIMEOUT_NONE = 0
```

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

8.34 Spinnaker.h

Global header file for [Spinnaker](#).

Collaboration diagram for Spinnaker.h:



Global header file for [Spinnaker](#).

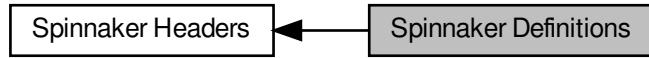
By including this file, all required header files for full [Spinnaker](#) operation will be included automatically. It is recommended that this file be used instead of manually including individual header files.

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

8.35 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.

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 = 3,
 IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT = 4,
 IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT = 5,
 IMAGE_PACKETID_INCONSISTENT = 6,
 IMAGE_MISSING_LEADER = 7,
 IMAGE_MISSING_TRAILER = 8,
 IMAGE_DATA_INCOMPLETE = 9,
 IMAGE_INFO_INCONSISTENT = 10,
 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,

- `PAYOUT_TYPE_IMAGE` = 1,
`PAYOUT_TYPE_RAW_DATA` = 2,
`PAYOUT_TYPE_FILE` = 3,
`PAYOUT_TYPE_CHUNK_DATA` = 4,
`PAYOUT_TYPE_JPEG` = 5,
`PAYOUT_TYPE_JPEG2000` = 6,
`PAYOUT_TYPE_H264` = 7,
`PAYOUT_TYPE_CHUNK_ONLY` = 8,
`PAYOUT_TYPE_DEVICE_SPECIFIC` = 9,
`PAYOUT_TYPE_MULTI_PART` = 10,
`PAYOUT_TYPE_CUSTOM_ID` = 1000,
`PAYOUT_TYPE_EXTENDED_CHUNK` = 1001 }
- 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` }

Functions

- `DEPRECATED_ENUM` ("This enum has been deprecated. Polarization images are now created through specific " "functions in the `ImageUtilityPolarization` class.") `PolarizationAlgorithm`
- `DEPRECATED_ENUM` ("This enum has been deprecated. `Image` scaling can now be applied through specific functions " "defined in the `ImageUtility` class.") `PolarizationResolution`

Variables

- `HeatMapColor`

8.35.1 Detailed Description

Definitions file for `Spinnaker`.

8.35.2 Enumeration Type Documentation

8.35.2.1 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	

8.35.2.2 BufferOwnership

enum `BufferOwnership`

Enumerator

BUFFER_OWNERSHIP_SYSTEM	
BUFFER_OWNERSHIP_USER	

8.35.2.3 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.

Enumerator

BILINEAR	Weighted average of surrounding 4 pixels in a 2x2 neighborhood.
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.
RIGOROUS	Slowest but produces good results.
WEIGHTED_DIRECTIONAL_FILTER	Weighted pixel average from different directions.

8.35.2.4 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	

Enumerator

GENICAM_ERR_PROPERTY	
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	

8.35.2.5 EventType

```
enum EventType
```

[Event](#) types in Spinnaker.

See also

[Event::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	

8.35.2.6 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	

8.35.2.7 ImageStatus

```
enum ImageStatus
```

Status of images returned from GetNextImage() call.

Enumerator

IMAGE_UNKNOWN_ERROR	Image has an unknown error.
IMAGE_NO_ERROR	Image is returned from GetNextImage() call without any errors.
IMAGE_CRC_CHECK_FAILED	Image failed CRC check.
IMAGE_DATA_OVERFLOW	Received more data than the size of the image.
IMAGE_MISSING_PACKETS	Image has missing packets.
IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT	Image leader is incomplete.
IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT	Image trailer is incomplete.
IMAGE_PACKETID_INCONSISTENT	Image has an inconsistent packet id.
IMAGE_MISSING_LEADER	Image leader is missing.
IMAGE_MISSING_TRAILER	Image trailer is missing.
IMAGE_DATA_INCOMPLETE	Image data is incomplete.
IMAGE_INFO_INCONSISTENT	Image info is corrupted.
IMAGE_CHUNK_DATA_INVALID	Image chunk data is invalid.
IMAGE_NO_SYSTEM_RESOURCES	Image cannot be processed due to lack of system resources.

8.35.2.8 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	

8.35.2.9 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	

8.35.2.10 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	

8.35.2.11 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	

8.35.2.12 StatisticsChannel

enum [StatisticsChannel](#)

Channels that allow statistics to be calculated.

Enumerator

GREY	
RED	
GREEN	

Enumerator

BLUE	
HUE	
SATURATION	
LIGHTNESS	
NUM_STATISTICS_CHANNELS	

8.35.3 Function Documentation

8.35.3.1 DEPRECATED_ENUM() [1/2]

```
Spinnaker::DEPRECATED_ENUM (
    "This enum has been deprecated. Polarization images are now created through
specific \"functions in the ImageUtilityPolarization class." )
```

No polarization.

Extracts a Mono8 pixel format image of the 0 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format image of the 45 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format image of the 90 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format image of the 135 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format Stokes' parameter image S0.

Extracts a BGRA8 pixel format Heatmap representation of the Stokes' parameter image S0.

Extracts a Mono8 pixel format Stokes' parameter image S1.

Extracts a BGRA8 pixel format Heatmap representation of the Stokes' parameter image S1.

Extracts a Mono8 pixel format Stokes' parameter image S2.

Extracts a BGRA8 pixel format Heatmap representation of the Stokes' parameter image S2.

Extracts a Mono8 pixel format image representation of the DoLP (Degree of Linear Polarization).

Extracts a BGRA8 pixel format Heatmap representation of the DoLP (Degree of Linear Polarization). Resulting polarization values are normalized between 0 and 1.

Extracts a Mono8 pixel format image representation of the AoP (Angle of Polarization).

Extracts a BGRA8 pixel format Heatmap representation of the AoP (Angle of Polarization). Resulting polarization values are normalized between 0 and 1.

8.35.3.2 DEPRECATED_ENUM() [2/2]

```
Spinnaker::DEPRECATED_ENUM (
    "This enum has been deprecated. Image scaling can now be applied through specific
functions " "defined in the ImageUtility class." )
```

Quarter Resolution.

Full Resolution.

8.35.4 Variable Documentation

8.35.4.1 HeatMapColor

HeatMapColor

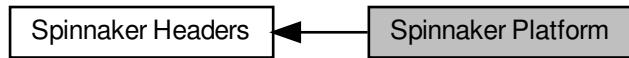
Initial value:

```
{
    HEATMAP_BLACK = 1,
    HEATMAP_BLUE = 2,
    HEATMAP_CYAN = 3,
    HEATMAP_GREEN = 4,
    HEATMAP_YELLOW = 5,
    HEATMAP_RED = 6,
    HEATMAP_WHITE = 7
}
```

8.36 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")))`

8.36.1 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.

8.36.2 Macro Definition Documentation

8.36.2.1 SPINNAKER_API

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

8.36.2.2 SPINNAKER_API_ABSTRACT

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

8.36.2.3 SPINNAKER_LOCAL

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

8.37 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.

8.37.1 Detailed Description

8.38 Spinnaker Video Definitions

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

Collaboration diagram for Spinnaker Video Definitions:



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

8.39 System Class

Collaboration diagram for System Class:



Classes

- class [System](#)

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

8.39.1 Detailed Description

8.40 SystemEvent Class

Collaboration diagram for SystemEvent Class:



Classes

- class [SystemEvent](#)
A handler to interface arrival and removal events on the system.

8.40.1 Detailed Description

8.41 SystemPtr Class

Collaboration diagram for SystemPtr Class:



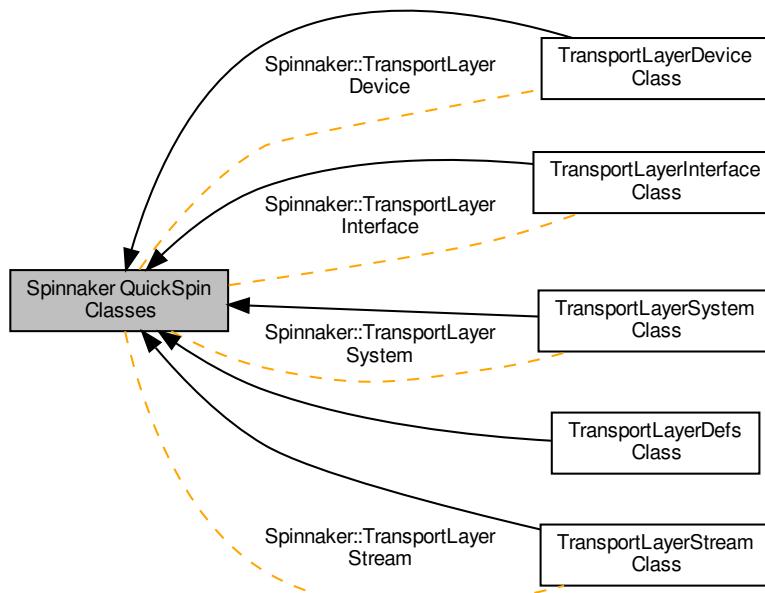
Classes

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

8.41.1 Detailed Description

8.42 Spinnaker QuickSpin Classes

Collaboration diagram for Spinnaker QuickSpin Classes:



Modules

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

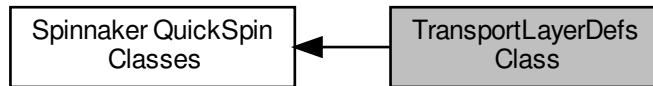
Classes

- 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.

8.42.1 Detailed Description

8.43 TransportLayerDefs Class

Collaboration diagram for TransportLayerDefs Class:



Enumerations

- enum StreamTypeEnum {
 StreamType_Mixed,
 StreamType_Custom,
 StreamType_GEV,
 StreamType_CL,
 StreamType_IIDC,
 StreamType_UVC,
 StreamType_CXP,
 StreamType_CLHS,
 StreamType_U3V,
 StreamType_ETHERNET,
 StreamType_PCI,
 NUMSTREAMTYPE }

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

- enum StreamDefaultBufferCountModeEnum {
 StreamDefaultBufferCountMode_Manual,
 StreamDefaultBufferCountMode_Auto,
 NUMSTREAMDEFAULTBUFFERCOUNTMODE }
- enum StreamBufferCountModeEnum {
 StreamBufferCountMode_Manual,
 StreamBufferCountMode_Auto,
 NUMSTREAMBUFFERCOUNTMODE }
- enum StreamBufferHandlingModeEnum {
 StreamBufferHandlingMode_OldestFirst,
 StreamBufferHandlingMode_OldestFirstOverwrite,
 StreamBufferHandlingMode_NewestFirst,
 StreamBufferHandlingMode_NewestFirstOverwrite,
 StreamBufferHandlingMode_NewestOnly,
 NUMSTREAMBUFFERHANDLINGMODE }
- enum DeviceTypeEnum {
 DeviceType_Mixed,
 DeviceType_Custom,
 DeviceType_GEV,
 DeviceType_CL,
 DeviceType_IIDC,
 DeviceType_UVC,
 DeviceType_CXP,
 DeviceType_CLHS,

```

DeviceType_U3V,
DeviceType_ETHERNET,
DeviceType_PCI,
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 DeviceEndianessMechanismEnum {
DeviceEndianessMechanism_Legacy,
DeviceEndianessMechanism_Standard,
NUMDEVICEENDIANESSMECHANISM }

• enum DeviceCurrentSpeedEnum {
DeviceCurrentSpeed_UnknownSpeed,
DeviceCurrentSpeed_LowSpeed,
DeviceCurrentSpeed_FullSpeed,
DeviceCurrentSpeed_HighSpeed,
DeviceCurrentSpeed_SuperSpeed,
NUMDEVICECURRENTSPEED }

• enum POEStatusEnum {
POEStatus_NotSupported,
POEStatus_PowerOff,
POEStatus_PowerOn,
NUMPOESTATUS }

• enum FilterDriverStatusEnum {
FilterDriverStatus_NotSupported,
FilterDriverStatus_Disabled,
FilterDriverStatus_Enabled,
NUMFILTERDRIVERSTATUS }

```

8.43.1 Detailed Description

8.43.2 Enumeration Type Documentation

8.43.2.1 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	

8.43.2.2 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	

8.43.2.3 DeviceEndianessMechanismEnum

```
enum DeviceEndianessMechanismEnum
```

< Identifies the endianness handling mode.

Enumerator

DeviceEndianessMechanism_Legacy	Handling the device endianness according to GenICam Schema 1.0
DeviceEndianessMechanism_Standard	Handling the device endianness according to GenICam Schema 1.1 and later
NUMDEVICEENDIANESSMECHANISM	

8.43.2.4 DeviceTypeEnum

enum `DeviceTypeEnum`

< Transport layer type of the device.

Enumerator

<code>DeviceType_Mixed</code>	TL - Mixed
<code>DeviceType_Custom</code>	TL - Custom
<code>DeviceType_GEV</code>	TL - GEV
<code>DeviceType_CL</code>	TL - CL
<code>DeviceType_IIDC</code>	TL - IIDC
<code>DeviceType_UVC</code>	TL - UVC
<code>DeviceType_CXP</code>	TL - CXP
<code>DeviceType_CLHS</code>	TL - CLHS
<code>DeviceType_U3V</code>	TL - U3V
<code>DeviceType_ETHERNET</code>	TL - ETHERNET
<code>DeviceType_PCI</code>	TL - PCI
<code>NUMDEVICETYPE</code>	

8.43.2.5 FilterDriverStatusEnum

enum `FilterDriverStatusEnum`

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

Enumerator

<code>FilterDriverStatus_NotSupported</code>	Not Supported
<code>FilterDriverStatus_Disabled</code>	FLIR Light Weight Filter Driver is disabled
<code>FilterDriverStatus_Enabled</code>	FLIR Light Weight Filter Driver is enabled
<code>NUMFILTERDRIVERSTATUS</code>	

8.43.2.6 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	

8.43.2.7 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	

8.43.2.8 GUIXMLLocationEnum

```
enum GUIXMLLocationEnum
```

< Sets the location to load GUI XML.

Enumerator

GUIXMLLocation_Device	Load XML from device
GUIXMLLocation_Host	Load XML from host
NUMGUIXMLLOCATION	

8.43.2.9 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	

8.43.2.10 StreamBufferCountModeEnum

enum `StreamBufferCountModeEnum`

< Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints.

Enumerator

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

8.43.2.11 StreamBufferHandlingModeEnum

enum `StreamBufferHandlingModeEnum`

< Available buffer handling modes of this data stream:

Enumerator

<code>StreamBufferHandlingMode_OldestFirst</code>	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.
<code>StreamBufferHandlingMode_OldestFirstOverwrite</code>	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).
<code>StreamBufferHandlingMode_NewestFirst</code>	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.
<code>StreamBufferHandlingMode_NewestFirstOverwrite</code>	DEPRECATED. This is replaced by <code>NewestOnly</code> .
<code>StreamBufferHandlingMode_NewestOnly</code>	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.
<code>NUMSTREAMBUFFERHANDLINGMODE</code>	

8.43.2.12 StreamDefaultBufferCountModeEnum

enum `StreamDefaultBufferCountModeEnum`

< DEPRECATED; Replaced by StreamBufferCountMode. Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints.

Enumerator

<code>StreamDefaultBufferCountMode_Manual</code>	DEPRECATED. The number of buffers used for the stream are set by the user.
<code>StreamDefaultBufferCountMode_Auto</code>	DEPRECATED. The number of buffers used for the stream is automatically calculated.
<code>NUMSTREAMDEFAULTBUFFERCOUNTMODE</code>	

8.43.2.13 StreamTypeEnum

enum `StreamTypeEnum`

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

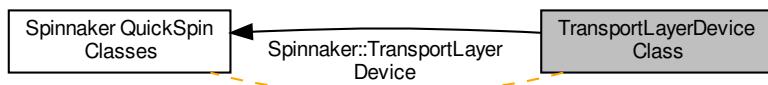
< Stream type of the device.

Enumerator

<code>StreamType_Mixed</code>	Stream Type - Mixed
<code>StreamType_Custom</code>	Stream Type - Custom
<code>StreamType_GEV</code>	Stream Type - GEV
<code>StreamType_CL</code>	Stream Type - CL
<code>StreamType_IIDC</code>	Stream Type - IIDC
<code>StreamType_UVC</code>	Stream Type - UVC
<code>StreamType_CXP</code>	Stream Type - CXP
<code>StreamType_CLHS</code>	Stream Type - CLHS
<code>StreamType_U3V</code>	Stream Type - U3V
<code>StreamType_ETHERNET</code>	Stream Type - ETHERNET
<code>StreamType_PCI</code>	Stream Type - PCI
<code>NUMSTREAMTYPE</code>	

8.44 TransportLayerDevice Class

Collaboration diagram for TransportLayerDevice Class:



Classes

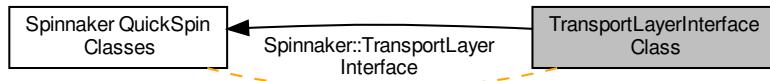
- class [TransportLayerDevice](#)

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

8.44.1 Detailed Description

8.45 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.

8.45.1 Detailed Description

8.46 TransportLayerStream Class

Collaboration diagram for TransportLayerStream Class:



Classes

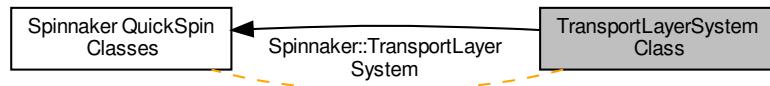
- class [TransportLayerStream](#)

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

8.46.1 Detailed Description

8.47 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.

8.47.1 Detailed Description

8.48 Camera Base Interface Class

Collaboration diagram for Camera Base Interface Class:



Classes

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

8.48.1 Detailed Description

8.49 IChunkData Class

Collaboration diagram for IChunkData Class:



Classes

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

8.49.1 Detailed Description

8.50 IImage Class

Collaboration diagram for IImage Class:



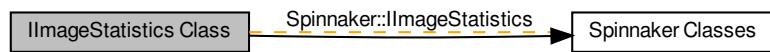
Classes

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

8.50.1 Detailed Description

8.51 IImageStatistics Class

Collaboration diagram for IImageStatistics Class:



Classes

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

8.51.1 Detailed Description

8.52 IIInterface Class

Collaboration diagram for IIInterface Class:



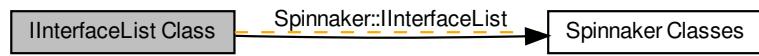
Classes

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

8.52.1 Detailed Description

8.53 IIInterfaceList Class

Collaboration diagram for IIInterfaceList Class:



Classes

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

8.53.1 Detailed Description

8.54 ISystem Class

Collaboration diagram for ISystem Class:



Classes

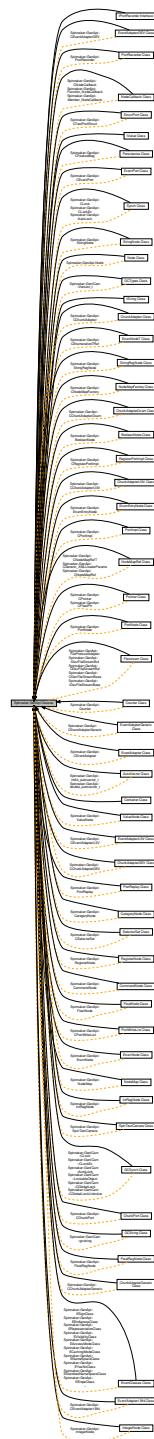
- class [ISystem](#)

The interface file for [System](#).

8.54.1 Detailed Description

8.55 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 >](#)

- class [Member_NodeCallback< Client, Member >](#)

Container for a function pointer.
- class [NodeMap](#)

Container for a member function pointer.
- class [CNodeMapFactory](#)

Smart pointer template for NodeMaps with create function.
- 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](#)
- 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](#)
- class [ValueNode](#)

Interface for value properties.

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.

8.55.1 Detailed Description

8.55.2 Typedef Documentation

8.55.2.1 CNodeMapRef

```
typedef NodeMap CNodeMapRef
```

8.55.2.2 CNodeRef

```
typedef Node CNodeRef
```

8.55.2.3 CSelectorRef

```
typedef Node CSelectorRef
```

8.55.3 Function Documentation

8.55.3.1 _ClearXMLCache()

```
bool Spinnaker::GenApi::_ClearXMLCache ( ) [inline]
```

8.55.3.2 `_Connect()` [1/2]

```
bool Spinnaker::GenApi::_Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) [inline]
```

8.55.3.3 `_Connect()` [2/2]

```
bool Spinnaker::GenApi::_Connect (
    IPort * pPort ) [inline]
```

8.55.3.4 `_Destroy()`

```
void _Destroy ( ) [inline]
```

Destroys the node map.

8.55.3.5 `_GetDeviceName()`

```
GenICam::gcstring Spinnaker::GenApi::_GetDeviceName ( ) [inline]
```

8.55.3.6 `_GetNode()`

```
INode* Spinnaker::GenApi::_GetNode (
    const GenICam::gcstring & key ) [inline]
```

8.55.3.7 `_GetNodes()`

```
void Spinnaker::GenApi::_GetNodes (
    NodeList_t & Nodes ) [inline]
```

8.55.3.8 `_GetSupportedSchemaVersions()`

```
void Spinnaker::GenApi::_GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) [inline]
```

8.55.3.9 _InvalidateNodes()

```
void Spinnaker::GenApi::_InvalidateNodes ( ) [inline]
```

8.55.3.10 _LoadXMLFromFile()

```
void Spinnaker::GenApi::_LoadXMLFromFile (
    const GenICam::gcstring & FileName ) [inline]
```

8.55.3.11 _LoadXMLFromFileInject()

```
void Spinnaker::GenApi::_LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName ) [inline]
```

8.55.3.12 _LoadXMLFromString()

```
void Spinnaker::GenApi::_LoadXMLFromString (
    const GenICam::gcstring & XMLData ) [inline]
```

8.55.3.13 _LoadXMLFromStringInject()

```
void Spinnaker::GenApi::_LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData,
    const GenICam::gcstring & InjectXMLData ) [inline]
```

8.55.3.14 _LoadXMLFromZIPData()

```
void Spinnaker::GenApi::_LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize ) [inline]
```

8.55.3.15 _LoadXMLFromZIPFile()

```
void Spinnaker::GenApi::_LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName ) [inline]
```

8.55.3.16 `_Poll()`

```
void Spinnaker::GenApi::_Poll (
    int64_t ElapsedTime ) [inline]
```

8.55.3.17 `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)

8.55.3.18 `CNodeMapRefT()` [1/3]

```
CNodeMapRefT (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

8.55.3.19 `CNodeMapRefT()` [2/3]

```
CNodeMapRefT (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

8.55.3.20 `CNodeMapRefT()` [3/3]

```
CNodeMapRefT (
    const CNodeMapRefT< TCameraParams > & Them )
```

Copy constructor.

8.55.3.21 `EatComments()`

```
SPINNAKER_API std::istream& Spinnaker::GenApi::EatComments (
    std::istream & is )
```

Helper function ignoring lines starting with comment character '#'.

8.55.3.22 operator<<()

```
SPINNAKER_API std::ostream& Spinnaker::GenApi::operator<< (
    std::ostream & os,
    const CFeatureBag & FeatureBag )
```

writes out persistent data to a stream

8.55.3.23 operator=() [1/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    const CNodeMapRefT< TCameraParams > & Them )
```

Assignment.

8.55.3.24 operator=() [2/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    INodeMap * pNodeMap )
```

Assignment of an INodeMap*.

8.55.3.25 operator>>()

```
SPINNAKER_API std::istream& Spinnaker::GenApi::operator>> (
    std::istream & is,
    CFeatureBag & FeatureBag )
```

Reads in persistent data from a stream.

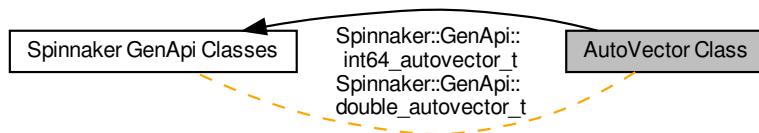
8.55.3.26 ~CNodeMapRefT()

```
~CNodeMapRefT ( ) [inline], [virtual]
```

Destructor.

8.56 AutoVector Class

Collaboration diagram for AutoVector Class:



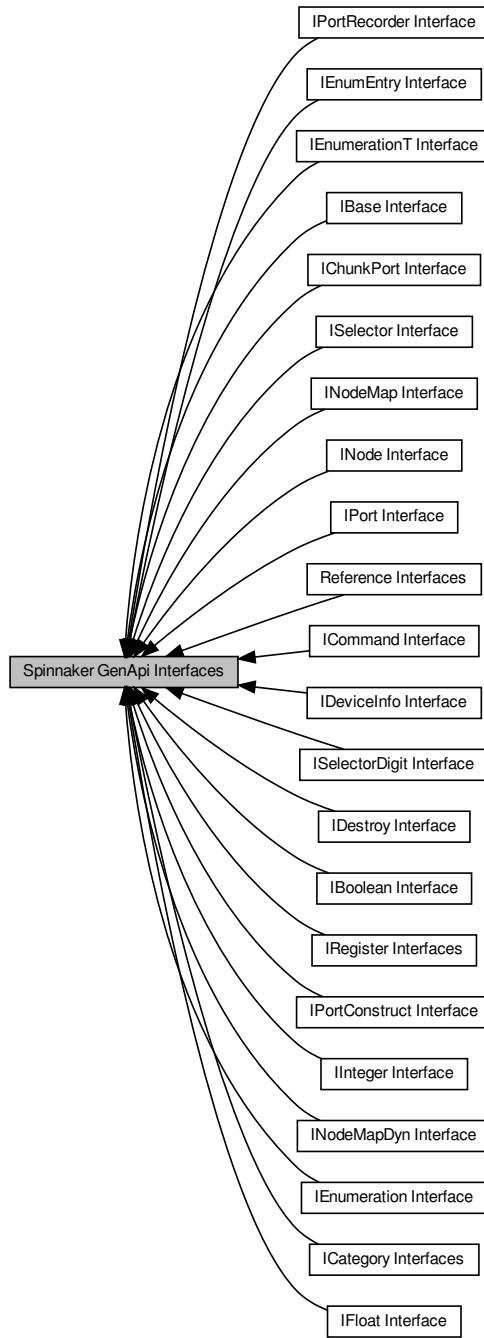
Classes

- class [int64_autovector_t](#)
Vector of integers with reference counting.
- class [double_autovector_t](#)
Vector of doubles with reference counting.

8.56.1 Detailed Description

8.57 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

8.57.1 Detailed Description

8.57.2 Typedef Documentation

8.57.2.1 CallbackHandleType

`typedef intptr_t CallbackHandleType`

the callback handle for nodes

8.57.2.2 NodeList_t

`typedef node_vector NodeList_t`

a list of node references

8.58 IBase Interface

Collaboration diagram for IBase Interface:



Variables

- `interface SPINNAKER_API_ABSTRACT IBase`
Base interface common to all nodes.

8.58.1 Detailed Description

8.58.2 Variable Documentation

8.58.2.1 IBase

```
interface SPINNAKER_API_ABSTRACT IBase
```

Initial value:

```
{  
    virtual EAccessMode GetAccessMode() const = 0
```

Base interface common to all nodes.

8.59 BooleanNode Class

Collaboration diagram for BooleanNode Class:



Classes

- class [BooleanNode](#)
Interface for string properties.

Typedefs

- [typedef BooleanNode CBooleanRef](#)

8.59.1 Detailed Description

8.59.2 Typedef Documentation

8.59.2.1 CBooleanRef

```
typedef BooleanNode CBooleanRef
```

8.60 CategoryNode Class

Collaboration diagram for CategoryNode Class:



Classes

- class [CategoryNode](#)
Interface for string properties.

Typedefs

- [typedef CategoryNode CCategoryRef](#)

8.60.1 Detailed Description

8.60.2 Typedef Documentation

8.60.2.1 CCategoryRef

```
typedef CategoryNode CCategoryRef
```

8.61 ChunkAdapter Class

Collaboration diagram for ChunkAdapter Class:



Classes

- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.

8.61.1 Detailed Description

8.62 ChunkAdapterDcam Class

Collaboration diagram for ChunkAdapterDcam Class:



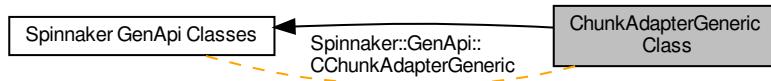
Classes

- class [CChunkAdapterDcam](#)
Connects a chunked DCAM buffer to a node map.

8.62.1 Detailed Description

8.63 ChunkAdapterGeneric Class

Collaboration diagram for ChunkAdapterGeneric Class:



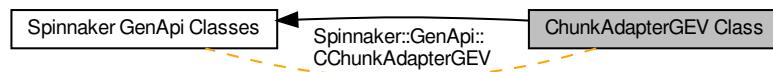
Classes

- class [CChunkAdapterGeneric](#)

8.63.1 Detailed Description

8.64 ChunkAdapterGEV Class

Collaboration diagram for ChunkAdapterGEV Class:



Classes

- class [CChunkAdapterGEV](#)

Connects a chunked DCAM buffer to a node map.

8.64.1 Detailed Description

8.65 ChunkPort Class

Collaboration diagram for ChunkPort Class:



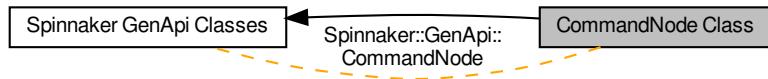
Classes

- class [CChunkPort](#)
Port attachable to a chunk in a buffer.

8.65.1 Detailed Description

8.66 CommandNode Class

Collaboration diagram for CommandNode Class:



Classes

- class [CommandNode](#)
Interface for string properties.

Typedefs

- [typedef CommandNode CCommandRef](#)

8.66.1 Detailed Description

8.66.2 Typedef Documentation

8.66.2.1 CCommandRef

```
typedef CommandNode CCommandRef
```

8.67 Container Class

Collaboration diagram for Container Class:



8.68 Counter Class

Collaboration diagram for Counter Class:



Classes

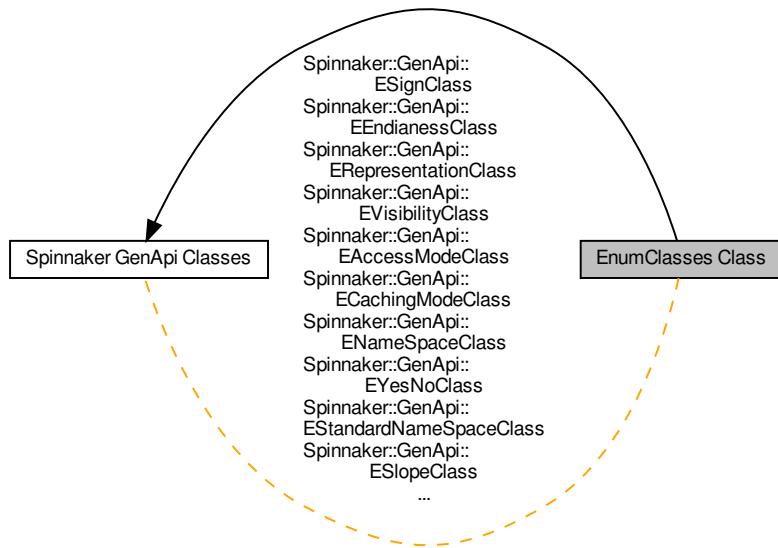
- class [Counter](#)

Definition of a simple [Counter](#) class.

8.68.1 Detailed Description

8.69 EnumClasses Class

Collaboration diagram for EnumClasses Class:



Classes

- 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

8.69.1 Detailed Description

8.70 EnumEntryNode Class

Collaboration diagram for EnumEntryNode Class:



Classes

- class [EnumEntryNode](#)
Interface for string properties.

Typedefs

- [typedef EnumEntryNode CEnumEntryRef](#)

8.70.1 Detailed Description

8.70.2 Typedef Documentation

8.70.2.1 CEnumEntryRef

```
typedef EnumEntryNode CEnumEntryRef
```

8.71 EnumNode Class

Collaboration diagram for EnumNode Class:



Classes

- class [EnumNode](#)
Interface for string properties.

Typedefs

- typedef [EnumNode CEnumerationRef](#)

8.71.1 Detailed Description

8.71.2 Typedef Documentation

8.71.2.1 CEnumerationRef

```
typedef EnumNode CEnumerationRef
```

8.72 EnumNodeT Class

Collaboration diagram for EnumNodeT Class:



Classes

- class [CEnumerationTRef< EnumT >](#)
Interface for string properties.

8.72.1 Detailed Description

8.73 EventAdapter Class

Collaboration diagram for EventAdapter Class:



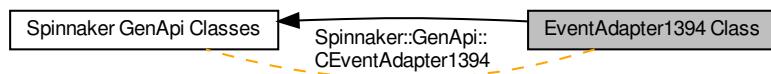
Classes

- class [CEventAdapter](#)
Delivers Events to ports.

8.73.1 Detailed Description

8.74 EventAdapter1394 Class

Collaboration diagram for EventAdapter1394 Class:



Classes

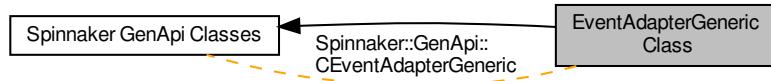
- class [CEventAdapter1394](#)

Distribute the events to the node map.

8.74.1 Detailed Description

8.75 EventAdapterGeneric Class

Collaboration diagram for EventAdapterGeneric Class:



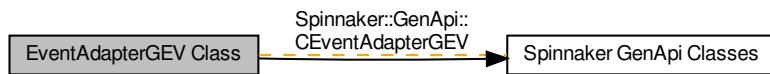
Classes

- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.

8.75.1 Detailed Description

8.76 EventAdapterGEV Class

Collaboration diagram for EventAdapterGEV Class:



Classes

- class [CEventAdapterGEV](#)
Connects a GigE [Event](#) to a node map.

8.76.1 Detailed Description

8.77 EventAdapterU3V Class

Collaboration diagram for EventAdapterU3V Class:



Classes

- class [CEventAdapterU3V](#)
Connects a U3V [Event](#) to a node map.

8.77.1 Detailed Description

8.78 EventPort Class

Collaboration diagram for EventPort Class:



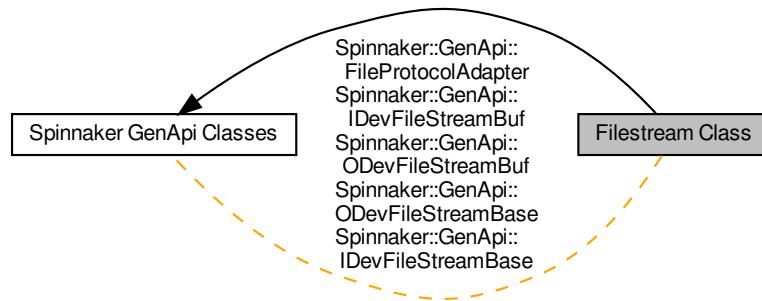
Classes

- class [CEventPort](#)
Port attachable to an event.

8.78.1 Detailed Description

8.79 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 >](#)

8.79.1 Detailed Description

8.80 FloatNode Class

Collaboration diagram for FloatNode Class:



Classes

- class [FloatNode](#)
Interface for string properties.

Typedefs

- typedef [FloatNode CFloatRef](#)

8.80.1 Detailed Description

8.80.2 Typedef Documentation

8.80.2.1 CFloatRef

```
typedef FloatNode CFloatRef
```

8.81 FloatRegNode Class

Collaboration diagram for FloatRegNode Class:



Classes

- class [FloatRegNode](#)
Interface for string properties.

8.81.1 Detailed Description

8.82 GCString Class

Collaboration diagram for GCString Class:



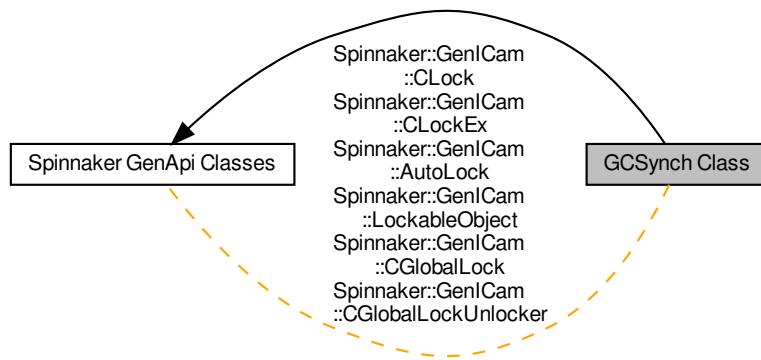
Classes

- class [gcstring](#)

8.82.1 Detailed Description

8.83 GCSynch Class

Collaboration diagram for GCSynch Class:



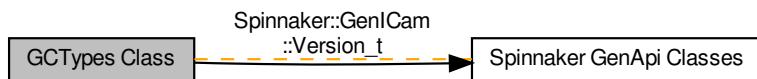
Classes

- 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.

8.83.1 Detailed Description

8.84 GCTypes Class

Collaboration diagram for GCTypes Class:



Classes

- struct [Version_t](#)
Version.

TypeDefs

- typedef float [float32_t](#)
32 bit floating point
- typedef double [float64_t](#)
64 bit floating point

8.84.1 Detailed Description

8.84.2 Typedef Documentation

8.84.2.1 [float32_t](#)

```
typedef float float32\_t
```

32 bit floating point

8.84.2.2 [float64_t](#)

```
typedef double float64\_t
```

64 bit floating point

8.85 Spinnaker GenApi Utilities

Collaboration diagram for Spinnaker GenApi Utilities:



Modules

- [GCUtilities Utility](#)

8.85.1 Detailed Description

8.86 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= " ")`

- **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.

8.86.1 Detailed Description

8.86.2 Function Documentation

8.86.2.1 DoesEnvironmentVariableExist()

```
SPINNAKER_API bool Spinnaker::GenICam::DoesEnvironmentVariableExist (
    const Spinnaker::GenICam::gcstring & VariableName )
```

Returns true if an environment variable exists.

8.86.2.2 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

8.86.2.3 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.

8.86.2.4 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.

8.86.2.5 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_CONF_IG_Vx_y. If this environment variable does not exist, an exception will be thrown.

8.86.2.6 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

8.86.2.7 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
--------------------------	--------------

8.86.2.8 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

8.86.2.9 INTEGRAL_CAST()

```
T Spinnaker::GenICam::INTEGRAL_CAST (
    int64_t ll ) [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)

8.86.2.10 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)

8.86.2.11 ReplaceEnvironmentVariables()

```
SPINNAKER_API void Spinnaker::GenICam::ReplaceEnvironmentVariables (
    gcstring & Buffer,
    bool ReplaceBlankBy20 = false )
```

Replaces in a string and replace '' with %20.

8.86.2.12 SetGenICamCacheFolder()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamCacheFolder (
    const gcstring & path )
```

Stores the path of the [GenICam](#) cache folder.

8.86.2.13 SetGenICamCLProtocolFolder()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamCLProtocolFolder (
    const gcstring & path )
```

Stores the path of the CLProtocol folder.

8.86.2.14 SetGenICamLogConfig()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamLogConfig (
    const gcstring & path )
```

Stores the path of the [GenICam](#) logging properties file.

8.86.2.15 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

8.86.2.16 UrlDecode()

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlDecode (
    const gcstring & Input )
```

Replaces xx escapes by their char equivalent.

8.86.2.17 UrlEncode()

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlEncode (
    const gcstring & Input )
```

Converts \ to / and replaces all unsafe characters by their xx equivalent.

8.87 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

8.87.1 Detailed Description

8.87.2 Function Documentation

8.87.2.1 GetValue()

```
GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) const [pure virtual]
```

Get node value.

Parameters

<code>Verify</code>	Enables Range verification (default = false). The AccessMode is always checked
<code>IgnoreCache</code>	If true the value is read ignoring any caches (default = false)

Returns

The value read

8.87.2.2 operator()()

```
GenICam::gcstring operator() ( ) const [virtual]
```

Get node value.

Execute the command.

8.87.2.3 operator=()

```
virtual void Spinnaker::GenApi::operator= (
    bool Value ) [virtual]
```

Set node value.

8.87.3 Variable Documentation**8.87.3.1 IBoolean**

```
interface SPINNAKER_API_ABSTRACT IBoolean
```

Interface for Boolean properties.

8.87.3.2 Verify

```
interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0
```

8.88 ICategory Interfaces

Collaboration diagram for ICategory Interfaces:



Variables

- [interface SPINNAKER_API_ABSTRACT ICategory](#)
Gives access to a category node.

8.88.1 Detailed Description

8.88.2 Variable Documentation

8.88.2.1 ICategory

```
interface SPINNAKER_API_ABSTRACT ICategory
```

Gives access to a category node.

8.89 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.

8.89.1 Detailed Description

8.89.2 Macro Definition Documentation

8.89.2.1 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)

8.89.2.2 CHUNK_BASE_ADDRESS_REGISTER_LEN

```
#define CHUNK_BASE_ADDRESS_REGISTER_LEN 8
```

Length of the CHUNK_BASE_ADDRESS_REGISTER pseudo register.

8.89.2.3 CHUNK_LENGTH_REGISTER

```
#define CHUNK_LENGTH_REGISTER (GC_INT64_MAX - 15)
```

Address of a int64_t pseudo register containing the length of the chunk.

8.89.2.4 CHUNK_LENGTH_REGISTER_LEN

```
#define CHUNK_LENGTH_REGISTER_LEN 8
```

Length of the CHUNK_LENGTH_REGISTER pseudo register.

8.89.3 Function Documentation

8.89.3.1 CacheChunkData()

```
virtual EYesNo Spinnaker::GenApi::CacheChunkData( ) const [pure virtual]
```

Indicates if the chunk adapter must hold a cached version of the chunk data.

8.89.4 Variable Documentation

8.89.4.1 IChunkPort

```
interface SPINNAKER_API_ABSTRACT IChunkPort
```

Interface for ports attached to a chunk.

8.90 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.

8.90.1 Detailed Description

8.90.2 Function Documentation

8.90.2.1 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

8.90.3 Variable Documentation

8.90.3.1 ICommand

```
interface SPINNAKER_API_ABSTRACT ICommand
```

[Interface](#) for command like properties.

8.91 IDestroy Interface

Collaboration diagram for IDestroy Interface:



Variables

- interface SPINNAKER_API_ABSTRACT IDestroy
Interface to destroy an object.

8.91.1 Detailed Description

8.91.2 Variable Documentation

8.91.2.1 IDestroy

```
interface SPINNAKER_API_ABSTRACT IDestroy
```

Initial value:

```
{  
    virtual void Destroy() = 0
```

Interface to destroy an object.

8.92 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)

8.92.1 Detailed Description

8.92.2 Function Documentation

8.92.2.1 GetDeviceVersion()

```
virtual void Spinnaker::GenApi::GetDeviceVersion (
    GenICam::Version_t & Version ) [pure virtual]
```

Get the version of the device description file.

8.92.2.2 GetGenApiVersion()

```
virtual void Spinnaker::GenApi::GetGenApiVersion (
    GenICam::Version_t & Version,
    uint16_t & Build ) [pure virtual]
```

Get the version of the DLL's [GenApi](#) implementation.

8.92.2.3 GetProductGuid()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetProductGuid () [pure virtual]
```

Get the Guid describing the product.

8.92.2.4 GetSchemaVersion()

```
virtual void Spinnaker::GenApi::GetSchemaVersion (
    GenICam::Version_t & Version ) [pure virtual]
```

Get the schema version number.

8.92.2.5 GetStandardNameSpace()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetStandardNameSpace () [pure virtual]
```

Get the standard name space.

8.92.2.6 GetToolTip()

```
GenICam::gcstring GetToolTip () [pure virtual]
```

Get tool tip.

Get a short description of the node.

8.92.2.7 GetVendorName()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetVendorName () [pure virtual]
```

Get the vendor name.

8.92.2.8 GetVersionGuid()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetVersionGuid( ) [pure virtual]
```

Get the Guid describing the product version.

8.92.3 Variable Documentation

8.92.3.1 IDeviceInfo

```
interface SPINNAKER_API_ABSTRACT IDeviceInfo
```

Initial value:

```
{  
    virtual GenICam::gcstring GetModelName() = 0
```

[Interface](#) to get information about the device (= nodemap)

8.93 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.

8.93.1 Detailed Description

8.93.2 Function Documentation

8.93.2.1 GetNumericValue()

```
virtual double Spinnaker::GenApi::GetNumericValue ( ) [pure virtual]
```

Get double number associated with the entry.

8.93.2.2 GetSymbolic()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetSymbolic ( ) const [pure virtual]
```

Get symbolic enum value.

8.93.2.3 IsSelfClearing()

```
virtual bool Spinnaker::GenApi::IsSelfClearing() [pure virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

8.93.3 Variable Documentation

8.93.3.1 IEnumEntry

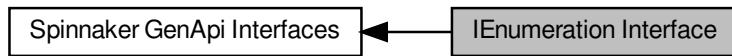
```
interface SPINNAKER_API_ABSTRACT IEnumEntry
```

[Interface](#) of single enum value.

Maps of Enum Values to symbolic values

8.94 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.

8.94.1 Detailed Description

8.94.2 Function Documentation

8.94.2.1 GetCurrentEntry()

```
IEnumEntry * GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
```

Get the current entry.

8.94.2.2 GetEntries()

```
virtual void Spinnaker::GenApi::GetEntries (
    NodeList_t & Entries ) [pure virtual]
```

Get list of entry nodes.

8.94.2.3 GetEntry()

```
IEnumEntry * GetEntry (
    const int64_t IntValue ) [pure virtual]
```

Get an entry node by its IntValue.

8.94.2.4 GetEntryByName()

```
virtual IEnumEntry* Spinnaker::GenApi::GetEntryByName (
    const GenICam::gcstring & Symbolic ) [pure virtual]
```

Get an entry node by name.

8.94.2.5 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

8.94.2.6 operator*()

```
GenICam::gcstring operator* ( ) [pure virtual]
```

Get string node value.

Get node value.

8.94.2.7 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)

8.94.3 Variable Documentation

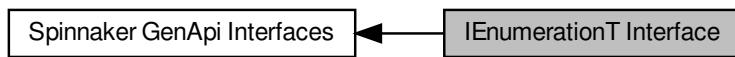
8.94.3.1 IEnumeration

```
interface SPINNAKER_API_ABSTRACT IEnumeration
```

[Interface](#) for enumeration properties.

8.95 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 GenlCam::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.

8.95.1 Detailed Description

8.95.2 Function Documentation

8.95.2.1 GetEntry()

```
virtual IEnumEntry* Spinnaker::GenApi::GetEntry (
    const EnumT Value ) [pure virtual]
```

returns the EnumEntry object belonging to the Value

8.95.2.2 operator=() [1/2]

```
virtual IEnumeration& Spinnaker::GenApi::operator= (
    EnumT Value ) [pure virtual]
```

Set node value.

8.95.2.3 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

8.95.3 Variable Documentation

8.95.3.1 IEnumerationT

```
interface SPINNAKER_API_ABSTRACT IEnumerationT
```

Interface for enumeration properties.

8.95.3.2 IEnumReference

```
interface SPINNAKER_API_ABSTRACT IEnumReference
```

Initial value:

```
{  
    virtual void SetValue(EnumT Value, bool Verify = true) = 0
```

Interface to construct an enum reference.

8.96 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.

8.96.1 Detailed Description

8.96.2 Function Documentation

8.96.2.1 GetDisplayNotation()

```
virtual EDisplayNotation Spinnaker::GenApi::GetDisplayNotation ( ) const [pure virtual]
```

Get the way the float should be converted to a string.

8.96.2.2 GetDisplayPrecision()

```
virtual int64_t Spinnaker::GenApi::GetDisplayPrecision ( ) const [pure virtual]
```

Get the precision to be used when converting the float to a string.

8.96.2.3 GetInc()

```
int64_t GetInc ( ) [pure virtual]
```

Get the constant increment if there is any.

Get increment.

8.96.2.4 GetIncMode()

```
EIncMode GetIncMode ( ) [pure virtual]
```

Get increment mode.

8.96.2.5 GetListOfValidValues()

```
int64_autovector_t GetListOfValidValues (
    bool bounded = true ) [pure virtual]
```

Get list of valid value.

8.96.2.6 GetMax()

```
int64_t GetMax ( ) [pure virtual]
```

Get maximum value allowed.

8.96.2.7 GetMin()

```
int64_t GetMin ( ) [pure virtual]
```

Get minimum value allowed.

8.96.2.8 GetRepresentation()

```
ERepresentation GetRepresentation ( ) [pure virtual]
```

Get recommended representation.

8.96.2.9 GetUnit()

```
GenICam::gcstring GetUnit ( ) const [pure virtual]
```

Get the physical unit name.

8.96.2.10 HasInc()

```
virtual bool Spinnaker::GenApi::HasInc ( ) [pure virtual]
```

True if the float has a constant increment.

8.96.2.11 ImposeMax()

```
virtual void Spinnaker::GenApi::ImposeMax ( double Value ) [pure virtual]
```

Restrict maximum value.

8.96.2.12 ImposeMin()

```
virtual void Spinnaker::GenApi::ImposeMin (
    double Value ) [pure virtual]
```

Restrict minimum value.

8.96.2.13 operator=()

```
virtual IFloat& Spinnaker::GenApi::operator= (
    double Value ) [pure virtual]
```

Set node value.

8.96.3 Variable Documentation

8.96.3.1 IFloat

```
interface SPINNAKER_API_ABSTRACT IFloat
```

Interface for float properties.

8.97 IInteger Interface

Collaboration diagram for IInteger Interface:



Functions

- virtual `IInteger & 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 IInteger`
Interface for integer properties.

8.97.1 Detailed Description

8.97.2 Function Documentation

8.97.2.1 ImposeMax()

```
virtual void Spinnaker::GenApi::ImposeMax (
    int64_t Value ) [pure virtual]
```

Restrict maximum value.

8.97.2.2 ImposeMin()

```
virtual void Spinnaker::GenApi::ImposeMin (
    int64_t Value ) [pure virtual]
```

Restrict minimum value.

8.97.2.3 operator=()

```
virtual IIInteger& Spinnaker::GenApi::operator= (
    int64_t Value ) [pure virtual]
```

Set node value.

8.97.3 Variable Documentation

8.97.3.1 IInteger

```
interface SPINNAKER_API_ABSTRACT IIInteger
```

[Interface](#) for integer properties.

8.98 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 `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.

8.98.1 Detailed Description

8.98.2 Function Documentation

8.98.2.1 `Combine()` [1/3]

```
EAccessMode Spinnaker::GenApi::Combine (
    EAccessMode Peter,
    EAccessMode Paul ) [inline]
```

Computes which access mode the two guards allow together.

8.98.2.2 `Combine()` [2/3]

```
EVisibility Spinnaker::GenApi::Combine (
    EVisibility Peter,
    EVisibility Paul ) [inline]
```

Computes which visibility the two guards allow together.

8.98.2.3 `Combine()` [3/3]

```
ECachingMode Spinnaker::GenApi::Combine (
    ECachingMode Peter,
    ECachingMode Paul ) [inline]
```

Computes which CachingMode results from a combination.

8.98.2.4 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

8.98.2.5 GetAlias()

```
virtual INode* Spinnaker::GenApi::GetAlias ( ) const [pure virtual]
```

Retrieves the a node which describes the same feature in a different way.

8.98.2.6 GetCachingMode()

```
virtual ECachingMode Spinnaker::GenApi::GetCachingMode ( ) const [pure virtual]
```

Get Caching Mode.

8.98.2.7 GetCastAlias()

```
virtual INode* Spinnaker::GenApi::GetCastAlias ( ) const [pure virtual]
```

Retrieves the a node which describes the same feature so that it can be casted.

8.98.2.8 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

<code>out</code>	<code>Children</code>	List of children nodes
	<code>LinkType</code>	The link type

8.98.2.9 GetDescription()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDescription ( ) const [pure virtual]
```

Get a long description of the node.

8.98.2.10 GetDisplayName()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDisplayName ( ) const [pure virtual]
```

Get a name string for display.

8.98.2.11 GetDocuURL()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDocuURL ( ) const [pure virtual]
```

Gets a URL pointing to the documentation of that feature.

8.98.2.12 GetEventID()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetEventID ( ) const [pure virtual]
```

Get the EventId of the node.

8.98.2.13 GetNameSpace()

```
virtual GenApi::ENamespace Spinnaker::GenApi::GetNameSpace ( ) const [pure virtual]
```

Get name space.

8.98.2.14 GetNodeMap()

```
virtual INodeMap* Spinnaker::GenApi::GetNodeMap ( ) const [pure virtual]
```

Retrieves the central node map.

8.98.2.15 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
-----	----------------	----------------------

8.98.2.16 GetPollingTime()

```
virtual int64_t Spinnaker::GenApi::GetPollingTime ( ) const [pure virtual]
```

recommended polling time (for non-cacheable nodes)

8.98.2.17 GetPrincipalInterfaceType()

```
virtual EInterfaceType Spinnaker::GenApi::GetPrincipalInterfaceType ( ) const [pure virtual]
```

Get the type of the main interface of a node.

8.98.2.18 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.

8.98.2.19 GetPropertyNames()

```
virtual void Spinnaker::GenApi::GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [pure virtual]
```

Returns a list of the names all properties set during initialization.

8.98.2.20 GetVisibility()

```
virtual EVisibility Spinnaker::GenApi::GetVisibility ( ) const [pure virtual]
```

Get the recommended visibility of the node.

8.98.2.21 ImposeAccessMode()

```
virtual void Spinnaker::GenApi::ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [pure virtual]
```

Imposes an access mode to the natural access mode of the node.

8.98.2.22 ImposeVisibility()

```
virtual void Spinnaker::GenApi::ImposeVisibility (
    EVisibility ImposedVisibility ) [pure virtual]
```

Imposes a visibility to the natural visibility of the node.

8.98.2.23 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

8.98.2.24 IsAccessModeCacheable()

```
virtual EYesNo Spinnaker::GenApi::IsAccessModeCacheable ( ) const [pure virtual]
```

True if the AccessMode can be cached.

8.98.2.25 IsAvailable() [1/3]

```
bool Spinnaker::GenApi::IsAvailable (
    EAccessMode AccessMode ) [inline]
```

Tests if available.

8.98.2.26 IsAvailable() [2/3]

```
bool Spinnaker::GenApi::IsAvailable (
    const IBase * p ) [inline]
```

Checks if a node is available.

8.98.2.27 IsAvailable() [3/3]

```
bool Spinnaker::GenApi::IsAvailable (
    const IBase & r ) [inline]
```

Checks if a node is available.

8.98.2.28 IsCachable()

```
virtual bool Spinnaker::GenApi::IsCachable () const [pure virtual]
```

Is the node value cacheable.

8.98.2.29 IsCacheable()

```
bool Spinnaker::GenApi::IsCacheable (
    ECachingMode CachingMode ) [inline]
```

Tests Cacheability.

8.98.2.30 IsDeprecated()

```
virtual bool Spinnaker::GenApi::IsDeprecated () const [pure virtual]
```

True if the node should not be used any more.

8.98.2.31 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".

8.98.2.32 IsImplemented() [1/3]

```
bool Spinnaker::GenApi::IsImplemented (
    EAccessMode AccessMode ) [inline]
```

Tests if implemented.

8.98.2.33 IsImplemented() [2/3]

```
bool Spinnaker::GenApi::IsImplemented (
    const IBase * p ) [inline]
```

Checks if a node is implemented.

8.98.2.34 IsImplemented() [3/3]

```
bool Spinnaker::GenApi::IsImplemented (
    const IBase & r ) [inline]
```

Checks if a node is implemented.

8.98.2.35 IsReadable() [1/3]

```
bool Spinnaker::GenApi::IsReadable (
    EAccessMode AccessMode ) [inline]
```

Tests if readable.

8.98.2.36 IsReadable() [2/3]

```
bool Spinnaker::GenApi::IsReadable (
    const IBase * p ) [inline]
```

Checks if a node is readable.

8.98.2.37 IsReadable() [3/3]

```
bool Spinnaker::GenApi::IsReadable (
    const IBase & r ) [inline]
```

Checks if a node is readable.

8.98.2.38 IsStreamable()

```
virtual bool Spinnaker::GenApi::IsStreamable () const [pure virtual]
```

True if the node is streamable.

8.98.2.39 IsVisible()

```
bool Spinnaker::GenApi::IsVisible (
    EVisibility Visibility,
    EVisibility MaxVisibility ) [inline]
```

Tests Visibility CAVE : this relies on the EVisibility enum's coding.

8.98.2.40 IsWritable() [1/3]

```
bool Spinnaker::GenApi::IsWritable (
    EAccessMode AccessMode ) [inline]
```

Tests if writable.

8.98.2.41 IsWritable() [2/3]

```
bool Spinnaker::GenApi::IsWritable (
    const IBase * p ) [inline]
```

Checks if a node is writable.

8.98.2.42 IsWritable() [3/3]

```
bool Spinnaker::GenApi::IsWritable (
    const IBase & r ) [inline]
```

Checks if a node is writable.

8.98.2.43 operator"!=()

```
virtual bool Spinnaker::GenApi::operator!= (
    int nullPtr ) const [pure virtual]
```

8.98.2.44 operator==()

```
virtual bool Spinnaker::GenApi::operator== (
    int nullPtr ) const [pure virtual]
```

8.98.2.45 RegisterCallback()

```
virtual CallbackHandleType Spinnaker::GenApi::RegisterCallback (  
    CNodeCallback * pCallback ) [pure virtual]
```

Register change callback Takes ownership of the **CNodeCallback** object.

8.98.3 Variable Documentation

8.98.3.1 INode

```
interface SPINNAKER_API_ABSTRACT INode
```

Interface common to all nodes.

8.98.3.2 IReference

```
interface SPINNAKER_API_ABSTRACT IReference
```

Initial value:

```
{  
    virtual GenICam::gcstring GetName(bool FullQualified = false) const = 0
```

Interface to construct a reference.

8.99 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.

8.99.1 Detailed Description

8.99.2 Function Documentation

8.99.2.1 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.

8.99.2.2 Connect() [2/2]

```
virtual bool Spinnaker::GenApi::Connect (
    IPort * pPort ) const [pure virtual]
```

Connects a port to the standard port "Device".

8.99.2.3 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".

8.99.2.4 GetLock()

```
virtual Clock& Spinnaker::GenApi::GetLock () const [pure virtual]
```

Returns the lock which guards the node map.

8.99.2.5 GetNode()

```
virtual INode* Spinnaker::GenApi::GetNode (
    const GenICam::gcstring & Name ) const [pure virtual]
```

Retrieves the node from the central map by Name.

8.99.2.6 GetNumNodes()

```
virtual uint64_t Spinnaker::GenApi::GetNumNodes( ) const [pure virtual]
```

Get the number of nodes in the map.

8.99.2.7 InvalidateNodes()

```
virtual void Spinnaker::GenApi::InvalidateNodes( ) const [pure virtual]
```

Invalidates all nodes.

8.99.2.8 Poll()

```
virtual void Spinnaker::GenApi::Poll( int64_t ElapsedTime ) [pure virtual]
```

Fires nodes which have a polling time.

8.99.3 Variable Documentation

8.99.3.1 INodeMap

```
interface SPINNAKER_API_ABSTRACT INodeMap
```

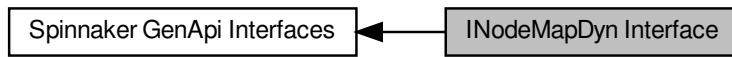
Initial value:

```
{  
    virtual void GetNodes(NodeList_t & Nodes) const = 0
```

Interface to access the node map.

8.100 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=[xvDefault](#))=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.

8.100.1 Detailed Description

8.100.2 Function Documentation

8.100.2.1 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.

8.100.2.2 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>" were <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

8.100.2.3 LoadXMLFromFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromFile (
    const GenICam::gcstring & FileName ) [pure virtual]
```

Loads an XML from a file.

8.100.2.4 LoadXMLFromFileInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName ) [pure virtual]
```

Loads an XML from a file with injection.

8.100.2.5 LoadXMLFromString()

```
virtual void Spinnaker::GenApi::LoadXMLFromString (
    const GenICam::gcstring & XMLData ) [pure virtual]
```

Loads an XML from a string.

8.100.2.6 LoadXMLFromStringInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData,
    const GenICam::gcstring & InjectXMLData ) [pure virtual]
```

Loads an XML from a string with injection.

8.100.2.7 LoadXMLFromZIPData()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize ) [pure virtual]
```

Loads an XML from a ZIP data buffer.

8.100.2.8 LoadXMLFromZIPFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName ) [pure virtual]
```

Loads an XML from a ZIP file.

8.100.2.9 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

8.100.2.10 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

8.100.2.11 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>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

8.100.3 Variable Documentation

8.100.3.1 INodeMapDyn

```
interface SPINNAKER_API_ABSTRACT INodeMapDyn
```

[Interface](#) to access the node map.

8.101 IntegerNode Class

Collaboration diagram for IntegerNode Class:



Classes

- class [IntegerNode](#)
Interface for string properties.

Typedefs

- typedef [IntegerNode CIntegerRef](#)

8.101.1 Detailed Description

8.101.2 Typedef Documentation

8.101.2.1 CIntegerRef

```
typedef IntegerNode CIntegerRef
```

8.102 IntRegNode Class

Collaboration diagram for IntRegNode Class:



Classes

- class [IntRegNode](#)
Interface for string properties.

8.102.1 Detailed Description

8.103 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](#)

8.103.1 Detailed Description

8.103.2 Function Documentation

8.103.2.1 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.

8.103.3 Variable Documentation

8.103.3.1 Address

```
interface SPINNAKER_API_ABSTRACT int64_t Address
```

8.103.3.2 IPort

```
interface SPINNAKER_API_ABSTRACT IPort
```

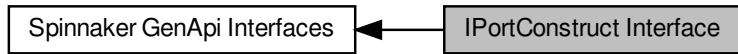
Interface for ports.

8.103.3.3 Length

```
interface SPINNAKER_API_ABSTRACT int64_t Length = 0
```

8.104 IPortConstruct Interface

Collaboration diagram for IPortConstruct Interface:



Functions

- virtual EYesNo GetSwapEndianess ()=0
Determines if the port adapter must perform an endianess swap.

Variables

- interface SPINNAKER_API IPortConstruct
Interface for ports.

8.104.1 Detailed Description

8.104.2 Function Documentation

8.104.2.1 GetSwapEndianess()

```
virtual EYesNo Spinnaker:::GenApi::GetSwapEndianess ( ) [pure virtual]
```

Determines if the port adapter must perform an endianess swap.

8.104.3 Variable Documentation

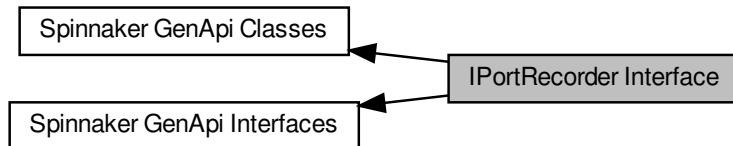
8.104.3.1 IPortConstruct

```
interface SPINNAKER_API IPortConstruct
```

Interface for ports.

8.105 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.

8.105.1 Detailed Description

8.105.2 Function Documentation

8.105.2.1 GetCookie()

```
virtual int64_t Spinnaker::GenApi::GetCookie ( ) [pure virtual]
```

Gets the cookie a port implementation may have set for caching a command list.

8.105.2.2 Replay()

```
virtual void Spinnaker::GenApi::Replay (
    IPort * pPort ) [pure virtual]
```

Replays the write command to the given port interface.

8.105.2.3 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.

8.105.2.4 StopRecording()

```
virtual void Spinnaker::GenApi::StopRecording ( ) [pure virtual]
```

Stops recording.

8.105.3 Variable Documentation

8.105.3.1 Invalidate

```
interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0
```

8.105.3.2 IPortRecorder

```
interface SPINNAKER_API_ABSTRACT IPortRecorder
```

Interface for recording write commands on a port.

8.105.3.3 IPortReplay

```
interface SPINNAKER_API_ABSTRACT IPortReplay
```

Interface for replaying write commands on a port.

8.105.3.4 IPortWriteList

```
interface SPINNAKER_API_ABSTRACT IPortWriteList
```

Initial value:

```
{  
    virtual void Write(const void* pBuffer, int64_t Address, int64_t  
Length) = 0
```

8.106 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.

8.106.1 Detailed Description

8.106.2 Function Documentation

8.106.2.1 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)

Generated by Doxygen

Returns

The value read

8.106.2.2 GetAddress()

```
virtual int64_t Spinnaker::GenApi::GetAddress () [pure virtual]
```

Retrieves the Address of the register.

8.106.2.3 GetLength()

```
virtual int64_t Spinnaker::GenApi::GetLength () [pure virtual]
```

Retrieves the Length of the register [Bytes].

8.106.3 Variable Documentation**8.106.3.1 IRegister**

```
interface SPINNAKER_API_ABSTRACT IRegister
```

[Interface](#) for registers.

8.107 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.

8.107.1 Detailed Description

8.107.2 Function Documentation

8.107.2.1 GetSelectedFeatures()

```
virtual void Spinnaker::GenApi::GetSelectedFeatures (
    FeatureList_t & ) const [pure virtual]
retrieve the group of selected features
```

8.107.2.2 GetSelectingFeatures()

```
virtual void Spinnaker::GenApi::GetSelectingFeatures (
    FeatureList_t & ) const [pure virtual]
retrieve the group of features selecting this node
```

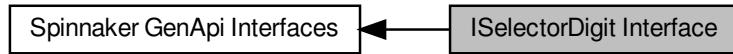
8.107.3 Variable Documentation

8.107.3.1 ISelector

```
interface SPINNAKER_API_ABSTRACT ISelector
Interface for groups of features selected by a single one.
```

8.108 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.

8.108.1 Detailed Description

8.108.2 Function Documentation

8.108.2.1 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 selector changed since the last GetNext are contained
--------------------	--

8.108.2.2 Restore()

```
virtual void Spinnaker::GenApi::Restore ( ) [pure virtual]
```

Restores the selectors' values found at creation.

8.108.2.3 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

8.108.2.4 ToString()

```
virtual GenICam::gcstring Spinnaker::GenApi::ToString ( ) [pure virtual]
```

Returns a string representation of the digit.

8.108.3 Variable Documentation

8.108.3.1 ISelectorDigit

```
interface SPINNAKER_API_ABSTRACT ISelectorDigit
```

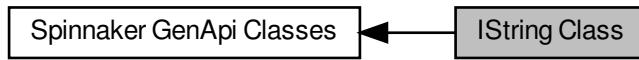
Initial value:

```
{  
    virtual bool SetFirst() = 0
```

Interface of a "digit" of the "counter" formed by the selector set.

8.109 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.

8.109.1 Detailed Description

8.109.2 Function Documentation

8.109.2.1 GetMaxLength()

```
virtual int64_t Spinnaker::GenApi::GetMaxLength ( ) [pure virtual]
```

Retrieves the maximum length of the string in bytes.

8.109.3 Variable Documentation

8.109.3.1 IString

```
interface SPINNAKER_API_ABSTRACT IString
```

Interface for string properties.

8.110 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.

8.110.1 Detailed Description

8.110.2 Function Documentation

8.110.2.1 FromString()

```

virtual void Spinnaker::GenApi::FromString (
    const GenICam::gcstring & ValueStr,
    bool Verify = true ) [pure virtual]
  
```

Set content of the node as string.

Parameters

<code>ValueStr</code>	The value to set
<code>Verify</code>	Enables AccessMode and Range verification (default = true)

8.110.2.2 IsValueCacheValid()

```
virtual bool Spinnaker::GenApi::IsValueCacheValid ( ) const [pure virtual]
```

Checks if the value comes from cache or is requested from another node.

8.110.2.3 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

8.110.3 Variable Documentation

8.110.3.1 IValue

```
interface SPINNAKER_API_ABSTRACT IValue
```

Interface for value properties.

8.111 Node Class

Collaboration diagram for Node Class:



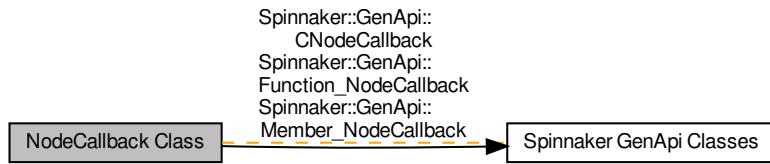
Classes

- class [Node](#)
class common to all nodes

8.111.1 Detailed Description

8.112 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.

8.112.1 Detailed Description

8.112.2 Enumeration Type Documentation

8.112.2.1 ECallbackType

enum `ECallbackType`

the type of callback

Enumerator

<code>cbPostInsideLock</code>	
<code>cbPostOutsideLock</code>	callback is fired on leaving the tree inside the lock-guarded area

8.112.3 Function Documentation

8.112.3.1 Deregister()

```
SPINNAKER_API void Spinnaker::GenApi::Deregister (
    GenApi::CallbackHandleType pCallbackInfo )
```

Unregistering callback by handle.

8.112.3.2 make_NodeCallback() [1/2]

```
CNodeCallback* Spinnaker::GenApi::make_NodeCallback (
    INode * pNode,
    Function function,
    ECallbackType CallbackType )
```

make a new callback object for C functions

8.112.3.3 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

8.112.3.4 Register() [1/2]

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Function f,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C-function as a callback.

8.112.3.5 Register() [2/2]

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Client & c,
    Member m,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C++-member function a callback.

8.113 NodeMap Class

Collaboration diagram for NodeMap Class:



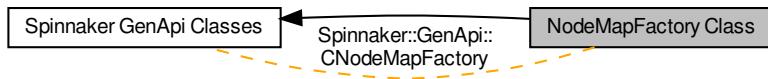
Classes

- class [NodeMap](#)
Smart pointer template for NodeMaps with create function.

8.113.1 Detailed Description

8.114 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.*

8.114.1 Detailed Description

8.114.2 Enumeration Type Documentation

8.114.2.1 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.

8.114.2.2 EContentType_t

```
enum EContentType_t
```

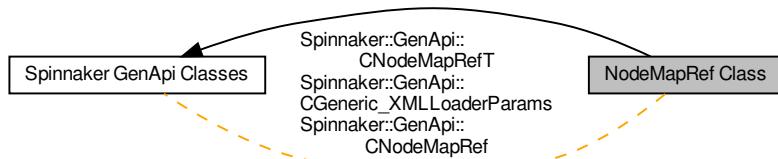
Lists the processable file types.

Enumerator

ContentType_Xml	XML camera description file text.
ContentType_ZippedXml	Zipped XML camera description file text.

8.115 NodeMapRef Class

Collaboration diagram for NodeMapRef Class:



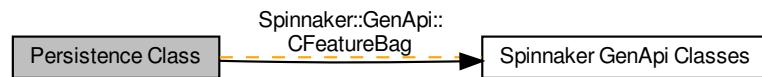
Classes

- 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.

8.115.1 Detailed Description

8.116 Persistence Class

Collaboration diagram for Persistence Class:



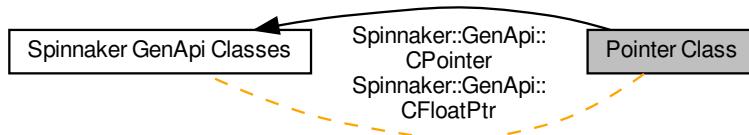
Classes

- class [CFeatureBag](#)
Bag holding streamable features of a nodetree.

8.116.1 Detailed Description

8.117 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< IPORtrecord > CPortRecorderPtr**

- `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 IDDeviceInfo 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.

8.117.1 Detailed Description

8.117.2 Typedef Documentation

8.117.2.1 CBasePtr

```
typedef CPointer<IBase> CBasePtr
```

SmartPointer for IBase interface pointer.

8.117.2.2 CBooleanPtr

```
typedef CPointer<IBoolean> CBooleanPtr
```

SmartPointer for IBoolean interface pointer.

8.117.2.3 CCategoryPtr

```
typedef CPointer<ICategory> CCategoryPtr
```

SmartPointer for ICategory interface pointer.

8.117.2.4 CChunkPortPtr

```
typedef CPointer<IChunkPort> CChunkPortPtr
```

SmartPointer for IChunkPort interface pointer.

8.117.2.5 CCommandPtr

```
typedef CPointer< ICommand> CCommandPtr
```

SmartPointer for ICommand interface pointer.

8.117.2.6 CDeviceInfoPtr

```
typedef CPointer<IDeviceInfo, INodeMap> CDeviceInfoPtr
```

SmartPointer for IDeviceInfo interface pointer.

8.117.2.7 CEnumEntryPtr

```
typedef CPointer<IEnumEntry> CEnumEntryPtr
```

SmartPointer for IEnumEntry interface pointer.

8.117.2.8 CEnumerationPtr

```
typedef CPointer<IEnumeration> CEnumerationPtr
```

SmartPointer for IEnumeration interface pointer.

8.117.2.9 CIIntegerPtr

```
typedef CPointer<IIInteger> CIIntegerPtr
```

SmartPointer for IIInteger interface pointer.

8.117.2.10 CNodeMapDynPtr

```
typedef CPointer<INodeMapDyn, INodeMap> CNodeMapDynPtr
```

SmartPointer for INodeMapDyn interface pointer.

8.117.2.11 CNodeMapPtr

```
typedef CPointer<INodeMap, INodeMap> CNodeMapPtr
```

SmartPointer for INodeMap interface pointer.

8.117.2.12 CNodePtr

```
typedef CPointer<INode, IBase> CNodePtr
```

SmartPointer for INode interface pointer.

8.117.2.13 CPortConstructPtr

```
typedef CPointer<IPortConstruct> CPortConstructPtr
```

SmartPointer for IPotConstruct interface pointer.

8.117.2.14 CPortPtr

```
typedef CPointer<IPort> CPortPtr
```

SmartPointer for IPo

8.117.2.15 CPortRecorderPtr

```
typedef CPointer<IPortRecorder> CPortRecorderPtr
```

SmartPointer for IPo

8.117.2.16 CPortReplayPtr

```
typedef CPointer<IPortReplay> CPortReplayPtr
```

SmartPointer for IPo

8.117.2.17 CPortWriteListPtr

```
typedef CPointer<IPortWriteList, IPo
```

SmartPointer for IPo

8.117.2.18 CRegisterPtr

```
typedef CPointer<IRegister> CRegisterPtr
```

SmartPointer for IRegis

8.117.2.19 CSelectorPtr

```
typedef CPointer<ISelector> CSelectorPtr
```

SmartPointer for ISelec

8.117.2.20 CStringPtr

```
typedef CPointer<IString> CStringPtr
```

SmartPointer for IString interface pointer.

8.117.2.21 CValuePtr

```
typedef CPointer<IValue> CValuePtr
```

SmartPointer for IValue interface pointer.

8.117.3 Function Documentation

8.117.3.1 GetInterfaceName()

```
GenICam::gcstring Spinnaker::GenApi::GetInterfaceName (
    IBase * pBase ) [inline]
```

Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

8.117.3.2 IsAvailable()

```
bool Spinnaker::GenApi::IsAvailable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Available.

8.117.3.3 IsImplemented()

```
bool Spinnaker::GenApi::IsImplemented (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Implemented.

8.117.3.4 IsReadable()

```
bool Spinnaker::GenApi::IsReadable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is readable.

8.117.3.5 IsWritable()

```
bool Spinnaker::GenApi::IsWritable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Writable.

8.118 PortImpl Class

Collaboration diagram for PortImpl Class:



Classes

- class [CPortImpl](#)

Standard implementation for a port.

8.118.1 Detailed Description

8.119 PortNode Class

Collaboration diagram for PortNode Class:



Classes

- class [PortNode](#)
Interface for value properties.

Typedefs

- [typedef PortNode CPortRef](#)

8.119.1 Detailed Description

8.119.2 Typedef Documentation

8.119.2.1 CPortRef

```
typedef PortNode CPortRef
```

8.120 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.

8.120.1 Detailed Description

8.120.2 Typedef Documentation

8.120.2.1 CPortRecorderRef

```
typedef PortRecorder CPortRecorderRef
```

Reference to an IPortRecorder pointer.

8.121 PortReplay Class

Collaboration diagram for PortReplay Class:



Classes

- class [PortReplay](#)
Interface for replaying write commands on a port.

8.121.1 Detailed Description

8.122 PortWriteList Class

Collaboration diagram for PortWriteList Class:



Classes

- class [CPortWriteList](#)
Container holding a list of port write commands.

8.122.1 Detailed Description

8.123 Reference Interfaces

Collaboration diagram for Reference Interfaces:



Functions

- virtual void [SetNumEnums](#) (int NumEnums)=0
sets the number of enum values

8.123.1 Detailed Description

8.123.2 Function Documentation

8.123.2.1 SetNumEnums()

```
virtual void Spinnaker::GenApi::SetNumEnums ( int NumEnums ) [pure virtual]
```

sets the number of enum values

8.124 RegisterNode Class

Collaboration diagram for RegisterNode Class:



Classes

- class [RegisterNode](#)
Interface for string properties.

Typedefs

- [typedef RegisterNode CRegisterRef](#)

8.124.1 Detailed Description

8.124.2 Typedef Documentation

8.124.2.1 CRegisterRef

```
typedef RegisterNode CRegisterRef
```

8.125 RegisterPortImpl Class

Collaboration diagram for RegisterPortImpl Class:



Classes

- class [CRegisterPortImpl](#)
Standard implementation for a port using a register based transport layer.

8.125.1 Detailed Description

8.126 SelectorSet Class

Collaboration diagram for SelectorSet Class:



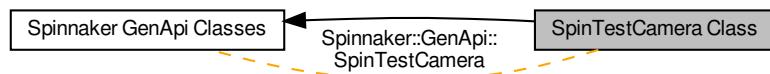
Classes

- class [CSelectorSet](#)
The set of selectors selecting a given node.

8.126.1 Detailed Description

8.127 SpinTestCamera Class

Collaboration diagram for SpinTestCamera Class:



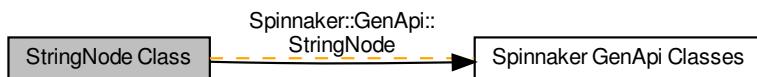
Classes

- class [SpinTestCamera](#)

8.127.1 Detailed Description

8.128 StringNode Class

Collaboration diagram for StringNode Class:



Classes

- class [StringNode](#)
Interface for string properties.

TypeDefs

- [typedef StringNode CStringRef](#)

8.128.1 Detailed Description

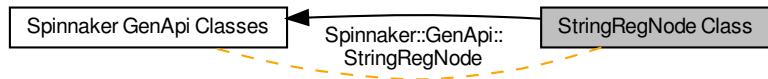
8.128.2 Typedef Documentation

8.128.2.1 CStringRef

```
typedef StringNode CStringRef
```

8.129 StringRegNode Class

Collaboration diagram for StringRegNode Class:



Classes

- class [StringRegNode](#)
Interface for string properties.

8.129.1 Detailed Description

8.130 StructPort Class

Collaboration diagram for StructPort Class:



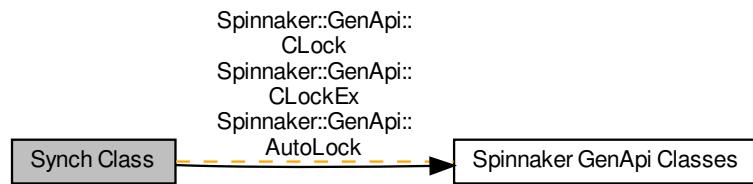
Classes

- class [CTestPortStruct< CDataStruct >](#)
Implements a register spaces based on a C++ struct.

8.130.1 Detailed Description

8.131 Synch Class

Collaboration diagram for Synch Class:



Classes

- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)

8.131.1 Detailed Description

8.132 Spinnaker GenApi Enums

Collaboration diagram for Spinnaker GenApi Enums:



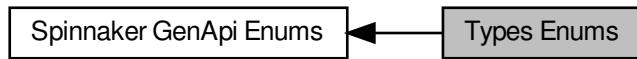
Modules

- [Types Enums](#)

8.132.1 Detailed Description

8.133 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, _UndefinedAccessMode, _CycleDetectAccessMode }`
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,

IICC,

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` = 0x8000000L }

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` {

`intf1Value`,

```
intflBase,  
intflInteger,  
intflBoolean,  
intflCommand,  
intflFloat,  
intflString,  
intflRegister,  
intflCategory,  
intflEnumeration,  
intflEnumEntry,  
intflPort }
```

typedef for interface type

- enum ELinkType {
 ctParentNodes,
 ctReadingChildren,
 ctWritingChildren,
 ctlInvalidatingChildren,
 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.

8.133.1 Detailed Description

8.133.2 Macro Definition Documentation

8.133.2.1 _UndefinedRepresentation

```
#define _UndefinedRepresentation _UndefinedRepresentation
```

8.133.3 Typedef Documentation

8.133.3.1 `StringList_t`

```
typedef GenICam::gcstring_vector StringList_t
```

A list of strings.

8.133.4 Enumeration Type Documentation

8.133.4.1 `EAccessMode`

```
enum EAccessMode
```

access mode of a node

Enumerator

NI	
NA	Not implemented.
WO	Not available.
RO	Write Only.
RW	Read Only.
_UndefinedAccessMode	Read and Write.
_CycleDetectAccessMode	Object is not yet initialized. used internally for AccessMode cycle detection

8.133.4.2 `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

8.133.4.3 `EDisplayNotation`

```
enum EDisplayNotation
```

typedef for float notation

Enumerator

fnAutomatic	
fnFixed	the notation is 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

8.133.4.4 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

8.133.4.5 EGenApiSchemaVersion

```
enum EGenApiSchemaVersion
```

GenApi schema version.

Enumerator

v1_0	
v1_1	
_Undefined	

8.133.4.6 EIIncMode

```
enum EIIncMode
```

typedef for increment mode

Enumerator

nolncrement	
fixedIncrement	The feature has no increment.
listIncrement	The feature has a fix increment.

8.133.4.7 EInputDirection

enum `EInputDirection`

typedef for link type

Enumerator

<code>idFrom</code>	
<code>idTo</code>	Indicates a swiss knife that it is used as worker for a converter computing FROM.
<code>idNone</code>	Indicates a swiss knife that it is used as worker for a converter computing TO. SwissKnife is not used within a converter

8.133.4.8 EInterfaceType

enum `EInterfaceType`

typedef for interface type

Enumerator

<code>intfIValue</code>	
<code>intfIBase</code>	IValue interface.
<code>intfIInteger</code>	IBase interface.
<code>intfIBoolean</code>	IInteger interface.
<code>intfICommand</code>	IBoolean interface.
<code>intfIFloat</code>	ICommand interface.
<code>intfIString</code>	IFloat interface.
<code>intfIRegister</code>	IString interface.
<code>intfICategory</code>	IRegister interface.
<code>intfIEnumeration</code>	ICategory interface.
<code>intfIEnumEntry</code>	IEnumeration interface.
<code>intfIPort</code>	IEnumEntry interface. IPort interface

8.133.4.9 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 node (i.e. which are dependent on this node) All indirectly connected terminal nodes

8.133.4.10 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

8.133.4.11 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.

8.133.4.12 ESign

```
enum ESign
```

signed or unsigned integers

Enumerator

Signed	
Unsigned	Integer is signed.
_UndefinedSign	Integer is unsigned. Object is not yet initialized

8.133.4.13 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

8.133.4.14 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

8.133.4.15 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.

8.133.4.16 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

8.133.4.17 EYesNo

```
enum EYesNo
```

Defines the choices of a Yes/No alternative.

Enumerator

Yes	
No	yes
_UndefinedYesNo	no

8.134 ValueNode Class

Collaboration diagram for ValueNode Class:



Classes

- class [ValueNode](#)
Interface for value properties.

Typedefs

- typedef [ValueNode CValueRef](#)

8.134.1 Detailed Description

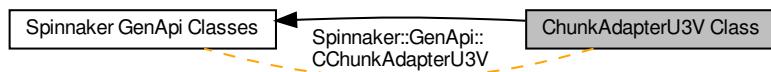
8.134.2 Typedef Documentation

8.134.2.1 CValueRef

```
typedef ValueNode CValueRef
```

8.135 ChunkAdapterU3V Class

Collaboration diagram for ChunkAdapterU3V Class:



Classes

- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.

8.135.1 Detailed Description

Chapter 9

Namespace Documentation

9.1 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,
 GATEWAY_INVALID,
 SUBNET_MASK_INVALID,
 GATEWAY_SUBNET_SAME_IP,
 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)
- [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 `GetGatewayAddress` (const std::string &ipAddrStr, unsigned int subnetMaskLength)
- **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` ()
- **ADAPTERCONFIG_API** std::string `GetAutoStartGateway` ()

9.1.1 Enumeration Type Documentation

9.1.1.1 AdapterConfigErr

enum `AdapterConfigErr`

Enumerator

IP_ADDRESS_INVALID	
IP_ADDRESS_IS_NOT_V4	
IP_ADDRESS_TOO_LARGE	
IP_ADDRESS_TOO_SMALL	
HOST_ADDRESS_ZERO	
GATEWAY_INVALID	
SUBNET_MASK_INVALID	
GATEWAY_SUBNET_SAME_IP	
VALID_SUBNET_NOT_FOUND	

9.1.2 Function Documentation

9.1.2.1 AutoPopulateAdapterInfo()

```
ADAPTERCONFIG_API void AdapterConfig::AutoPopulateAdapterInfo (
    std::vector< AdapterInfo > & adaptersToConfigure,
    const std::vector< AdapterInfo > & allAdapters )
```

9.1.2.2 AutoPopulateAdvancedProperties()

```
ADAPTERCONFIG_API void AdapterConfig::AutoPopulateAdvancedProperties (
    std::vector< AdapterInfo > & adaptersToConfigure )
```

9.1.2.3 ConfigureAdapter()

```
ADAPTERCONFIG_API void AdapterConfig::ConfigureAdapter (
    AdapterInfo & adapter,
    bool configureIP,
    bool configureAdvancedProperties )
```

9.1.2.4 GetAuto10GDesc()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAuto10GDesc ()
```

9.1.2.5 GetAutoGigabitDesc()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoGigabitDesc ()
```

9.1.2.6 GetAutoStartGateway()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoStartGateway ()
```

9.1.2.7 GetAutoStartIp()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoStartIp ()
```

9.1.2.8 GetAutoSubnetMask()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoSubnetMask ()
```

9.1.2.9 GetAutoSubnetMaskLength()

```
ADAPTERCONFIG_API unsigned int AdapterConfig::GetAutoSubnetMaskLength ( )
```

9.1.2.10 GetConfigLogFileName()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetConfigLogFileName ( )
```

9.1.2.11 GetEnumerationLogFileName()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetEnumerationLogFileName ( )
```

9.1.2.12 GetGatewayAddress()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetGatewayAddress ( 
    const std::string & ipAddrStr,
    unsigned int subnetMaskLength )
```

9.1.2.13 GetMaxIpAddress()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetMaxIpAddress ( )
```

9.1.2.14 GetMinIpAddress()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetMinIpAddress ( )
```

9.1.2.15 GetSubnetMaskLength()

```
ADAPTERCONFIG_API unsigned int AdapterConfig::GetSubnetMaskLength ( 
    const std::string & subnetMask )
```

9.1.2.16 IsOnSameSubnet()

```
ADAPTERCONFIG_API bool AdapterConfig::IsOnSameSubnet (
    const std::string & ipAddrStr1,
    const std::string & ipAddrStr2,
    const unsigned int subnetMaskLength )
```

9.1.2.17 IsValidIpAddress()

```
ADAPTERCONFIG_API bool AdapterConfig::IsValidIpAddress (
    const std::string & ipAddr )
```

9.1.2.18 IsValidSubnetMask()

```
ADAPTERCONFIG_API bool AdapterConfig::IsValidSubnetMask (
    const std::string & subnetMask )
```

9.1.2.19 PopulateAdapterIpInfo()

```
ADAPTERCONFIG_API void AdapterConfig::PopulateAdapterIpInfo (
    IpInfo startingIpInfo,
    std::vector< AdapterInfo > & adaptersToConfigure,
    const std::vector< AdapterInfo > & allAdapters )
```

9.1.2.20 RetrieveAllAdapters()

```
ADAPTERCONFIG_API std::vector<AdapterInfo> AdapterConfig::RetrieveAllAdapters ( )
```

9.1.2.21 ValidateIpAddress()

```
ADAPTERCONFIG_API void AdapterConfig::ValidateIpAddress (
    const std::string & ipAddr,
    unsigned int subnetMaskLength )
```

9.2 Spinnaker Namespace Reference

Namespaces

- [GenApi](#)
- [GenICam](#)
- [Video](#)

Classes

- struct [ActionCommandResult](#)
Action Command Result.
- class [ArrivalEvent](#)
An event handler for capturing the device arrival event.
- 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.
- class [ChunkData](#)
The chunk data which contains additional information about an image.
- class [DeviceEvent](#)
A handler to device events.
- class [Event](#)
The base class for all event types.
- class [Exception](#)
The [Exception](#) object represents an error that is returned from the library.
- class [IArrivalEvent](#)
- 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 [IDeviceEvent](#)
- class [IImage](#)
The interface file for [Image](#).
- class [IImageEvent](#)
- class [IImageStatistics](#)
The interface file for image statistics.
- class [IInterface](#)
The interface file for [Interface](#).

- class [IInterfaceArrivalEvent](#)
- class [IInterfaceEvent](#)
- class [IInterfaceList](#)

The interface file for [InterfaceList](#) class.
- class [IInterfaceRemovalEvent](#)
- class [ILoggingEvent](#)
- class [Image](#)

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

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 [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 [InterfaceArrivalEvent](#)

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

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 [InterfaceRemovalEvent](#)

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

The interface file for [System](#).
- class [ISystemEvent](#)
- struct [JPEGOption](#)

Options for saving JPEG image.
- struct [JPQ2Option](#)

Options for saving JPEG2000 image.
- struct [LibraryVersion](#)

Provides easier access to the current version of [Spinnaker](#).
- class [LoggingEvent](#)

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

- class [LoggingEventData](#)
The LoggingEventData object.
- class [LoggingEventDataPtr](#)
A reference tracked pointer to the LoggingEvent object.
- struct [PGMOption](#)
Options for saving PGM images.
- struct [PNGOption](#)
Options for saving PNG images.
- struct [PPMOption](#)
Options for saving PPM images.
- class [RemovalEvent](#)
An event handler for capturing the device removal event.
- class [System](#)
The system object is used to retrieve the list of interfaces and cameras available.
- class [SystemEvent](#)
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_DEVICEREGISTERSENDIANCESS }

• 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_DEVICETLTYPES }

• 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 FileModeEnums {
    FileMode_Read,
    FileMode_Write,
    FileMode_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 `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_DEVICESTREAMCHANNELENDIANCESS }

• 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_SCF1WRWGB,
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_GEVGVCPEXTENDEDCODESSELECTOR }

• 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_CXP CONNECTIONTESTMODE }

• enum CxpPoCxpStatusEnums {
    CxpPoCxpStatus_Auto,
    CxpPoCxpStatus_Off,
    CxpPoCxpStatus_Tripped,
    NUM_CXPOCXPSTATUS }

• 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` = 3,
`IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT` = 4,
`IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT` = 5,
`IMAGE_PACKETID_INCONSISTENT` = 6,
`IMAGE_MISSING_LEADER` = 7,
`IMAGE_MISSING_TRAILER` = 8,
`IMAGE_DATA_INCOMPLETE` = 9,
`IMAGE_INFO_INCONSISTENT` = 10,
`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 }

- 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 `StreamTypeEnum` {
 `StreamType_Mixed`,
 `StreamType_Custom`,
 `StreamType_GEV`,
 `StreamType_CL`,
 `StreamType_IIDC`,
 `StreamType_UVC`,
 `StreamType_CXP`,
 `StreamType_CLHS`,
 `StreamType_U3V`,
 `StreamType_ETHERNET`,
 `StreamType_PCI`,
 `NUMSTREAMTYPE` }

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

- enum `StreamDefaultBufferCountModeEnum` {
 `StreamDefaultBufferCountMode_Manual`,
 `StreamDefaultBufferCountMode_Auto`,
 `NUMSTREAMDEFAULTBUFFERCOUNTMODE` }
- enum `StreamBufferCountModeEnum` {
 `StreamBufferCountMode_Manual`,
 `StreamBufferCountMode_Auto`,
 `NUMSTREAMBUFFERCOUNTMODE` }
- enum `StreamBufferHandlingModeEnum` {
 `StreamBufferHandlingMode_OldestFirst`,
 `StreamBufferHandlingMode_OldestFirstOverwrite`,
 `StreamBufferHandlingMode_NewestFirst`,
 `StreamBufferHandlingMode_NewestFirstOverwrite`,
 `StreamBufferHandlingMode_NewestOnly`,
 `NUMSTREAMBUFFERHANDLINGMODE` }
- enum `DeviceTypeEnum` {
 `DeviceType_Mixed`,
 `DeviceType_Custom`,

```

DeviceType_GEV,
DeviceType_CL,
DeviceType_IIDC,
DeviceType_UVC,
DeviceType_CXP,
DeviceType_CLHS,
DeviceType_U3V,
DeviceType_ETHERNET,
DeviceType_PCI,
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 DeviceEndianessMechanismEnum {
DeviceEndianessMechanism_Legacy,
DeviceEndianessMechanism_Standard,
NUMDEVICEENDIANESSMECHANISM }

• enum DeviceCurrentSpeedEnum {
DeviceCurrentSpeed_UnknownSpeed,
DeviceCurrentSpeed_LowSpeed,
DeviceCurrentSpeed_FullSpeed,
DeviceCurrentSpeed_HighSpeed,
DeviceCurrentSpeed_SuperSpeed,
NUMDEVICECURRENTSPEED }

• enum POEStatusEnum {
POEStatus_NotSupported,
POEStatus_PowerOff,
POEStatus_PowerOn,
NUMPOESTATUS }

• enum FilterDriverStatusEnum {
FilterDriverStatus_NotSupported,
FilterDriverStatus_Disabled,
FilterDriverStatus_Enabled,
NUMFILTERDRIVERSTATUS }

```

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.

- **DEPRECATED_ENUM** ("This enum has been deprecated. Polarization images are now created through specific " "functions in the [ImageUtilityPolarization](#) class.") [PolarizationAlgorithm](#)
- **DEPRECATED_ENUM** ("This enum has been deprecated. [Image](#) scaling can now be applied through specific functions " "defined in the [ImageUtility](#) class.") [PolarizationResolution](#)

Variables

- const uint64_t [EVENT_TIMEOUT_NONE](#) = 0
Timeout values for getting next image, device, or interface event.
- const uint64_t [EVENT_TIMEOUT_INFINITE](#) = 0xFFFFFFFFFFFFFFFFF
- [HeatMapColor](#)

9.3 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_avector_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](#)

- [ERepresentationClass](#)
Interface for string properties.
- [ESignClass](#)
Holds conversion methods for the sign enumeration.
- [ESlopeClass](#)
Holds conversion methods for the converter formulas.
- [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- [FileProtocolAdapter](#)
Adapter between the std::iostreambuf and the SFNC Features representing the device file system.
- [FloatNode](#)
Interface for string properties.
- [FloatRegNode](#)
Interface for string properties.
- [Function_NodeCallback](#)
Container for a function pointer.
- [IDevFileStreamBase](#)
- [IDevFileStreamBuf](#)
- [int64_autovector_t](#)
Vector of integers with reference counting.
- [IntegerNode](#)
Interface for string properties.
- [IntRegNode](#)
Interface for string properties.
- [Member_NodeCallback](#)
Container for a member function pointer.
- [Node](#)
class common to all nodes
- [NodeMap](#)
Smart pointer template for NodeMaps with create function.
- [ODevFileStreamBase](#)
- [ODevFileStreamBuf](#)
- [PortNode](#)
Interface for value properties.
- [PortRecorder](#)
Interface for recording write commands on a port.
- [PortReplay](#)
Interface for replaying write commands on a port.
- [RegisterNode](#)
Interface for string properties.
- [SpinTestCamera](#)
- [StringNode](#)
Interface for string properties.
- [StringRegNode](#)
Interface for string properties.
- [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 GenICam::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 `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 = 0x8000000L }

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,
intfIInteger,
intfIBoolean,
intfICommand,
intfIFloat,
intfIString,
intfIRegister,
intfICategory,
intfIEnumeration,
intfIEnumEntry,
intfIPort }

typedef for interface type

• enum ELinkType {
ctParentNodes,
ctReadingChildren,
ctWritingChildren,
ctlInvalidatingChildren,
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 `GenICam::gcstring GetVersionGuid` ()=0
 - Get the Guid describing the product version.*
- 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.*
- 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.*
- 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_vector_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.*
 - virtual IInteger & [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 [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 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)

- **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**

Invalidate 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=xvDefault)=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**

- virtual void `ExtractIndependentSubtree` (const `GenICam::gcstring` &XMLData, const `GenICam::gcstring` &InjectXMLData, const `GenICam::gcstring` &SubTreeRootNodeName, `GenICam::gcstring` &ExtractedSubtree)=0

Extracts an XML file into a target file.
- 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.
- virtual void `Write` (const void *pBuffer, int64_t Address, int64_t Length)=0

Writes a chunk of bytes to the port.
- virtual EYesNo `GetSwapEndianess` ()=0

Determines if the port adapter must perform an endianess 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

Creates a node from a string.

- *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 '#'.
 - **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.
 - bool **IsWritable** (const Spinnaker::GenApi::CPointer< T, B > &ptr)

Checks if a node is Writable.
 - bool **IsImplemented** (const Spinnaker::GenApi::CPointer< T, B > &ptr)

Checks if a node is Implemented.
 - 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.

9.3.1 Typedef Documentation

9.3.1.1 IDevFileStream

```
typedef IDevFileStreamBase<char, std::char_traits<char>> IDevFileStream
```

9.3.1.2 ODevFileStream

```
typedef ODevFileStreamBase<char, std::char_traits<char> > ODevFileStream
```

9.3.2 Enumeration Type Documentation

9.3.2.1 GVCP_MESSAGE_TAGS

```
enum GVCP_MESSAGE_TAGS
```

Enumerator

TAG_EVENT_CMD	
TAG_EVENTDATA_CMD	

9.3.3 Function Documentation

9.3.3.1 PersistFeature()

```
virtual void Spinnaker::GenApi::PersistFeature (
    IValue & item ) [pure virtual]
```

Stores a feature.

9.3.3.2 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 )
```

9.3.4 Variable Documentation

9.3.4.1 COMMAND_MAGIC

```
const uint8_t COMMAND_MAGIC = 0x42
```

9.3.4.2 GENCP_COMMAND_HEADER_SIZE

```
const size_t GENCP_COMMAND_HEADER_SIZE = sizeof(U3V_COMMAND_HEADER)
```

9.3.4.3 GENCP_EVENT_BASIC_SIZE

```
const size_t GENCP_EVENT_BASIC_SIZE = sizeof(U3V_EVENT_MESSAGE)
```

9.3.4.4 GENCP_EVENT_CMD_ID

```
const uint16_t GENCP_EVENT_CMD_ID = 0x0C00
```

9.3.4.5 IPersistScript

```
interface SPINNAKER_API_ABSTRACT IPersistScript
```

Initial value:

```
{  
    virtual void SetInfo(GenICam::gcstring & Info) = 0
```

Basic interface to persist values to.

9.3.4.6 U3V_EVENT_PREFIX

```
const uint32_t U3V_EVENT_PREFIX = 0x45563355
```

9.4 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)**

- **SPINNAKER_API gcstring GetGenICamCLProtocolFolder (void)**

Retrieve the path of the GenICam logging properties file.
- **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. The path to the CLProtocol folder can be stored by calling SetGenICamCLProtocolFolder().
- **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.

9.4.1 Function Documentation

9.4.1.1 getline() [1/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

STL getline.

9.4.1.2 getline() [2/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str,
    char delim ) [inline]
```

STL getline.

9.4.1.3 ThrowBadAlloc()

```
SPINNAKER_API void Spinnaker::GenICam::ThrowBadAlloc ( )
```

9.5 Spinnaker::Video Namespace Reference

Classes

- struct [AVIOption](#)
Options for saving AVI files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [MJPEGOption](#)
Options for saving MJPG files.
- class [SpinVideo](#)
Provides the functionality for the user to record images to an AVI/MP4 file.

Chapter 10

Class Documentation

10.1 ActionCommandResult Struct Reference

Action Command Result.

Public Attributes

- unsigned int [DeviceAddress](#)
- [ActionCommandStatus Status](#)

10.1.1 Detailed Description

Action Command Result.

10.1.2 Member Data Documentation

10.1.2.1 DeviceAddress

unsigned int [DeviceAddress](#)

10.1.2.2 Status

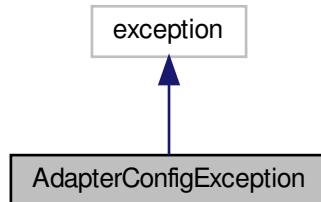
[ActionCommandStatus Status](#)

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

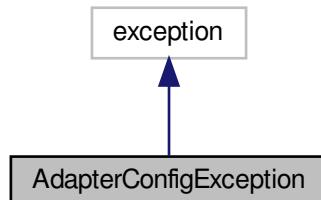
- [include/SpinnakerDefs.h](#)

10.2 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`

10.2.1 Constructor & Destructor Documentation

10.2.1.1 AdapterConfigException() [1/2]

```
AdapterConfigException (const AdapterConfig::AdapterConfigErr errCode) [inline]
```

10.2.1.2 AdapterConfigException() [2/2]

```
AdapterConfigException (
    const AdapterConfig::AdapterConfigErr errCode,
    std::string param ) [inline]
```

10.2.2 Member Function Documentation

10.2.2.1 ErrCode()

```
AdapterConfig::AdapterConfigErr ErrCode () const [inline]
```

10.2.2.2 GetParamStr()

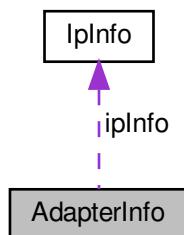
```
std::string GetParamStr () const [inline]
```

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

- [include/AdapterConfig.h](#)

10.3 AdapterInfo Struct Reference

Collaboration diagram for AdapterInfo:



Public Member Functions

- [AdapterInfo \(\)](#)

Public Attributes

- std::string [adapterName](#)
- std::string [adapterGUID](#)
- std::string [adapterDescription](#)
- bool [dhcpEnabled](#)
- [IplInfo iplInfo](#)
- 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](#)

10.3.1 Constructor & Destructor Documentation

10.3.1.1 [AdapterInfo\(\)](#)

```
AdapterInfo ( ) [inline]
```

10.3.2 Member Data Documentation

10.3.2.1 [adapterDescription](#)

```
std::string adapterDescription
```

10.3.2.2 [adapterGUID](#)

```
std::string adapterGUID
```

10.3.2.3 adapterName

```
std::string adapterName
```

10.3.2.4 dhcpEnabled

```
bool dhcpEnabled
```

10.3.2.5 ipInfo

```
IpInfo ipInfo
```

10.3.2.6 jumboPackets

```
unsigned int jumboPackets
```

10.3.2.7 jumboPacketsRegKey

```
std::string jumboPacketsRegKey
```

10.3.2.8 jumboPacketValidValues

```
std::vector<unsigned int> jumboPacketValidValues
```

10.3.2.9 receiveBuffers

```
unsigned int receiveBuffers
```

10.3.2.10 receiveBuffersMax

```
unsigned int receiveBuffersMax
```

10.3.2.11 receiveBuffersMin

```
unsigned int receiveBuffersMin
```

10.3.2.12 receiveBuffersRegKey

```
std::string receiveBuffersRegKey
```

10.3.2.13 receiveBuffersStep

```
unsigned int receiveBuffersStep
```

10.3.2.14 transmitBuffers

```
unsigned int transmitBuffers
```

10.3.2.15 transmitBuffersMax

```
unsigned int transmitBuffersMax
```

10.3.2.16 transmitBuffersMin

```
unsigned int transmitBuffersMin
```

10.3.2.17 transmitBuffersRegKey

```
std::string transmitBuffersRegKey
```

10.3.2.18 transmitBuffersStep

```
unsigned int transmitBuffersStep
```

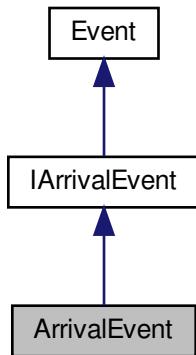
The documentation for this struct was generated from the following file:

- [include/AdapterConfig.h](#)

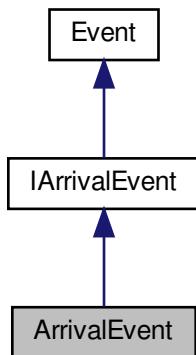
10.4 ArrivalEvent Class Reference

An event handler for capturing the device arrival event.

Inheritance diagram for ArrivalEvent:



Collaboration diagram for ArrivalEvent:



Public Member Functions

- `ArrivalEvent ()`
Default constructor.
- `virtual ~ArrivalEvent ()`
Virtual destructor.
- `virtual void OnDeviceArrival (uint64_t serialNumber)=0`
Callback to the device arrival event.

Protected Member Functions

- `ArrivalEvent & operator= (const ArrivalEvent &)`
Assignment operator.

Additional Inherited Members

10.4.1 Detailed Description

An event handler for capturing the device arrival event.

10.4.2 Constructor & Destructor Documentation

10.4.2.1 `ArrivalEvent()`

`ArrivalEvent ()`

Default constructor.

10.4.2.2 `~ArrivalEvent()`

`virtual ~ArrivalEvent () [virtual]`

Virtual destructor.

10.4.3 Member Function Documentation

10.4.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device arrival event.

Implements [IArrivalEvent](#).

10.4.3.2 operator=()

```
ArrivalEvent& operator= (
    const ArrivalEvent & ) [protected]
```

Assignment operator.

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

- [include/ArrivalEvent.h](#)

10.5 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.

10.5.1 Detailed Description

Delivers information about the attached chunks and nodes.

10.5.2 Member Data Documentation

10.5.2.1 NumAttachedChunks

```
int NumAttachedChunks
```

Number of chunks from the buffer attached to a chunk port.

10.5.2.2 NumChunkPorts

```
int NumChunkPorts
```

Number of chunk ports found in the node map.

10.5.2.3 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](#)

10.6 AutoLock Class Reference

Public Member Functions

- [AutoLock \(CLOCK &lock\)](#)
- [~AutoLock \(\)](#)

10.6.1 Constructor & Destructor Documentation

10.6.1.1 AutoLock()

```
AutoLock (
```



```
    CLOCK & lock ) [inline]
```

10.6.1.2 ~AutoLock()

```
~AutoLock () [inline]
```

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

- include/SpinGenApi/GCSynch.h

10.7 AutoLock Class Reference

Public Member Functions

- [AutoLock \(CLock &lock\)](#)
- [~AutoLock \(\)](#)

10.7.1 Constructor & Destructor Documentation

10.7.1.1 AutoLock()

```
AutoLock (
    CLock & lock ) [inline]
```

10.7.1.2 ~AutoLock()

```
~AutoLock () [inline]
```

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

- include/SpinGenApi/Synch.h

10.8 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.

10.8.1 Detailed Description

Options for saving AVI files.

10.8.2 Constructor & Destructor Documentation

10.8.2.1 `AVIOption()`

```
AVIOption ( ) [inline]
```

10.8.3 Member Data Documentation

10.8.3.1 `frameRate`

```
float frameRate
```

Frame rate of the stream.

10.8.3.2 `reserved`

```
unsigned int reserved[256]
```

Reserved for future use.

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

- `include/SpinVideoDefs.h`

10.9 `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.

10.9.1 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.

10.9.2 Constructor & Destructor Documentation

10.9.2.1 BasePtr() [1/2]

```
BasePtr ( void ) throw ()
```

Default constructor.

10.9.2.2 ~BasePtr()

```
virtual ~BasePtr ( void ) [virtual]
```

10.9.2.3 BasePtr() [2/2]

```
BasePtr ( const BasePtr< T, B > & other ) throw ()
```

Copy constructor.

10.9.3 Member Function Documentation**10.9.3.1 get()**

```
virtual T* get ( ) const [virtual]  
get\(\)
```

10.9.3.2 IsValid()

```
virtual bool IsValid ( ) const throw () [virtual]
```

True if the pointer is valid.

10.9.3.3 operator bool()

```
virtual operator bool (
    void ) const throw ) [virtual]
```

True if the pointer is valid.

10.9.3.4 operator T*()

```
virtual operator T* (
    void ) const [virtual]
```

Dereferencing.

10.9.3.5 operator->()

```
virtual T* operator-> (
    void ) const [virtual]
```

Dereferencing.

10.9.3.6 operator=() [1/4]

```
virtual BasePtr& operator= (
    const BasePtr< T, B > & rhs ) [virtual]
```

Assign INode Pointer.

10.9.3.7 operator=() [2/4]

```
virtual BasePtr& operator= (
    const int nMustBeNull ) [virtual]
```

10.9.3.8 operator=() [3/4]

```
virtual BasePtr& operator= (
    const long nMustBeNull ) [virtual]
```

10.9.3.9 operator=() [4/4]

```
virtual BasePtr& operator= (
    const std::nullptr_t nullPtr ) [virtual]
```

10.9.3.10 operator==(()) [1/4]

```
virtual bool operator== (
    const BasePtr< T, B > & rT ) const [virtual]
```

Pointer equal.

10.9.3.11 operator==(()) [2/4]

```
virtual bool operator== (
    std::nullptr_t ) const [virtual]
```

Pointer equal.

10.9.3.12 operator==(()) [3/4]

```
virtual bool operator== (
    int nMustBeNull ) const [virtual]
```

Pointer equal.

10.9.3.13 operator==(()) [4/4]

```
virtual bool operator== (
    long nMustBeNull ) const [virtual]
```

Pointer equal.

10.9.4 Member Data Documentation

10.9.4.1 m_pT

```
PointerData* m_pT [protected]
```

Underlying raw pointer.

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

- [include/BasePtr.h](#)

10.10 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.

10.10.1 Detailed Description

Options for saving Bitmap image.

10.10.2 Constructor & Destructor Documentation

10.10.2.1 BMPOption()

```
BMPOption () [inline]
```

10.10.3 Member Data Documentation

10.10.3.1 indexedColor_8bit

```
bool indexedColor_8bit
```

10.10.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

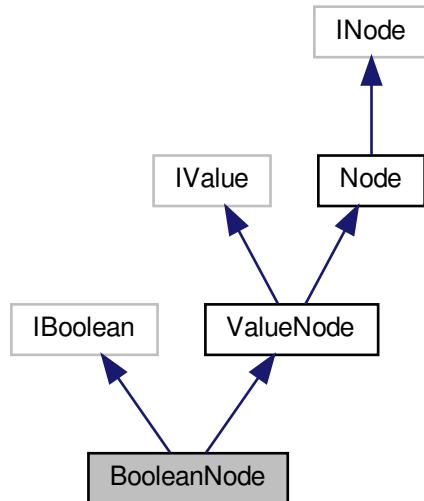
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

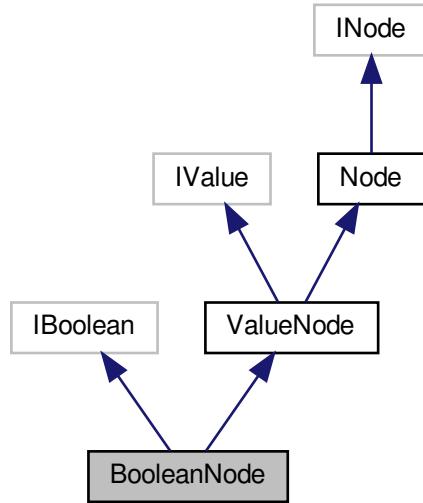
10.11 BooleanNode Class Reference

[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

10.11.1 Detailed Description

[Interface](#) for string properties.

10.11.2 Constructor & Destructor Documentation

10.11.2.1 BooleanNode() [1/2]

```
BooleanNode ( )
```

10.11.2.2 BooleanNode() [2/2]

```
BooleanNode (
    std::shared_ptr< Node::NodeImpl > pBoolean )
```

10.11.2.3 ~BooleanNode()

```
virtual ~BooleanNode ( ) [virtual]
```

10.11.3 Member Function Documentation

10.11.3.1 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.

10.11.3.2 operator=()

```
virtual void operator= (
    bool Value ) [virtual]
```

Set node value.

10.11.3.3 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

10.11.3.4 SetValue()

```
void SetValue (
    bool Value,
    bool Verify = true )
```

Set node value.

Parameters

<code>Value</code>	The value to set.
<code>Verify</code>	Enables AccessMode and Range verification (default = true).

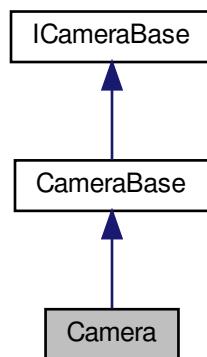
The documentation for this class was generated from the following file:

- include/SpinGenApi/[BooleanNode.h](#)

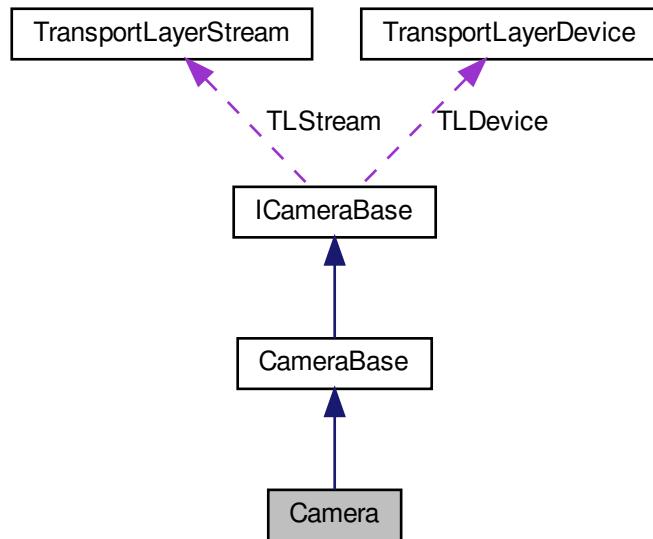
10.12 Camera Class Reference

The camera object class.

Inheritance diagram for Camera:



Collaboration diagram for Camera:



Public Member Functions

- `~Camera ()`
- void `Init ()`

Public Attributes

- `GenApi::IInteger & LUTIndex`
Description:
Control the index (offset) of the coefficient to access in the selected LUT.
- `GenApi::IBoolean & LUTEnable`
Description:
Activates the selected LUT.
- `GenApi::IInteger & LUTValue`
Description:
Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.
- `GenApi::IEnumeratorT< LUTSelectorEnums > & LUTSelector`
Description:
Selects which LUT to control.
- `GenApi::IFloat & ExposureTime`
Description:
Exposure time in microseconds when Exposure Mode is Timed.
- `GenApi::ICommand & AcquisitionStop`
Description: *This command stops the acquisition of images.*
- `GenApi::IFloat & AcquisitionResultingFrameRate`
Description: *Resulting frame rate in Hertz.*
- `GenApi::IFloat & AcquisitionLineRate`

- Description:* Controls the rate (in Hertz) at which the Lines in a Frame are captured.
- [GenApi::ICommand & AcquisitionStart](#)

Description: This command starts the acquisition of images.
- [GenApi::ICommand & TriggerSoftware](#)

Description:
Generates an internal trigger if Trigger Source is set to Software.
- [GenApi::IEnumerationT< ExposureModeEnums > & ExposureMode](#)

Description:
Sets the operation mode of the Exposure.
- [GenApi::IEnumerationT< AcquisitionModeEnums > & AcquisitionMode](#)

Description: Sets the acquisition mode of the device.
- [GenApi::IInteger & AcquisitionFrameCount](#)

Description:
Number of images to acquire during a multi frame acquisition.
- [GenApi::IEnumerationT< TriggerSourceEnums > & TriggerSource](#)

Description:
Specifies the internal signal or physical input line to use as the trigger source.
- [GenApi::IEnumerationT< TriggerActivationEnums > & TriggerActivation](#)

Description: Specifies the activation mode of the trigger.
- [GenApi::IEnumerationT< SensorShutterModeEnums > & SensorShutterMode](#)

Description: Sets the shutter mode of the device.
- [GenApi::IFloat & TriggerDelay](#)

Description:
Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.
- [GenApi::IEnumerationT< TriggerModeEnums > & TriggerMode](#)

Description:
Controls whether or not trigger is active.
- [GenApi::IFloat & AcquisitionFrameRate](#)

Description: User controlled acquisition frame rate in Hertz Visibility:
- [GenApi::IEnumerationT< TriggerOverlapEnums > & TriggerOverlap](#)

Description: Specifies the overlap mode of the trigger.
- [GenApi::IEnumerationT< 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::IInteger & AcquisitionBurstFrameCount](#)

Description:
This feature is used only if the FrameBurstStart trigger is enabled and the FrameBurstEnd trigger is disabled.
- [GenApi::IInteger & EventTest](#)

Description: Returns the unique identifier of the Test type of [Event](#).
- [GenApi::IInteger & EventTestTimestamp](#)

Description: Returns the Timestamp of the Test [Event](#).
- [GenApi::IInteger & EventExposureEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End [Event](#).
- [GenApi::IInteger & EventExposureEnd](#)

Description: Returns the unique identifier of the Exposure End type of [Event](#).
- [GenApi::IInteger & EventExposureEndTimestamp](#)

Description: Returns the Timestamp of the Exposure End [Event](#).
- [GenApi::IInteger & EventError](#)

Description: Returns the unique identifier of the Error type of [Event](#).
- [GenApi::IInteger & EventErrorTimestamp](#)

Description: Returns the Timestamp of the Error [Event](#).

- Description:* Returns the Timestamp of the Error Event.

 - GenApi::IInteger & EventErrorCode

Description: Returns the error code for the error that happened Visibility:
 - GenApi::IInteger & 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::IInteger & EventSerialPortReceive

Description: Returns the unique identifier of the Serial Port Receive type of Event.
 - GenApi::IInteger & EventSerialPortReceiveTimestamp

Description: Returns the Timestamp of the Serial Port Receive Event.
 - GenApi::IString & EventSerialData

Description: Returns the serial data that was received.
 - GenApi::IInteger & 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::IInteger & 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 & LogicBlockLUTOOutputValue

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.
 - GenApi::IInteger & LogicBlockLUTOOutputValueAll

Description: Sets the value of all the output bits in the selected LUT.
 - GenApi::IEnumerationT< 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::IEnumerationT< ColorTransformationSelectorEnums > & ColorTransformationSelector

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.
 - GenApi::IEnumerationT< 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::IEnumerationT< ColorTransformationValueSelectorEnums > & ColorTransformationValueSelector

- **GenApi::IInteger & TimestampLatchValue**

Description: Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module
- **GenApi::ICommand & TimestampReset**

Description: Returns the latched value of the timestamp counter.
- **GenApi::IString & DeviceUserID**

Description: Resets the current value of the device timestamp counter.
- **GenApi::IString & DeviceTemperature**

Description: User-programmable device identifier.
- **GenApi::IFloat & DeviceTemperature**

Description: Device temperature in degrees Celsius (C).
- **GenApi::IInteger & MaxDeviceResetTime**

Description: Time to wait until device reset complete (ms).
- **GenApi::IInteger & 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::IEnumerationT< DeviceRegistersEndiannessEnums > & DeviceRegistersEndianness**

Description: Endianess of the registers of the device.
- **GenApi::IString & DeviceManufacturerInfo**

Description: Manufacturer information about the device.
- **GenApi::IInteger & DeviceLinkSpeed**

Description: Indicates the speed of transmission negotiated on the specified Link.
- **GenApi::IInteger & LinkUptime**

Description: Time since the last phy negotiation (enumeration).
- **GenApi::IInteger & 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::IInteger & 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::IInteger & 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::IInteger & 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::IInteger & 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::IInteger & DeviceUptime

Description: Total time since the device was powered up in seconds.
- GenApi::IInteger & DeviceLinkCurrentThroughput

Description: Current bandwidth of streamed data.
- GenApi::IInteger & 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::IInteger & AasRoiOffsetY

Description:
Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.
- GenApi::IInteger & 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::IInteger & 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 & GevSCPSDoNotFragment](#)
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::IInteger & GevGVCPPendingTimeout](#)
Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.
- [GenApi::IEnumerationT< 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::IInteger & GevMACAddress](#)
Description: MAC address of the logical link.
- [GenApi::IInteger & GevPersistentSubnetMask](#)
Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.
- [GenApi::IInteger & GevMCPHostPort](#)
Description: The port to which the device must send messages Visibility:
- [GenApi::IInteger & 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::IInteger & GevSCPInterfaceIndex](#)
Description: Index of the logical link to use.
- [GenApi::IBoolean & GevSupportedOption](#)
Description: Returns if the selected GEV option is supported.

- [GenApi::IEnumerationT< 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::IInteger & 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::IInteger & 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::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.
- [GenApi::IInteger & GevHeartbeatTimeout](#)
Description: Indicates the current heartbeat timeout in milliseconds.
- [GenApi::IInteger & GevPersistentDefaultGateway](#)
Description: Controls the persistent default gateway for this logical link.
- [GenApi::IEnumerationT< GevCCPEnums > & GevCCP](#)
Description: Controls the device access privilege of an application.
- [GenApi::IInteger & GevMCDA](#)
Description: Controls the destination IP address of the message channel Visibility:
- [GenApi::IInteger & 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::IInteger & GevSCPDirection](#)
Description: Transmit or Receive of the channel Visibility:
- [GenApi::IBoolean & GevSCPSFireTestPacket](#)
Description: Sends a test packet.
- [GenApi::IString & GevSecondURL](#)
Description: The second choice of URL to the XML device description file.
- [GenApi::IEnumerationT< GevSupportedOptionSelectorEnums > & GevSupportedOptionSelector](#)
Description: Selects the GEV option to interrogate for existing support.
- [GenApi::IBoolean & GevGVCPHeartbeatDisable](#)
Description: Disables the GVCP heartbeat.
- [GenApi::IInteger & GevMCRC](#)
Description: Indicates the number of retries of the message channel.
- [GenApi::IBoolean & GevSCPSBigEndian](#)
Description: Endianess of multi-byte pixel data for this stream.
- [GenApi::IInteger & GevNumberOfInterfaces](#)
Description: Indicates the number of physical network interfaces supported by this device.
- [GenApi::IInteger & TLParamsLocked](#)
Description: Visibility:
- [GenApi::IInteger & PayloadSize](#)
Description: Provides the number of bytes transferred for each image or chunk on the stream channel.
- [GenApi::IInteger & PacketResendRequestCount](#)
Description: Counts the number of resend requests received from the host.
- [GenApi::IBoolean & SharpeningEnable](#)

- Description:
Enables/disables the sharpening feature.
- GenApi::IEnumerationT< BlackLevelSelectorEnums > & BlackLevelSelector
 - Description:
Selects which black level to control.
- GenApi::IBoolean & GammaEnable
 - Description: *Enables/disables gamma correction.*
- GenApi::IBoolean & SharpeningAuto
 - Description:
Enables/disables the auto sharpening feature.
- GenApi::IBoolean & BlackLevelClampingEnable
 - Description:
Enable the black level auto clamping feature which performing dark current compensation.
- GenApi::IFloat & BalanceRatio
 - Description:
Controls the balance ratio of the selected color relative to green.
- GenApi::IEnumerationT< BalanceWhiteAutoEnums > & BalanceWhiteAuto
 - Description:
White Balance compensates for color shifts caused by different lighting conditions.
- GenApi::IFloat & SharpeningThreshold
 - Description:
Controls the minimum intensity gradient change to invoke sharpening.
- GenApi::IEnumerationT< GainAutoEnums > & GainAuto
 - Description:
Sets the automatic gain mode.
- GenApi::IFloat & Sharpening
 - Description:
Controls the amount to sharpen a signal.
- GenApi::IFloat & 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::IInteger & 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::IInteger & 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::IInteger & 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::IInteger & DefectTableCoordinateX](#)
Description:
Returns the X coordinate of the defective pixel at DefectTableIndex within the defective pixel table.
- [GenApi::IInteger & 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::IInteger & 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::IInteger & SerialTransmitQueueMaxCharacterCount](#)
Description: Returns the maximum number of characters in the serial port transmit queue.
- [GenApi::IInteger & 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::IInteger & SerialReceiveFramingErrorCount](#)
Description: Returns the number of framing errors that have occurred on the serial port.
- [GenApi::IInteger & SerialTransmitQueueCurrentCharacterCount](#)
Description: Returns the number of characters currently in the serial port transmit queue.
- [GenApi::IInteger & 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::IInteger & SerialReceiveQueueMaxCharacterCount](#)
Description: Returns the maximum number of characters in the serial port receive queue.

- `GenApi::IInteger & 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::IInteger & 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::IInteger & SequencerSetActive`
Description: Displays the currently active sequencer set.
- `GenApi::IInteger & SequencerSetNext`
Description: Specifies the next sequencer set.
- `GenApi::ICommand & SequencerSetLoad`
Description:
Loads currently selected sequencer to the current device configuration.
- `GenApi::IInteger & SequencerPathSelector`
Description:
Selects branching path to be used for subsequent settings.
- `GenApi::IBoolean & SequencerFeatureEnable`
Description:
Enables the selected feature and makes it active in all sequencer sets.
- `GenApi::IInteger & TransferBlockCount`
Description: Specifies the number of data blocks (images) that the device should stream before stopping.
- `GenApi::ICommand & TransferStart`
Description: Starts the streaming of data blocks (images) out of the device.
- `GenApi::IInteger & TransferQueueMaxBlockCount`
Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:
- `GenApi::IInteger & TransferQueueCurrentBlockCount`
Description: Returns number of data blocks (images) currently in the transfer queue.
- `GenApi::IEnumerationT< TransferQueueModeEnums > & TransferQueueMode`
Description: Specifies the operation mode of the transfer queue.
- `GenApi::IEnumerationT< TransferOperationModeEnums > & TransferOperationMode`
Description: Selects the operation mode of the transfer.
- `GenApi::ICommand & TransferStop`
Description: Stops the streaming of data block (images).
- `GenApi::IInteger & TransferQueueOverflowCount`

- **GenApi::IEnumerationT< TransferControlModeEnums > & TransferControlMode**
Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.
- **GenApi::IFloat & ChunkBlackLevel**
Description: Selects the control method for the transfers.
- **GenApi::IFloat & ChunkFrameID**
Description: Returns the black level used to capture the image.
- **GenApi::IInteger & ChunkFrameID**
Description: Returns the image frame ID.
- **GenApi::IString & ChunkSerialData**
Description: Returns the serial data that was received.
- **GenApi::IFloat & ChunkExposureTime**
Description: Returns the exposure time used to capture the image.
- **GenApi::IBoolean & ChunkSerialReceiveOverflow**
Description: Returns the status of the chunk serial receive overflow.
- **GenApi::IInteger & ChunkTimestamp**
Description: Returns the Timestamp of the image.
- **GenApi::IBoolean & ChunkModeActive**
Description: Activates the inclusion of Chunk data in the payload of the image.
- **GenApi::IInteger & ChunkExposureEndLineStatusAll**
Description: Returns the status of all the I/O lines at the end of exposure event.
- **GenApi::IEnumerationT< ChunkGainSelectorEnums > & ChunkGainSelector**
Description: Selects which gain to retrieve Visibility:
- **GenApi::IEnumerationT< ChunkSelectorEnums > & ChunkSelector**
Description: Selects which chunk data to enable or disable.
- **GenApi::IEnumerationT< ChunkBlackLevelSelectorEnums > & ChunkBlackLevelSelector**
Description: Selects which black level to retrieve Visibility:
- **GenApi::IInteger & ChunkWidth**
Description: Returns the width of the image included in the payload.
- **GenApi::IInteger & ChunkImage**
Description: Returns the image payload.
- **GenApi::IInteger & ChunkHeight**
Description: Returns the height of the image included in the payload.
- **GenApi::IEnumerationT< ChunkPixelFormatEnums > & ChunkPixelFormat**
Description: Format of the pixel provided by the camera Visibility:
- **GenApi::IFloat & ChunkGain**
Description: Returns the gain used to capture the image.
- **GenApi::IInteger & ChunkSequencerSetActive**
Description: Returns the index of the active set of the running sequencer included in the payload.
- **GenApi::IInteger & ChunkCRC**
Description: Returns the CRC of the image payload.
- **GenApi::IInteger & ChunkOffsetX**
Description: Returns the Offset X of the image included in the payload.
- **GenApi::IInteger & ChunkOffsetY**
Description: Returns the Offset Y of the image included in the payload.
- **GenApi::IBoolean & ChunkEnable**
Description: Enables the inclusion of the selected Chunk data in the payload of the image.
- **GenApi::IInteger & ChunkSerialDataLength**
Description: Returns the length of the received serial data that was included in the payload.
- **GenApi::IInteger & FileAccessOffset**
Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.
- **GenApi::IInteger & FileAccessLength**

- Description:* Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

 - GenApi::IEnumerationT< FileOperationStatusEnums > & FileOperationStatus

Description: Represents the file operation execution status.
 - GenApi::ICommand & FileOperationExecute

Description:

This is a command that executes the selected file operation on the selected file.
 - GenApi::IEnumerationT< FileModeEnums > & FileMode

Description:

The mode of the file when it is opened.
 - GenApi::IInteger & FileOperationResult

Description: Represents the file operation result.
 - GenApi::IEnumerationT< FileOperationSelectorEnums > & FileOperationSelector

Description:

Sets operation to execute on the selected file when the execute command is given.
 - GenApi::IEnumerationT< FileSelectorEnums > & FileSelector

Description:

Selects which file is being operated on.
 - GenApi::IInteger & FileSize

Description: Represents the size of the selected file in bytes.
 - GenApi::IEnumerationT< BinningSelectorEnums > & BinningSelector

Description:

Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.
 - GenApi::IInteger & PixelDynamicRangeMin

Description: Minimum value that can be returned during the digitization process.
 - GenApi::IInteger & PixelDynamicRangeMax

Description: Maximum value that can be returned during the digitization process.
 - GenApi::IInteger & OffsetY

Description:

Vertical offset from the origin to the ROI (in pixels).
 - GenApi::IInteger & BinningHorizontal

Description:

Number of horizontal photo-sensitive cells to combine together.
 - GenApi::IInteger & Width

Description:

Width of the image provided by the device (in pixels).
 - GenApi::IEnumerationT< 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::IBoolean & ReverseX

Description: Horizontally flips the image sent by the device.
 - GenApi::IBoolean & ReverseY

Description: Vertically flips the image sent by the device.
 - GenApi::IEnumerationT< TestPatternEnums > & TestPattern

Description:

Selects the type of test pattern that is generated by the device as image source.
 - GenApi::IEnumerationT< PixelColorFilterEnums > & PixelColorFilter

Description: Type of color filter that is applied to the image.
 - GenApi::IInteger & WidthMax

Description:

Maximum width of the image (in pixels).
 - GenApi::IEnumerationT< AdcBitDepthEnums > & AdcBitDepth

Description:

Selects which ADC bit depth to use.

- [GenApi::IInteger & BinningVertical](#)
Description:
Number of vertical photo-sensitive cells to combine together.
- [GenApi::IEnumerationT< DecimationHorizontalModeEnums > & DecimationHorizontalMode](#)
Description:
The mode used to reduce the horizontal resolution when DecimationHorizontal is used.
- [GenApi::IEnumerationT< BinningVerticalModeEnums > & BinningVerticalMode](#)
Description: Visibility:
- [GenApi::IInteger & OffsetX](#)
Description:
Horizontal offset from the origin to the ROI (in pixels).
- [GenApi::IInteger & HeightMax](#)
Description: Maximum height of the image (in pixels).
- [GenApi::IInteger & DecimationHorizontal](#)
Description:
Horizontal decimation of the image.
- [GenApi::IEnumerationT< PixelSizeEnums > & PixelSize](#)
Description: Total size in bits of a pixel of the image.
- [GenApi::IInteger & SensorHeight](#)
Description: Effective height of the sensor in pixels.
- [GenApi::IEnumerationT< 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::IInteger & DecimationVertical](#)
Description:
Vertical decimation of the image.
- [GenApi::IInteger & 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::IInteger & 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::IInteger & GuiXmlManifestAddress](#)
Description: Location of the GUI XML manifest table.
- [GenApi::IInteger & 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::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).
- [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::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).
- [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::IInteger & CounterValue](#)
Description: Current counter value Visibility:
- [GenApi::IEnumerationT< CounterSelectorEnums > & CounterSelector](#)
Description: Selects which counter to configure Visibility:
- [GenApi::IInteger & 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::IInteger & 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::IInteger & 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::IInteger](#) & [DeviceSFNCVersionMajor](#)

Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
 - [GenApi::IInteger](#) & [DeviceSFNCVersionMinor](#)

Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
 - [GenApi::IInteger](#) & [DeviceSFNCVersionSubMinor](#)

Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
 - [GenApi::IInteger](#) & [DeviceManifestEntrySelector](#)

Description: Selects the manifest entry to reference.
 - [GenApi::IInteger](#) & [DeviceManifestXMLMajorVersion](#)

Description: Indicates the major version number of the [GenICam XML](#) file of the selected manifest entry.
 - [GenApi::IInteger](#) & [DeviceManifestXMLMinorVersion](#)

Description: Indicates the minor version number of the [GenICam XML](#) file of the selected manifest entry.
 - [GenApi::IInteger](#) & [DeviceManifestXMLSubMinorVersion](#)

Description: Indicates the subminor version number of the [GenICam XML](#) file of the selected manifest entry.
 - [GenApi::IInteger](#) & [DeviceManifestSchemaMajorVersion](#)

Description: Indicates the major version number of the schema file of the selected manifest entry.
 - [GenApi::IInteger](#) & [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::IInteger](#) & [DeviceTLVersionSubMinor](#)

Description: Sub minor version of the Transport Layer of the device.
 - [GenApi::IInteger](#) & [DeviceGenCPVersionMajor](#)

Description: Major version of the GenCP protocol supported by the device.
 - [GenApi::IInteger](#) & [DeviceGenCPVersionMinor](#)

Description: Minor version of the GenCP protocol supported by the device.
 - [GenApi::IInteger](#) & [DeviceConnectionSelector](#)

Description: Selects which Connection of the device to control.
 - [GenApi::IInteger](#) & [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::IInteger](#) & [DeviceLinkSelector](#)

Description: Selects which Link of the device to control.
 - [GenApi::IEnumerationT< DeviceLinkThroughputLimitModeEnums >](#) & [DeviceLinkThroughputLimitMode](#)

Description: Controls if the DeviceLinkThroughputLimit is active.
 - [GenApi::IInteger](#) & [DeviceLinkConnectionCount](#)

Description: Returns the number of physical connection of the device used by a particular Link.
 - [GenApi::IEnumerationT< DeviceLinkHeartbeatModeEnums >](#) & [DeviceLinkHeartbeatMode](#)

- **GenApi::IFloat & DeviceLinkHeartbeatTimeout**
Description: Activate or deactivate the Link's heartbeat.
- **GenApi::IFloat & DeviceLinkCommandTimeout**
Description: Controls the current heartbeat timeout of the specific Link.
- **GenApi::IFloat & DeviceStreamChannelSelector**
Description: Indicates the command timeout of the specified Link.
- **GenApi::IInteger & DeviceStreamChannelLink**
Description: Selects the stream channel to control.
- **GenApi::IEnumerationT< DeviceStreamChannelTypeEnums > & DeviceStreamChannelType**
Description: Reports the type of the stream channel.
- **GenApi::IInteger & DeviceStreamChannelLinkIndex**
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::IInteger & 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::IInteger & 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**

- [GenApi::IInteger & LinePitch](#)

Description: Controls if the selected component streaming is active.
- [GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > & PixelFormatInfoSelector](#)

Description: Total number of bytes between 2 successive lines.
- [GenApi::IEnumerationT< DeinterlacingEnums > & Deinterlacing](#)

Description: Select the pixel format for which the information will be returned.
- [GenApi::IInteger & PixelFormatInfoID](#)

Description: Returns the value used by the streaming channels to identify the selected pixel format.
- [GenApi::IEnumerationT< ImageCompressionRateOptionEnums > & ImageCompressionRateOption](#)

Description: Two rate controlling options are offered: fixed bit rate or fixed quality.
- [GenApi::IInteger & 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::IInteger & EncoderDivide](#)
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::IInteger & EncoderValue](#)
Description: Reads or writes the current value of the position counter of the selected Encoder.
- [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.
- [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::IInteger & ActionDeviceKey**

Description: Provides the device key that allows the device to check the validity of action commands.
- **GenApi::IInteger & ActionQueueSize**

Description: Indicates the size of the scheduled action commands queue.
- **GenApi::IInteger & ActionSelector**

Description: Selects to which Action Signal further Action settings apply.
- **GenApi::IInteger & ActionGroupMask**

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.
- **GenApi::IInteger & ActionGroupKey**

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.
- **GenApi::IInteger & EventAcquisitionTrigger**

Description: Returns the unique Identifier of the Acquisition Trigger type of [Event](#).
- **GenApi::IInteger & EventAcquisitionTriggerTimestamp**

Description: Returns the Timestamp of the Acquisition Trigger [Event](#).
- **GenApi::IInteger & EventAcquisitionTriggerFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger [Event](#).
- **GenApi::IInteger & EventAcquisitionStart**

Description: Returns the unique Identifier of the Acquisition Start type of [Event](#).
- **GenApi::IInteger & EventAcquisitionStartTimestamp**

Description: Returns the Timestamp of the Acquisition Start [Event](#).
- **GenApi::IInteger & EventAcquisitionStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start [Event](#).
- **GenApi::IInteger & EventAcquisitionEnd**

Description: Returns the unique Identifier of the Acquisition End type of [Event](#).
- **GenApi::IInteger & EventAcquisitionEndTimestamp**

Description: Returns the Timestamp of the Acquisition End [Event](#).
- **GenApi::IInteger & EventAcquisitionEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End [Event](#).
- **GenApi::IInteger & EventAcquisitionTransferStart**

Description: Returns the unique Identifier of the Acquisition Transfer Start type of [Event](#).
- **GenApi::IInteger & EventAcquisitionTransferStartTimestamp**

Description: Returns the Timestamp of the Acquisition Transfer Start [Event](#).
- **GenApi::IInteger & EventAcquisitionTransferStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start [Event](#).
- **GenApi::IInteger & EventAcquisitionTransferEnd**

Description: Returns the unique Identifier of the Acquisition Transfer End type of [Event](#).
- **GenApi::IInteger & EventAcquisitionTransferEndTimestamp**

Description: Returns the Timestamp of the Acquisition Transfer End [Event](#).
- **GenApi::IInteger & EventAcquisitionTransferEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End [Event](#).
- **GenApi::IInteger & EventAcquisitionError**

Description: Returns the unique Identifier of the Acquisition Error type of [Event](#).
- **GenApi::IInteger & EventAcquisitionErrorTimestamp**

Description: Returns the Timestamp of the Acquisition Error [Event](#).
- **GenApi::IInteger & EventAcquisitionErrorFrameID**

- **GenApi::IInteger & EventFrameTrigger**
Description: Returns the unique Identifier of the FrameTrigger type of [Event](#).
- **GenApi::IInteger & EventFrameTriggerTimestamp**
Description: Returns the Timestamp of the FrameTrigger [Event](#).
- **GenApi::IInteger & EventFrameTriggerFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger [Event](#).
- **GenApi::IInteger & EventFrameStart**
Description: Returns the unique Identifier of the Frame Start type of [Event](#).
- **GenApi::IInteger & EventFrameStartTimestamp**
Description: Returns the Timestamp of the Frame Start [Event](#).
- **GenApi::IInteger & EventFrameStartFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start [Event](#).
- **GenApi::IInteger & EventFrameEnd**
Description: Returns the unique Identifier of the Frame End type of [Event](#).
- **GenApi::IInteger & EventFrameEndTimestamp**
Description: Returns the Timestamp of the Frame End [Event](#).
- **GenApi::IInteger & EventFrameEndFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End [Event](#).
- **GenApi::IInteger & EventFrameBurstStart**
Description: Returns the unique Identifier of the Frame Burst Start type of [Event](#).
- **GenApi::IInteger & EventFrameBurstStartTimestamp**
Description: Returns the Timestamp of the Frame Burst Start [Event](#).
- **GenApi::IInteger & EventFrameBurstStartFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start [Event](#).
- **GenApi::IInteger & EventFrameBurstEnd**
Description: Returns the unique Identifier of the Frame Burst End type of [Event](#).
- **GenApi::IInteger & EventFrameBurstEndTimestamp**
Description: Returns the Timestamp of the Frame Burst End [Event](#).
- **GenApi::IInteger & EventFrameBurstEndFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End [Event](#).
- **GenApi::IInteger & EventFrameTransferStart**
Description: Returns the unique Identifier of the Frame Transfer Start type of [Event](#).
- **GenApi::IInteger & EventFrameTransferStartTimestamp**
Description: Returns the Timestamp of the Frame Transfer Start [Event](#).
- **GenApi::IInteger & EventFrameTransferStartFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start [Event](#).
- **GenApi::IInteger & EventFrameTransferEnd**
Description: Returns the unique Identifier of the Frame Transfer End type of [Event](#).
- **GenApi::IInteger & EventFrameTransferEndTimestamp**
Description: Returns the Timestamp of the Frame Transfer End [Event](#).
- **GenApi::IInteger & EventFrameTransferEndFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End [Event](#).
- **GenApi::IInteger & EventExposureStart**
Description: Returns the unique Identifier of the Exposure Start type of [Event](#).
- **GenApi::IInteger & EventExposureStartTimestamp**
Description: Returns the Timestamp of the Exposure Start [Event](#).
- **GenApi::IInteger & EventExposureStartFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start [Event](#).
- **GenApi::IInteger & EventStream0TransferStart**
Description: Returns the unique Identifier of the Stream 0 Transfer Start type of [Event](#).

- `GenApi::IInteger & EventStream0TransferStartTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer Start [Event](#).
- `GenApi::IInteger & EventStream0TransferStartFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start [Event](#).
- `GenApi::IInteger & EventStream0TransferEnd`
Description: Returns the unique Identifier of the Stream 0 Transfer End type of [Event](#).
- `GenApi::IInteger & EventStream0TransferEndTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer End [Event](#).
- `GenApi::IInteger & EventStream0TransferEndFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End [Event](#).
- `GenApi::IInteger & EventStream0TransferPause`
Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of [Event](#).
- `GenApi::IInteger & EventStream0TransferPauseTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer Pause [Event](#).
- `GenApi::IInteger & EventStream0TransferPauseFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause [Event](#).
- `GenApi::IInteger & EventStream0TransferResume`
Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of [Event](#).
- `GenApi::IInteger & EventStream0TransferResumeTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer Resume [Event](#).
- `GenApi::IInteger & EventStream0TransferResumeFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockStart`
Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockStartTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer Block Start [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockStartFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockEnd`
Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockEndTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer Block End [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockEndFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockTrigger`
Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockTriggerTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger [Event](#).
- `GenApi::IInteger & EventStream0TransferBlockTriggerFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger [Event](#).
- `GenApi::IInteger & EventStream0TransferBurstStart`
Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of [Event](#).
- `GenApi::IInteger & EventStream0TransferBurstStartTimestamp`
Description: Returns the Timestamp of the Stream 0 Transfer Burst Start [Event](#).
- `GenApi::IInteger & EventStream0TransferBurstStartFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start [Event](#).
- `GenApi::IInteger & EventStream0TransferBurstEnd`

- **GenApi::IInteger & EventStream0TransferBurstEndTimestamp**
Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of [Event](#).
 - **GenApi::IInteger & EventStream0TransferBurstEndFrameID**
Description: Returns the Timestamp of the Stream 0 Transfer Burst End [Event](#).
 - **GenApi::IInteger & EventStream0TransferOverflow**
Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of [Event](#).
 - **GenApi::IInteger & EventStream0TransferOverflowTimestamp**
Description: Returns the Timestamp of the Stream 0 Transfer Overflow [Event](#).
 - **GenApi::IInteger & EventStream0TransferOverflowFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow [Event](#).
 - **GenApi::IInteger & EventSequencerSetChange**
Description: Returns the unique Identifier of the Sequencer Set Change type of [Event](#).
 - **GenApi::IInteger & EventSequencerSetChangeTimestamp**
Description: Returns the Timestamp of the Sequencer Set Change [Event](#).
 - **GenApi::IInteger & EventSequencerSetChangeFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change [Event](#).
- **GenApi::IInteger & EventCounter0Start**
Description: Returns the unique Identifier of the Counter 0 Start type of [Event](#).
 - **GenApi::IInteger & EventCounter0StartTimestamp**
Description: Returns the Timestamp of the Counter 0 Start [Event](#).
 - **GenApi::IInteger & EventCounter0StartFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start [Event](#).
- **GenApi::IInteger & EventCounter1Start**
Description: Returns the unique Identifier of the Counter 1 Start type of [Event](#).
 - **GenApi::IInteger & EventCounter1StartTimestamp**
Description: Returns the Timestamp of the Counter 1 Start [Event](#).
 - **GenApi::IInteger & EventCounter1StartFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start [Event](#).
- **GenApi::IInteger & EventCounter0End**
Description: Returns the unique Identifier of the Counter 0 End type of [Event](#).
 - **GenApi::IInteger & EventCounter0EndTimestamp**
Description: Returns the Timestamp of the Counter 0 End [Event](#).
 - **GenApi::IInteger & EventCounter0EndFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End [Event](#).
- **GenApi::IInteger & EventCounter1End**
Description: Returns the unique Identifier of the Counter 1 End type of [Event](#).
 - **GenApi::IInteger & EventCounter1EndTimestamp**
Description: Returns the Timestamp of the Counter 1 End [Event](#).
 - **GenApi::IInteger & EventCounter1EndFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End [Event](#).
- **GenApi::IInteger & EventTimer0Start**
Description: Returns the unique Identifier of the Timer 0 Start type of [Event](#).
 - **GenApi::IInteger & EventTimer0StartTimestamp**
Description: Returns the Timestamp of the Timer 0 Start [Event](#).
 - **GenApi::IInteger & EventTimer0StartFrameID**
Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start [Event](#).
- **GenApi::IInteger & EventTimer1Start**
Description: Returns the unique Identifier of the Timer 1 Start type of [Event](#).

- `GenApi::IInteger & EventTimer1StartTimestamp`
Description: Returns the Timestamp of the Timer 1 Start [Event](#).
- `GenApi::IInteger & EventTimer1StartFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start [Event](#).
- `GenApi::IInteger & EventTimer0End`
Description: Returns the unique Identifier of the Timer 0 End type of [Event](#).
- `GenApi::IInteger & EventTimer0EndTimestamp`
Description: Returns the Timestamp of the Timer 0 End [Event](#).
- `GenApi::IInteger & EventTimer0EndFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End [Event](#).
- `GenApi::IInteger & EventTimer1End`
Description: Returns the unique Identifier of the Timer 1 End type of [Event](#).
- `GenApi::IInteger & EventTimer1EndTimestamp`
Description: Returns the Timestamp of the Timer 1 End [Event](#).
- `GenApi::IInteger & EventEncoder0Stopped`
Description: Returns the unique Identifier of the Encoder 0 Stopped type of [Event](#).
- `GenApi::IInteger & EventEncoder0StoppedTimestamp`
Description: Returns the Timestamp of the Encoder 0 Stopped [Event](#).
- `GenApi::IInteger & EventEncoder0StoppedFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped [Event](#).
- `GenApi::IInteger & EventEncoder1Stopped`
Description: Returns the unique Identifier of the Encoder 1 Stopped type of [Event](#).
- `GenApi::IInteger & EventEncoder1StoppedTimestamp`
Description: Returns the Timestamp of the Encoder 1 Stopped [Event](#).
- `GenApi::IInteger & EventEncoder1StoppedFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped [Event](#).
- `GenApi::IInteger & EventEncoder0Restarted`
Description: Returns the unique Identifier of the Encoder 0 Restarted type of [Event](#).
- `GenApi::IInteger & EventEncoder0RestartedTimestamp`
Description: Returns the Timestamp of the Encoder 0 Restarted [Event](#).
- `GenApi::IInteger & EventEncoder0RestartedFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted [Event](#).
- `GenApi::IInteger & EventEncoder1Restarted`
Description: Returns the unique Identifier of the Encoder 1 Restarted type of [Event](#).
- `GenApi::IInteger & EventEncoder1RestartedTimestamp`
Description: Returns the Timestamp of the Encoder 1 Restarted [Event](#).
- `GenApi::IInteger & EventEncoder1RestartedFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted [Event](#).
- `GenApi::IInteger & EventLine0RisingEdge`
Description: Returns the unique Identifier of the Line 0 Rising Edge type of [Event](#).
- `GenApi::IInteger & EventLine0RisingEdgeTimestamp`
Description: Returns the Timestamp of the Line 0 Rising Edge [Event](#).
- `GenApi::IInteger & EventLine0RisingEdgeFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge [Event](#).
- `GenApi::IInteger & EventLine1RisingEdge`
Description: Returns the unique Identifier of the Line 1 Rising Edge type of [Event](#).
- `GenApi::IInteger & EventLine1RisingEdgeTimestamp`
Description: Returns the Timestamp of the Line 1 Rising Edge [Event](#).
- `GenApi::IInteger & EventLine1RisingEdgeFrameID`
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge [Event](#).

- `GenApi::IInteger & EventLine0FallingEdge`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge [Event](#).
- `GenApi::IInteger & EventLine0FallingEdgeTimestamp`

Description: Returns the unique Identifier of the Line 0 Falling Edge type of [Event](#).
- `GenApi::IInteger & EventLine0FallingEdgeFrameID`

Description: Returns the Timestamp of the Line 0 Falling Edge [Event](#).
- `GenApi::IInteger & EventLine1FallingEdge`

Description: Returns the unique Identifier of the Line 0 Falling Edge type of [Event](#).
- `GenApi::IInteger & EventLine1FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Falling Edge [Event](#).
- `GenApi::IInteger & EventLine1FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge [Event](#).
- `GenApi::IInteger & EventLine0AnyEdge`

Description: Returns the unique Identifier of the Line 0 Any Edge type of [Event](#).
- `GenApi::IInteger & EventLine0AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Any Edge [Event](#).
- `GenApi::IInteger & EventLine0AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge [Event](#).
- `GenApi::IInteger & EventLine1AnyEdge`

Description: Returns the unique Identifier of the Line 1 Any Edge type of [Event](#).
- `GenApi::IInteger & EventLine1AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Any Edge [Event](#).
- `GenApi::IInteger & EventLine1AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge [Event](#).
- `GenApi::IInteger & EventLinkTrigger0`

Description: Returns the unique Identifier of the Link Trigger 0 type of [Event](#).
- `GenApi::IInteger & EventLinkTrigger0Timestamp`

Description: Returns the Timestamp of the Link Trigger 0 [Event](#).
- `GenApi::IInteger & EventLinkTrigger0FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 [Event](#).
- `GenApi::IInteger & EventLinkTrigger1`

Description: Returns the unique Identifier of the Link Trigger 1 type of [Event](#).
- `GenApi::IInteger & EventLinkTrigger1Timestamp`

Description: Returns the Timestamp of the Link Trigger 1 [Event](#).
- `GenApi::IInteger & EventLinkTrigger1FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 [Event](#).
- `GenApi::IInteger & EventActionLate`

Description: Returns the unique Identifier of the Action Late type of [Event](#).
- `GenApi::IInteger & EventActionLateTimestamp`

Description: Returns the Timestamp of the Action Late [Event](#).
- `GenApi::IInteger & EventActionLateFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late [Event](#).
- `GenApi::IInteger & EventLinkSpeedChange`

Description: Returns the unique Identifier of the Link Speed Change type of [Event](#).
- `GenApi::IInteger & EventLinkSpeedChangeTimestamp`

Description: Returns the Timestamp of the Link Speed Change [Event](#).
- `GenApi::IInteger & EventLinkSpeedChangeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change [Event](#).
- `GenApi::IRegister & FileAccessBuffer`

- **GenApi::IInteger & SourceCount**
Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
- **GenApi::IEnumerationT< SourceSelectorEnums > & SourceSelector**
Description: Controls or returns the number of sources supported by the device.
- **GenApi::IEnumerationT< TransferSelectorEnums > & TransferSelector**
Description: Selects the source to control.
- **GenApi::IEnumerationT< TransferBurstCount > & TransferBurstCount**
Description: Selects which stream transfers are currently controlled by the selected Transfer features.
- **GenApi::IInteger & 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::IInteger & 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::IInteger & 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::IInteger & ChunkPixelDynamicRangeMin](#)
Description: Returns the minimum value of dynamic range of the image included in the payload.
- [GenApi::IInteger & ChunkPixelDynamicRangeMax](#)
Description: Returns the maximum value of dynamic range of the image included in the payload.
- [GenApi::IInteger & ChunkTimestampLatchValue](#)
Description: Returns the last Timestamp latched with the TimestampLatch command.
- [GenApi::IInteger & ChunkLineStatusAll](#)
Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.
- [GenApi::IEnumerationT< ChunkCounterSelectorEnums > & ChunkCounterSelector](#)
Description: Selects which counter to retrieve data from.
- [GenApi::IInteger & ChunkCounterValue](#)
Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.
- [GenApi::IEnumerationT< 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::IEnumerationT< ChunkEncoderSelectorEnums > & ChunkEncoderSelector](#)
Description: Selects which Encoder to retrieve data from.
- [GenApi::IInteger & ChunkScanLineSelector](#)
Description: Index for vector representation of one chunk value per line in an image.
- [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.
- [GenApi::IEnumerationT< ChunkEncoderStatusEnums > & ChunkEncoderStatus](#)
Description: Returns the motion status of the selected encoder.
- [GenApi::IEnumerationT< ChunkExposureTimeSelectorEnums > & ChunkExposureTimeSelector](#)
Description: Selects which exposure time is read by the ChunkExposureTime feature.
- [GenApi::IInteger & ChunkLinePitch](#)
Description: Returns the LinePitch of the image included in the payload.
- [GenApi::IEnumerationT< ChunkSourceIDEnums > & ChunkSourceID](#)
Description: Returns the identifier of Source that the image comes from.
- [GenApi::IEnumerationT< ChunkRegionIDEnums > & ChunkRegionID](#)

- **GenApi::IInteger & ChunkTransferBlockID**

Description: Returns the unique identifier of the transfer block used to transport the payload.
- **GenApi::IEnumerationT< ChunkTransferStreamIDEnums > & ChunkTransferStreamID**

Description: Returns identifier of the stream that generated this block.
- **GenApi::IInteger & ChunkTransferQueueCurrentBlockCount**

Description: Returns the current number of blocks in the transfer queue.
- **GenApi::IInteger & ChunkStreamChannelID**

Description: Returns identifier of the stream channel used to carry the block.
- **GenApi::IEnumerationT< ChunkScan3dDistanceUnitEnums > & ChunkScan3dDistanceUnit**

Description: Returns the Distance Unit of the payload image.
- **GenApi::IEnumerationT< ChunkScan3dOutputModeEnums > & ChunkScan3dOutputMode**

Description: Returns the Calibrated Mode of the payload image.
- **GenApi::IEnumerationT< ChunkScan3dCoordinateSystemEnums > & ChunkScan3dCoordinateSystem**

Description: Returns the Coordinate System of the image included in the payload.
- **GenApi::IEnumerationT< ChunkScan3dCoordinateSystemReferenceEnums > & ChunkScan3dCoordinateSystemReference**

Description: Returns the Coordinate System Position of the image included in the payload.
- **GenApi::IEnumerationT< 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::IInteger & GevSCZoneCount

Description: Reports the number of zones per block transmitted on the selected stream channel.
- GenApi::IInteger & 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::IInteger & aPAUSEMACCtrlFramesTransmitted

Description: Reports the number of transmitted PAUSE frames.
- GenApi::IInteger & aPAUSEMACCtrlFramesReceived

Description: Reports the number of received PAUSE frames.
- GenApi::IEnumerationT< CICodecConfigurationEnums > & CICodecConfiguration

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::IInteger & 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::IInteger & CxpConnectionTestErrorCount](#)

Description: Reports the current connection error count for test packets received by the device on the connection selected by CxpConnectionSelector.

- [GenApi::IInteger & CxpConnectionTestPacketCount](#)

Description: Reports the current count for test packets received 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::IInteger & 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

10.12.1 Detailed Description

The camera object class.

10.12.2 Constructor & Destructor Documentation

10.12.2.1 ~Camera()

[~Camera \(\)](#)

10.12.2.2 Camera()

`Camera () [protected]`

10.12.3 Member Function Documentation

10.12.3.1 Init()

`void Init () [virtual]`

Implements [ICameraBase](#).

10.12.4 Member Data Documentation

10.12.4.1 AasRoiEnable

`GenApi::IBoolean& AasRoiEnable`

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithm Selector feature.

Visibility:

10.12.4.2 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:

10.12.4.3 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:

10.12.4.4 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:

10.12.4.5 AasRoiWidth

`GenApi::IInteger& AasRoiWidth`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

10.12.4.6 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

10.12.4.7 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

10.12.4.8 AcquisitionBurstFrameCount

`GenApi::IInteger& 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:

10.12.4.9 AcquisitionFrameCount

`GenApi::IInteger& AcquisitionFrameCount`

Description:

Number of images to acquire during a multi frame acquisition.

Visibility:

10.12.4.10 AcquisitionFrameRate

`GenApi::IFloat& AcquisitionFrameRate`

Description: User controlled acquisition frame rate in Hertz Visibility:

10.12.4.11 AcquisitionFrameRateEnable

`GenApi::IBoolean& AcquisitionFrameRateEnable`

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

Visibility:

10.12.4.12 AcquisitionLineRate

`GenApi::IFloat& AcquisitionLineRate`

Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.

Visibility:

10.12.4.13 AcquisitionMode

```
GenApi::IEnumerationT<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:

10.12.4.14 AcquisitionResultingFrameRate

```
GenApi::IFloat& 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:

10.12.4.15 AcquisitionStart

```
GenApi:: ICommand& AcquisitionStart
```

Description: This command starts the acquisition of images.

Visibility:

10.12.4.16 AcquisitionStatus

```
GenApi::IBoolean& AcquisitionStatus
```

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.

Visibility: Expert

10.12.4.17 AcquisitionStatusSelector

```
GenApi::IEnumerationT<AcquisitionStatusSelectorEnums>& AcquisitionStatusSelector
```

Description: Selects the internal acquisition signal to read using AcquisitionStatus.

Visibility: Expert

10.12.4.18 AcquisitionStop

```
GenApi:: ICommand& AcquisitionStop
```

Description: This command stops the acquisition of images.

Visibility:

10.12.4.19 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

10.12.4.20 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

10.12.4.21 ActionGroupMask

`GenApi::IInteger& ActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

10.12.4.22 ActionQueueSize

`GenApi::IInteger& 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

10.12.4.23 ActionSelector

`GenApi::IInteger& ActionSelector`

Description: Selects to which Action Signal further Action settings apply.

Visibility: Guru

10.12.4.24 ActionUnconditionalMode

`GenApi::IEnumerationT<ActionUnconditionalModeEnums>& ActionUnconditionalMode`

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

Visibility: Guru

10.12.4.25 AdaptiveCompressionEnable

```
GenApi::IBoolean& AdaptiveCompressionEnable
```

Description: Controls whether lossless compression adapts to the image content.

If disabled, a fixed encoding table is used. Visibility:

10.12.4.26 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:

10.12.4.27 aPAUSEMACCtrlFramesReceived

```
GenApi::IInteger& aPAUSEMACCtrlFramesReceived
```

Description: Reports the number of received PAUSE frames.

Visibility: Guru

10.12.4.28 aPAUSEMACCtrlFramesTransmitted

```
GenApi::IInteger& aPAUSEMACCtrlFramesTransmitted
```

Description: Reports the number of transmitted PAUSE frames.

Visibility: Guru

10.12.4.29 AutoAlgorithmSelector

```
GenApi::IEnumerationT<AutoAlgorithmSelectorEnums>& AutoAlgorithmSelector
```

Description:

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

Visibility:

10.12.4.30 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:

10.12.4.31 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:

10.12.4.32 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:

10.12.4.33 AutoExposureExposureTimeLowerLimit

`GenApi::IFloat& AutoExposureExposureTimeLowerLimit`

Description:

The smallest exposure time that auto exposure can set.

Visibility:

10.12.4.34 AutoExposureExposureTimeUpperLimit

`GenApi::IFloat& AutoExposureExposureTimeUpperLimit`

Description:

The largest exposure time that auto exposure can set.

Visibility:

10.12.4.35 AutoExposureGainLowerLimit

`GenApi::IFloat& AutoExposureGainLowerLimit`

Description:

The smallest gain that auto exposure can set.

Visibility:

10.12.4.36 AutoExposureGainUpperLimit

`GenApi::IFloat& AutoExposureGainUpperLimit`

Description:

The largest gain that auto exposure can set.

Visibility:

10.12.4.37 AutoExposureGreyValueLowerLimit

`GenApi::IFloat& AutoExposureGreyValueLowerLimit`

Description:

The lowest value in percentage that the target mean may reach.

Visibility:

10.12.4.38 AutoExposureGreyValueUpperLimit

`GenApi::IFloat & AutoExposureGreyValueUpperLimit`

Description:

The highest value in percentage that the target mean may reach.

Visibility:

10.12.4.39 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:

10.12.4.40 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:

10.12.4.41 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:

10.12.4.42 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:

10.12.4.43 BalanceRatio

```
GenApi::IFloat& BalanceRatio
```

Description:

Controls the balance ratio of the selected color relative to green.

Used for white balancing.

Visibility:

10.12.4.44 BalanceRatioSelector

```
GenApi::IEnumerationT<BalanceRatioSelectorEnums>& BalanceRatioSelector
```

Description:

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

Visibility:

10.12.4.45 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:

10.12.4.46 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:

10.12.4.47 BalanceWhiteAutoLowerLimit

```
GenApi::IFloat& BalanceWhiteAutoLowerLimit
```

Description:

Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.

Visibility:

10.12.4.48 BalanceWhiteAutoProfile

```
GenApi::IEnumerationT<BalanceWhiteAutoProfileEnums>& BalanceWhiteAutoProfile
```

Description: Selects the profile used by BalanceWhiteAuto.

Visibility:

10.12.4.49 BalanceWhiteAutoUpperLimit

```
GenApi::IFloat& BalanceWhiteAutoUpperLimit
```

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

Visibility:

10.12.4.50 BinningHorizontal

```
GenApi::IInteger& 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:

10.12.4.51 BinningHorizontalMode

```
GenApi::IEnumerationT<BinningHorizontalModeEnums>& BinningHorizontalMode
```

Description: Visibility:

10.12.4.52 BinningSelector

```
GenApi::IEnumerationT<BinningSelectorEnums>& BinningSelector
```

Description:

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

Visibility:

10.12.4.53 BinningVertical

`GenApi::IInteger& 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:

10.12.4.54 BinningVerticalMode

`GenApi::IEnumerationT<BinningVerticalModeEnums>& BinningVerticalMode`

Description: Visibility:

10.12.4.55 BlackLevel

`GenApi::IFloat& BlackLevel`

Description:

Controls the offset of the video signal in percent.

Visibility:

10.12.4.56 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

10.12.4.57 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

10.12.4.58 BlackLevelClampingEnable

```
GenApi::IBoolean& BlackLevelClampingEnable
```

Description:

Enable the black level auto clamping feature which performing dark current compensation.

Visibility:

10.12.4.59 BlackLevelRaw

```
GenApi::IInteger& BlackLevelRaw
```

Description:

Controls the offset of the video signal in camera specific units.

Visibility:

10.12.4.60 BlackLevelSelector

```
GenApi::IEnumerationT<BlackLevelSelectorEnums>& BlackLevelSelector
```

Description:

Selects which black level to control.

Only All can be set by the user. Analog and Digital are read-only.

Visibility:

10.12.4.61 ChunkBlackLevel

```
GenApi::IFloat& ChunkBlackLevel
```

Description: Returns the black level used to capture the image.

Visibility:

10.12.4.62 ChunkBlackLevelSelector

```
GenApi::IEnumerationT<ChunkBlackLevelSelectorEnums>& ChunkBlackLevelSelector
```

Description: Selects which black level to retrieve Visibility:

10.12.4.63 ChunkCounterSelector

`GenApi::IEnumerationT<ChunkCounterSelectorEnums>& ChunkCounterSelector`

Description: Selects which counter to retrieve data from.

Visibility: Expert

10.12.4.64 ChunkCounterValue

`GenApi::IInteger& ChunkCounterValue`

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

10.12.4.65 ChunkCRC

`GenApi::IInteger& ChunkCRC`

Description: Returns the CRC of the image payload.

Visibility:

10.12.4.66 ChunkEnable

`GenApi::IBoolean& ChunkEnable`

Description: Enables the inclusion of the selected Chunk data in the payload of the image.

Visibility:

10.12.4.67 ChunkEncoderSelector

`GenApi::IEnumerationT<ChunkEncoderSelectorEnums>& ChunkEncoderSelector`

Description: Selects which Encoder to retrieve data from.

Visibility: Expert

10.12.4.68 ChunkEncoderStatus

`GenApi::IEnumerationT<ChunkEncoderStatusEnums>& ChunkEncoderStatus`

Description: Returns the motion status of the selected encoder.

Visibility: Expert

10.12.4.69 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

10.12.4.70 ChunkExposureEndLineStatusAll

```
GenApi::IInteger& ChunkExposureEndLineStatusAll
```

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

10.12.4.71 ChunkExposureTime

```
GenApi::IFloat& ChunkExposureTime
```

Description: Returns the exposure time used to capture the image.

Visibility:

10.12.4.72 ChunkExposureTimeSelector

```
GenApi::IEnumerationT<ChunkExposureTimeSelectorEnums>& ChunkExposureTimeSelector
```

Description: Selects which exposure time is read by the ChunkExposureTime feature.

Visibility: Expert

10.12.4.73 ChunkFrameID

```
GenApi::IInteger& ChunkFrameID
```

Description: Returns the image frame ID.

Visibility:

10.12.4.74 ChunkGain

```
GenApi::IFloat& ChunkGain
```

Description: Returns the gain used to capture the image.

Visibility:

10.12.4.75 ChunkGainSelector

```
GenApi::IEnumerationT<ChunkGainSelectorEnums>& ChunkGainSelector
```

Description: Selects which gain to retrieve Visibility:

10.12.4.76 ChunkHeight

```
GenApi::IInteger& ChunkHeight
```

Description: Returns the height of the image included in the payload.

Visibility:

10.12.4.77 ChunkImage

```
GenApi::IInteger& ChunkImage
```

Description: Returns the image payload.

Visibility:

10.12.4.78 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

10.12.4.79 ChunkInferenceBoundingBoxResult

```
GenApi::IRegister& ChunkInferenceBoundingBoxResult
```

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

10.12.4.80 ChunkInferenceConfidence

```
GenApi::IFloat& ChunkInferenceConfidence
```

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

10.12.4.81 ChunkInferenceResult

```
GenApi::IInteger& ChunkInferenceResult
```

Description: Returns the chunk data inference result.

Visibility: Expert

10.12.4.82 ChunkLinePitch

```
GenApi::IInteger& ChunkLinePitch
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

10.12.4.83 ChunkLineStatusAll

```
GenApi::IInteger& ChunkLineStatusAll
```

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

10.12.4.84 ChunkModeActive

```
GenApi::IBoolean& ChunkModeActive
```

Description: Activates the inclusion of Chunk data in the payload of the image.

Visibility:

10.12.4.85 ChunkOffsetX

```
GenApi::IInteger& ChunkOffsetX
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

10.12.4.86 ChunkOffsetY

```
GenApi::IInteger& ChunkOffsetY
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

10.12.4.87 ChunkPartSelector

`GenApi::IInteger& ChunkPartSelector`

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

10.12.4.88 ChunkPixelDynamicRangeMax

`GenApi::IInteger& ChunkPixelDynamicRangeMax`

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

10.12.4.89 ChunkPixelDynamicRangeMin

`GenApi::IInteger& ChunkPixelDynamicRangeMin`

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

10.12.4.90 ChunkPixelFormat

`GenApi::IEnumerationT<ChunkPixelFormatEnums>& ChunkPixelFormat`

Description: Format of the pixel provided by the camera Visibility:

10.12.4.91 ChunkRegionID

`GenApi::IEnumerationT<ChunkRegionIDEnums>& ChunkRegionID`

Description: Returns the identifier of Region that the image comes from.

Visibility: Expert

10.12.4.92 ChunkScan3dAxisMax

`GenApi::IFloat& ChunkScan3dAxisMax`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.12.4.93 ChunkScan3dAxisMin

```
GenApi::IFloat& ChunkScan3dAxisMin
```

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.12.4.94 ChunkScan3dCoordinateOffset

```
GenApi::IFloat& ChunkScan3dCoordinateOffset
```

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.12.4.95 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

10.12.4.96 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

10.12.4.97 ChunkScan3dCoordinateScale

```
GenApi::IFloat& ChunkScan3dCoordinateScale
```

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.12.4.98 ChunkScan3dCoordinateSelector

```
GenApi::IEnumerationT<ChunkScan3dCoordinateSelectorEnums>& ChunkScan3dCoordinateSelector
```

Description: Selects which Coordinate to retrieve data from.

Visibility: Expert

10.12.4.99 ChunkScan3dCoordinateSystem

```
GenApi::IEnumerationT<ChunkScan3dCoordinateSystemEnums>& ChunkScan3dCoordinateSystem
```

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

Visibility: Expert

10.12.4.100 ChunkScan3dCoordinateSystemReference

```
GenApi::IEnumerationT<ChunkScan3dCoordinateSystemReferenceEnums>& ChunkScan3dCoordinateSystemReference
```

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

Visibility: Expert

10.12.4.101 ChunkScan3dCoordinateTransformSelector

```
GenApi::IEnumerationT<ChunkScan3dCoordinateTransformSelectorEnums>& ChunkScan3dCoordinateTransformSelector
```

Description: Selector for transform values.

Visibility: Expert

10.12.4.102 ChunkScan3dDistanceUnit

```
GenApi::IEnumerationT<ChunkScan3dDistanceUnitEnums>& ChunkScan3dDistanceUnit
```

Description: Returns the Distance Unit of the payload image.

Visibility: Expert

10.12.4.103 ChunkScan3dInvalidDataFlag

```
GenApi::IBoolean& ChunkScan3dInvalidDataFlag
```

Description: Returns if a specific non-valid data flag is used in the data stream.

Visibility: Expert

10.12.4.104 ChunkScan3dInvalidDataValue

```
GenApi::IFloat& ChunkScan3dInvalidDataValue
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

10.12.4.105 ChunkScan3dOutputMode

```
GenApi::IEnumerationT<ChunkScan3dOutputModeEnums>& ChunkScan3dOutputMode
```

Description: Returns the Calibrated Mode of the payload image.

Visibility: Expert

10.12.4.106 ChunkScan3dTransformValue

```
GenApi::IFloat& ChunkScan3dTransformValue
```

Description: Returns the transform value.

Visibility: Expert

10.12.4.107 ChunkScanLineSelector

```
GenApi::IInteger& ChunkScanLineSelector
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

10.12.4.108 ChunkSelector

```
GenApi::IEnumerationT<ChunkSelectorEnums>& ChunkSelector
```

Description: Selects which chunk data to enable or disable.

Visibility:

10.12.4.109 ChunkSequencerSetActive

```
GenApi::IInteger& ChunkSequencerSetActive
```

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

10.12.4.110 ChunkSerialData

```
GenApi::IString& ChunkSerialData
```

Description: Returns the serial data that was received.

Visibility:

10.12.4.111 ChunkSerialDataLength

```
GenApi::IInteger& ChunkSerialDataLength
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

10.12.4.112 ChunkSerialReceiveOverflow

```
GenApi::IBoolean& ChunkSerialReceiveOverflow
```

Description: Returns the status of the chunk serial receive overflow.

Visibility:

10.12.4.113 ChunkSourceID

```
GenApi::IEnumerationT<ChunkSourceIDEnums>& ChunkSourceID
```

Description: Returns the identifier of Source that the image comes from.

Visibility: Expert

10.12.4.114 ChunkStreamChannelID

```
GenApi::IInteger& ChunkStreamChannelID
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

10.12.4.115 ChunkTimerSelector

```
GenApi::IEnumerationT<ChunkTimerSelectorEnums>& ChunkTimerSelector
```

Description: Selects which Timer to retrieve data from.

Visibility: Expert

10.12.4.116 ChunkTimerValue

```
GenApi::IFloat& ChunkTimerValue
```

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

10.12.4.117 ChunkTimestamp

```
GenApi::IInteger& ChunkTimestamp
```

Description: Returns the Timestamp of the image.

Visibility:

10.12.4.118 ChunkTimestampLatchValue

```
GenApi::IInteger& ChunkTimestampLatchValue
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

10.12.4.119 ChunkTransferBlockID

```
GenApi::IInteger& ChunkTransferBlockID
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

10.12.4.120 ChunkTransferQueueCurrentBlockCount

```
GenApi::IInteger& ChunkTransferQueueCurrentBlockCount
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

10.12.4.121 ChunkTransferStreamID

```
GenApi::IEnumerationT<ChunkTransferStreamIDEnums>& ChunkTransferStreamID
```

Description: Returns identifier of the stream that generated this block.

Visibility: Expert

10.12.4.122 ChunkWidth

```
GenApi::IInteger& ChunkWidth
```

Description: Returns the width of the image included in the payload.

Visibility:

10.12.4.123 CConfiguration

GenApi::IEnumerationT<CConfigurationEnums>& CConfiguration

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

10.12.4.124 CTImeSlotsCount

GenApi::IEnumerationT<CTimeSlotsCountEnums>& CTImeSlotsCount

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

10.12.4.125 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:

10.12.4.126 ColorTransformationSelector

GenApi::IEnumerationT<ColorTransformationSelectorEnums>& ColorTransformationSelector

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

Visibility:

10.12.4.127 ColorTransformationValue

GenApi::IFloat& ColorTransformationValue

Description:

Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

Visibility:

10.12.4.128 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:

10.12.4.129 CompressionRatio

```
GenApi::IFloat& CompressionRatio
```

Description: Reports the ratio between the uncompressed image size and compressed image size.

Visibility:

10.12.4.130 CounterDelay

```
GenApi::IInteger& CounterDelay
```

Description: Sets the delay (or number of events) before the CounterStart event is generated.

Visibility:

10.12.4.131 CounterDuration

```
GenApi::IInteger& CounterDuration
```

Description: Sets the duration (or number of events) before the CounterEnd event is generated.

Visibility:

10.12.4.132 CounterEventActivation

```
GenApi::IEnumerationT<CounterEventActivationEnums>& CounterEventActivation
```

Description: Selects the activation mode of the event to increment the Counter.

Visibility:

10.12.4.133 CounterEventSource

```
GenApi::IEnumerationT<CounterEventSourceEnums>& CounterEventSource
```

Description: Selects the event that will increment the counter Visibility:

10.12.4.134 CounterReset

```
GenApi:: ICommand& 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

10.12.4.135 CounterResetActivation

```
GenApi:: IEnumerationT<CounterResetActivationEnums>& CounterResetActivation
```

Description: Selects the Activation mode of the Counter Reset Source signal.

Visibility:

10.12.4.136 CounterResetSource

```
GenApi:: IEnumerationT<CounterResetSourceEnums>& CounterResetSource
```

Description: Selects the signal that will be the source to reset the Counter.

Visibility:

10.12.4.137 CounterSelector

```
GenApi:: IEnumerationT<CounterSelectorEnums>& CounterSelector
```

Description: Selects which counter to configure Visibility:

10.12.4.138 CounterStatus

```
GenApi:: IEnumerationT<CounterStatusEnums>& CounterStatus
```

Description: Returns the current status of the Counter.

Visibility:

10.12.4.139 CounterTriggerActivation

```
GenApi:: IEnumerationT<CounterTriggerActivationEnums>& CounterTriggerActivation
```

Description: Selects the activation mode of the trigger to start the Counter.

Visibility:

10.12.4.140 CounterTriggerSource

```
GenApi::IEnumerationT<CounterTriggerSourceEnums>& CounterTriggerSource
```

Description: Selects the source of the trigger to start the counter Visibility:

10.12.4.141 CounterValue

```
GenApi::IInteger& CounterValue
```

Description: Current counter value Visibility:

10.12.4.142 CounterValueAtReset

```
GenApi::IInteger& CounterValueAtReset
```

Description: Value of the selected Counter when it was reset by a trigger.

Visibility:

10.12.4.143 CxpConnectionSelector

```
GenApi::IInteger& CxpConnectionSelector
```

Description: Selects the CoaXPress physical connection to control.

Visibility: Expert

10.12.4.144 CxpConnectionTestErrorCount

```
GenApi::IInteger& CxpConnectionTestErrorCount
```

Description: Reports the current connection error count for test packets received by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

10.12.4.145 CxpConnectionTestMode

```
GenApi::IEnumerationT<CxpConnectionTestModeEnums>& CxpConnectionTestMode
```

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

Visibility: Expert

10.12.4.146 CxpConnectionTestPacketCount

```
GenApi::IInteger& CxpConnectionTestPacketCount
```

Description: Reports the current count for test packets received by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

10.12.4.147 CxpLinkConfiguration

```
GenApi::IEnumerationT<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

10.12.4.148 CxpLinkConfigurationPreferred

```
GenApi::IEnumerationT<CxpLinkConfigurationPreferredEnums>& CxpLinkConfigurationPreferred
```

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

Visibility: Expert

10.12.4.149 CxpLinkConfigurationStatus

```
GenApi::IEnumerationT<CxpLinkConfigurationStatusEnums>& CxpLinkConfigurationStatus
```

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

Visibility: Beginner

10.12.4.150 CxpPoCxpAuto

```
GenApi:: ICommand& CxpPoCxpAuto
```

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

10.12.4.151 CxpPoCxpStatus

```
GenApi::IEnumerationT<CxpPoCxpStatusEnums>& CxpPoCxpStatus
```

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

Visibility: Expert

10.12.4.152 CxpPoCxpTripReset

```
GenApi:: ICommand& CxpPoCxpTripReset
```

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

Visibility: Expert

10.12.4.153 CxpPoCxpTurnOff

```
GenApi:: ICommand& CxpPoCxpTurnOff
```

Description: Disable Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

10.12.4.154 DecimationHorizontal

```
GenApi:: IInteger& 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:

10.12.4.155 DecimationHorizontalMode

```
GenApi:: IEnumerationT<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:

10.12.4.156 DecimationSelector

```
GenApi::IEnumerationT<DecimationSelectorEnums>& DecimationSelector
```

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

Visibility:

10.12.4.157 DecimationVertical

```
GenApi::IInteger& 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:

10.12.4.158 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:

10.12.4.159 DefectCorrectionMode

```
GenApi::IEnumerationT<DefectCorrectionModeEnums>& DefectCorrectionMode
```

Description: Controls the method used for replacing defective pixels.

Visibility:

10.12.4.160 DefectCorrectStaticEnable

```
GenApi::IBoolean& DefectCorrectStaticEnable
```

Description: Enables/Disables table-based defective pixel correction.

Visibility:

10.12.4.161 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:

10.12.4.162 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:

10.12.4.163 DefectTableCoordinateY

```
GenApi::IInteger& 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:

10.12.4.164 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:

10.12.4.165 DefectTableIndex

`GenApi:: IInteger& DefectTableIndex`

Description:

Controls the offset of the element to access in the defective pixel location table.

Visibility:

10.12.4.166 DefectTablePixelCount

`GenApi:: IInteger& DefectTablePixelCount`

Description:

The number of defective pixel locations in the current table.

Visibility:

10.12.4.167 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:

10.12.4.168 Deinterlacing

`GenApi:: IEnumeration<DeinterlacingEnums>& Deinterlacing`

Description: Controls how the device performs de-interlacing.

Visibility: Beginner

10.12.4.169 DeviceCharacterSet

```
GenApi::IEnumerationT<DeviceCharacterSetEnums>& DeviceCharacterSet
```

Description:

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

Visibility:

10.12.4.170 DeviceClockFrequency

```
GenApi::IFloat& DeviceClockFrequency
```

Description: Returns the frequency of the selected Clock.

Visibility: Expert

10.12.4.171 DeviceClockSelector

```
GenApi::IEnumerationT<DeviceClockSelectorEnums>& DeviceClockSelector
```

Description: Selects the clock frequency to access from the device.

Visibility: Expert

10.12.4.172 DeviceConnectionSelector

```
GenApi::IInteger& DeviceConnectionSelector
```

Description: Selects which Connection of the device to control.

Visibility: Beginner

10.12.4.173 DeviceConnectionSpeed

```
GenApi::IInteger& DeviceConnectionSpeed
```

Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.

10.12.4.174 DeviceConnectionStatus

```
GenApi::IEnumerationT<DeviceConnectionStatusEnums>& DeviceConnectionStatus
```

Description: Indicates the status of the specified Connection.

Visibility: Expert

10.12.4.175 DeviceEventChannelCount

`GenApi::IInteger& DeviceEventChannelCount`

Description:

Indicates the number of event channels supported by the device.

Visibility:

10.12.4.176 DeviceFamilyName

`GenApi::IString& DeviceFamilyName`

Description: Identifier of the product family of the device.

Visibility: Beginner

10.12.4.177 DeviceFeaturePersistenceEnd

`GenApi:: ICommand& DeviceFeaturePersistenceEnd`

Description: Indicate to the device the end of feature persistence.

Visibility: Guru

10.12.4.178 DeviceFeaturePersistenceStart

`GenApi:: ICommand& DeviceFeaturePersistenceStart`

Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.

Visibility: Guru

10.12.4.179 DeviceFirmwareVersion

`GenApi::IString& DeviceFirmwareVersion`

Description: Version of the firmware on the device.

Visibility:

10.12.4.180 DeviceGenCPVersionMajor

`GenApi::IInteger& DeviceGenCPVersionMajor`

Description: Major version of the GenCP protocol supported by the device.

Visibility: Beginner

10.12.4.181 DeviceGenCPVersionMinor

`GenApi::IInteger& DeviceGenCPVersionMinor`

Description: Minor version of the GenCP protocol supported by the device.

Visibility: Beginner

10.12.4.182 DeviceID

`GenApi::IString& DeviceID`

Description: Device identifier (serial number).

Visibility:

10.12.4.183 DeviceIndicatorMode

`GenApi::IEnumerationT<DeviceIndicatorModeEnums>& DeviceIndicatorMode`

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

Visibility:

10.12.4.184 DeviceLinkBandwidthReserve

`GenApi::IFloat& DeviceLinkBandwidthReserve`

Description:

Percentage of streamed data bandwidth reserved for packet resend.

Visibility:

10.12.4.185 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

10.12.4.186 DeviceLinkConnectionCount

`GenApi::IInteger& DeviceLinkConnectionCount`

Description: Returns the number of physical connection of the device used by a particular Link.

Visibility: Beginner

10.12.4.187 DeviceLinkCurrentThroughput

```
GenApi::IInteger& DeviceLinkCurrentThroughput
```

Description: Current bandwidth of streamed data.

Visibility:

10.12.4.188 DeviceLinkHeartbeatMode

```
GenApi::IEnumerationT<DeviceLinkHeartbeatModeEnums>& DeviceLinkHeartbeatMode
```

Description: Activate or deactivate the Link's heartbeat.

Visibility: Expert

10.12.4.189 DeviceLinkHeartbeatTimeout

```
GenApi::IFloat& DeviceLinkHeartbeatTimeout
```

Description: Controls the current heartbeat timeout of the specific Link.

Visibility: Guru

10.12.4.190 DeviceLinkSelector

```
GenApi::IInteger& DeviceLinkSelector
```

Description: Selects which Link of the device to control.

Visibility: Beginner

10.12.4.191 DeviceLinkSpeed

```
GenApi::IInteger& DeviceLinkSpeed
```

Description:

Indicates the speed of transmission negotiated on the specified Link.

(Bps)

Visibility:

10.12.4.192 DeviceLinkThroughputLimit

```
GenApi::IInteger& 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:

10.12.4.193 DeviceLinkThroughputLimitMode

```
GenApi::IEnumeration<DeviceLinkThroughputLimitModeEnums>& DeviceLinkThroughputLimitMode
```

Description: 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. Visibility: Expert

10.12.4.194 DeviceManifestEntrySelector

```
GenApi::IInteger& DeviceManifestEntrySelector
```

Description: Selects the manifest entry to reference.

Visibility: Guru

10.12.4.195 DeviceManifestPrimaryURL

```
GenApi::IString& DeviceManifestPrimaryURL
```

Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

10.12.4.196 DeviceManifestSchemaMajorVersion

```
GenApi::IInteger& DeviceManifestSchemaMajorVersion
```

Description: Indicates the major version number of the schema file of the selected manifest entry.

Visibility: Guru

10.12.4.197 DeviceManifestSchemaMinorVersion

```
GenApi::IInteger& DeviceManifestSchemaMinorVersion
```

Description: Indicates the minor version number of the schema file of the selected manifest entry.

Visibility: Guru

10.12.4.198 DeviceManifestSecondaryURL

```
GenApi::IString& DeviceManifestSecondaryURL
```

Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

10.12.4.199 DeviceManifestXMLMajorVersion

```
GenApi::IInteger& DeviceManifestXMLMajorVersion
```

Description: Indicates the major version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

10.12.4.200 DeviceManifestXMLMinorVersion

```
GenApi::IInteger& DeviceManifestXMLMinorVersion
```

Description: Indicates the minor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

10.12.4.201 DeviceManifestXMLSubMinorVersion

```
GenApi::IInteger& DeviceManifestXMLSubMinorVersion
```

Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

10.12.4.202 DeviceManufacturerInfo

```
GenApi::IString& DeviceManufacturerInfo
```

Description: Manufacturer information about the device.

Visibility:

10.12.4.203 DeviceMaxThroughput

```
GenApi::IInteger& 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:

10.12.4.204 DeviceModelName

```
GenApi::IString& DeviceModelName
```

Description: Model of the device.

Visibility:

10.12.4.205 DevicePowerSupplySelector

```
GenApi::IEnumerationT<DevicePowerSupplySelectorEnums>& DevicePowerSupplySelector
```

Description:

Selects the power supply source to control or read.

Visibility:

10.12.4.206 DeviceRegistersCheck

```
GenApi:: ICommand& DeviceRegistersCheck
```

Description: Perform the validation of the current register set for consistency.

This will update the DeviceRegistersValid flag. Visibility: Expert

10.12.4.207 DeviceRegistersEndianness

```
GenApi::IEnumerationT<DeviceRegistersEndiannessEnums>& DeviceRegistersEndianness
```

Description: Endianess of the registers of the device.

Visibility:

10.12.4.208 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

10.12.4.209 DeviceRegistersStreamingStart

```
GenApi:: ICommand& DeviceRegistersStreamingStart
```

Description: Prepare the device for registers streaming without checking for consistency.

Visibility: Guru

10.12.4.210 DeviceRegistersValid

```
GenApi:: IBoolean& DeviceRegistersValid
```

Description: Returns if the current register set is valid and consistent.

Visibility: Expert

10.12.4.211 DeviceReset

```
GenApi:: ICommand& DeviceReset
```

Description: This is a command that immediately resets and reboots the device.

Visibility:

10.12.4.212 DeviceScanType

```
GenApi:: IEnumeration<DeviceScanTypeEnums>& DeviceScanType
```

Description: Scan type of the sensor of the device.

Visibility:

10.12.4.213 DeviceSerialNumber

```
GenApi:: IString& DeviceSerialNumber
```

Description:

Device's serial number.

This string is a unique identifier of the device.

Visibility:

10.12.4.214 DeviceSerialPortBaudRate

```
GenApi::IEnumerationT<DeviceSerialPortBaudRateEnums>& DeviceSerialPortBaudRate
```

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

Visibility: Expert

10.12.4.215 DeviceSerialPortSelector

```
GenApi::IEnumerationT<DeviceSerialPortSelectorEnums>& DeviceSerialPortSelector
```

Description: Selects which serial port of the device to control.

Visibility: Expert

10.12.4.216 DeviceSFNCVersionMajor

```
GenApi::IInteger& DeviceSFNCVersionMajor
```

Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

10.12.4.217 DeviceSFNCVersionMinor

```
GenApi::IInteger& DeviceSFNCVersionMinor
```

Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

10.12.4.218 DeviceSFNCVersionSubMinor

```
GenApi::IInteger& DeviceSFNCVersionSubMinor
```

Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

10.12.4.219 DeviceStreamChannelCount

```
GenApi::IInteger& DeviceStreamChannelCount
```

Description:

Indicates the number of streaming channels supported by the device.

Visibility:

10.12.4.220 DeviceStreamChannelEndianness

`GenApi::IEnumerationT<DeviceStreamChannelEndiannessEnums>& DeviceStreamChannelEndianness`

Description: Endianess of multi-byte pixel data for this stream.

Visibility: Guru

10.12.4.221 DeviceStreamChannelLink

`GenApi::IInteger& DeviceStreamChannelLink`

Description: Index of device's Link to use for streaming the specified stream channel.

Visibility: Guru

10.12.4.222 DeviceStreamChannelPacketSize

`GenApi::IInteger& 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

10.12.4.223 DeviceStreamChannelSelector

`GenApi::IInteger& DeviceStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility: Expert

10.12.4.224 DeviceStreamChannelType

`GenApi::IEnumerationT<DeviceStreamChannelTypeEnums>& DeviceStreamChannelType`

Description: Reports the type of the stream channel.

Visibility: Guru

10.12.4.225 DeviceTapGeometry

`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.

Visibility: Expert

10.12.4.226 DeviceTemperature

```
GenApi::IFloat& DeviceTemperature
```

Description: Device temperature in degrees Celsius (C).

Visibility:

10.12.4.227 DeviceTemperatureSelector

```
GenApi::IEnumerationT<DeviceTemperatureSelectorEnums>& DeviceTemperatureSelector
```

Description:

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

Visibility:

10.12.4.228 DeviceTLType

```
GenApi::IEnumerationT<DeviceTLTypeEnums>& DeviceTLType
```

Description: Transport Layer type of the device.

Visibility:

10.12.4.229 DeviceTLVersionMajor

```
GenApi::IInteger& DeviceTLVersionMajor
```

Description:

Major version of the Transport Layer of the device.

Visibility:

10.12.4.230 DeviceTLVersionMinor

```
GenApi::IInteger& DeviceTLVersionMinor
```

Description:

Minor version of the Transport Layer of the device.

Visibility:

10.12.4.231 DeviceTLVersionSubMinor

`GenApi::IInteger& DeviceTLVersionSubMinor`

Description: Sub minor version of the Transport Layer of the device.

Visibility: Beginner

10.12.4.232 DeviceType

`GenApi::IEnumerationT<DeviceTypeEnums>& DeviceType`

Description: Returns the device type.

Visibility: Guru

10.12.4.233 DeviceUptime

`GenApi::IInteger& DeviceUptime`

Description: Total time since the device was powered up in seconds.

Visibility:

10.12.4.234 DeviceUserID

`GenApi::IString& DeviceUserID`

Description: User-programmable device identifier.

Visibility:

10.12.4.235 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the manufacturer of the device.

Visibility:

10.12.4.236 DeviceVersion

`GenApi::IString& DeviceVersion`

Description: Version of the device.

Visibility:

10.12.4.237 EncoderDivider

```
GenApi::IInteger& EncoderDivider
```

Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.

Visibility: Expert

10.12.4.238 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

10.12.4.239 EncoderOutputMode

```
GenApi::IEnumerationT<EncoderOutputModeEnums>& EncoderOutputMode
```

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

Visibility: Expert

10.12.4.240 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

10.12.4.241 EncoderResetActivation

```
GenApi::IEnumerationT<EncoderResetActivationEnums>& EncoderResetActivation
```

Description: Selects the Activation mode of the Encoder Reset Source signal.

Visibility: Expert

10.12.4.242 EncoderResetSource

```
GenApi::IEnumerationT<EncoderResetSourceEnums>& EncoderResetSource
```

Description: Selects the signals that will be the source to reset the Encoder.

Visibility: Expert

10.12.4.243 EncoderSelector

```
GenApi::IEnumerationT<EncoderSelectorEnums>& EncoderSelector
```

Description: Selects which Encoder to configure.

Visibility: Expert

10.12.4.244 EncoderSourceA

```
GenApi::IEnumerationT<EncoderSourceAEnums>& EncoderSourceA
```

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

Visibility: Expert

10.12.4.245 EncoderSourceB

```
GenApi::IEnumerationT<EncoderSourceBEnums>& EncoderSourceB
```

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

Visibility: Expert

10.12.4.246 EncoderStatus

```
GenApi::IEnumerationT<EncoderStatusEnums>& EncoderStatus
```

Description: Returns the motion status of the encoder.

Visibility: Expert

10.12.4.247 EncoderTimeout

```
GenApi::IFloat& EncoderTimeout
```

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

Visibility: Expert

10.12.4.248 EncoderValue

```
GenApi::IInteger& EncoderValue
```

Description: Reads or writes the current value of the position counter of the selected Encoder.

Visibility: Expert

10.12.4.249 EncoderValueAtReset

```
GenApi::IInteger& EncoderValueAtReset
```

Description: Reads the value of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

Visibility: Expert

10.12.4.250 EnumerationCount

```
GenApi::IInteger& EnumerationCount
```

Description: Number of enumerations since uptime.

Visibility:

10.12.4.251 EventAcquisitionEnd

```
GenApi::IInteger& EventAcquisitionEnd
```

Description: Returns the unique Identifier of the Acquisition End type of [Event](#).

Visibility: Expert

10.12.4.252 EventAcquisitionEndFrameID

```
GenApi::IInteger& EventAcquisitionEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End [Event](#).

Visibility: Expert

10.12.4.253 EventAcquisitionEndTimestamp

```
GenApi::IInteger& EventAcquisitionEndTimestamp
```

Description: Returns the Timestamp of the Acquisition End [Event](#).

Visibility: Expert

10.12.4.254 EventAcquisitionError

```
GenApi::IInteger& EventAcquisitionError
```

Description: Returns the unique Identifier of the Acquisition Error type of [Event](#).

Visibility: Expert

10.12.4.255 EventAcquisitionErrorFrameID

```
GenApi::IInteger& EventAcquisitionErrorFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error [Event](#).

Visibility: Expert

10.12.4.256 EventAcquisitionErrorTimestamp

```
GenApi::IInteger& EventAcquisitionErrorTimestamp
```

Description: Returns the Timestamp of the Acquisition Error [Event](#).

Visibility: Expert

10.12.4.257 EventAcquisitionStart

```
GenApi::IInteger& EventAcquisitionStart
```

Description: Returns the unique Identifier of the Acquisition Start type of [Event](#).

Visibility: Expert

10.12.4.258 EventAcquisitionStartFrameID

```
GenApi::IInteger& EventAcquisitionStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start [Event](#).

Visibility: Expert

10.12.4.259 EventAcquisitionStartTimestamp

```
GenApi::IInteger& EventAcquisitionStartTimestamp
```

Description: Returns the Timestamp of the Acquisition Start [Event](#).

Visibility: Expert

10.12.4.260 EventAcquisitionTransferEnd

```
GenApi::IInteger& EventAcquisitionTransferEnd
```

Description: Returns the unique Identifier of the Acquisition Transfer End type of [Event](#).

Visibility: Expert

10.12.4.261 EventAcquisitionTransferEndFrameID

```
GenApi::IInteger& EventAcquisitionTransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End [Event](#).

Visibility: Expert

10.12.4.262 EventAcquisitionTransferEndTimestamp

```
GenApi::IInteger& EventAcquisitionTransferEndTimestamp
```

Description: Returns the Timestamp of the Acquisition Transfer End [Event](#).

Visibility: Expert

10.12.4.263 EventAcquisitionTransferStart

```
GenApi::IInteger& EventAcquisitionTransferStart
```

Description: Returns the unique Identifier of the Acquisition Transfer Start type of [Event](#).

Visibility: Expert

10.12.4.264 EventAcquisitionTransferStartFrameID

```
GenApi::IInteger& EventAcquisitionTransferStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start [Event](#).

Visibility: Expert

10.12.4.265 EventAcquisitionTransferStartTimestamp

```
GenApi::IInteger& EventAcquisitionTransferStartTimestamp
```

Description: Returns the Timestamp of the Acquisition Transfer Start [Event](#).

Visibility: Expert

10.12.4.266 EventAcquisitionTrigger

```
GenApi::IInteger& EventAcquisitionTrigger
```

Description: Returns the unique Identifier of the Acquisition Trigger type of [Event](#).

Visibility: Expert

10.12.4.267 EventAcquisitionTriggerFrameID

```
GenApi::IInteger& EventAcquisitionTriggerFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger [Event](#).

Visibility: Expert

10.12.4.268 EventAcquisitionTriggerTimestamp

```
GenApi::IInteger& EventAcquisitionTriggerTimestamp
```

Description: Returns the Timestamp of the Acquisition Trigger [Event](#).

Visibility: Expert

10.12.4.269 EventActionLate

```
GenApi::IInteger& EventActionLate
```

Description: Returns the unique Identifier of the Action Late type of [Event](#).

Visibility: Expert

10.12.4.270 EventActionLateFrameID

```
GenApi::IInteger& EventActionLateFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late [Event](#).

Visibility: Expert

10.12.4.271 EventActionLateTimestamp

```
GenApi::IInteger& EventActionLateTimestamp
```

Description: Returns the Timestamp of the Action Late [Event](#).

Visibility: Expert

10.12.4.272 EventCounter0End

```
GenApi::IInteger& EventCounter0End
```

Description: Returns the unique Identifier of the Counter 0 End type of [Event](#).

Visibility: Expert

10.12.4.273 EventCounter0EndFrameID

```
GenApi::IInteger& EventCounter0EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End [Event](#).

Visibility: Expert

10.12.4.274 EventCounter0EndTimestamp

```
GenApi::IInteger& EventCounter0EndTimestamp
```

Description: Returns the Timestamp of the Counter 0 End [Event](#).

Visibility: Expert

10.12.4.275 EventCounter0Start

```
GenApi::IInteger& EventCounter0Start
```

Description: Returns the unique Identifier of the Counter 0 Start type of [Event](#).

Visibility: Expert

10.12.4.276 EventCounter0StartFrameID

```
GenApi::IInteger& EventCounter0StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start [Event](#).

Visibility: Expert

10.12.4.277 EventCounter0StartTimestamp

```
GenApi::IInteger& EventCounter0StartTimestamp
```

Description: Returns the Timestamp of the Counter 0 Start [Event](#).

Visibility: Expert

10.12.4.278 EventCounter1End

```
GenApi::IInteger& EventCounter1End
```

Description: Returns the unique Identifier of the Counter 1 End type of [Event](#).

Visibility: Expert

10.12.4.279 EventCounter1EndFrameID

```
GenApi::IInteger& EventCounter1EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End [Event](#).

Visibility: Expert

10.12.4.280 EventCounter1EndTimestamp

```
GenApi::IInteger& EventCounter1EndTimestamp
```

Description: Returns the Timestamp of the Counter 1 End [Event](#).

Visibility: Expert

10.12.4.281 EventCounter1Start

```
GenApi::IInteger& EventCounter1Start
```

Description: Returns the unique Identifier of the Counter 1 Start type of [Event](#).

Visibility: Expert

10.12.4.282 EventCounter1StartFrameID

```
GenApi::IInteger& EventCounter1StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start [Event](#).

Visibility: Expert

10.12.4.283 EventCounter1StartTimestamp

```
GenApi::IInteger& EventCounter1StartTimestamp
```

Description: Returns the Timestamp of the Counter 1 Start [Event](#).

Visibility: Expert

10.12.4.284 EventEncoder0Restarted

```
GenApi::IInteger& EventEncoder0Restarted
```

Description: Returns the unique Identifier of the Encoder 0 Restarted type of [Event](#).

Visibility: Expert

10.12.4.285 EventEncoder0RestartedFrameID

```
GenApi::IInteger& EventEncoder0RestartedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted [Event](#).

Visibility: Expert

10.12.4.286 EventEncoder0RestartedTimestamp

```
GenApi::IInteger& EventEncoder0RestartedTimestamp
```

Description: Returns the Timestamp of the Encoder 0 Restarted [Event](#).

Visibility: Expert

10.12.4.287 EventEncoder0Stopped

```
GenApi::IInteger& EventEncoder0Stopped
```

Description: Returns the unique Identifier of the Encoder 0 Stopped type of [Event](#).

Visibility: Expert

10.12.4.288 EventEncoder0StoppedFrameID

```
GenApi::IInteger& EventEncoder0StoppedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped [Event](#).

Visibility: Expert

10.12.4.289 EventEncoder0StoppedTimestamp

```
GenApi::IInteger& EventEncoder0StoppedTimestamp
```

Description: Returns the Timestamp of the Encoder 0 Stopped [Event](#).

Visibility: Expert

10.12.4.290 EventEncoder1Restarted

```
GenApi::IInteger& EventEncoder1Restarted
```

Description: Returns the unique Identifier of the Encoder 1 Restarted type of [Event](#).

Visibility: Expert

10.12.4.291 EventEncoder1RestartedFrameID

```
GenApi::IInteger& EventEncoder1RestartedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted [Event](#).

Visibility: Expert

10.12.4.292 EventEncoder1RestartedTimestamp

```
GenApi::IInteger& EventEncoder1RestartedTimestamp
```

Description: Returns the Timestamp of the Encoder 1 Restarted [Event](#).

Visibility: Expert

10.12.4.293 EventEncoder1Stopped

```
GenApi::IInteger& EventEncoder1Stopped
```

Description: Returns the unique Identifier of the Encoder 1 Stopped type of [Event](#).

Visibility: Expert

10.12.4.294 EventEncoder1StoppedFrameID

```
GenApi::IInteger& EventEncoder1StoppedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped [Event](#).

Visibility: Expert

10.12.4.295 EventEncoder1StoppedTimestamp

```
GenApi::IInteger& EventEncoder1StoppedTimestamp
```

Description: Returns the Timestamp of the Encoder 1 Stopped [Event](#).

Visibility: Expert

10.12.4.296 EventError

```
GenApi::IInteger& EventError
```

Description: Returns the unique identifier of the Error type of [Event](#).

Visibility:

10.12.4.297 EventErrorCode

```
GenApi::IInteger& EventErrorCode
```

Description: Returns the error code for the error that happened Visibility:

10.12.4.298 EventErrorFrameID

```
GenApi::IInteger& EventErrorFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Error [Event](#).

Visibility:

10.12.4.299 EventErrorTimestamp

```
GenApi::IInteger& EventErrorTimestamp
```

Description: Returns the Timestamp of the Error [Event](#).

Visibility:

10.12.4.300 EventExposureEnd

```
GenApi::IInteger& EventExposureEnd
```

Description: Returns the unique identifier of the Exposure End type of [Event](#).

Visibility:

10.12.4.301 EventExposureEndFrameID

```
GenApi::IInteger& EventExposureEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End [Event](#).

Visibility:

10.12.4.302 EventExposureEndTimestamp

```
GenApi::IInteger& EventExposureEndTimestamp
```

Description: Returns the Timestamp of the Exposure End [Event](#).

Visibility:

10.12.4.303 EventExposureStart

```
GenApi::IInteger& EventExposureStart
```

Description: Returns the unique Identifier of the Exposure Start type of [Event](#).

Visibility: Expert

10.12.4.304 EventExposureStartFrameID

```
GenApi::IInteger& EventExposureStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start [Event](#).

Visibility: Expert

10.12.4.305 EventExposureStartTimestamp

```
GenApi::IInteger& EventExposureStartTimestamp
```

Description: Returns the Timestamp of the Exposure Start [Event](#).

Visibility: Expert

10.12.4.306 EventFrameBurstEnd

```
GenApi::IInteger& EventFrameBurstEnd
```

Description: Returns the unique Identifier of the Frame Burst End type of [Event](#).

Visibility: Expert

10.12.4.307 EventFrameBurstEndFrameID

```
GenApi::IInteger& EventFrameBurstEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End [Event](#).

Visibility: Expert

10.12.4.308 EventFrameBurstEndTimestamp

```
GenApi::IInteger& EventFrameBurstEndTimestamp
```

Description: Returns the Timestamp of the Frame Burst End [Event](#).

Visibility: Expert

10.12.4.309 EventFrameBurstStart

```
GenApi::IInteger& EventFrameBurstStart
```

Description: Returns the unique Identifier of the Frame Burst Start type of [Event](#).

Visibility: Expert

10.12.4.310 EventFrameBurstStartFrameID

```
GenApi::IInteger& EventFrameBurstStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start [Event](#).

Visibility: Expert

10.12.4.311 EventFrameBurstStartTimestamp

```
GenApi::IInteger& EventFrameBurstStartTimestamp
```

Description: Returns the Timestamp of the Frame Burst Start [Event](#).

Visibility: Expert

10.12.4.312 EventFrameEnd

```
GenApi::IInteger& EventFrameEnd
```

Description: Returns the unique Identifier of the Frame End type of [Event](#).

Visibility: Expert

10.12.4.313 EventFrameEndFrameID

```
GenApi::IInteger& EventFrameEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End [Event](#).

Visibility: Expert

10.12.4.314 EventFrameEndTimestamp

```
GenApi::IInteger& EventFrameEndTimestamp
```

Description: Returns the Timestamp of the Frame End [Event](#).

Visibility: Expert

10.12.4.315 EventFrameStart

```
GenApi::IInteger& EventFrameStart
```

Description: Returns the unique Identifier of the Frame Start type of [Event](#).

Visibility: Expert

10.12.4.316 EventFrameStartFrameID

```
GenApi::IInteger& EventFrameStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start [Event](#).

Visibility: Expert

10.12.4.317 EventFrameStartTimestamp

```
GenApi::IInteger& EventFrameStartTimestamp
```

Description: Returns the Timestamp of the Frame Start [Event](#).

Visibility: Expert

10.12.4.318 EventFrameTransferEnd

```
GenApi::IInteger& EventFrameTransferEnd
```

Description: Returns the unique Identifier of the Frame Transfer End type of [Event](#).

Visibility: Expert

10.12.4.319 EventFrameTransferEndFrameID

```
GenApi::IInteger& EventFrameTransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End [Event](#).

Visibility: Expert

10.12.4.320 EventFrameTransferEndTimestamp

```
GenApi::IInteger& EventFrameTransferEndTimestamp
```

Description: Returns the Timestamp of the Frame Transfer End [Event](#).

Visibility: Expert

10.12.4.321 EventFrameTransferStart

```
GenApi::IInteger& EventFrameTransferStart
```

Description: Returns the unique Identifier of the Frame Transfer Start type of [Event](#).

Visibility: Expert

10.12.4.322 EventFrameTransferStartFrameID

```
GenApi::IInteger& EventFrameTransferStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start [Event](#).

Visibility: Expert

10.12.4.323 EventFrameTransferStartTimestamp

```
GenApi::IInteger& EventFrameTransferStartTimestamp
```

Description: Returns the Timestamp of the Frame Transfer Start [Event](#).

Visibility: Expert

10.12.4.324 EventFrameTrigger

```
GenApi::IInteger& 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

10.12.4.325 EventFrameTriggerFrameID

```
GenApi::IInteger& EventFrameTriggerFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger [Event](#).

Visibility: Expert

10.12.4.326 EventFrameTriggerTimestamp

```
GenApi::IInteger& EventFrameTriggerTimestamp
```

Description: Returns the Timestamp of the FrameTrigger [Event](#).

It can be used to determine precisely when the event occurred. Visibility: Expert

10.12.4.327 EventLine0AnyEdge

`GenApi::IInteger& EventLine0AnyEdge`

Description: Returns the unique Identifier of the Line 0 Any Edge type of [Event](#).

Visibility: Expert

10.12.4.328 EventLine0AnyEdgeFrameID

`GenApi::IInteger& EventLine0AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge [Event](#).

Visibility: Expert

10.12.4.329 EventLine0AnyEdgeTimestamp

`GenApi::IInteger& EventLine0AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Any Edge [Event](#).

Visibility: Expert

10.12.4.330 EventLine0FallingEdge

`GenApi::IInteger& EventLine0FallingEdge`

Description: Returns the unique Identifier of the Line 0 Falling Edge type of [Event](#).

Visibility: Expert

10.12.4.331 EventLine0FallingEdgeFrameID

`GenApi::IInteger& EventLine0FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge [Event](#).

Visibility: Expert

10.12.4.332 EventLine0FallingEdgeTimestamp

`GenApi::IInteger& EventLine0FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Falling Edge [Event](#).

Visibility: Expert

10.12.4.333 EventLine0RisingEdge

```
GenApi::IInteger& EventLine0RisingEdge
```

Description: Returns the unique Identifier of the Line 0 Rising Edge type of [Event](#).

Visibility: Expert

10.12.4.334 EventLine0RisingEdgeFrameID

```
GenApi::IInteger& EventLine0RisingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge [Event](#).

Visibility: Expert

10.12.4.335 EventLine0RisingEdgeTimestamp

```
GenApi::IInteger& EventLine0RisingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 0 Rising Edge [Event](#).

Visibility: Expert

10.12.4.336 EventLine1AnyEdge

```
GenApi::IInteger& EventLine1AnyEdge
```

Description: Returns the unique Identifier of the Line 1 Any Edge type of [Event](#).

Visibility: Expert

10.12.4.337 EventLine1AnyEdgeFrameID

```
GenApi::IInteger& EventLine1AnyEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge [Event](#).

Visibility: Expert

10.12.4.338 EventLine1AnyEdgeTimestamp

```
GenApi::IInteger& EventLine1AnyEdgeTimestamp
```

Description: Returns the Timestamp of the Line 1 Any Edge [Event](#).

Visibility: Expert

10.12.4.339 EventLine1FallingEdge

`GenApi::IInteger& EventLine1FallingEdge`

Description: Returns the unique Identifier of the Line 1 Falling Edge type of [Event](#).

Visibility: Expert

10.12.4.340 EventLine1FallingEdgeFrameID

`GenApi::IInteger& EventLine1FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge [Event](#).

Visibility: Expert

10.12.4.341 EventLine1FallingEdgeTimestamp

`GenApi::IInteger& EventLine1FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Falling Edge [Event](#).

Visibility: Expert

10.12.4.342 EventLine1RisingEdge

`GenApi::IInteger& EventLine1RisingEdge`

Description: Returns the unique Identifier of the Line 1 Rising Edge type of [Event](#).

Visibility: Expert

10.12.4.343 EventLine1RisingEdgeFrameID

`GenApi::IInteger& EventLine1RisingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge [Event](#).

Visibility: Expert

10.12.4.344 EventLine1RisingEdgeTimestamp

`GenApi::IInteger& EventLine1RisingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Rising Edge [Event](#).

Visibility: Expert

10.12.4.345 EventLinkSpeedChange

```
GenApi::IInteger& EventLinkSpeedChange
```

Description: Returns the unique Identifier of the Link Speed Change type of [Event](#).

Visibility: Expert

10.12.4.346 EventLinkSpeedChangeFrameID

```
GenApi::IInteger& EventLinkSpeedChangeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change [Event](#).

Visibility: Expert

10.12.4.347 EventLinkSpeedChangeTimestamp

```
GenApi::IInteger& EventLinkSpeedChangeTimestamp
```

Description: Returns the Timestamp of the Link Speed Change [Event](#).

Visibility: Expert

10.12.4.348 EventLinkTrigger0

```
GenApi::IInteger& EventLinkTrigger0
```

Description: Returns the unique Identifier of the Link Trigger 0 type of [Event](#).

Visibility: Expert

10.12.4.349 EventLinkTrigger0FrameID

```
GenApi::IInteger& EventLinkTrigger0FrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 [Event](#).

Visibility: Expert

10.12.4.350 EventLinkTrigger0Timestamp

```
GenApi::IInteger& EventLinkTrigger0Timestamp
```

Description: Returns the Timestamp of the Link Trigger 0 [Event](#).

Visibility: Expert

10.12.4.351 EventLinkTrigger1

```
GenApi::IInteger& EventLinkTrigger1
```

Description: Returns the unique Identifier of the Link Trigger 1 type of [Event](#).

Visibility: Expert

10.12.4.352 EventLinkTrigger1FrameID

```
GenApi::IInteger& EventLinkTrigger1FrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 [Event](#).

Visibility: Expert

10.12.4.353 EventLinkTrigger1Timestamp

```
GenApi::IInteger& EventLinkTrigger1Timestamp
```

Description: Returns the Timestamp of the Link Trigger 1 [Event](#).

Visibility: Expert

10.12.4.354 EventNotification

```
GenApi::IEnumerationT<EventNotificationEnums>& EventNotification
```

Description: Enables/Disables the selected event.

Visibility:

10.12.4.355 EventSelector

```
GenApi::IEnumerationT<EventSelectorEnums>& EventSelector
```

Description: Selects which [Event](#) to enable or disable.

Visibility:

10.12.4.356 EventSequencerSetChange

```
GenApi::IInteger& EventSequencerSetChange
```

Description: Returns the unique Identifier of the Sequencer Set Change type of [Event](#).

Visibility: Expert

10.12.4.357 EventSequencerSetChangeFrameID

```
GenApi::IInteger& EventSequencerSetChangeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change [Event](#).

Visibility: Expert

10.12.4.358 EventSequencerSetChangeTimestamp

```
GenApi::IInteger& EventSequencerSetChangeTimestamp
```

Description: Returns the Timestamp of the Sequencer Set Change [Event](#).

Visibility: Expert

10.12.4.359 EventSerialData

```
GenApi::IString& EventSerialData
```

Description: Returns the serial data that was received.

Visibility:

10.12.4.360 EventSerialDataLength

```
GenApi::IInteger& EventSerialDataLength
```

Description: Returns the length of the received serial data that was included in the event payload.

Visibility:

10.12.4.361 EventSerialPortReceive

```
GenApi::IInteger& EventSerialPortReceive
```

Description: Returns the unique identifier of the Serial Port Receive type of [Event](#).

Visibility:

10.12.4.362 EventSerialPortReceiveTimestamp

```
GenApi::IInteger& EventSerialPortReceiveTimestamp
```

Description: Returns the Timestamp of the Serial Port Receive [Event](#).

Visibility:

10.12.4.363 EventSerialReceiveOverflow

```
GenApi::IBoolean& EventSerialReceiveOverflow
```

Description: Returns the status of the event serial receive overflow.

Visibility:

10.12.4.364 EventStream0TransferBlockEnd

```
GenApi::IInteger& EventStream0TransferBlockEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of [Event](#).

Visibility: Expert

10.12.4.365 EventStream0TransferBlockEndFrameID

```
GenApi::IInteger& EventStream0TransferBlockEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End [Event](#).

Visibility: Expert

10.12.4.366 EventStream0TransferBlockEndTimestamp

```
GenApi::IInteger& EventStream0TransferBlockEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block End [Event](#).

Visibility: Expert

10.12.4.367 EventStream0TransferBlockStart

```
GenApi::IInteger& EventStream0TransferBlockStart
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of [Event](#).

Visibility: Expert

10.12.4.368 EventStream0TransferBlockStartFrameID

```
GenApi::IInteger& EventStream0TransferBlockStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start [Event](#).

Visibility: Expert

10.12.4.369 EventStream0TransferBlockStartTimestamp

```
GenApi::IInteger& EventStream0TransferBlockStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block Start [Event](#).

Visibility: Expert

10.12.4.370 EventStream0TransferBlockTrigger

```
GenApi::IInteger& EventStream0TransferBlockTrigger
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of [Event](#).

Visibility: Expert

10.12.4.371 EventStream0TransferBlockTriggerFrameID

```
GenApi::IInteger& EventStream0TransferBlockTriggerFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger [Event](#).

Visibility: Expert

10.12.4.372 EventStream0TransferBlockTriggerTimestamp

```
GenApi::IInteger& EventStream0TransferBlockTriggerTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger [Event](#).

Visibility: Expert

10.12.4.373 EventStream0TransferBurstEnd

```
GenApi::IInteger& EventStream0TransferBurstEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of [Event](#).

Visibility: Expert

10.12.4.374 EventStream0TransferBurstEndFrameID

```
GenApi::IInteger& EventStream0TransferBurstEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End [Event](#).

Visibility: Expert

10.12.4.375 EventStream0TransferBurstEndTimestamp

`GenApi::IInteger& EventStream0TransferBurstEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst End [Event](#).

Visibility: Expert

10.12.4.376 EventStream0TransferBurstStart

`GenApi::IInteger& EventStream0TransferBurstStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of [Event](#).

Visibility: Expert

10.12.4.377 EventStream0TransferBurstStartFrameID

`GenApi::IInteger& EventStream0TransferBurstStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start [Event](#).

Visibility: Expert

10.12.4.378 EventStream0TransferBurstStartTimestamp

`GenApi::IInteger& EventStream0TransferBurstStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start [Event](#).

Visibility: Expert

10.12.4.379 EventStream0TransferEnd

`GenApi::IInteger& EventStream0TransferEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer End type of [Event](#).

Visibility: Expert

10.12.4.380 EventStream0TransferEndFrameID

`GenApi::IInteger& EventStream0TransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End [Event](#).

Visibility: Expert

10.12.4.381 EventStream0TransferEndTimestamp

```
GenApi::IInteger& EventStream0TransferEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer End [Event](#).

Visibility: Expert

10.12.4.382 EventStream0TransferOverflow

```
GenApi::IInteger& EventStream0TransferOverflow
```

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of [Event](#).

Visibility: Expert

10.12.4.383 EventStream0TransferOverflowFrameID

```
GenApi::IInteger& EventStream0TransferOverflowFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow [Event](#).

Visibility: Expert

10.12.4.384 EventStream0TransferOverflowTimestamp

```
GenApi::IInteger& EventStream0TransferOverflowTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Overflow [Event](#).

Visibility: Expert

10.12.4.385 EventStream0TransferPause

```
GenApi::IInteger& EventStream0TransferPause
```

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of [Event](#).

Visibility: Expert

10.12.4.386 EventStream0TransferPauseFrameID

```
GenApi::IInteger& EventStream0TransferPauseFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause [Event](#).

Visibility: Expert

10.12.4.387 EventStream0TransferPauseTimestamp

`GenApi::IInteger& EventStream0TransferPauseTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Pause [Event](#).

Visibility: Expert

10.12.4.388 EventStream0TransferResume

`GenApi::IInteger& EventStream0TransferResume`

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of [Event](#).

Visibility: Expert

10.12.4.389 EventStream0TransferResumeFrameID

`GenApi::IInteger& EventStream0TransferResumeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume [Event](#).

Visibility: Expert

10.12.4.390 EventStream0TransferResumeTimestamp

`GenApi::IInteger& EventStream0TransferResumeTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Resume [Event](#).

Visibility: Expert

10.12.4.391 EventStream0TransferStart

`GenApi::IInteger& EventStream0TransferStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of [Event](#).

Visibility: Expert

10.12.4.392 EventStream0TransferStartFrameID

`GenApi::IInteger& EventStream0TransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start [Event](#).

Visibility: Expert

10.12.4.393 EventStream0TransferStartTimestamp

```
GenApi::IInteger& EventStream0TransferStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Start [Event](#).

Visibility: Expert

10.12.4.394 EventTest

```
GenApi::IInteger& EventTest
```

Description: Returns the unique identifier of the Test type of [Event](#).

Visibility:

10.12.4.395 EventTestTimestamp

```
GenApi::IInteger& EventTestTimestamp
```

Description: Returns the Timestamp of the Test [Event](#).

Visibility:

10.12.4.396 EventTimer0End

```
GenApi::IInteger& EventTimer0End
```

Description: Returns the unique Identifier of the Timer 0 End type of [Event](#).

Visibility: Expert

10.12.4.397 EventTimer0EndFrameID

```
GenApi::IInteger& EventTimer0EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End [Event](#).

Visibility: Expert

10.12.4.398 EventTimer0EndTimestamp

```
GenApi::IInteger& EventTimer0EndTimestamp
```

Description: Returns the Timestamp of the Timer 0 End [Event](#).

Visibility: Expert

10.12.4.399 EventTimer0Start

```
GenApi::IInteger& EventTimer0Start
```

Description: Returns the unique Identifier of the Timer 0 Start type of [Event](#).

Visibility: Expert

10.12.4.400 EventTimer0StartFrameID

```
GenApi::IInteger& EventTimer0StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start [Event](#).

Visibility: Expert

10.12.4.401 EventTimer0StartTimestamp

```
GenApi::IInteger& EventTimer0StartTimestamp
```

Description: Returns the Timestamp of the Timer 0 Start [Event](#).

Visibility: Expert

10.12.4.402 EventTimer1End

```
GenApi::IInteger& EventTimer1End
```

Description: Returns the unique Identifier of the Timer 1 End type of [Event](#).

Visibility: Expert

10.12.4.403 EventTimer1EndFrameID

```
GenApi::IInteger& EventTimer1EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End [Event](#).

Visibility: Expert

10.12.4.404 EventTimer1EndTimestamp

```
GenApi::IInteger& EventTimer1EndTimestamp
```

Description: Returns the Timestamp of the Timer 1 End [Event](#).

Visibility: Expert

10.12.4.405 EventTimer1Start

`GenApi::IInteger& EventTimer1Start`

Description: Returns the unique Identifier of the Timer 1 Start type of [Event](#).

Visibility: Expert

10.12.4.406 EventTimer1StartFrameID

`GenApi::IInteger& EventTimer1StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start [Event](#).

Visibility: Expert

10.12.4.407 EventTimer1StartTimestamp

`GenApi::IInteger& EventTimer1StartTimestamp`

Description: Returns the Timestamp of the Timer 1 Start [Event](#).

Visibility: Expert

10.12.4.408 ExposureActiveMode

`GenApi::IEnumerationT<ExposureActiveModeEnums>& ExposureActiveMode`

Description: Control sensor active exposure mode.

Visibility:

10.12.4.409 ExposureAuto

`GenApi::IEnumerationT<ExposureAutoEnums>& ExposureAuto`

Description: Sets the automatic exposure mode Visibility:

10.12.4.410 ExposureMode

`GenApi::IEnumerationT<ExposureModeEnums>& ExposureMode`

Description:

Sets the operation mode of the Exposure.

Visibility:

10.12.4.411 ExposureTime

`GenApi::IFloat& ExposureTime`

Description:

Exposure time in microseconds when Exposure Mode is Timed.

Visibility:

10.12.4.412 ExposureTimeMode

`GenApi::IEnumerationT<ExposureTimeModeEnums>& ExposureTimeMode`

Description: Sets the configuration mode of the ExposureTime feature.

Visibility: Beginner

10.12.4.413 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

10.12.4.414 FactoryReset

`GenApi:: ICommand& FactoryReset`

Description: Returns all user tables to factory default Visibility:

10.12.4.415 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

10.12.4.416 FileAccessLength

`GenApi::IInteger& FileAccessLength`

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

10.12.4.417 FileAccessOffset

`GenApi::IInteger& FileAccessOffset`

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

10.12.4.418 FileMode

`GenApi::IEnumerationT<FileModeEnums>& FileMode`

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:

10.12.4.419 FileOperationExecute

`GenApi:: ICommand& FileOperationExecute`

Description:

This is a command that executes the selected file operation on the selected file.

Visibility:

10.12.4.420 FileOperationResult

`GenApi::IInteger& FileOperationResult`

Description: Represents the file operation result.

For Read or Write operations, the number of successfully read/written bytes is returned. Visibility:

10.12.4.421 FileOperationSelector

`GenApi::IEnumerationT<FileOperationSelectorEnums>& FileOperationSelector`

Description:

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

Visibility:

10.12.4.422 FileOperationStatus

`GenApi::IEnumerationT<FileOperationStatusEnums>& FileOperationStatus`

Description: Represents the file operation execution status.

Visibility:

10.12.4.423 FileSelector

`GenApi::IEnumerationT<FileSelectorEnums>& FileSelector`

Description:

Selects which file is being operated on.

This must be set before performing any file operations.

Visibility:

10.12.4.424 FileSize

`GenApi::IIInteger& FileSize`

Description: Represents the size of the selected file in bytes.

Visibility:

10.12.4.425 Gain

`GenApi::IFloat& Gain`

Description:

Controls the amplification of the video signal in dB.

Visibility:

10.12.4.426 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:

10.12.4.427 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

10.12.4.428 GainSelector

```
GenApi::IEnumerationT<GainSelectorEnums>& GainSelector
```

Description: Selects which gain to control.

The All selection is a total amplification across all channels (or taps).

Visibility:

10.12.4.429 Gamma

```
GenApi::IFloat& Gamma
```

Description: Controls the gamma correction of pixel intensity.

Visibility:

10.12.4.430 GammaEnable

```
GenApi::IBoolean& GammaEnable
```

Description: Enables/disables gamma correction.

Visibility:

10.12.4.431 GevActiveLinkCount

```
GenApi::IInteger& GevActiveLinkCount
```

Description: Indicates the current number of active logical links.

Visibility: Expert

10.12.4.432 GevCCP

```
GenApi::IEnumerationT<GevCCPEnums>& GevCCP
```

Description: Controls the device access privilege of an application.

Visibility:

10.12.4.433 GevCurrentDefaultGateway

```
GenApi::IInteger& GevCurrentDefaultGateway
```

Description: Reports the default gateway IP address to be used on the given logical link.

Visibility:

10.12.4.434 GevCurrentIPAddress

```
GenApi::IInteger& GevCurrentIPAddress
```

Description: Reports the IP address for the given logical link.

Visibility:

10.12.4.435 GevCurrentIPConfigurationDHCP

```
GenApi::IBoolean& GevCurrentIPConfigurationDHCP
```

Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.

Visibility:

10.12.4.436 GevCurrentIPConfigurationLLA

```
GenApi::IBoolean& GevCurrentIPConfigurationLLA
```

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Visibility:

10.12.4.437 GevCurrentIPConfigurationPersistentIP

```
GenApi::IBoolean& GevCurrentIPConfigurationPersistentIP
```

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.

Visibility:

10.12.4.438 GevCurrentPhysicalLinkConfiguration

```
GenApi::IEnumerationT<GevCurrentPhysicalLinkConfigurationEnums>& GevCurrentPhysicalLinkConfiguration
```

Description: Indicates the current physical link configuration of the device.

Visibility: Expert

10.12.4.439 GevCurrentSubnetMask

```
GenApi::IInteger& GevCurrentSubnetMask
```

Description: Reports the subnet mask of the given logical link.

Visibility:

10.12.4.440 GevDiscoveryAckDelay

```
GenApi::IInteger& GevDiscoveryAckDelay
```

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

Visibility: Expert

10.12.4.441 GevFirstURL

```
GenApi::IString& GevFirstURL
```

Description: The first choice of URL for the XML device description file.

Visibility:

10.12.4.442 GevGVCPExtendedStatusCodes

```
GenApi::IBoolean& GevGVCPExtendedStatusCodes
```

Description: Enables the generation of extended status codes.

Visibility: Guru

10.12.4.443 GevGVCPExtendedStatusCodesSelector

```
GenApi::IEnumerationT<GevGVCPExtendedStatusCodesSelectorEnums>& GevGVCPExtendedStatusCodes←
Selector
```

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

Visibility: Guru

10.12.4.444 GevGVCPHeartbeatDisable

```
GenApi::IBoolean& GevGVCPHeartbeatDisable
```

Description: Disables the GVCP heartbeat.

Visibility:

10.12.4.445 GevGVCPPendingAck

`GenApi::IBoolean& GevGVCPPendingAck`

Description: Enables the generation of PENDING_ACK.

Visibility:

10.12.4.446 GevGVCPPendingTimeout

`GenApi::IInteger& GevGVCPPendingTimeout`

Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.

Visibility:

10.12.4.447 GevGVSPExtendedIDMode

`GenApi::IEnumerationT<GevGVSPExtendedIDModeEnums>& GevGVSPExtendedIDMode`

Description: Enables the extended IDs mode.

Visibility: Expert

10.12.4.448 GevHeartbeatTimeout

`GenApi::IInteger& GevHeartbeatTimeout`

Description: Indicates the current heartbeat timeout in milliseconds.

Visibility:

10.12.4.449 GevIEEE1588

`GenApi::IBoolean& GevIEEE1588`

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

Visibility:

10.12.4.450 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:

10.12.4.451 GevIEEE1588Mode

```
GenApi::IEnumerationT<GevIEEE1588ModeEnums>& GevIEEE1588Mode
```

Description: Provides the mode of the IEEE 1588 clock.

Visibility:

10.12.4.452 GevIEEE1588Status

```
GenApi::IEnumerationT<GevIEEE1588StatusEnums>& GevIEEE1588Status
```

Description: Provides the status of the IEEE 1588 clock.

Visibility:

10.12.4.453 GevInterfaceSelector

```
GenApi::IInteger& GevInterfaceSelector
```

Description: Selects which logical link to control.

Visibility:

10.12.4.454 GevIPConfigurationStatus

```
GenApi::IEnumerationT<GevIPConfigurationStatusEnums>& GevIPConfigurationStatus
```

Description: Reports the current IP configuration status.

Visibility: Beginner

10.12.4.455 GevMACAddress

```
GenApi::IInteger& GevMACAddress
```

Description: MAC address of the logical link.

Visibility:

10.12.4.456 GevMCDA

```
GenApi::IInteger& GevMCDA
```

Description: Controls the destination IP address of the message channel

10.12.4.457 GevMCPHostPort

`GenApi::IInteger& GevMCPHostPort`

Description: The port to which the device must send messages Visibility:

10.12.4.458 GevMCRC

`GenApi::IInteger& GevMCRC`

Description: Indicates the number of retries of the message channel.

Visibility:

10.12.4.459 GevMCSP

`GenApi::IInteger& GevMCSP`

Description: Indicates the source port of the message channel.

Visibility:

10.12.4.460 GevMCTT

`GenApi::IInteger& GevMCTT`

Description: Indicates the transmission timeout of the message channel.

Visibility:

10.12.4.461 GevNumberOfInterfaces

`GenApi::IInteger& GevNumberOfInterfaces`

Description: Indicates the number of physical network interfaces supported by this device.

Visibility:

10.12.4.462 GevPAUSEFrameReception

`GenApi::IBoolean& GevPAUSEFrameReception`

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

Visibility: Expert

10.12.4.463 GevPAUSEFrameTransmission

```
GenApi::IBoolean& GevPAUSEFrameTransmission
```

Description: Controls whether PAUSE Frames can be generated on the given logical link.

Visibility: Expert

10.12.4.464 GevPersistentDefaultGateway

```
GenApi::IInteger& GevPersistentDefaultGateway
```

Description: Controls the persistent default gateway for this logical link.

Visibility:

10.12.4.465 GevPersistentIPAddress

```
GenApi::IInteger& GevPersistentIPAddress
```

Description: Controls the Persistent IP address for this logical link.

Visibility:

10.12.4.466 GevPersistentSubnetMask

```
GenApi::IInteger& GevPersistentSubnetMask
```

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.

Visibility:

10.12.4.467 GevPhysicalLinkConfiguration

```
GenApi::IEnumerationT<GevPhysicalLinkConfigurationEnums>& GevPhysicalLinkConfiguration
```

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

Visibility: Expert

10.12.4.468 GevPrimaryApplicationIPAddress

```
GenApi::IInteger& GevPrimaryApplicationIPAddress
```

Description: Returns the address of the primary application.

Visibility: Guru

10.12.4.469 GevPrimaryApplicationSocket

`GenApi::IInteger& GevPrimaryApplicationSocket`

Description: Returns the UDP source port of the primary application.

Visibility: Guru

10.12.4.470 GevPrimaryApplicationSwitchoverKey

`GenApi::IInteger& GevPrimaryApplicationSwitchoverKey`

Description: Controls the key to use to authenticate primary application switchover requests.

Visibility: Guru

10.12.4.471 GevSCCFGAllInTransmission

`GenApi::IBoolean& GevSCCFGAllInTransmission`

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

Visibility: Guru

10.12.4.472 GevSCCFGExtendedChunkData

`GenApi::IBoolean& GevSCCFGExtendedChunkData`

Description: Enables cameras to use the extended chunk data payload type for this stream channel.

Visibility:

10.12.4.473 GevSCCFGPacketResendDestination

`GenApi::IBoolean& 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

10.12.4.474 GevSCCFGUnconditionalStreaming

`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).

Visibility:

10.12.4.475 GevSCDA

```
GenApi::IInteger& 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:

10.12.4.476 GevSCPD

```
GenApi::IInteger& 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:

10.12.4.477 GevSCPDirection

```
GenApi::IInteger& GevSCPDirection
```

Description: Transmit or Receive of the channel Visibility:

10.12.4.478 GevSCPHostPort

```
GenApi::IInteger& 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:

10.12.4.479 GevSCPIfaceIndex

```
GenApi::IInteger& GevSCPIfaceIndex
```

Description: Index of the logical link to use.

Visibility:

10.12.4.480 GevSCPSBigEndian

```
GenApi::IBoolean& GevSCPSBigEndian
```

Description: Endianess of multi-byte pixel data for this stream.

Visibility:

10.12.4.481 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:

10.12.4.482 GevSCPSFireTestPacket

`GenApi::IBoolean& GevSCPSFireTestPacket`

Description: Sends a test packet.

Visibility:

10.12.4.483 GevSCSPacketSize

`GenApi::IInteger& GevSCSPacketSize`

Description: Specifies the stream packet size (in bytes) to send on this channel.

Visibility:

10.12.4.484 GevSCSP

`GenApi::IInteger& GevSCSP`

Description: Indicates the source port of the stream channel.

Visibility:

10.12.4.485 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

10.12.4.486 GevSCZoneCount

`GenApi::IInteger& GevSCZoneCount`

Description: Reports the number of zones per block transmitted on the selected stream channel.

Visibility: Guru

10.12.4.487 GevSCZoneDirectionAll

```
GenApi::IInteger& GevSCZoneDirectionAll
```

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.

Visibility: Guru

10.12.4.488 GevSecondURL

```
GenApi::IString& GevSecondURL
```

Description: The second choice of URL to the XML device description file.

Visibility:

10.12.4.489 GevStreamChannelSelector

```
GenApi::IInteger& GevStreamChannelSelector
```

Description: Selects the stream channel to control.

Visibility:

10.12.4.490 GevSupportedOption

```
GenApi::IBoolean& GevSupportedOption
```

Description: Returns if the selected GEV option is supported.

Visibility:

10.12.4.491 GevSupportedOptionSelector

```
GenApi::IEnumerationT<GevSupportedOptionSelectorEnums>& GevSupportedOptionSelector
```

Description: Selects the GEV option to interrogate for existing support.

Visibility:

10.12.4.492 GevTimestampTickFrequency

```
GenApi::IInteger& GevTimestampTickFrequency
```

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

Visibility:

10.12.4.493 GuiXmlManifestAddress

`GenApi::IInteger& GuiXmlManifestAddress`

Description: Location of the GUI XML manifest table.

Visibility:

10.12.4.494 Height

`GenApi::IInteger& Height`

Description:

Height of the image provided by the device (in pixels).

Visibility:

10.12.4.495 HeightMax

`GenApi::IInteger& 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:

10.12.4.496 ImageComponentEnable

`GenApi::IBoolean& ImageComponentEnable`

Description: Controls if the selected component streaming is active.

Visibility: Beginner

10.12.4.497 ImageComponentSelector

`GenApi::IEnumerationT<ImageComponentSelectorEnums>& ImageComponentSelector`

Description: Selects a component to activate data streaming from.

Visibility: Beginner

10.12.4.498 ImageCompressionBitrate

`GenApi::IFloat& ImageCompressionBitrate`

Description: Control the rate of the produced compressed stream.

Visibility: Expert

10.12.4.499 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

10.12.4.500 ImageCompressionMode

```
GenApi::IEnumerationT<ImageCompressionModeEnums>& ImageCompressionMode
```

Description: Visibility:

10.12.4.501 ImageCompressionQuality

```
GenApi::IInteger& ImageCompressionQuality
```

Description: Control the quality of the produced compressed stream.

Visibility: Expert

10.12.4.502 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

10.12.4.503 IspEnable

```
GenApi::IBoolean& IspEnable
```

Description:

Controls whether the image processing core is used for optional pixel format mode (i.e.

mono).

Visibility:

10.12.4.504 LineFilterWidth

```
GenApi::IFloat& LineFilterWidth
```

Description: Filter width in microseconds for the selected line and filter combination Visibility:

10.12.4.505 LineFormat

```
GenApi::IEnumerationT<LineFormatEnums>& LineFormat
```

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

Visibility:

10.12.4.506 LineInputFilterSelector

```
GenApi::IEnumerationT<LineInputFilterSelectorEnums>& LineInputFilterSelector
```

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

Visibility:

10.12.4.507 LineInverter

```
GenApi::IBoolean& LineInverter
```

Description: Controls the inversion of the signal of the selected input or output line.

Visibility:

10.12.4.508 LineMode

```
GenApi::IEnumerationT<LineModeEnums>& LineMode
```

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

Visibility:

10.12.4.509 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

10.12.4.510 LineSelector

```
GenApi::IEnumerationT<LineSelectorEnums>& LineSelector
```

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:

10.12.4.511 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:

10.12.4.512 LineStatus

```
GenApi::IBoolean& LineStatus
```

Description: Returns the current status of the selected input or output Line Visibility:

10.12.4.513 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:

10.12.4.514 LinkErrorCount

```
GenApi::IInteger& LinkErrorCount
```

Description: Counts the number of error on the link.

Visibility:

10.12.4.515 LinkUptime

```
GenApi::IInteger& LinkUptime
```

Description: Time since the last phy negotiation (enumeration).

Visibility:

10.12.4.516 LogicBlockLUTInputActivation

```
GenApi::IEnumerationT<LogicBlockLUTInputActivationEnums>& LogicBlockLUTInputActivation
```

Description: Selects the activation mode of the Logic Input Source signal.

Visibility:

10.12.4.517 LogicBlockLUTInputSelector

```
GenApi::IEnumerationT<LogicBlockLUTInputSelectorEnums>& LogicBlockLUTInputSelector
```

Description: Controls which LogicBlockLUT Input Source & Activation to access.

Visibility:

10.12.4.518 LogicBlockLUTInputSource

```
GenApi::IEnumerationT<LogicBlockLUTInputSourceEnums>& LogicBlockLUTInputSource
```

Description: Selects the source for the input into the Logic LUT.

Visibility:

10.12.4.519 LogicBlockLUTOutputValue

```
GenApi::IBoolean& LogicBlockLUTOutputValue
```

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

Visibility:

10.12.4.520 LogicBlockLUTOutputValueAll

```
GenApi::IInteger& LogicBlockLUTOutputValueAll
```

Description: Sets the value of all the output bits in the selected LUT.

Visibility:

10.12.4.521 LogicBlockLUTRowIndex

```
GenApi::IInteger& LogicBlockLUTRowIndex
```

Description: Controls the row of the truth table to access in the selected LUT.

Visibility:

10.12.4.522 LogicBlockLUTSelector

```
GenApi::IEnumerationT<LogicBlockLUTSelectorEnums>& LogicBlockLUTSelector
```

Description: Selects which LogicBlock LUT to configure Visibility:

10.12.4.523 LogicBlockSelector

```
GenApi::IEnumerationT<LogicBlockSelectorEnums>& LogicBlockSelector
```

Description: Selects which LogicBlock to configure Visibility:

10.12.4.524 LUTEnable

```
GenApi::IBoolean& LUTEnable
```

Description:

Activates the selected LUT.

Visibility:

10.12.4.525 LUTIndex

```
GenApi::IInteger& LUTIndex
```

Description:

Control the index (offset) of the coefficient to access in the selected LUT.

Visibility:

10.12.4.526 LUTSelector

```
GenApi::IEnumerationT<LUTSelectorEnums>& LUTSelector
```

Description:

Selects which LUT to control.

Visibility:

10.12.4.527 LUTValue

```
GenApi::IInteger& LUTValue
```

Description:

Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.

Visibility:

10.12.4.528 LUTValueAll

```
GenApi::IRegister& LUTValueAll
```

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.

Visibility: Guru

10.12.4.529 MaxDeviceResetTime

```
GenApi::IInteger& MaxDeviceResetTime
```

Description: Time to wait until device reset complete (ms).

Visibility:

10.12.4.530 OffsetX

```
GenApi::IInteger& OffsetX
```

Description:

Horizontal offset from the origin to the ROI (in pixels).

Visibility:

10.12.4.531 OffsetY

```
GenApi::IInteger& OffsetY
```

Description:

Vertical offset from the origin to the ROI (in pixels).

Visibility:

10.12.4.532 PacketResendRequestCount

```
GenApi::IInteger& PacketResendRequestCount
```

Description: Counts the number of resend requests received from the host.

Visibility:

10.12.4.533 PayloadSize

```
GenApi::IInteger& PayloadSize
```

Description: Provides the number of bytes transferred for each image or chunk on the stream channel.

Visibility:

10.12.4.534 PixelColorFilter

```
GenApi::IEnumeration<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:

10.12.4.535 PixelDynamicRangeMax

```
GenApi::IInteger& 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:

10.12.4.536 PixelDynamicRangeMin

```
GenApi::IInteger& 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:

10.12.4.537 PixelFormat

```
GenApi::IEnumerationT<PixelFormatEnums>& PixelFormat
```

Description: Format of the pixel provided by the camera.

Visibility:

10.12.4.538 PixelFormatInfoID

```
GenApi::IInteger& PixelFormatInfoID
```

Description: Returns the value used by the streaming channels to identify the selected pixel format.

Visibility: Guru

10.12.4.539 PixelFormatInfoSelector

```
GenApi::IEnumerationT<PixelFormatInfoSelectorEnums>& PixelFormatInfoSelector
```

Description: Select the pixel format for which the information will be returned.

Visibility: Guru

10.12.4.540 PixelSize

```
GenApi::IEnumerationT<PixelSizeEnums>& PixelSize
```

Description: Total size in bits of a pixel of the image.

Visibility:

10.12.4.541 PowerSupplyCurrent

```
GenApi::IFloat& PowerSupplyCurrent
```

Description:

Indicates the output current of the selected power supply (A).

Visibility:

10.12.4.542 PowerSupplyVoltage

```
GenApi::IFloat& PowerSupplyVoltage
```

Description:

Indicates the current voltage of the selected power supply (V).

Visibility:

10.12.4.543 RegionDestination

```
GenApi::IEnumerationT<RegionDestinationEnums>& RegionDestination
```

Description: Control the destination of the selected region.

Visibility: Expert

10.12.4.544 RegionMode

```
GenApi::IEnumerationT<RegionModeEnums>& RegionMode
```

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

Visibility: Beginner

10.12.4.545 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

10.12.4.546 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:

10.12.4.547 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:

10.12.4.548 RgbTransformLightSource

```
GenApi::IEnumeration<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:

10.12.4.549 Saturation

```
GenApi::IFloat& Saturation
```

Description: Controls the color saturation.

Visibility:

10.12.4.550 SaturationEnable

```
GenApi::IBoolean& SaturationEnable
```

Description: Enables/disables Saturation adjustment.

Visibility:

10.12.4.551 Scan3dAxisMax

```
GenApi::IFloat& Scan3dAxisMax
```

Description: Maximum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

10.12.4.552 Scan3dAxisMin

```
GenApi::IFloat& Scan3dAxisMin
```

Description: Minimum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

10.12.4.553 Scan3dCoordinateOffset

```
GenApi::IFloat& Scan3dCoordinateOffset
```

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

10.12.4.554 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

10.12.4.555 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

10.12.4.556 Scan3dCoordinateScale

```
GenApi::IFloat& Scan3dCoordinateScale
```

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

10.12.4.557 Scan3dCoordinateSelector

```
GenApi::IEnumerationT<Scan3dCoordinateSelectorEnums>& Scan3dCoordinateSelector
```

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

Visibility: Expert

10.12.4.558 Scan3dCoordinateSystem

```
GenApi::IEnumerationT<Scan3dCoordinateSystemEnums>& Scan3dCoordinateSystem
```

Description: Specifies the Coordinate system to use for the device.

Visibility: Beginner

10.12.4.559 Scan3dCoordinateSystemReference

```
GenApi::IEnumerationT<Scan3dCoordinateSystemReferenceEnums>& Scan3dCoordinateSystemReference
```

Description: Defines coordinate system reference location.

Visibility: Expert

10.12.4.560 Scan3dCoordinateTransformSelector

```
GenApi::IEnumerationT<Scan3dCoordinateTransformSelectorEnums>& Scan3dCoordinateTransformSelector
```

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

Visibility: Expert

10.12.4.561 Scan3dDistanceUnit

```
GenApi::IEnumerationT<Scan3dDistanceUnitEnums>& Scan3dDistanceUnit
```

Description: Specifies the unit used when delivering calibrated distance data.

Visibility: Beginner

10.12.4.562 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

10.12.4.563 Scan3dInvalidDataValue

```
GenApi::IFloat& Scan3dInvalidDataValue
```

Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.

Visibility: Expert

10.12.4.564 Scan3dOutputMode

```
GenApi::IEnumerationT<Scan3dOutputModeEnums>& Scan3dOutputMode
```

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

Visibility: Expert

10.12.4.565 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

10.12.4.566 SensorDescription

```
GenApi::IString& SensorDescription
```

Description: Returns Sensor Description Visibility:

10.12.4.567 SensorDigitizationTaps

```
GenApi::IEnumerationT<SensorDigitizationTapsEnums>& SensorDigitizationTaps
```

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

Visibility: Expert

10.12.4.568 SensorHeight

```
GenApi::IInteger& SensorHeight
```

Description: Effective height of the sensor in pixels.

Visibility:

10.12.4.569 SensorShutterMode

```
GenApi::IEnumerationT<SensorShutterModeEnums>& SensorShutterMode
```

Description: Sets the shutter mode of the device.

Visibility:

10.12.4.570 SensorTaps

```
GenApi::IEnumerationT<SensorTapsEnums>& SensorTaps
```

Description: Number of taps of the camera sensor.

Visibility: Expert

10.12.4.571 SensorWidth

```
GenApi::IInteger& SensorWidth
```

Description: Effective width of the sensor in pixels.

Visibility:

10.12.4.572 SequencerConfigurationMode

```
GenApi::IEnumerationT<SequencerConfigurationModeEnums>& SequencerConfigurationMode
```

Description:

Controls whether or not a sequencer is in configuration mode.

Visibility:

10.12.4.573 SequencerConfigurationValid

```
GenApi::IEnumerationT<SequencerConfigurationValidEnums>& SequencerConfigurationValid
```

Description:

Display whether the current sequencer configuration is valid to run.

Visibility:

10.12.4.574 SequencerFeatureEnable

```
GenApi::IBoolean& SequencerFeatureEnable
```

Description:

Enables the selected feature and makes it active in all sequencer sets.

Visibility:

10.12.4.575 SequencerMode

```
GenApi::IEnumerationT<SequencerModeEnums>& SequencerMode
```

Description: Controls whether or not a sequencer is active.

Visibility:

10.12.4.576 SequencerPathSelector

```
GenApi::IInteger& SequencerPathSelector
```

Description:

Selects branching path to be used for subsequent settings.

Visibility:

10.12.4.577 SequencerSetActive

```
GenApi::IInteger& SequencerSetActive
```

Description: Displays the currently active sequencer set.

Visibility:

10.12.4.578 SequencerSetLoad

```
GenApi:: ICommand& SequencerSetLoad
```

Description:

Loads currently selected sequencer to the current device configuration.

Visibility:

10.12.4.579 SequencerSetNext

```
GenApi::IInteger& SequencerSetNext
```

Description: Specifies the next sequencer set.

Visibility:

10.12.4.580 SequencerSetSave

```
GenApi:: ICommand& SequencerSetSave
```

Description:

Saves the current device configuration to the currently selected sequencer set.

Visibility:

10.12.4.581 SequencerSetSelector

`GenApi::IInteger& SequencerSetSelector`

Description:

Selects the sequencer set to which subsequent settings apply.

Visibility:

10.12.4.582 SequencerSetStart

`GenApi::IInteger& SequencerSetStart`

Description: Sets the first sequencer set to be used.

Visibility:

10.12.4.583 SequencerSetValid

`GenApi::IEnumerationT<SequencerSetValidEnums>& SequencerSetValid`

Description:

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

Visibility:

10.12.4.584 SequencerTriggerActivation

`GenApi::IEnumerationT<SequencerTriggerActivationEnums>& SequencerTriggerActivation`

Description:

Specifies the activation mode of the sequencer trigger.

Visibility:

10.12.4.585 SequencerTriggerSource

`GenApi::IEnumerationT<SequencerTriggerSourceEnums>& SequencerTriggerSource`

Description:

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

Visibility:

10.12.4.586 SerialPortBaudRate

```
GenApi::IEnumerationT<SerialPortBaudRateEnums>& SerialPortBaudRate
```

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

Visibility:

10.12.4.587 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:

10.12.4.588 SerialPortParity

```
GenApi::IEnumerationT<SerialPortParityEnums>& SerialPortParity
```

Description: This feature controls the parity used by the selected serial port.

Visibility:

10.12.4.589 SerialPortSelector

```
GenApi::IEnumerationT<SerialPortSelectorEnums>& SerialPortSelector
```

Description: Selects which serial port of the device to control.

Visibility:

10.12.4.590 SerialPortSource

```
GenApi::IEnumerationT<SerialPortSourceEnums>& SerialPortSource
```

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

Visibility:

10.12.4.591 SerialPortStopBits

```
GenApi::IEnumerationT<SerialPortStopBitsEnums>& SerialPortStopBits
```

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

Visibility:

10.12.4.592 SerialReceiveFramingErrorCount

```
GenApi::IInteger& SerialReceiveFramingErrorCount
```

Description: Returns the number of framing errors that have occurred on the serial port.

Visibility:

10.12.4.593 SerialReceiveParityErrorCount

```
GenApi::IInteger& SerialReceiveParityErrorCount
```

Description: Returns the number of parity errors that have occurred on the serial port.

Visibility:

10.12.4.594 SerialReceiveQueueClear

```
GenApi:: ICommand& SerialReceiveQueueClear
```

Description: This is a command that clears the device serial port receive queue.

Visibility:

10.12.4.595 SerialReceiveQueueCurrentCharacterCount

```
GenApi::IInteger& SerialReceiveQueueCurrentCharacterCount
```

Description: Returns the number of characters currently in the serial port receive queue.

Visibility:

10.12.4.596 SerialReceiveQueueMaxCharacterCount

```
GenApi::IInteger& SerialReceiveQueueMaxCharacterCount
```

Description: >Returns the maximum number of characters in the serial port receive queue.

Visibility:

10.12.4.597 SerialTransmitQueueCurrentCharacterCount

```
GenApi::IInteger& SerialTransmitQueueCurrentCharacterCount
```

Description: Returns the number of characters currently in the serial port transmit queue.

Visibility:

10.12.4.598 SerialTransmitQueueMaxCharacterCount

`GenApi::IInteger& SerialTransmitQueueMaxCharacterCount`

Description: >Returns the maximum number of characters in the serial port transmit queue.

Visibility:

10.12.4.599 Sharpening

`GenApi::IFloat& 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:

10.12.4.600 SharpeningAuto

`GenApi::IBoolean& 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:

10.12.4.601 SharpeningEnable

`GenApi::IBoolean& SharpeningEnable`

Description:

Enables/disables the sharpening feature.

Sharpening is disabled by default.

Visibility:

10.12.4.602 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:

10.12.4.603 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

10.12.4.604 SoftwareSignalSelector

`GenApi::IEnumerationT<SoftwareSignalSelectorEnums>& SoftwareSignalSelector`

Description: Selects which Software Signal features to control.

Visibility: Beginner

10.12.4.605 SourceCount

`GenApi::IIInteger& SourceCount`

Description: Controls or returns the number of sources supported by the device.

Visibility: Beginner

10.12.4.606 SourceSelector

`GenApi::IEnumerationT<SourceSelectorEnums>& SourceSelector`

Description: Selects the source to control.

Visibility: Beginner

10.12.4.607 Test0001

```
GenApi::IInteger& Test0001
```

Description: For testing only.

Visibility:

10.12.4.608 TestEventGenerate

```
GenApi:: ICommand& TestEventGenerate
```

Description: This command generates a test event and sends it to the host.

Visibility:

10.12.4.609 TestPattern

```
GenApi::IEnumerationT<TestPatternEnums>& TestPattern
```

Description:

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

Visibility:

10.12.4.610 TestPatternGeneratorSelector

```
GenApi::IEnumerationT<TestPatternGeneratorSelectorEnums>& TestPatternGeneratorSelector
```

Description:

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

Visibility:

10.12.4.611 TestPendingAck

```
GenApi::IInteger& 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

10.12.4.612 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

10.12.4.613 TimerDuration

`GenApi::IFloat& TimerDuration`

Description: Sets the duration (in microseconds) of the Timer pulse.

Visibility: Expert

10.12.4.614 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

10.12.4.615 TimerSelector

`GenApi::IEnumerationT<TimerSelectorEnums>& TimerSelector`

Description: Selects which Timer to configure.

Visibility: Expert

10.12.4.616 TimerStatus

`GenApi::IEnumerationT<TimerStatusEnums>& TimerStatus`

Description: Returns the current status of the Timer.

Visibility: Expert

10.12.4.617 TimerTriggerActivation

`GenApi::IEnumerationT<TimerTriggerActivationEnums>& TimerTriggerActivation`

Description: Selects the activation mode of the trigger to start the Timer.

Visibility: Expert

10.12.4.618 TimerTriggerSource

```
GenApi::IEnumerationT<TimerTriggerSourceEnums>& TimerTriggerSource
```

Description: Selects the source of the trigger to start the Timer.

Visibility: Expert

10.12.4.619 TimerValue

```
GenApi::IFloat& TimerValue
```

Description: Reads or writes the current value (in microseconds) of the selected Timer.

Visibility: Expert

10.12.4.620 Timestamp

```
GenApi::IInteger& Timestamp
```

Description: Reports the current value of the device timestamp counter.

Visibility: Expert

10.12.4.621 TimestampLatch

```
GenApi:: ICommand& TimestampLatch
```

Description: Latches the current timestamp counter into TimestampLatchValue.

Visibility:

10.12.4.622 TimestampLatchValue

```
GenApi::IInteger& TimestampLatchValue
```

Description: Returns the latched value of the timestamp counter.

Visibility:

10.12.4.623 TimestampReset

```
GenApi:: ICommand& TimestampReset
```

Description: Resets the current value of the device timestamp counter.

Visibility:

10.12.4.624 TLParamsLocked

```
GenApi::IInteger& TLParamsLocked
```

Description: Visibility:

10.12.4.625 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

10.12.4.626 TransferBlockCount

```
GenApi::IInteger& 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:

10.12.4.627 TransferBurstCount

```
GenApi::IInteger& TransferBurstCount
```

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.

Visibility: Expert

10.12.4.628 TransferComponentSelector

```
GenApi::IEnumerationT<TransferComponentSelectorEnums>& TransferComponentSelector
```

Description: Selects the color component for the control of the TransferStreamChannel feature.

Visibility: Guru

10.12.4.629 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:

10.12.4.630 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:

10.12.4.631 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

10.12.4.632 TransferQueueCurrentBlockCount

```
GenApi::IInteger& TransferQueueCurrentBlockCount
```

Description: Returns number of data blocks (images) currently in the transfer queue.

Visibility:

10.12.4.633 TransferQueueMaxBlockCount

```
GenApi::IInteger& TransferQueueMaxBlockCount
```

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

10.12.4.634 TransferQueueMode

```
GenApi::IEnumerationT<TransferQueueModeEnums>& TransferQueueMode
```

Description: Specifies the operation mode of the transfer queue.

Visibility:

10.12.4.635 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:

10.12.4.636 TransferResume

```
GenApi:: ICommand& TransferResume
```

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.

Visibility: Guru

10.12.4.637 TransferSelector

```
GenApi:: IEnumerationT<TransferSelectorEnums>& TransferSelector
```

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

Visibility: Expert

10.12.4.638 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:

10.12.4.639 TransferStatus

```
GenApi:: IBoolean& TransferStatus
```

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.

Visibility: Guru

10.12.4.640 TransferStatusSelector

```
GenApi:: IEnumerationT<TransferStatusSelectorEnums>& TransferStatusSelector
```

Description: Selects which status of the transfer module to read.

Visibility: Guru

10.12.4.641 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:

10.12.4.642 TransferStreamChannel

```
GenApi::IInteger& 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

10.12.4.643 TransferTriggerActivation

```
GenApi::IEnumerationT<TransferTriggerActivationEnums>& TransferTriggerActivation
```

Description: Specifies the activation mode of the transfer control trigger.

Visibility: Guru

10.12.4.644 TransferTriggerMode

```
GenApi::IEnumerationT<TransferTriggerModeEnums>& TransferTriggerMode
```

Description: Controls if the selected trigger is active.

Visibility: Guru

10.12.4.645 TransferTriggerSelector

```
GenApi::IEnumerationT<TransferTriggerSelectorEnums>& TransferTriggerSelector
```

Description: Selects the type of transfer trigger to configure.

Visibility: Guru

10.12.4.646 TransferTriggerSource

```
GenApi::IEnumerationT<TransferTriggerSourceEnums>& TransferTriggerSource
```

Description: Specifies the signal to use as the trigger source for transfers.

Visibility: Guru

10.12.4.647 TriggerActivation

```
GenApi::IEnumerationT<TriggerActivationEnums>& TriggerActivation
```

Description: Specifies the activation mode of the trigger.

Visibility:

10.12.4.648 TriggerDelay

`GenApi::IFloat& TriggerDelay`

Description:

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

Visibility:

10.12.4.649 TriggerDivider

`GenApi::IInteger& TriggerDivider`

Description: Specifies a division factor for the incoming trigger pulses.

Visibility: Expert

10.12.4.650 TriggerEventTest

`GenApi:: ICommand& TriggerEventTest`

Description: This command generates a test event and sends it to the host.

Visibility:

10.12.4.651 TriggerMode

`GenApi::IEnumerationT<TriggerModeEnums>& TriggerMode`

Description:

Controls whether or not trigger is active.

Visibility:

10.12.4.652 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

10.12.4.653 TriggerOverlap

```
GenApi::IEnumerationT<TriggerOverlapEnums>& TriggerOverlap
```

Description: Specifies the overlap mode of the trigger.

Visibility:

10.12.4.654 TriggerSelector

```
GenApi::IEnumerationT<TriggerSelectorEnums>& TriggerSelector
```

Description: Selects the type of trigger to configure.

Visibility:

10.12.4.655 TriggerSoftware

```
GenApi:: ICommand& TriggerSoftware
```

Description:

Generates an internal trigger if Trigger Source is set to Software.

Visibility:

10.12.4.656 TriggerSource

```
GenApi::IEnumerationT<TriggerSourceEnums>& TriggerSource
```

Description:

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

Visibility:

10.12.4.657 UserOutputSelector

```
GenApi::IEnumerationT<UserOutputSelectorEnums>& UserOutputSelector
```

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

Visibility:

10.12.4.658 UserOutputValue

```
GenApi::IBoolean& UserOutputValue
```

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).

Visibility:

10.12.4.659 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:

10.12.4.660 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

10.12.4.661 UserSetDefault

```
GenApi::IEnumerationT<UserSetDefaultEnums>& UserSetDefault
```

Description:

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

Visibility:

10.12.4.662 UserSetFeatureEnable

```
GenApi::IBoolean& UserSetFeatureEnable
```

Description: Whether or not the selected feature is saved to user sets.

Visibility:

10.12.4.663 UserSetLoad

`GenApi:: ICommand& UserSetLoad`

Description:

Loads the User Set specified by UserSetSelector to the device and makes it active.

Visibility:

10.12.4.664 UserSetSave

`GenApi:: ICommand& UserSetSave`

Description:

Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.

Visibility:

10.12.4.665 UserSetSelector

`GenApi:: IEnumeration<UserSetSelectorEnums>& UserSetSelector`

Description:

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

Visibility:

10.12.4.666 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:

10.12.4.667 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

10.12.4.668 WhiteClipSelector

```
GenApi::IEnumeration<WhiteClipSelectorEnums>& WhiteClipSelector
```

Description: Selects which White Clip to control.

Visibility: Expert

10.12.4.669 Width

```
GenApi::IInteger& Width
```

Description:

Width of the image provided by the device (in pixels).

Visibility:

10.12.4.670 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:

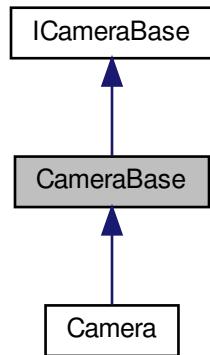
The documentation for this class was generated from the following file:

- [include/](#)[Camera.h](#)

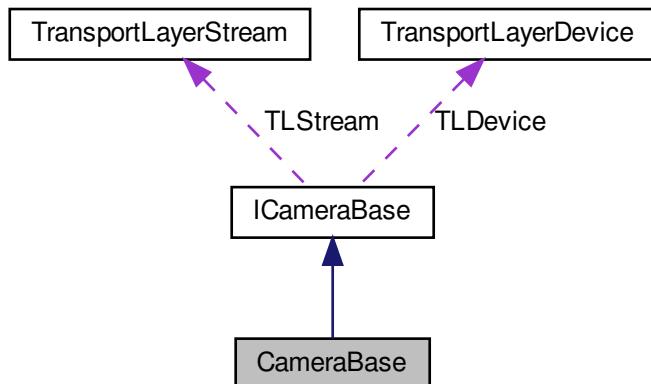
10.13 CameraBase Class Reference

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 Connect to camera, retrieve XML and generate node map.

- void [DeInit \(\)](#)
DeInit Disconnect camera port and free 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 [RegisterEvent \(Event &evtToRegister\)](#)
RegisterEvent(Event &) Registers a specific event for the camera.
- void [RegisterEvent \(Event &evtToRegister, const GenICam::gcstring &eventName\)](#)
RegisterEvent(Event &, const GenICam::gcstring&) Registers a specific event for the camera.
- void [UnregisterEvent \(Event &evtToUnregister\)](#)
UnregisterEvent Unregisters an event for the camera Events should be unregistered first before calling camera DeInit().

- `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 `InterfaceImpl`

Additional Inherited Members

10.13.1 Detailed Description

The base class for the camera object.

10.13.2 Constructor & Destructor Documentation

10.13.2.1 ~CameraBase()

```
virtual ~CameraBase (
    void ) [virtual]
```

Virtual Destructor.

10.13.2.2 CameraBase() [1/2]

```
CameraBase (
    void ) [protected]
```

Default constructor.

10.13.2.3 CameraBase() [2/2]

```
CameraBase (
    const CameraBase & ) [protected]
```

Copy constructor.

10.13.3 Member Function Documentation

10.13.3.1 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](#).

10.13.3.2 DeInit()

```
void DeInit ( ) [virtual]
```

DeInit Disconnect camera port and free [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\(\)](#)

[UnregisterEvent\(Event & evtToUnregister\)](#)

Implements [ICameraBase](#).

10.13.3.3 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](#).

10.13.3.4 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](#).

10.13.3.5 ForceIP()

```
void ForceIP ( ) [virtual]
```

ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Implements [ICameraBase](#).

10.13.3.6 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](#).

10.13.3.7 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](#).

10.13.3.8 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](#).

10.13.3.9 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](#).

10.13.3.10 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](#).

10.13.3.11 GetNumDataStreams()

```
unsigned int GetNumDataStreams ( ) [virtual]
```

GetNumDataStreams Returns the number of streams that a device supports.

Returns

The number of data streams

Implements [ICameraBase](#).

10.13.3.12 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](#).

10.13.3.13 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](#).

10.13.3.14 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](#).

10.13.3.15 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](#).

10.13.3.16 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](#).

10.13.3.17 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](#).

10.13.3.18 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](#).

10.13.3.19 Init()

```
void Init ( ) [virtual]
```

Init Connect to camera, retrieve XML and generate node map.

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](#).

10.13.3.20 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](#).

10.13.3.21 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](#).

10.13.3.22 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](#).

10.13.3.23 operator=()

```
CameraBase& operator= (
    const CameraBase & ) [protected]
```

Assignment operator.

10.13.3.24 ReadPort()

```
void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

10.13.3.25 RegisterEvent() [1/2]

```
void RegisterEvent (
    Event & evtToRegister ) [virtual]
```

[RegisterEvent\(Event &\)](#) Registers a specific event for the camera.

The camera has to be initialized first with a call to [Init\(\)](#) before registering for events.

See also

[Init\(\)](#)

Parameters

<code>evtToRegister</code>	The event to register for the camera
----------------------------	--------------------------------------

Implements [ICameraBase](#).

10.13.3.26 RegisterEvent() [2/2]

```
void RegisterEvent (
    Event & evtToRegister,
    const GenICam::gcstring & eventName ) [virtual]
```

[RegisterEvent\(Event &, const GenICam::gcstring&\)](#) Registers a specific event for the camera.

See also

[Init\(\)](#)

Parameters

<i>evtToRegister</i>	The event to register for the camera
<i>eventName</i>	The event name to register

Implements [ICameraBase](#).

10.13.3.27 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](#).

10.13.3.28 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](#).

10.13.3.29 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](#).

10.13.3.30 UnregisterEvent()

```
void UnregisterEvent (
    Event & evtToUnregister ) [virtual]
```

UnregisterEvent Unregisters an event for the camera Events should be unregistered first 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

[DeInit\(\)](#)

Parameters

<code>evtToUnregister</code>	The event to unregister from the camera
------------------------------	---

Implements [ICameraBase](#).

10.13.3.31 WritePort()

```
void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

10.13.4 Friends And Related Function Documentation

10.13.4.1 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

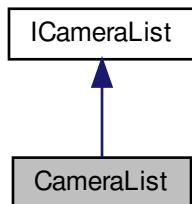
The documentation for this class was generated from the following file:

- [include/ CameraBase.h](#)

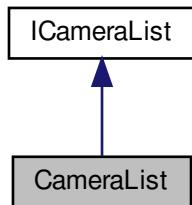
10.14 CameraList Class Reference

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`](#)

- `CameraPtr GetBySerial (std::string serialNumber) const`
Returns a pointer to a camera object at the "index".
- `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 Append (CameraList &otherList)`
Appends a camera list to the current list.

Additional Inherited Members

10.14.1 Detailed Description

Used to hold a list of camera objects.

10.14.2 Constructor & Destructor Documentation

10.14.2.1 CameraList() [1/2]

```
CameraList (
    void )
```

Default constructor.

10.14.2.2 ~CameraList()

```
virtual ~CameraList (
    void ) [virtual]
```

Virtual destructor.

10.14.2.3 CameraList() [2/2]

```
CameraList (
    const CameraList & iface )
```

Copy constructor.

10.14.3 Member Function Documentation

10.14.3.1 Append()

```
void Append (
```

`CameraList & otherList) [virtual]`

Appends a camera list to the current list.

Parameters

<i>otherList</i>	The other list to append to this list
------------------	---------------------------------------

Implements [ICameraList](#).

10.14.3.2 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](#).

10.14.3.3 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 an camera object.

Implements [ICameraList](#).

10.14.3.4 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 an camera object.

Implements [ICameraList](#).

10.14.3.5 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](#).

10.14.3.6 operator=()

```
CameraList& operator= (
    const CameraList & iface )
```

Assignment operator.

10.14.3.7 operator[]()

```
CameraPtr operator[ ] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [ICameraList](#).

10.14.3.8 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](#).

10.14.3.9 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](#).

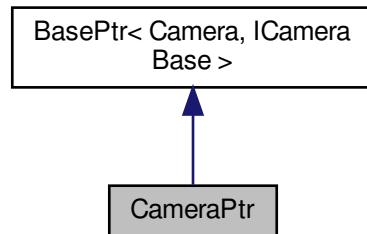
The documentation for this class was generated from the following file:

- [include/ CameraList.h](#)

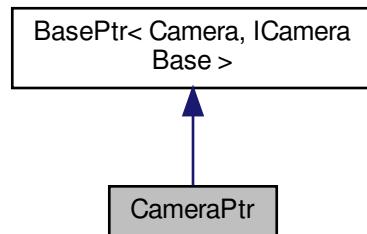
10.15 CameraPtr Class Reference

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

10.15.1 Detailed Description

A reference tracked pointer to a camera object.

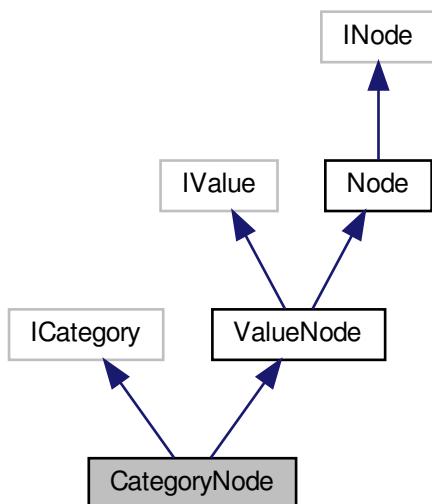
The documentation for this class was generated from the following file:

- [include/CameraPtr.h](#)

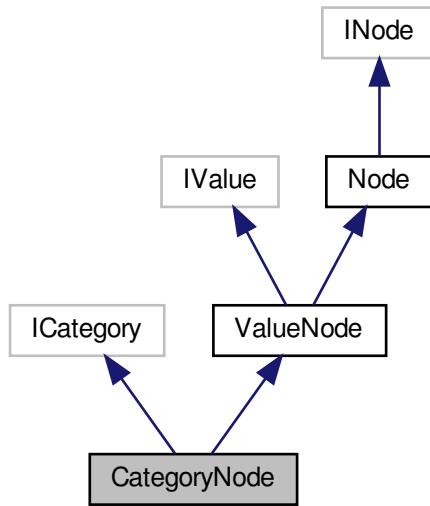
10.16 CategoryNode Class Reference

[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

10.16.1 Detailed Description

[Interface](#) for string properties.

10.16.2 Constructor & Destructor Documentation

10.16.2.1 CategoryNode() [1/2]

[CategoryNode \(\)](#)

10.16.2.2 CategoryNode() [2/2]

```
CategoryNode (
    std::shared_ptr< Node::NodeImpl > pCategory )
```

10.16.2.3 ~CategoryNode()

```
virtual ~CategoryNode ( ) [virtual]
```

10.16.3 Member Function Documentation

10.16.3.1 GetFeatures()

```
virtual void GetFeatures (
    FeatureList_t & Features ) const [virtual]
```

Get all features of the category (including sub-categories)

10.16.3.2 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

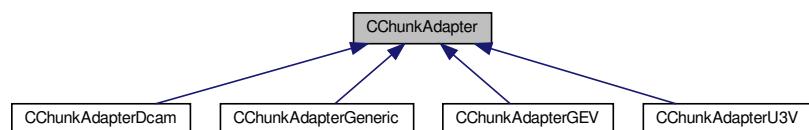
The documentation for this class was generated from the following file:

- include/SpinGenApi/[CategoryNode.h](#)

10.17 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`

10.17.1 Detailed Description

Connects a chunked buffer to a node map.

10.17.2 Constructor & Destructor Documentation

10.17.2.1 `~CChunkAdapter()`

```
virtual ~CChunkAdapter ( ) [virtual]
```

Destructor.

10.17.2.2 CChunkAdapter()

```
CChunkAdapter (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 ) [protected]
```

Serves as default constructor.

10.17.3 Member Function Documentation

10.17.3.1 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](#).

10.17.3.2 AttachNodeMap()

```
void AttachNodeMap (
    INodeMap * pNodeMap )
```

Attaches to a node map and retrieves the chunk ports.

10.17.3.3 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](#).

10.17.3.4 ClearCaches()

```
void ClearCaches ( )
```

Clears the chunk caches.

10.17.3.5 DetachBuffer()

```
void DetachBuffer ( )
```

Detaches a buffer.

10.17.3.6 DetachNodeMap()

```
void DetachNodeMap ( )
```

Detaches from the node map.

10.17.3.7 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the buffer.

10.17.4 Member Data Documentation

10.17.4.1 m_pChunkAdapter

```
void* m_pChunkAdapter [protected]
```

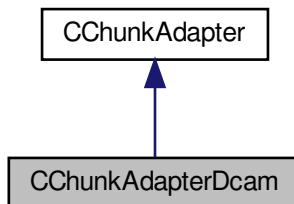
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapter.h](#)

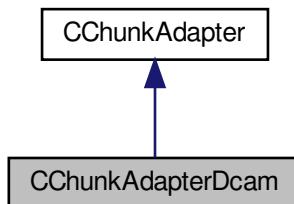
10.18 CChunkAdapterDcam Class Reference

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

10.18.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

10.18.2 Constructor & Destructor Documentation

10.18.2.1 CChunkAdapterDcam()

```
CChunkAdapterDcam (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

10.18.2.2 ~CChunkAdapterDcam()

```
virtual ~CChunkAdapterDcam ( ) [virtual]
```

Destructor.

10.18.3 Member Function Documentation

10.18.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.18.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

10.18.3.3 CheckCRC()

```
bool CheckCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

Checks CRC sum of buffer.

10.18.3.4 HasCRC()

```
bool HasCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

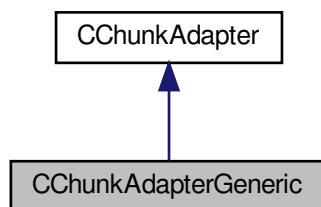
Checks if buffer has a CRC attached.

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

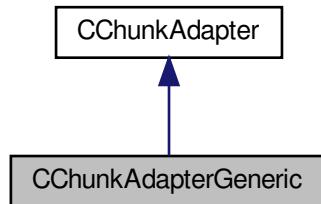
- include/SpinGenApi/[ChunkAdapterDcam.h](#)

10.19 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

10.19.1 Constructor & Destructor Documentation

10.19.1.1 CChunkAdapterGeneric()

```
CChunkAdapterGeneric (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

10.19.1.2 ~CChunkAdapterGeneric()

```
virtual ~CChunkAdapterGeneric ( ) [virtual]
```

10.19.2 Member Function Documentation

10.19.2.1 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](#).

10.19.2.2 AttachBuffer() [2/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkData_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

10.19.2.3 AttachBuffer() [3/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkDataStr_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

10.19.2.4 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

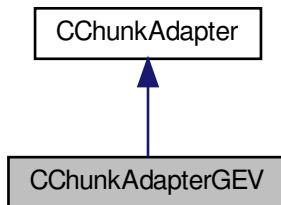
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

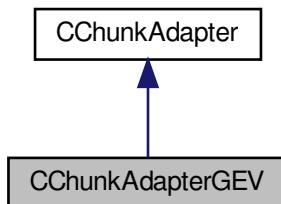
10.20 CChunkAdapterGEV Class Reference

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

10.20.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

10.20.2 Constructor & Destructor Documentation

10.20.2.1 CChunkAdapterGEV()

```
CChunkAdapterGEV (   
    INodeMap * pNodeMap = NULL,  
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

10.20.2.2 ~CChunkAdapterGEV()

```
virtual ~CChunkAdapterGEV ( ) [virtual]
```

Destructor.

10.20.3 Member Function Documentation

10.20.3.1 AttachBuffer()

```
virtual void AttachBuffer (   
    uint8_t * pBuffer,  
    int64_t BufferLength,  
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.20.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (   
    uint8_t * pBuffer,  
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

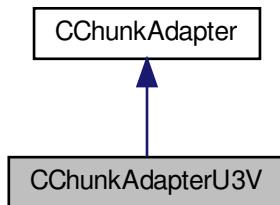
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

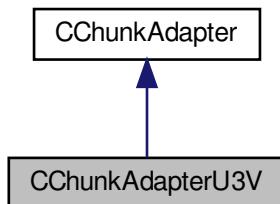
10.21 CChunkAdapterU3V Class Reference

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

10.21.1 Detailed Description

Connects a chunked U3V buffer to a node map.

10.21.2 Constructor & Destructor Documentation

10.21.2.1 CChunkAdapterU3V()

```
CChunkAdapterU3V (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

10.21.2.2 ~CChunkAdapterU3V()

```
virtual ~CChunkAdapterU3V ( ) [virtual]
```

Destructor.

10.21.3 Member Function Documentation

10.21.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.21.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

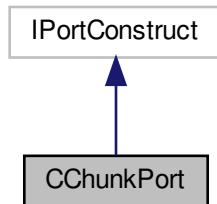
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

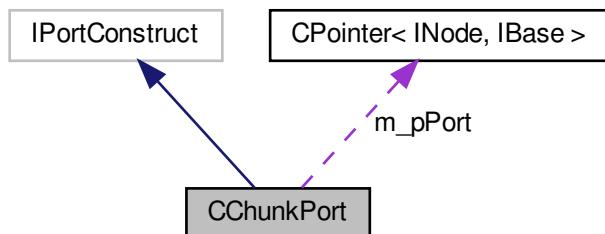
10.22 CChunkPort Class Reference

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 GetSwapEndianess \(\)](#)
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](#)

10.22.1 Detailed Description

Port attachable to a chunk in a buffer.

10.22.2 Constructor & Destructor Documentation

10.22.2.1 CChunkPort()

```
CChunkPort (
    IPort * pPort = NULL )
```

Constructor; can attach to a port.

10.22.2.2 ~CChunkPort()

```
~CChunkPort ( )
```

Destructor; detaches from the port.

10.22.3 Member Function Documentation**10.22.3.1 AttachChunk()**

```
void AttachChunk (
    uint8_t * pBaseAddress,
    int64_t ChunkOffset,
    int64_t Length,
    bool Cache )
```

Attaches the Chunk to the ChunkPort.

10.22.3.2 AttachPort()

```
bool AttachPort (
    ::Spinnaker::GenApi::IPort * pPort )
```

Attaches the ChunkPort to the Port.

10.22.3.3 CheckChunkID() [1/2]

```
bool CheckChunkID (
    uint8_t * pChunkIDBuffer,
    int ChunkIDLength )
```

Checks if a ChunkID matches.

10.22.3.4 CheckChunkID() [2/2]

```
bool CheckChunkID (
    uint64_t ChunkID )
```

Checks if a ChunkID matches, version using uint64_t ID representation.

10.22.3.5 ClearCache()

```
void ClearCache ( )
```

Clears the chunk cache.

10.22.3.6 DetachChunk()

```
void DetachChunk ( )
```

Detaches the Chunk from the ChunkPort.

10.22.3.7 DetachPort()

```
void DetachPort ( )
```

Detaches the ChunkPort to the Port.

10.22.3.8 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

10.22.3.9 GetChunkIDLength()

```
int GetChunkIDLength ( )
```

Gets the ChunkID length.

10.22.3.10 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

10.22.3.11 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

10.22.3.12 InvalidateNode()

```
void InvalidateNode ( )
```

10.22.3.13 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

10.22.3.14 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

10.22.3.15 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the chunk.

10.22.3.16 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

10.22.4 Member Data Documentation

10.22.4.1 m_pChunkPort

```
void* m_pChunkPort [protected]
```

10.22.4.2 m_pPort

```
CNodePtr m_pPort [protected]
```

10.22.4.3 m_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

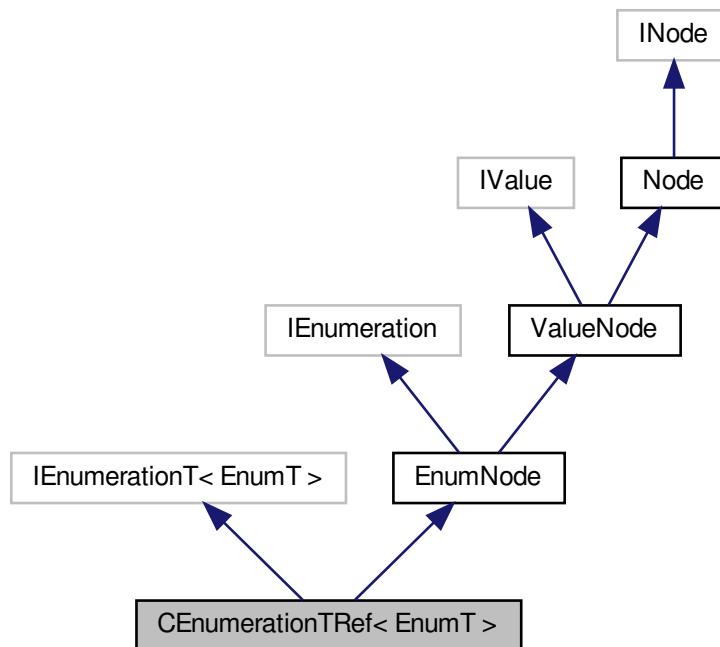
The documentation for this class was generated from the following file:

- include/SpinGenApi/ChunkPort.h

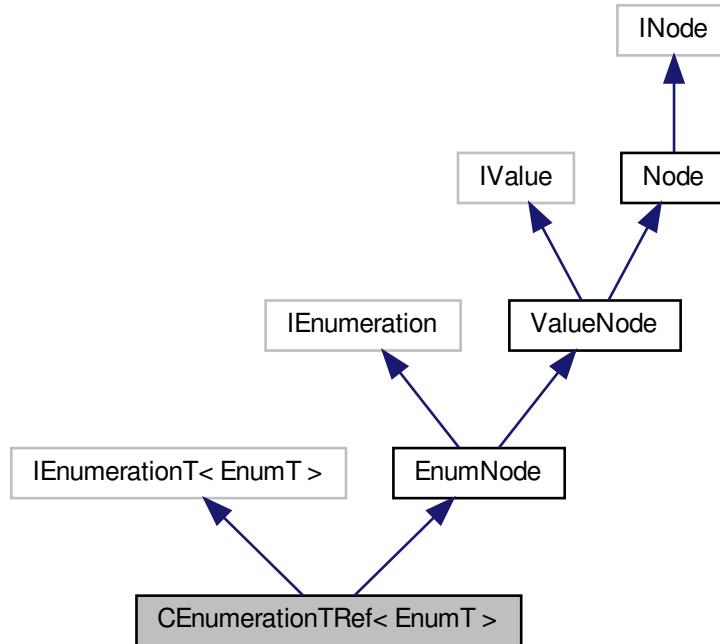
10.23 CEnumerationTRef< EnumT > Class Template Reference

[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 GenICam::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

10.23.1 Detailed Description

```
template<class EnumT>
class Spinnaker::GenApi::CEnumerationTRef< EnumT >
```

[Interface](#) for string properties.

10.23.2 Constructor & Destructor Documentation

10.23.2.1 CEnumerationTRef() [1/2]

```
CEnumerationTRef ( )
```

10.23.2.2 CEnumerationTRef() [2/2]

```
CEnumerationTRef (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

10.23.2.3 ~CEnumerationTRef()

```
virtual ~CEnumerationTRef ( ) [virtual]
```

10.23.3 Member Function Documentation

10.23.3.1 GetCurrentEntry()

```
virtual IEnumEntry* GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get the current entry.

Reimplemented from [EnumNode](#).

10.23.3.2 GetEntry() [1/2]

```
virtual IEnumEntry* GetEntry (
    const EnumT Value ) [virtual]
```

returns the EnumEntry object belonging to the Value

10.23.3.3 GetEntry() [2/2]

```
virtual IEnumEntry* GetEntry (
    const int64_t IntValue ) [virtual]
```

Get an entry node by its IntValue.

Reimplemented from [EnumNode](#).

10.23.3.4 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

10.23.3.5 operator()

```
virtual EnumT operator() ( ) [virtual]
```

Get node value.

10.23.3.6 operator=() [1/2]

```
virtual IEnumeration& operator= (
    EnumT Value ) [virtual]
```

Set node value.

10.23.3.7 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](#).

10.23.3.8 SetEnumReference()

```
virtual void SetEnumReference (
    int Index,
    GenICam::gcstring Name ) [virtual]
```

sets the Enum value corresponding to a value

10.23.3.9 SetNumEnums()

```
virtual void SetNumEnums (
    int NumEnums ) [virtual]
```

sets the number of enum values

10.23.3.10 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for EnumerationT

Reimplemented from [EnumNode](#).

10.23.3.11 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)

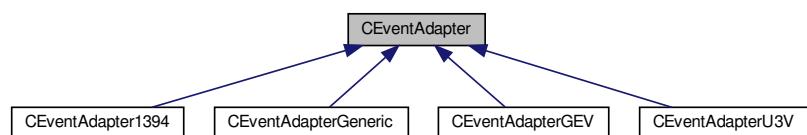
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumNodeT.h](#)

10.24 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](#)

10.24.1 Detailed Description

Delivers Events to ports.

10.24.2 Constructor & Destructor Documentation

10.24.2.1 CEventAdapter()

```
CEventAdapter (
    INodeMap * pNodeMap = NULL )
```

Constructor.

10.24.2.2 ~CEventAdapter()

```
virtual ~CEventAdapter ( ) [virtual]
```

Destructor.

10.24.3 Member Function Documentation

10.24.3.1 AttachNodeMap()

```
virtual void AttachNodeMap (
    INodeMap * pNodeMap ) [virtual]
```

Attaches to a node map and retrieves the chunk ports.

10.24.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes ) [pure virtual]
```

Deliver message.

Implemented in [CEventAdapterGEV](#), [CEventAdapterU3V](#), [CEventAdapter1394](#), and [CEventAdapterGeneric](#).

10.24.3.3 DetachNodeMap()

```
virtual void DetachNodeMap ( ) [virtual]
```

Detaches from the node emap.

10.24.4 Member Data Documentation

10.24.4.1 m_pEventAdapter

```
void* m_pEventAdapter [protected]
```

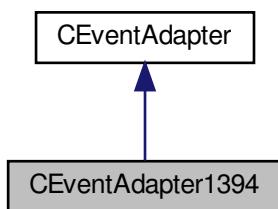
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter.h](#)

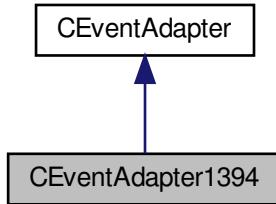
10.25 CEventAdapter1394 Class Reference

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

10.25.1 Detailed Description

Distribute the events to the node map.

10.25.2 Constructor & Destructor Documentation

10.25.2.1 CEventAdapter1394()

```
CEventAdapter1394 (
    INodeMap * pNodeMap = NULL ) [explicit]
```

constructor

10.25.2.2 ~CEventAdapter1394()

```
virtual ~CEventAdapter1394 ( ) [virtual]
```

10.25.3 Member Function Documentation

10.25.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    EventData1394 & Event,
    uint32_t numBytes )
```

distributes events to node map

10.25.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

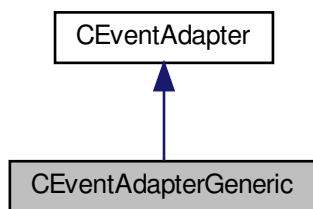
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter1394.h](#)

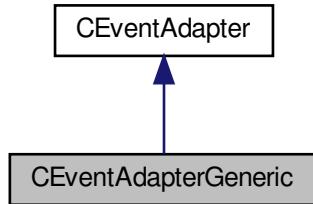
10.26 CEventAdapterGeneric Class Reference

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

10.26.1 Detailed Description

Connects a generic event to a node map.

10.26.2 Constructor & Destructor Documentation

10.26.2.1 CEventAdapterGeneric()

```
CEventAdapterGeneric (
```

```
    INodeMap * pNodeMap = NULL )
```

Constructor.

10.26.2.2 ~CEventAdapterGeneric()

```
virtual ~CEventAdapterGeneric ( ) [virtual]
```

Destructor.

10.26.3 Member Function Documentation

10.26.3.1 DeliverMessage() [1/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

10.26.3.2 DeliverMessage() [2/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    const GenICam::gcstring & EventID ) [virtual]
```

10.26.3.3 DeliverMessage() [3/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    uint64_t EventID ) [virtual]
```

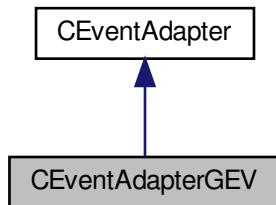
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGeneric.h](#)

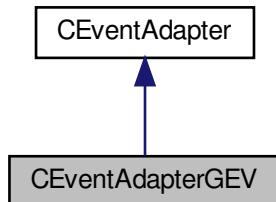
10.27 CEventAdapterGEV Class Reference

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 DeliverEventDataMessage \(const GVCP_EVENTDATA_REQUEST *pEventData\)`](#)
Delivers the [Event](#) + Data listed in the [EventData](#) packet.

Additional Inherited Members

10.27.1 Detailed Description

Connects a GigE [Event](#) to a node map.

10.27.2 Constructor & Destructor Documentation

10.27.2.1 CEventAdapterGEV()

```
CEventAdapterGEV (
    INodeMap * pNodeMap = NULL )
```

Constructor.

10.27.2.2 ~CEventAdapterGEV()

```
virtual ~CEventAdapterGEV ( ) [virtual]
```

Destructor.

10.27.3 Member Function Documentation

10.27.3.1 DeliverEventMessage() [1/2]

```
void DeliverEventMessage (
    const GVCP_EVENT_REQUEST * pEvent )
```

Delivers the Events listed in the [Event](#) packet.

10.27.3.2 DeliverEventMessage() [2/2]

```
void DeliverEventMessage (
    const GVCP_EVENTDATA_REQUEST * pEventData )
```

Delivers the [Event](#) + Data listed in the EventData packet.

10.27.3.3 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

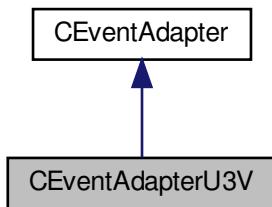
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

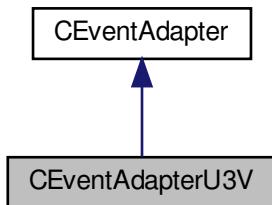
10.28 CEventAdapterU3V Class Reference

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

10.28.1 Detailed Description

Connects a U3V [Event](#) to a node map.

10.28.2 Constructor & Destructor Documentation

10.28.2.1 CEventAdapterU3V()

```
CEventAdapterU3V (
    INodeMap * pNodeMap = NULL )
```

Constructor.

10.28.2.2 ~CEventAdapterU3V()

```
virtual ~CEventAdapterU3V ( ) [virtual]
```

Destructor.

10.28.3 Member Function Documentation

10.28.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    const U3V_EVENT_MESSAGE * pEventMessage )
```

Delivers the [Event](#) + Data listed in the packet.

10.28.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

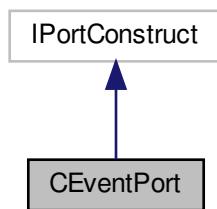
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

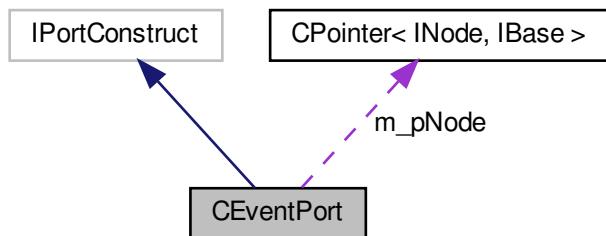
10.29 CEventPort Class Reference

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 GetSwapEndianess ()**
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**

10.29.1 Detailed Description

Port attachable to an event.

10.29.2 Constructor & Destructor Documentation

10.29.2.1 CEventPort()

```
CEventPort (   
    INode * pNode = NULL )
```

Constructor; can attach to a node.

10.29.2.2 ~CEventPort()

```
~CEventPort ( )
```

Destructor; detaches from the port.

10.29.3 Member Function Documentation

10.29.3.1 AttachEvent()

```
void AttachEvent (   
    uint8_t * pBaseAddress,   
    int64_t Length )
```

Attaches the an [Event](#) to the EventPort.

10.29.3.2 AttachNode()

```
bool AttachNode (   
    ::Spinnaker::GenApi::INode * pNode )
```

Attaches to the [Node](#).

10.29.3.3 CheckEventID() [1/2]

```
bool CheckEventID (   
    uint8_t * pEventIDBuffer,   
    int EventIDLength )
```

Checks if a EventID matches.

10.29.3.4 CheckEventID() [2/2]

```
bool CheckEventID (
    uint64_t EventID )
```

Checks if a EventID matches, version using uint64_t ID representation.

10.29.3.5 DetachEvent()

```
void DetachEvent ( )
```

Detaches the [Event](#) from the EventPort.

10.29.3.6 DetachNode()

```
void DetachNode ( )
```

Detaches from the [Node](#).

10.29.3.7 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

10.29.3.8 GetEventIDLength()

```
int GetEventIDLength ( )
```

Gets the EventID length.

10.29.3.9 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

10.29.3.10 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

10.29.3.11 InvalidateNode()

```
void InvalidateNode ( )
```

10.29.3.12 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

10.29.3.13 SetPortImpl()

```
virtual void SetPortImpl (
    ::Spinnaker::GenApi::IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

10.29.3.14 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

10.29.4 Member Data Documentation

10.29.4.1 m_pEventPort

```
void* m_pEventPort [protected]
```

10.29.4.2 m_pNode

```
CNodePtr m_pNode [protected]
```

10.29.4.3 m_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

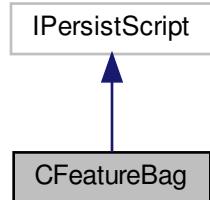
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventPort.h](#)

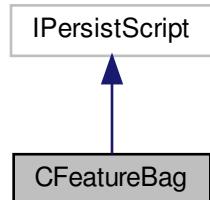
10.30 CFeatureBag Class Reference

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 ()`

10.30.1 Detailed Description

Bag holding streamable features of a nodetree.

10.30.2 Constructor & Destructor Documentation

10.30.2.1 CFeatureBag()

```
CFeatureBag ( )
```

10.30.2.2 ~CFeatureBag()

```
virtual ~CFeatureBag ( ) [virtual]
```

10.30.3 Member Function Documentation

10.30.3.1 GetFeatureBagHandle()

```
void* GetFeatureBagHandle ( )
```

10.30.3.2 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.

10.30.3.3 operator==()

```
bool operator== (
    const CFeatureBag & FeatureBag ) const
```

compares the content of two feature bags

10.30.3.4 PersistFeature()

```
virtual void PersistFeature (
    IValue & item ) [virtual]
```

Stores a feature.

10.30.3.5 SetInfo()

```
virtual void SetInfo (
    GenICam::gestring & Info ) [virtual]
```

sets information about the node map

10.30.3.6 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

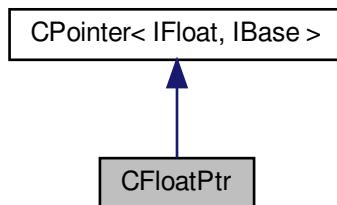
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Persistence.h](#)

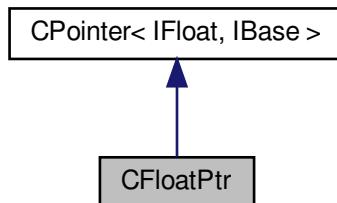
10.31 CFloatPtr Class Reference

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.
- [IEnumerator * GetEnumAlias \(\)](#)
gets the interface of an enum alias node.

Additional Inherited Members

10.31.1 Detailed Description

SmartPointer for IFloat interface pointer.

10.31.2 Constructor & Destructor Documentation

10.31.2.1 CFloatPtr() [1/2]

```
CFloatPtr ( ) throw () [inline]
```

Default constructor.

10.31.2.2 CFloatPtr() [2/2]

```
CFloatPtr ( IBase * pB ) [inline]
```

Constructor from IBase pointer type.

10.31.3 Member Function Documentation

10.31.3.1 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( ) [inline]
```

gets the interface of an enum alias node.

10.31.3.2 GetIntAlias()

```
IIInteger* GetIntAlias ( ) [inline]
```

gets the interface of an integer alias node.

10.31.3.3 operator=()

```
void operator= (
    IBase * pB ) [inline]
```

Assign IBase Pointer.

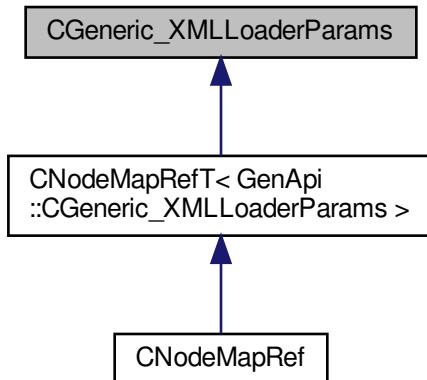
The documentation for this class was generated from the following file:

- include/SpinGenApi/Pointer.h

10.32 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 \(GenApi::INodeMap *\)](#)

10.32.1 Detailed Description

Empty base class used by class [CNodeMapRef](#) as generic template argument.

10.32.2 Member Function Documentation

10.32.2.1 `_Initialize()`

```
virtual void _Initialize (
    GenApi::INodeMap * ) [inline], [protected], [virtual]
```

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

- include/SpinGenApi/[NodeMapRef.h](#)

10.33 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](#)

10.33.1 Detailed Description

Named global lock which can be used over process boundaries.

10.33.2 Constructor & Destructor Documentation

10.33.2.1 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.

10.33.2.2 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.

10.33.2.3 ~CGlobalLock()

```
~CGlobalLock ( )
```

10.33.3 Member Function Documentation**10.33.3.1 IsValid()**

```
bool IsValid (
    void ) const
```

tests whether the lock is valid

10.33.3.2 Lock()

```
bool Lock (
    unsigned int timeout_ms )
```

enters the lock (may block)

10.33.3.3 TryLock()

```
bool TryLock (
    void )
```

tries to enter the lock and returns immediately when not possible

10.33.3.4 Unlock()

```
void Unlock (
    void )
```

leaves the lock

10.33.4 Member Data Documentation

10.33.4.1 m_DebugCount

```
long m_DebugCount [mutable], [protected]
```

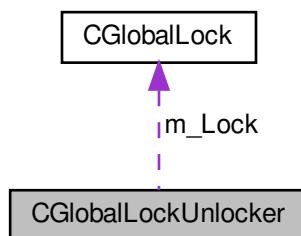
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

10.34 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`

10.34.1 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 contructors

10.34.2 Constructor & Destructor Documentation

10.34.2.1 CGlobalLockUnlocker()

```
CGlobalLockUnlocker (
    CGlobalLock & lock ) [inline]
```

10.34.2.2 ~CGlobalLockUnlocker()

```
~CGlobalLockUnlocker ( ) [inline]
```

10.34.3 Member Function Documentation

10.34.3.1 UnlockEarly()

```
void UnlockEarly (
    void ) [inline]
```

This function allows to unlock the object early before the object is destroyed.

10.34.4 Member Data Documentation

10.34.4.1 m_enabled

```
bool m_enabled [protected]
```

10.34.4.2 m_Lock

```
CGlobalLock& m_Lock [protected]
```

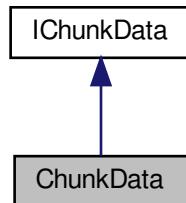
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

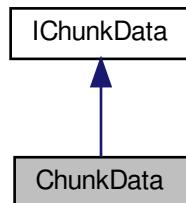
10.35 ChunkData Class Reference

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 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 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

10.35.1 Detailed Description

The chunk data which contains additional information about an image.

10.35.2 Constructor & Destructor Documentation

10.35.2.1 `ChunkData()` [1/2]

`ChunkData ()`

10.35.2.2 ChunkData() [2/2]

```
ChunkData (
    const ChunkData & src )
```

10.35.2.3 ~ChunkData()

```
virtual ~ChunkData (
    void ) [virtual]
```

10.35.3 Member Function Documentation**10.35.3.1 GetBlackLevel()**

```
float64_t GetBlackLevel ( ) const [virtual]
```

Description: Returns the black level used to capture the image.

Visibility:

Implements [IChunkData](#).

10.35.3.2 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](#).

10.35.3.3 GetCRC()

```
int64_t GetCRC ( ) const [virtual]
```

Description: Returns the CRC of the image payload.

Visibility:

Implements [IChunkData](#).

10.35.3.4 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](#).

10.35.3.5 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](#).

10.35.3.6 GetExposureTime()

```
float64_t GetExposureTime ( ) const [virtual]
```

Description: Returns the exposure time used to capture the image.

Visibility:

Implements [IChunkData](#).

10.35.3.7 GetFrameID()

```
int64_t GetFrameID ( ) const [virtual]
```

Description: Returns the image frame ID.

Visibility:

Implements [IChunkData](#).

10.35.3.8 GetGain()

```
float64_t GetGain ( ) const [virtual]
```

Description: Returns the gain used to capture the image.

Visibility:

Implements [IChunkData](#).

10.35.3.9 GetHeight()

```
int64_t GetHeight ( ) const [virtual]
```

Description: Returns the height of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.35.3.10 GetImage()

```
int64_t GetImage ( ) const [virtual]
```

Description: Returns the image payload.

Visibility:

Implements [IChunkData](#).

10.35.3.11 GetInferenceBoundingBoxResult()

```
InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const [virtual]
```

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.12 GetInferenceConfidence()

```
float64_t GetInferenceConfidence ( ) const [virtual]
```

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.13 GetInferenceResult()

```
int64_t GetInferenceResult ( ) const [virtual]
```

Description: Returns the chunk data inference result.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.14 GetLinePitch()

```
int64_t GetLinePitch ( ) const [virtual]
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.15 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](#).

10.35.3.16 GetOffsetX()

```
int64_t GetOffsetX ( ) const [virtual]
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.35.3.17 GetOffsetY()

```
int64_t GetOffsetY ( ) const [virtual]
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.35.3.18 GetPartSelector()

```
int64_t GetPartSelector ( ) const [virtual]
```

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.19 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](#).

10.35.3.20 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](#).

10.35.3.21 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](#).

10.35.3.22 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](#).

10.35.3.23 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](#).

10.35.3.24 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](#).

10.35.3.25 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](#).

10.35.3.26 GetScan3dInvalidDataValue()

```
float64_t GetScan3dInvalidDataValue ( ) const [virtual]
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.27 GetScan3dTransformValue()

```
float64_t GetScan3dTransformValue ( ) const [virtual]
```

Description: Returns the transform value.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.28 GetScanLineSelector()

```
int64_t GetScanLineSelector ( ) const [virtual]
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.29 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](#).

10.35.3.30 GetSerialDataLength()

```
int64_t GetSerialDataLength ( ) const [virtual]
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

Implements [IChunkData](#).

10.35.3.31 GetStreamChannelID()

```
int64_t GetStreamChannelID ( ) const [virtual]
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.32 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](#).

10.35.3.33 GetTimestamp()

```
int64_t GetTimestamp ( ) const [virtual]
```

Description: Returns the Timestamp of the image.

Visibility:

Implements [IChunkData](#).

10.35.3.34 GetTimestampLatchValue()

```
int64_t GetTimestampLatchValue ( ) const [virtual]
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.35 GetTransferBlockID()

```
int64_t GetTransferBlockID ( ) const [virtual]
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.36 GetTransferQueueCurrentBlockCount()

```
int64_t GetTransferQueueCurrentBlockCount ( ) const [virtual]
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

Implements [IChunkData](#).

10.35.3.37 GetWidth()

```
int64_t GetWidth ( ) const [virtual]
```

Description: Returns the width of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.35.3.38 SetChunks()

```
void SetChunks ( GenApi::INodeMap & pNodeMap ) [virtual]
```

Implements [IChunkData](#).

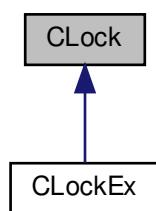
The documentation for this class was generated from the following file:

- [include/ChunkData.h](#)

10.36 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

10.36.1 Detailed Description

A lock class.

10.36.2 Constructor & Destructor Documentation

10.36.2.1 `CLock()`

`CLock ()`

Constructor.

10.36.2.2 `~CLock()`

`~CLock ()`

Destructor.

10.36.3 Member Function Documentation

10.36.3.1 `Lock()`

`void Lock ()`

enters the critical section (may block)

10.36.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

10.36.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

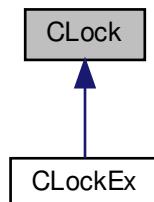
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

10.37 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`

10.37.1 Detailed Description

A lock class.

10.37.2 Constructor & Destructor Documentation

10.37.2.1 CLock() [1/2]

```
CLock ()
```

Constructor.

10.37.2.2 CLock() [2/2]

```
CLock (
    void * pLock )
```

Constructor.

10.37.2.3 ~CLock()

```
~CLock ()
```

Destructor.

10.37.3 Member Function Documentation

10.37.3.1 Lock()

```
void Lock ( )
```

enters the critical section (may block)

10.37.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

10.37.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

10.37.4 Friends And Related Function Documentation

10.37.4.1 NodeMap

```
friend class NodeMap [friend]
```

10.37.5 Member Data Documentation

10.37.5.1 m_bOwnLock

```
bool m_bOwnLock [protected]
```

10.37.5.2 m_lock

```
void* m_lock [protected]
```

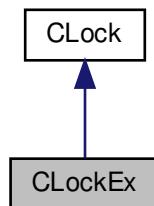
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

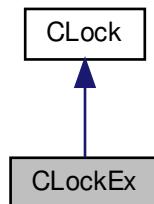
10.38 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Additional Inherited Members

10.38.1 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

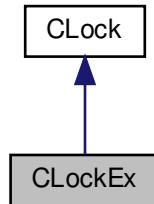
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

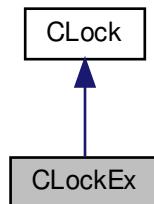
10.39 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Protected Attributes

- void * `m_lockEx`

Additional Inherited Members

10.39.1 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

10.39.2 Member Data Documentation

10.39.2.1 m_lockEx

```
void* m_lockEx [protected]
```

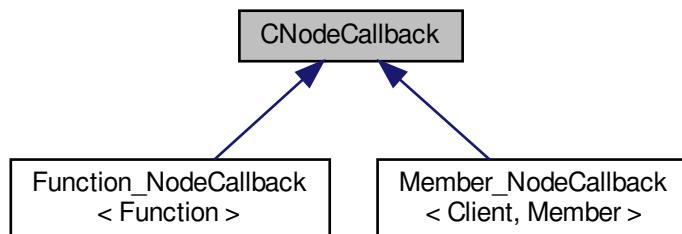
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

10.40 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 the 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

10.40.1 Detailed Description

callback body instance for INode pointers

10.40.2 Constructor & Destructor Documentation

10.40.2.1 CNodeCallback()

```
CNodeCallback (
    INode * pNode,
    ECallbackType CallbackType ) [inline]
```

10.40.2.2 ~CNodeCallback()

```
virtual ~CNodeCallback ( ) [inline], [virtual]
```

virtual destructor

10.40.3 Member Function Documentation

10.40.3.1 Destroy()

```
virtual void Destroy ( ) [pure virtual]
```

destroys the object

Implemented in `Member_NodeCallback< Client, Member >`, and `Function_NodeCallback< Function >`.

10.40.3.2 GetCallbackType()

```
ECallbackType GetCallbackType () [inline]
```

10.40.3.3 GetNode()

```
INode* GetNode () [inline]
```

returns the node the callback is registered to

10.40.3.4 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 >](#).

10.40.4 Member Data Documentation

10.40.4.1 m_CallbackType

```
ECallbackType m_CallbackType [protected]
```

the type of the callback

10.40.4.2 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](#)

10.41 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 \(EContentType_t FileType, const GenICam::gcstring &FileName, ECacheUsage_t CacheUsage=CacheUsage_Automatic, bool SuppressStringsOnLoad=false\)](#)
Creates the node map factory and simply stores the full path to the provided camera description file data.
- [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 \(const GenICam::gcstring &XmlData, ECacheUsage_t CacheUsage=CacheUsage_Automatic, bool SuppressStringsOnLoad=false\)](#)
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 \(const GenICam::gcstring &SubTreeRootNodeName, bool doRenameToRoot=false\)](#)
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 CNodeDataMap (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 \(const GenICam::gcstring &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true\)](#)
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)`

10.41.1 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.

10.41.2 Constructor & Destructor Documentation

10.41.2.1 [CNodeMapFactory\(\)](#) [1/5]

```
CNodeMapFactory( )
```

Creates an empty node map factory for assigning a non-empty node map factory later.

10.41.2.2 [~CNodeMapFactory\(\)](#)

```
virtual ~CNodeMapFactory( ) [virtual]
```

Destroys the node map factory data if all references to the data have been released.

10.41.2.3 [CNodeMapFactory\(\)](#) [2/5]

```
CNodeMapFactory(
    const CNodeMapFactory & )
```

Creates another reference to the node map factory data.

No data is copied.

10.41.2.4 [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.

10.41.2.5 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.

10.41.2.6 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. gcstring cfdData; //... fill cfdData ... 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).
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.

10.41.3 Member Function Documentation

10.41.3.1 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.

10.41.3.2 ApplyStyleSheet()

```
const GenICam::gcstring ApplyStyleSheet (
    const GenICam::gcstring & StyleSheetFileName )
```

Applies a style sheet to the pre-processed node map.

10.41.3.3 ClearCache()

```
static bool ClearCache ( ) [static]
```

Deletes all preprocessed camera description files from the cache.

10.41.3.4 CreateEmptyNodeMap()

```
static INodeMap* CreateEmptyNodeMap ( ) [static]
```

Creates an empty node map usable as placeholder, e.g.

if certain features are not supported by a module.

10.41.3.5 CreateNodeDataFromNodeMap()

```
static CNodeDataMap* CreateNodeDataFromNodeMap (
    INodeMap * pNodeMap ) [static]
```

10.41.3.6 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.

10.41.3.7 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.

10.41.3.8 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.

10.41.3.9 GetNodeStatistics()

```
void GetNodeStatistics (
    NodeStatistics_t & NodeStatistics )
```

10.41.3.10 GetSupportedSchemaVersions()

```
void GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) const
```

Each list entry is a string with the format "{Major}.{Minor}" were {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

10.41.3.11 IsCameraDescriptionFileDataReleased()

```
bool IsCameraDescriptionFileDataReleased ( ) const
```

Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.

10.41.3.12 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.

10.41.3.13 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).

10.41.3.14 IsPreprocessed()

```
bool IsPreprocessed () const
```

Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.

10.41.3.15 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.

10.41.3.16 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.

10.41.3.17 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.

10.41.3.18 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.

10.41.3.19 ToString()

```
GenICam::gcstring ToString( ) const
```

Outputs the pre-processed node map in string form (for debug purpose)

10.41.3.20 ToXml()

```
GenICam::gcstring ToXml( ) const
```

Outputs the pre-processed node map in XML form (mainly for debug purpose)

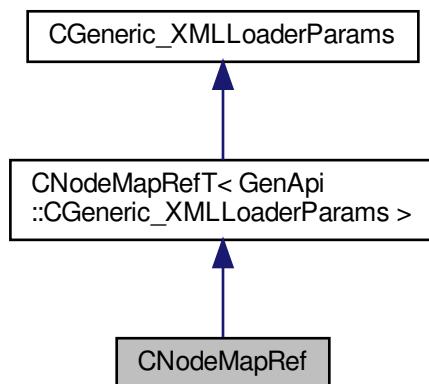
The documentation for this class was generated from the following file:

- include/SpinGenApi/NodeMapFactory.h

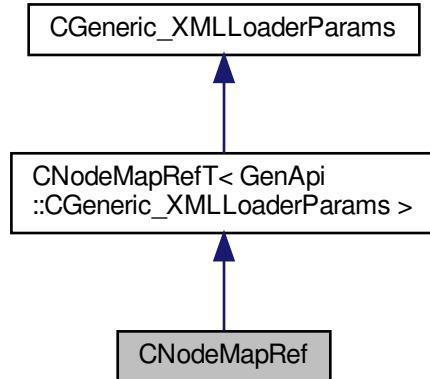
10.42 CNodeMapRef Class Reference

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

10.42.1 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.

10.42.2 Constructor & Destructor Documentation

10.42.2.1 CNodeMapRef() [1/3]

```
CNodeMapRef (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

10.42.2.2 CNodeMapRef() [2/3]

```
CNodeMapRef (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

10.42.2.3 CNodeMapRef() [3/3]

```
CNodeMapRef (
    const CNodeMapRef & Them ) [inline]
```

Copy constructor.

10.42.3 Member Function Documentation**10.42.3.1 operator=() [1/2]**

```
CNodeMapRef& operator= (
    const CNodeMapRef & Them ) [inline]
```

Assignment.

10.42.3.2 operator=() [2/2]

```
CNodeMapRef& operator= (
    INodeMap * pNodeMap ) [inline]
```

Assignment of an INodeMap*.

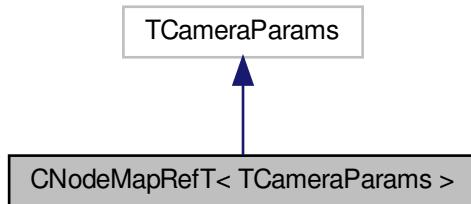
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

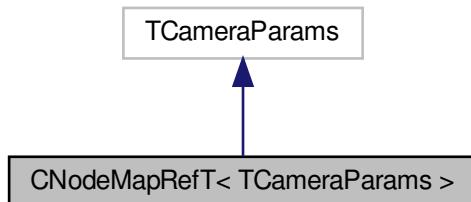
10.43 CNodeMapRefT< TCameraParams > Class Template Reference

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](#).

10.43.1 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)
----------------------	---

10.43.2 Member Function Documentation

10.43.2.1 `_ClearXMLCache()`

```
static bool _ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

10.43.2.2 `_Connect()` [1/2]

```
virtual bool _Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) [virtual]
```

Connects a port to a port node with given name.

10.43.2.3 `_Connect()` [2/2]

```
virtual bool _Connect (
    IPort * pPort ) [virtual]
```

Connects a port to the standard port "Device".

10.43.2.4 `_GetDeviceName()`

```
virtual GenICam::gcstring _GetDeviceName ( ) [virtual]
```

Get device name.

10.43.2.5 _GetNode()

```
virtual INode* _GetNode (
    const GenICam::gcstring & key ) [virtual]
```

Retrieves the node from the central map by name.

10.43.2.6 _GetNodes()

```
virtual void _GetNodes (
    NodeList_t & Nodes ) [virtual]
```

Retrieves all nodes in the node map.

10.43.2.7 _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}" were {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

10.43.2.8 _InvalidateNodes()

```
virtual void _InvalidateNodes ( ) [virtual]
```

Invalidates all nodes.

10.43.2.9 _LoadXMLFromFile()

```
void _LoadXMLFromFile (
    const GenICam::gcstring & FileName )
```

Creates the object from a XML file with given file name.

10.43.2.10 _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.

10.43.2.11 _LoadXMLFromString()

```
void _LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

10.43.2.12 _LoadXMLFromStringInject()

```
void _LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

10.43.2.13 _LoadXMLFromZIPData()

```
void _LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

10.43.2.14 _LoadXMLFromZIPFile()

```
void _LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

10.43.2.15 _Poll()

```
virtual void _Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

10.43.3 Member Data Documentation

10.43.3.1 _Ptr

```
INodeMap* _Ptr
```

Pointer to the [NodeMap](#).

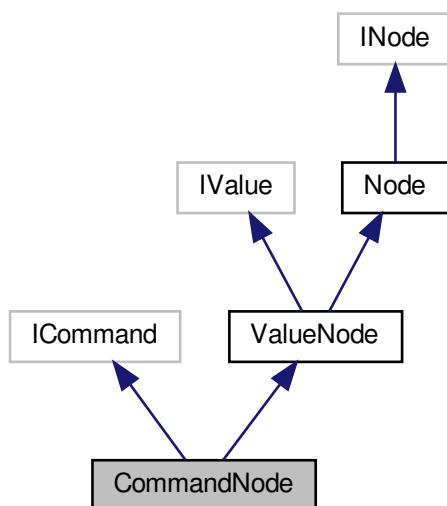
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

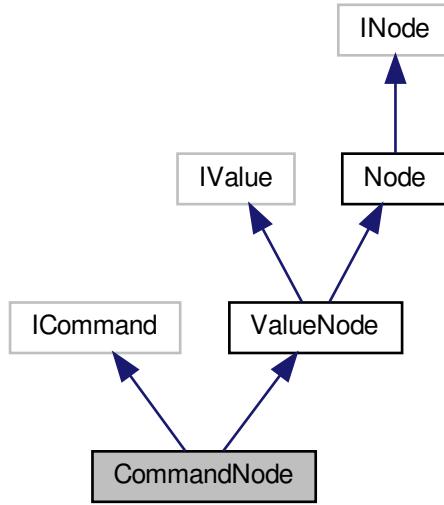
10.44 CommandNode Class Reference

[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

10.44.1 Detailed Description

[Interface](#) for string properties.

10.44.2 Constructor & Destructor Documentation

10.44.2.1 CommandNode() [1/2]

```
CommandNode ( )
```

10.44.2.2 CommandNode() [2/2]

```
CommandNode (
    std::shared_ptr< Node::NodeImpl > pCommand )
```

10.44.2.3 ~CommandNode()

```
virtual ~CommandNode ( ) [virtual]
```

10.44.3 Member Function Documentation**10.44.3.1 Execute()**

```
virtual void Execute (
    bool Verify = true ) [virtual]
```

Execute the command.

Parameters

<i>Verify</i>	Enables AccessMode and Range verification (default = true)
---------------	--

10.44.3.2 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

10.44.3.3 operator()

```
virtual void operator() () [virtual]
```

Execute the command.

10.44.3.4 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](#)

10.45 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 \(\)](#)

10.45.1 Detailed Description

Definition of a simple [Counter](#) class.

10.45.2 Constructor & Destructor Documentation

10.45.2.1 Counter()

```
Counter ( ) [inline]
```

10.45.3 Member Function Documentation

10.45.3.1 GetValue()

```
unsigned int GetValue ( ) const [inline]
```

10.45.3.2 IsZero()

```
bool IsZero ( ) [inline]
```

10.45.3.3 operator unsigned int()

```
operator unsigned int ( ) [inline]
```

10.45.3.4 operator++() [1/2]

```
unsigned int operator++ ( ) [inline]
```

10.45.3.5 operator++() [2/2]

```
unsigned int operator++ (
    int ) [inline]
```

10.45.3.6 operator--() [1/2]

```
unsigned int operator-- (
    int ) [inline]
```

10.45.3.7 operator--() [2/2]

```
unsigned int operator-- ( ) [inline]
```

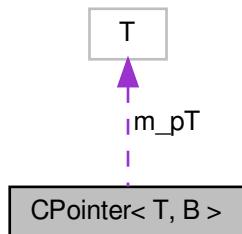
The documentation for this class was generated from the following file:

- include/SpinGenApi/Counter.h

10.46 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.

10.46.1 Detailed Description

```
template<class T, class B = IBase>
class Spinnaker::GenApi::CPointer< T, B >
```

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

10.46.2 Constructor & Destructor Documentation

10.46.2.1 CPointer() [1/2]

```
CPointer (
    void ) [inline]
```

Default constructor.

10.46.2.2 CPointer() [2/2]

```
CPointer (
    B * pB ) [inline]
```

Constructor from INode pointer type.

10.46.2.3 ~CPointer()

```
virtual ~CPointer (
    void ) [inline], [virtual]
```

10.46.3 Member Function Documentation**10.46.3.1 IsValid()**

```
bool IsValid ( ) const throw () [inline]
```

true if the pointer is valid

10.46.3.2 operator bool()

```
operator bool (
    void ) const throw () [inline]
```

true if the pointer is valid

10.46.3.3 operator T*()

```
operator T* (
    void ) const [inline]
```

Dereferencing.

10.46.3.4 operator"!=() [1/5]

```
bool operator!= (
    const CPointer< T, B > & rT ) const [inline]  
pointer unequal
```

10.46.3.5 operator"!=() [2/5]

```
bool operator!= (
    T * pT ) const [inline]  
pointer unequal
```

10.46.3.6 operator"!=() [3/5]

```
bool operator!= (
    const long int nMustBeNull ) const [inline]  
pointer unequal
```

10.46.3.7 operator"!=() [4/5]

```
bool operator!= (
    const int nMustBeNull ) const [inline]  
pointer unequal
```

10.46.3.8 operator"!=() [5/5]

```
bool operator!= (
    const std::nullptr_t nullPtr ) const [inline]  
pointer unequal
```

10.46.3.9 operator()()

```
T& operator() (
    void ) const [inline]
```

Dereferencing.

10.46.3.10 operator*()

```
T& operator* (
    void ) const [inline]
```

Dereferencing.

10.46.3.11 operator->()

```
T* operator-> (
    void ) const [inline]
```

Dereferencing.

10.46.3.12 operator=()

```
void operator= (
    B * pB ) [inline]
```

Assign INode Pointer.

10.46.3.13 operator==() [1/3]

```
bool operator== (
    T * pT ) const [inline]
```

pointer equal

10.46.3.14 operator==() [2/3]

```
bool operator== (
    const CPointer< T, B > & rT ) const [inline]
```

pointer equal

10.46.3.15 operator==() [3/3]

```
bool operator== (
    int nMustBeNull ) const [inline]
```

pointer equal

10.46.4 Member Data Documentation

10.46.4.1 m_pT

`T* m_pT [protected]`

Underlying raw pointer.

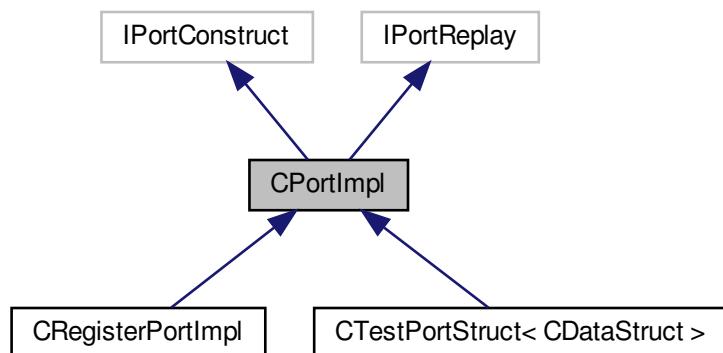
The documentation for this class was generated from the following file:

- include/SpinGenApi/Pointer.h

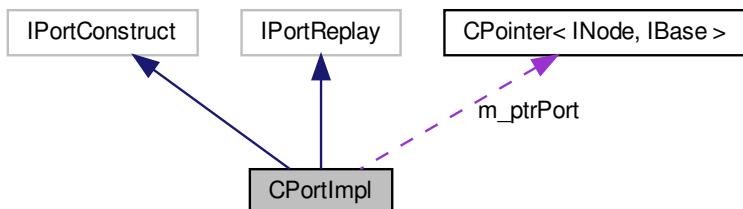
10.47 CPortImpl Class Reference

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 GetSwapEndianess ()`
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.

10.47.1 Detailed Description

Standard implementation for a port.

10.47.2 Constructor & Destructor Documentation

10.47.2.1 CPortImpl()

`CPortImpl () [inline]`

Constructor.

10.47.2.2 ~CPortImpl()

`virtual ~CPortImpl () [inline], [virtual]`

Destructor.

10.47.3 Member Function Documentation

10.47.3.1 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 >](#).

10.47.3.2 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

10.47.3.3 InvalidateNode()

```
void InvalidateNode ( ) [inline]
```

10.47.3.4 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 >](#).

10.47.3.5 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.

10.47.3.6 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented in [CRegisterPortImpl](#).

10.47.3.7 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 >](#).

10.47.4 Member Data Documentation

10.47.4.1 m_ptrPort

```
CNodePtr m_ptrPort [protected]
```

Pointer to the node holding a reference to this implementation.

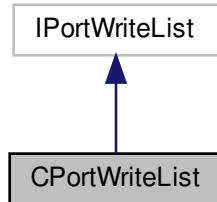
The documentation for this class was generated from the following file:

- include/SpinGenApi/PortImpl.h

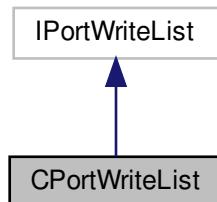
10.48 CPortWriteList Class Reference

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`

10.48.1 Detailed Description

Container holding a list of port write commands.

10.48.2 Constructor & Destructor Documentation

10.48.2.1 CPortWriteList()

```
CPortWriteList ()
```

Constructor.

10.48.2.2 ~CPortWriteList()

```
~CPortWriteList ()
```

Destructor.

10.48.3 Member Function Documentation

10.48.3.1 GetCookie()

```
virtual int64_t GetCookie () [virtual]
```

Gets the cookie a port implementation may have set for caching a command list.

10.48.3.2 GetPortWriteListHandle()

```
void* GetPortWriteListHandle ()
```

10.48.3.3 Replay()

```
virtual void Replay (
    IPort * pPort ) [virtual]
```

Replays the write command to the given port interface.

10.48.3.4 SetCookie()

```
virtual void SetCookie (
    const int64_t Value ) [virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

10.48.3.5 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

10.48.4 Member Data Documentation

10.48.4.1 m_pWriteList

```
void* m_pWriteList [protected]
```

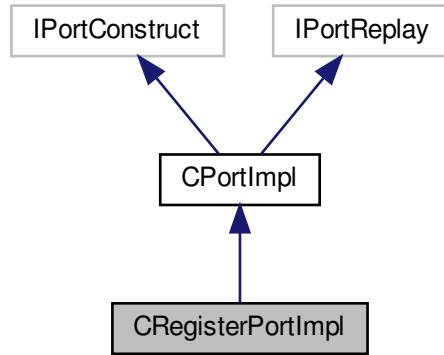
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortWriteList.h](#)

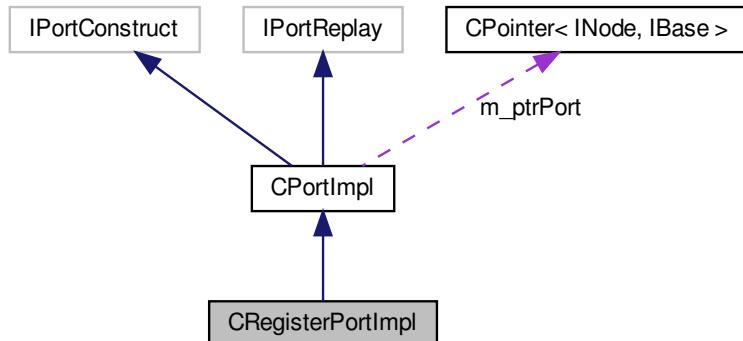
10.49 CRegisterPortImpl Class Reference

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

10.49.1 Detailed Description

Standard implementation for a port using a register based transport layer.

10.49.2 Constructor & Destructor Documentation

10.49.2.1 CRegisterPortImpl()

```
CRegisterPortImpl (
    int MaxNumQuadlets = 1,
    bool TransportLayerSwapsEndianess = false ) [inline]
```

Constructor.

10.49.2.2 ~CRegisterPortImpl()

```
virtual ~CRegisterPortImpl () [inline], [virtual]
```

Destructor.

10.49.3 Member Function Documentation

10.49.3.1 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](#).

10.49.3.2 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

10.49.3.3 ReadRegister()

```
virtual void ReadRegister (
    uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Reads an array of quadlets from the port.

10.49.3.4 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented from [CPortImpl](#).

10.49.3.5 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](#).

10.49.3.6 WriteRegister()

```
virtual void WriteRegister (
    const uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes an array of quadlets to the port.

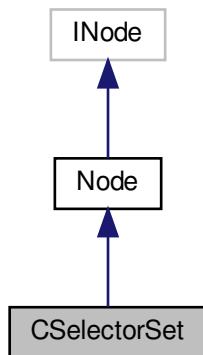
The documentation for this class was generated from the following file:

- include/SpinGenApi/[RegisterPortImpl.h](#)

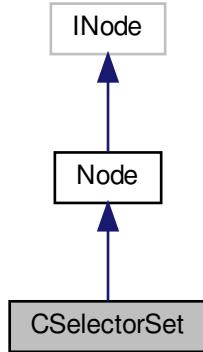
10.50 CSelectorSet Class Reference

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 GenlCam::gcstring ToString \(\)](#)
- [virtual void GetSelectorList \(FeatureList_t &SelectorList, bool Incremental=false\)](#)

Additional Inherited Members

10.50.1 Detailed Description

The set of selectors selecting a given node.

10.50.2 Constructor & Destructor Documentation

10.50.2.1 CSelectorSet()

```
CSelectorSet (
    IBase * pBase )
```

Constructor.

Parameters

<i>pBase</i>	Feature selected by the selector set
--------------	--------------------------------------

10.50.2.2 ~CSelectorSet()

```
~CSelectorSet ( )
```

Destructor.

10.50.3 Member Function Documentation**10.50.3.1 GetSelectorList()**

```
virtual void GetSelectorList (
    FeatureList_t & SelectorList,
    bool Incremental = false ) [virtual]
```

10.50.3.2 IsEmpty()

```
bool IsEmpty ( )
```

returns true if no selectors are present

10.50.3.3 Restore()

```
virtual void Restore ( ) [virtual]
```

10.50.3.4 SetFirst()

```
virtual bool SetFirst ( ) [virtual]
```

10.50.3.5 SetNext()

```
virtual bool SetNext (  
    bool Tick = true ) [virtual]
```

10.50.3.6 ToString()

```
virtual GenICam::gcstring ToString ( ) [virtual]
```

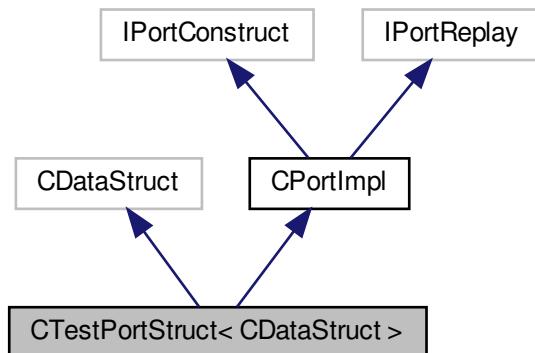
The documentation for this class was generated from the following file:

- include/SpinGenApi/SelectorSet.h

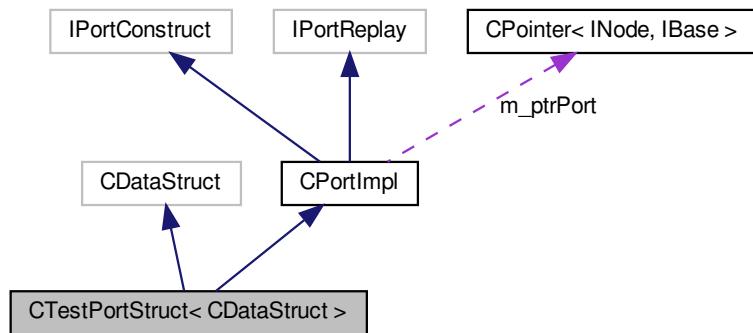
10.51 CTestPortStruct< CDataStruct > Class Template Reference

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 GetPrincipallInterfaceType () 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

10.51.1 Detailed Description

```
template<class CDataStruct>
class Spinnaker::GenApi::CTestPortStruct< CDataStruct >
```

Implements a register spaces based on a C++ struct.

10.51.2 Constructor & Destructor Documentation

10.51.2.1 CTestPortStruct()

```
CTestPortStruct (
    int64_t BaseAddress = 0 ) [inline]
```

10.51.3 Member Function Documentation

10.51.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]
```

Get the access mode of the node.

Implements [CPortImpl](#).

10.51.3.2 GetNumReads()

```
int64_t GetNumReads ( ) [inline]
```

Returns the number of reads since lastReset Statistics.

10.51.3.3 GetNumWrites()

```
int64_t GetNumWrites ( ) [inline]
```

Returns the number of writes since lastReset Statistics.

10.51.3.4 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [inline], [virtual]
```

Get the type of the main interface of a node.

10.51.3.5 MemSet()

```
void MemSet (
    const char FillValue ) [inline]
```

10.51.3.6 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

10.51.3.7 ResetStatistics()

```
void ResetStatistics ( ) [inline]
```

Resets the read/write statistics.

10.51.3.8 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](#).

10.51.4 Member Data Documentation

10.51.4.1 m_BaseAddress

```
int64_t m_BaseAddress [protected]
```

the base address used for the struct

10.51.4.2 m_NumReads

```
int64_t m_NumReads [protected]
```

Number of reads since last reset.

10.51.4.3 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](#)

10.52 DCAM_CHECKSUM Struct Reference

Public Attributes

- `uint32_t CRCChecksum`

10.52.1 Member Data Documentation

10.52.1.1 CRCChecksum

```
uint32_t CRCChecksum
```

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

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

10.53 DCAM_CHUNK_TRAILER Struct Reference

Public Attributes

- SPIN_GUID [ChunkID](#)
- `uint32_t ChunkLength`
- `uint32_t InverseChunkLength`

10.53.1 Member Data Documentation

10.53.1.1 ChunkID

```
SPIN_GUID ChunkID
```

10.53.1.2 ChunkLength

```
uint32_t ChunkLength
```

10.53.1.3 InverseChunkLength

```
uint32_t InverseChunkLength
```

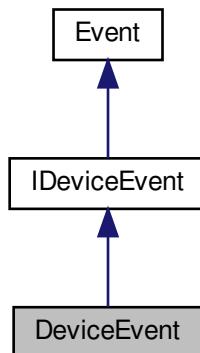
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

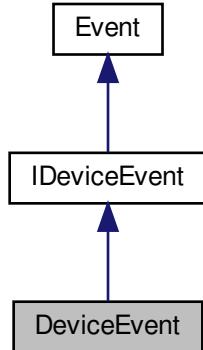
10.54 DeviceEvent Class Reference

A handler to device events.

Inheritance diagram for DeviceEvent:



Collaboration diagram for DeviceEvent:



Public Member Functions

- [`DeviceEvent \(\)`](#)
Default constructor.
- [`virtual ~DeviceEvent \(\)`](#)
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

- [`DeviceEvent & operator= \(const DeviceEvent &\)`](#)
Assignment operator.

Additional Inherited Members

10.54.1 Detailed Description

A handler to device events.

10.54.2 Constructor & Destructor Documentation

10.54.2.1 DeviceEvent()

```
DeviceEvent ( )
```

Default constructor.

10.54.2.2 ~DeviceEvent()

```
virtual ~DeviceEvent ( ) [virtual]
```

Virtual destructor.

10.54.3 Member Function Documentation

10.54.3.1 GetDeviceEventId()

```
uint64_t GetDeviceEventId ( ) const [virtual]
```

Get the ID of the device event.

Returns

The device event ID

Implements [IDeviceEvent](#).

10.54.3.2 GetDeviceEventName()

```
GenICam::gcstring GetDeviceEventName ( ) const [virtual]
```

Get the name of the device event.

Returns

The device event name

Implements [IDeviceEvent](#).

10.54.3.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (
    Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Device event callback.

Parameters

<code>eventName</code>	The name of the event
------------------------	-----------------------

Implements [IDeviceEvent](#).

10.54.3.4 operator=()

```
DeviceEvent& operator= (
    const DeviceEvent & ) [protected]
```

Assignment operator.

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

- [include/DeviceEvent.h](#)

10.55 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](#)

10.55.1 Detailed Description

Vector of doubles with reference counting.

10.55.2 Constructor & Destructor Documentation

10.55.2.1 `double_autovector_t()` [1/3]

```
double_autovector_t ( )
```

10.55.2.2 `double_autovector_t()` [2/3]

```
double_autovector_t (
    const double_autovector_t & obj )
```

10.55.2.3 `double_autovector_t()` [3/3]

```
double_autovector_t (
    size_t n ) [explicit]
```

10.55.2.4 `~double_autovector_t()`

```
virtual ~double_autovector_t (
    void ) [virtual]
```

10.55.3 Member Function Documentation

10.55.3.1 `operator delete()`

```
void operator delete (
    void * pWhere )
```

10.55.3.2 `operator new()`

```
void* operator new (
    size_t uiSize )
```

10.55.3.3 operator=()

```
double_autovector_t& operator= (
    const double_autovector_t & obj )
```

10.55.3.4 operator[]() [1/2]

```
double& operator[ ] (
    size_t uiIndex )
```

10.55.3.5 operator[]() [2/2]

```
const double& operator[ ] (
    size_t uiIndex ) const
```

10.55.3.6 size()

```
size_t size ( ) const
```

10.55.4 Member Data Documentation**10.55.4.1 _pCount**

```
ATOMIC_VARIABLE* _pCount [protected]
```

10.55.4.2 _pv

```
std::vector<double>* _pv [protected]
```

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

- include/SpinGenApi/Autovector.h

10.56 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.

10.56.1 Detailed Description

Holds conversion methods for the access mode enumeration.

10.56.2 Member Function Documentation

10.56.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EAccessMode * pValue ) [static]
```

Converts a string to enum value.

10.56.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EAccessMode * pValue ) [static]
```

Converts a string to an int32_t property.

10.56.2.3 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](#)

10.57 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.

10.57.1 Detailed Description

Holds conversion methods for the caching mode enumeration.

10.57.2 Member Function Documentation

10.57.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

Converts a string to enum value.

10.57.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

10.57.2.3 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](#)

10.58 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.

10.58.1 Detailed Description

Holds conversion methods for the notation type of floats.

10.58.2 Member Function Documentation

10.58.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EDisplayNotation * pValue ) [static]
```

Converts a string to enum value.

10.58.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EDisplayNotation * pValue ) [static]
```

Converts a string to an int32_t property.

10.58.2.3 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](#)

10.59 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.

10.59.1 Detailed Description

Holds conversion methods for the endianess enumeration.

10.59.2 Member Function Documentation

10.59.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EEndianess * pValue ) [static]
```

Converts a string to enum value.

10.59.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EEndianess * pValue ) [static]
```

Converts a string to an int32_t property.

10.59.2.3 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](#)

10.60 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.

10.60.1 Detailed Description

helper class converting EGenApiSchemaVersion from and to string

10.60.2 Member Function Documentation

10.60.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to enum value.

10.60.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to an int32_t property.

10.60.2.3 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](#)

10.61 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.

10.61.1 Detailed Description

Holds conversion methods for the notation type of floats.

10.61.2 Member Function Documentation

10.61.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to enum value.

10.61.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to an int32_t property.

10.61.2.3 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](#)

10.62 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.

10.62.1 Detailed Description

Holds conversion methods for the namespace enumeration.

10.62.2 Member Function Documentation

10.62.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ENameSpace * pValue ) [static]
```

Converts a string to enum value.

10.62.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ENameSpace * pValue ) [static]
```

Converts a string to an int32_t property.

10.62.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ENameSpace Value ) [static]
```

Converts a string to an int32_t property.

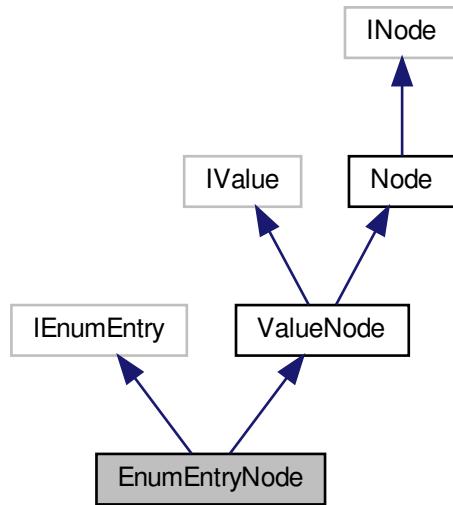
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

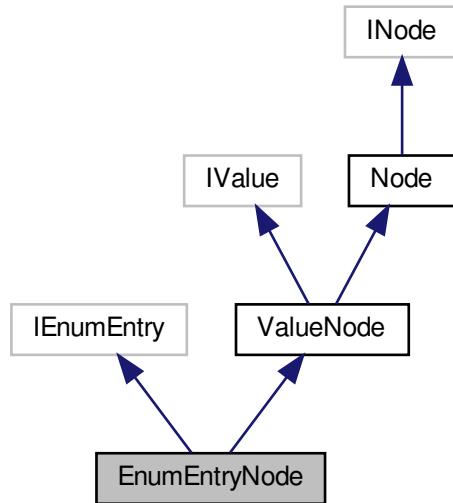
10.63 EnumEntryNode Class Reference

[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 GenICam::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

10.63.1 Detailed Description

[Interface](#) for string properties.

10.63.2 Constructor & Destructor Documentation

10.63.2.1 `EnumEntryNode()` [1/2]

`EnumEntryNode ()`

10.63.2.2 `EnumEntryNode()` [2/2]

```
EnumEntryNode (
    std::shared_ptr< Node::NodeImpl > pEnumEntry )
```

10.63.2.3 `~EnumEntryNode()`

`virtual ~EnumEntryNode () [virtual]`

10.63.3 Member Function Documentation

10.63.3.1 GetNumericValue()

```
virtual double GetNumericValue ( ) [virtual]
```

Get double number associated with the entry.

10.63.3.2 GetSymbolic()

```
virtual GenICam::gcstring GetSymbolic ( ) const [virtual]
```

Get symbolic enum value.

10.63.3.3 GetValue()

```
virtual int64_t GetValue ( ) [virtual]
```

Get numeric enum value.

10.63.3.4 IsSelfClearing()

```
virtual bool IsSelfClearing ( ) [virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

10.63.3.5 SetReference()

```
virtual void SetReference (  
    INode * pBase ) [virtual]
```

overload SetReference for EnumEntry

Reimplemented from [ValueNode](#).

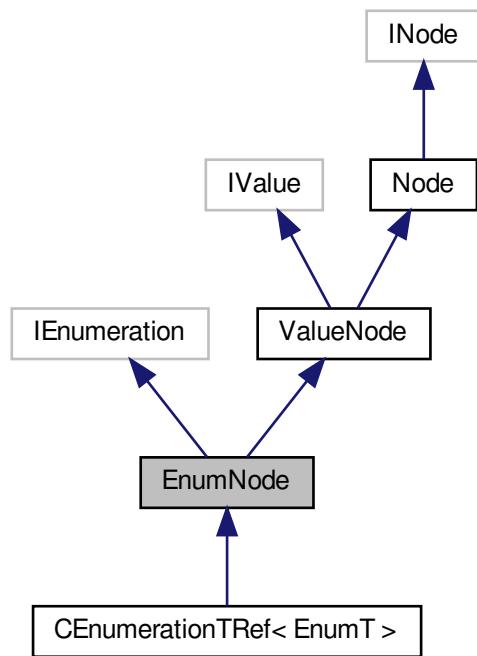
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumEntryNode.h](#)

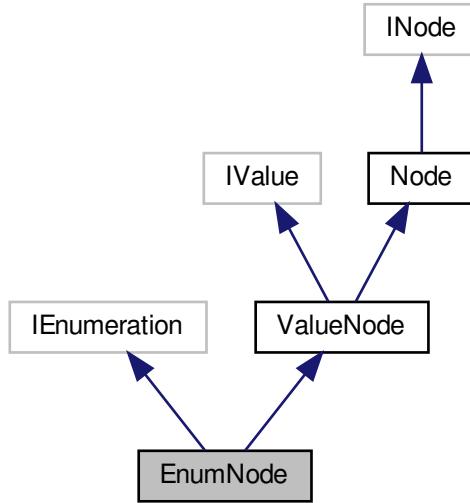
10.64 EnumNode Class Reference

[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](#)

10.64.1 Detailed Description

[Interface](#) for string properties.

10.64.2 Constructor & Destructor Documentation

10.64.2.1 [EnumNode\(\)](#) [1/2]

```
EnumNode ( )
```

10.64.2.2 [EnumNode\(\)](#) [2/2]

```
EnumNode (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

10.64.2.3 [~EnumNode\(\)](#)

```
virtual ~EnumNode ( ) [virtual]
```

10.64.3 Member Function Documentation

10.64.3.1 [GetCurrentEntry\(\)](#)

```
virtual IEnumEntry* GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get the current entry.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.64.3.2 GetEntries()

```
virtual void GetEntries (
    NodeList_t & Entries ) [virtual]
```

Get list of entry nodes.

10.64.3.3 GetEntry()

```
virtual IEnumEntry* GetEntry (
    const int64_t IntValue ) [virtual]
```

Get an entry node by its IntValue.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.64.3.4 GetEntryByName()

```
virtual IEnumEntry* GetEntryByName (
    const GenICam::gcstring & Symbolic ) [virtual]
```

Get an entry node by name.

10.64.3.5 GetIntValue()

```
virtual int64_t GetIntValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get integer node value.

Parameters

Verify	Enables Range verification (default = false). The AccessMode is always checked
IgnoreCache	If true the value is read ignoring any caches (default = false)

Returns

The value read

10.64.3.6 GetSymbolics()

```
virtual void GetSymbolics (
    StringList_t & Symbolics ) [virtual]
```

Get list of symbolic Values.

10.64.3.7 operator*()

```
virtual GenICam::gcstring operator* ( ) [virtual]
```

Get string node value.

10.64.3.8 operator=()

```
virtual IEnumeration& operator= (
    const GenICam::gcstring & ValueStr ) [virtual]
```

Set string node value.

Reimplemented in [CEnumerationTRef<EnumT>](#).

10.64.3.9 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)

10.64.3.10 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Enumeration

Reimplemented from [ValueNode](#).

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.64.4 Member Data Documentation

10.64.4.1 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](#)

10.65 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.

10.65.1 Detailed Description

Holds conversion methods for the representation enumeration.

10.65.2 Member Function Documentation

10.65.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to enum value.

10.65.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to an int32_t property.

10.65.2.3 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

10.66 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.

10.66.1 Detailed Description

Holds conversion methods for the sign enumeration.

10.66.2 Member Function Documentation

10.66.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to enum value.

10.66.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to an int32_t property.

10.66.2.3 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

10.67 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.

10.67.1 Detailed Description

Holds conversion methods for the converter formulas.

10.67.2 Member Function Documentation

10.67.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESlope * pValue ) [static]
```

Converts a string to enum value.

10.67.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ESlope * pValue ) [static]
```

Converts a string to an int32_t property.

10.67.2.3 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

10.68 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.

10.68.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

10.68.2 Member Function Documentation

10.68.2.1 [FromString\(\)](#)

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EStandardNameSpace * pValue ) [static]
```

Converts a string to enum value.

10.68.2.2 [ToString\(\)](#) [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EStandardNameSpace * pValue ) [static]
```

Converts a string to an int32_t property.

10.68.2.3 [ToString\(\)](#) [2/2]

```
static GenICam::gcstring ToString (
    EStandardNameSpace Value ) [static]
```

Converts a string to an int32_t property.

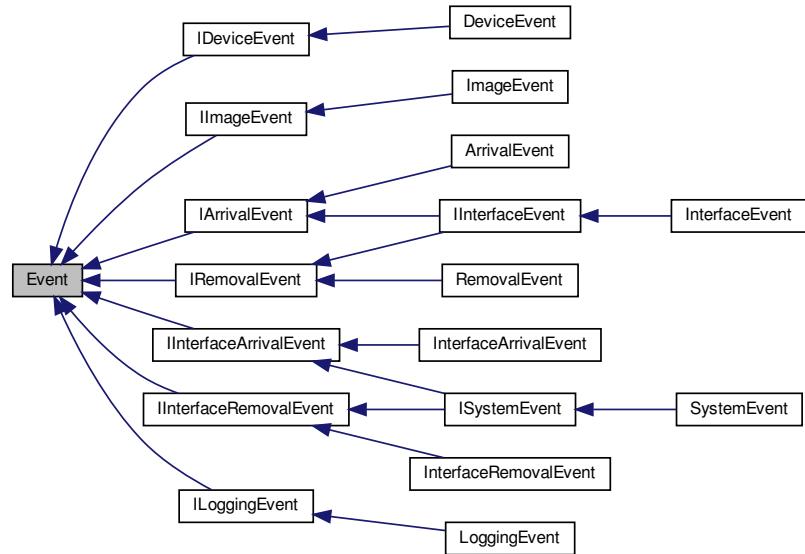
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

10.69 Event Class Reference

The base class for all event types.

Inheritance diagram for Event:



Public Member Functions

- virtual `~Event ()`
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

- `Event ()`
- `Event & operator= (const Event &)`
- void `SetEventPayload (uint8_t *offset, size_t length)`

Protected Attributes

- `EventData * m_pEventData`

Friends

- class [EventProcessor](#)
- class [IDataStream](#)
- class [Stream](#)

10.69.1 Detailed Description

The base class for all event types.

10.69.2 Constructor & Destructor Documentation

10.69.2.1 ~Event()

```
virtual ~Event( ) [virtual]
```

Virtual Destructor.

10.69.2.2 Event()

```
Event( ) [protected]
```

10.69.3 Member Function Documentation

10.69.3.1 GetEventPayloadData()

```
const uint8_t* GetEventPayloadData( )
```

Gets the event payload data.

Returns

The event payload data

10.69.3.2 GetEventPayloadDataSize()

```
const size_t GetEventPayloadDataSize ( )
```

Gets the event payload data size.

Returns

The event payload data size

10.69.3.3 GetEventType()

```
EventType GetEventType ( )
```

Gets the event type.

Returns

The event type

10.69.3.4 operator=()

```
Event& operator= (
    const Event & ) [protected]
```

10.69.3.5 SetEventPayload()

```
void SetEventPayload (
    uint8_t * offset,
    size_t length ) [protected]
```

10.69.3.6 SetEventType()

```
void SetEventType (
    EventType eventType )
```

Sets the event type.

Parameters

<code>eventType</code>	The event type
------------------------	----------------

10.69.4 Friends And Related Function Documentation

10.69.4.1 EventProcessor

```
friend class EventProcessor [friend]
```

10.69.4.2 IDataStream

```
friend class IDataStream [friend]
```

10.69.4.3 Stream

```
friend class Stream [friend]
```

10.69.5 Member Data Documentation

10.69.5.1 m_pEventData

```
EventData* m_pEventData [protected]
```

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

- [include/Event.h](#)

10.70 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.

10.70.1 Detailed Description

Holds conversion methods for the visibility enumeration.

10.70.2 Member Function Documentation

10.70.2.1 [FromString\(\)](#)

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to enum value.

10.70.2.2 [ToString\(\)](#) [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to an int32_t property.

10.70.2.3 [ToString\(\)](#) [2/2]

```
static GenICam::gcstring ToString (
    EVisibility Value ) [static]
```

Converts a string to an int32_t property.

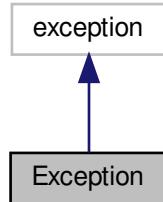
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

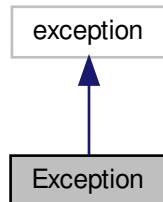
10.71 Exception Class Reference

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

10.71.1 Detailed Description

The `Exception` object represents an error that is returned from the library.

Overloaded operators allow comparisons against other `Exception` objects.

10.71.2 Constructor & Destructor Documentation

10.71.2.1 `Exception()` [1/4]

`Exception ()`

Default constructor.

10.71.2.2 `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

10.71.2.3 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
<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

10.71.2.4 Exception() [4/4]

```
Exception (
    const Exception & except )
```

Copy constructor.

10.71.2.5 ~Exception()

```
virtual ~Exception ( ) throw () [virtual]
```

Default destructor.

10.71.3 Member Function Documentation

10.71.3.1 GetBuildDate()

```
const char* GetBuildDate ( ) const
```

10.71.3.2 GetBuildTime()

```
const char* GetBuildTime ( ) const
```

10.71.3.3 GetError()

```
Error GetError ( ) const
```

10.71.3.4 GetErrorMessage()

```
const char* GetErrorMessage ( ) const
```

Accessor Functions.

10.71.3.5 GetFileName()

```
const char* GetFileName ( ) const
```

10.71.3.6 GetFullErrorMessage()

```
const char* GetFullErrorMessage ( ) const
```

Gets the error code and full error message including the line, file, function, build date, and time.

10.71.3.7 GetFunctionName()

```
const char* GetFunctionName ( ) const
```

10.71.3.8 GetLineNumber()

```
int GetLineNumber ( ) const
```

10.71.3.9 operator"!=()

```
bool operator!= (
    const Error err ) const
```

Inequality operator.

10.71.3.10 operator=()

```
Exception& operator= (
    const Exception & except )
```

Assignment operator.

10.71.3.11 operator==()

```
bool operator== (
    const Error err ) const
```

Equality operator.

10.71.3.12 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](#)
-

10.72 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.

10.72.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

10.72.2 Member Function Documentation

10.72.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EYesNo * pValue ) [static]
```

Converts a string to enum value.

10.72.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EYesNo * pValue ) [static]
```

Converts a string to an int32_t property.

10.72.2.3 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](#)

10.73 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.

10.73.1 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

10.73.2 Constructor & Destructor Documentation

10.73.2.1 FileProtocolAdapter()

```
FileProtocolAdapter ( )
```

Constructor.

10.73.2.2 ~FileProtocolAdapter()

```
virtual ~FileProtocolAdapter ( ) [virtual]
```

10.73.3 Member Function Documentation

10.73.3.1 attach()

```
bool attach (
    ::Spinnaker::GenApi::INodeMap * pInterface )
```

attach file protocol adapter to [NodeMap](#)

Parameters

<code>pInterface</code>	NodeMap of the device to which the FileProtocolAdapter is attached
-------------------------	--

Returns

true if attach was successful, false if not

10.73.3.2 closeFile()

```
bool closeFile (
    const char * pFileName )
```

close a file on the device

Parameters

<code>pFileName</code>	filename of the file to open. The filename must exist in the Enumeration FileSelector
------------------------	---

Returns

true on success, false on error

10.73.3.3 deleteFile()

```
bool deleteFile (
    const char * pFileName )
```

Delete the content of the file.

Parameters

<code>pFileName</code>	filename of the file to open. The filename must exist in the Enumeration FileSelector
------------------------	---

Returns

true on success, false on error

10.73.3.4 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 FileMode

Returns

max length of FileAccessBuffer in the given mode on the given file

10.73.3.5 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 FileMode

Returns

true on success, false on error

10.73.3.6 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

10.73.3.7 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

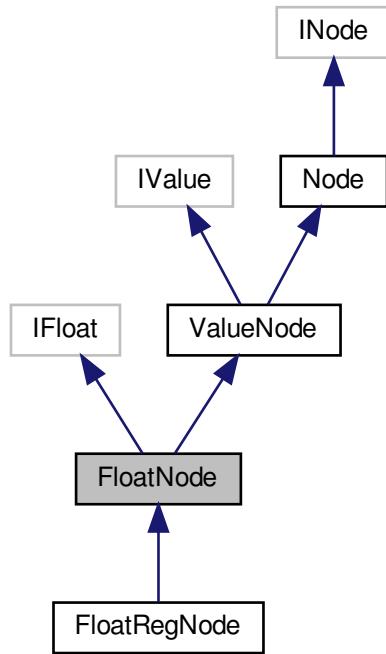
The documentation for this class was generated from the following file:

- include/SpinGenApi/Filestream.h

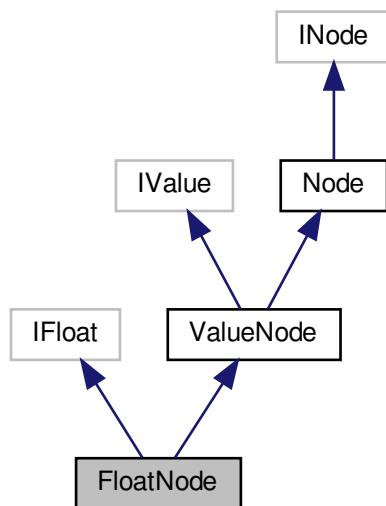
10.74 FloatNode Class Reference

[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_vector_t GetListOfValidValues (bool bounded=true)`
Get list of valid value.
- `virtual ERepresentation GetRepresentation ()`
Get recommended representation.
- `virtual GenICam::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.
- `IInteger * 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

10.74.1 Detailed Description

Interface for string properties.

10.74.2 Constructor & Destructor Documentation

10.74.2.1 `FloatNode()` [1/2]

```
FloatNode ( )
```

10.74.2.2 `FloatNode()` [2/2]

```
FloatNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

10.74.2.3 `~FloatNode()`

```
virtual ~FloatNode ( ) [virtual]
```

10.74.3 Member Function Documentation

10.74.3.1 `GetDisplayNotation()`

```
virtual EDisplayNotation GetDisplayNotation ( ) const [virtual]
```

Get the way the float should be converted to a string.

10.74.3.2 `GetDisplayPrecision()`

```
virtual int64_t GetDisplayPrecision ( ) const [virtual]
```

Get the precision to be used when converting the float to a string.

10.74.3.3 `GetEnumAlias()`

```
IEnumeration* GetEnumAlias ( )
```

gets the interface of an alias node.

10.74.3.4 GetInc()

```
virtual double GetInc ( ) [virtual]
```

Get the constant increment if there is any.

10.74.3.5 GetIncMode()

```
virtual EIncMode GetIncMode ( ) [virtual]
```

Get increment mode.

10.74.3.6 GetIntAlias()

```
IIInteger* GetIntAlias ( )
```

gets the interface of an alias node.

10.74.3.7 GetListOfValidValues()

```
virtual double_vector_t GetListOfValidValues ( bool bounded = true ) [virtual]
```

Get list of valid value.

10.74.3.8 GetMax()

```
virtual double GetMax ( ) [virtual]
```

Get maximum value allowed.

10.74.3.9 GetMin()

```
virtual double GetMin ( ) [virtual]
```

Get minimum value allowed.

10.74.3.10 GetRepresentation()

```
virtual ERepresentation GetRepresentation() [virtual]
```

Get recommended representation.

10.74.3.11 GetUnit()

```
virtual GenICam::gcstring GetUnit() const [virtual]
```

Get the physical unit name.

10.74.3.12 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

10.74.3.13 HasInc()

```
virtual bool HasInc() [virtual]
```

True if the float has a constant increment.

10.74.3.14 ImposeMax()

```
virtual void ImposeMax(
    double Value) [virtual]
```

Restrict maximum value.

10.74.3.15 ImposeMin()

```
virtual void ImposeMin (
    double Value ) [virtual]
```

Restrict minimum value.

10.74.3.16 operator()()

```
virtual double operator() () [virtual]
```

Get node value.

10.74.3.17 operator*()

```
virtual double operator* () [virtual]
```

Get node value.

10.74.3.18 operator=()

```
virtual IFloat& operator= (
    double Value ) [virtual]
```

Set node value.

10.74.3.19 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Float

Reimplemented from [ValueNode](#).

Reimplemented in [FloatRegNode](#).

10.74.3.20 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)

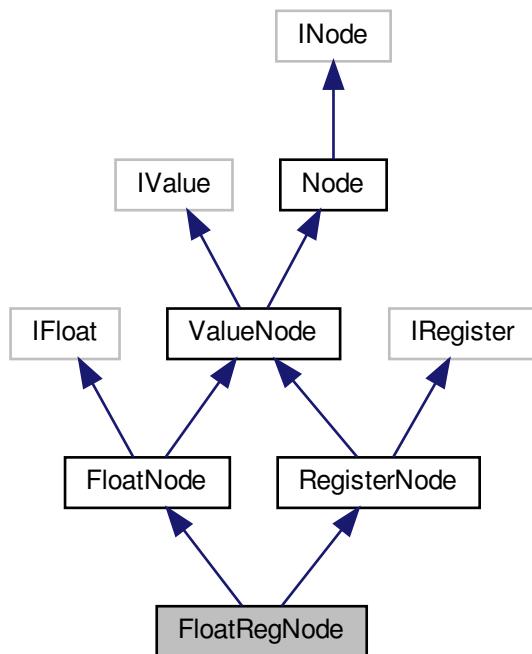
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatNode.h](#)

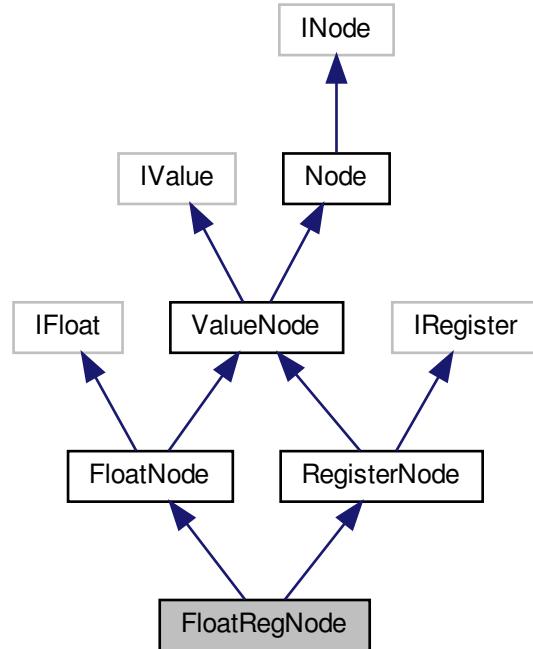
10.75 FloatRegNode Class Reference

[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

10.75.1 Detailed Description

[Interface](#) for string properties.

10.75.2 Constructor & Destructor Documentation

10.75.2.1 [FloatRegNode\(\)](#) [1/2]

```
FloatRegNode ( )
```

10.75.2.2 [FloatRegNode\(\)](#) [2/2]

```
FloatRegNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

10.75.2.3 [~FloatRegNode\(\)](#)

```
virtual ~FloatRegNode ( ) [virtual]
```

10.75.3 Member Function Documentation**10.75.3.1 [SetReference\(\)](#)**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [FloatNode](#).

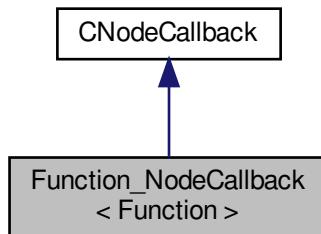
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatRegNode.h](#)

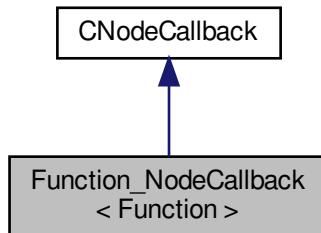
10.76 Function_NodeCallback< Function > Class Template Reference

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

10.76.1 Detailed Description

```
template<class Function>
class Spinnaker::GenApi::Function_NodeCallback< Function >
```

Container for a function pointer.

10.76.2 Constructor & Destructor Documentation

10.76.2.1 Function_NodeCallback()

```
Function_NodeCallback (
    INode * pNode,
    const Function & function,
    ECallbackType CallbackType ) [inline]
```

Constructor.

10.76.3 Member Function Documentation

10.76.3.1 Destroy()

```
virtual void Destroy () [inline], [virtual]
```

destroys teh object

Implements [CNodeCallback](#).

10.76.3.2 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](#)

10.77 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)`

10.77.1 Constructor & Destructor Documentation

10.77.1.1 `gcstring()` [1/5]

```
gcstring ( )
```

10.77.1.2 `gcstring()` [2/5]

```
gcstring (
    const char * pc )
```

10.77.1.3 `gcstring()` [3/5]

```
gcstring (
    const char * pc,
    size_t n )
```

10.77.1.4 `gcstring()` [4/5]

```
gcstring (
    size_t count,
    char ch )
```

10.77.1.5 gcstring() [5/5]

```
gcstring (
    const gcstring & str )
```

10.77.1.6 ~gcstring()

```
virtual ~gcstring (
    void ) [virtual]
```

10.77.2 Member Function Documentation**10.77.2.1 _npos()**

```
static size_t _npos (
    void ) [static]
```

10.77.2.2 append() [1/2]

```
virtual gcstring& append (
    const gcstring & str ) [virtual]
```

10.77.2.3 append() [2/2]

```
virtual gcstring& append (
    size_t count,
    char ch ) [virtual]
```

10.77.2.4 assign() [1/4]

```
virtual gcstring& assign (
    const gcstring & str ) [virtual]
```

10.77.2.5 assign() [2/4]

```
virtual gcstring& assign (
    size_t count,
    char ch )  [virtual]
```

10.77.2.6 assign() [3/4]

```
virtual gcstring& assign (
    const char * pc )  [virtual]
```

10.77.2.7 assign() [4/4]

```
virtual gcstring& assign (
    const char * pc,
    size_t n )  [virtual]
```

10.77.2.8 c_str()

```
virtual const char* c_str (
    void ) const  [virtual]
```

10.77.2.9 compare()

```
virtual int compare (
    const gcstring & str ) const  [virtual]
```

10.77.2.10 empty()

```
virtual bool empty (
    void ) const  [virtual]
```

10.77.2.11 find() [1/5]

```
virtual size_t find (
    char ch,
    size_t offset = 0 ) const [virtual]
```

10.77.2.12 find() [2/5]

```
virtual size_t find (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

10.77.2.13 find() [3/5]

```
virtual size_t find (
    const gcstring & str,
    size_t offset,
    size_t count ) const [virtual]
```

10.77.2.14 find() [4/5]

```
virtual size_t find (
    const char * pc,
    size_t offset = 0 ) const [virtual]
```

10.77.2.15 find() [5/5]

```
virtual size_t find (
    const char * pc,
    size_t offset,
    size_t count ) const [virtual]
```

10.77.2.16 find_first_not_of()

```
virtual size_t find_first_not_of (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

10.77.2.17 find_first_of()

```
virtual size_t find_first_of (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

10.77.2.18 length()

```
virtual size_t length (
    void ) const [virtual]
```

10.77.2.19 max_size()

```
virtual size_t max_size ( ) const [virtual]
```

10.77.2.20 operator const char *()

```
operator const char * (
    void ) const
```

10.77.2.21 operator delete() [1/2]

```
void operator delete (
    void * pWhere )
```

10.77.2.22 operator delete() [2/2]

```
void operator delete (
    void * pWhere,
    void * pNewWhere )
```

10.77.2.23 operator new() [1/2]

```
void* operator new (
    size_t uiSize )
```

10.77.2.24 operator new() [2/2]

```
void* operator new (
    size_t uiSize,
    void * pWhere )
```

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

```
bool operator!= (
    const gcstring & str ) const
```

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

```
bool operator!= (
    const char * pc ) const
```

10.77.2.27 operator+=() [1/5]

```
gcstring& operator+= (
    const gcstring & str )
```

10.77.2.28 operator+=() [2/5]

```
gcstring operator+= (
    const gcstring & str ) const
```

10.77.2.29 operator+=() [3/5]

```
gcstring& operator+= (
    const char * pc )
```

10.77.2.30 operator+=() [4/5]

```
gcstring& operator+= (
    char ch )
```

10.77.2.31 operator+=() [5/5]

```
gcstring operator+= (
    char ch ) const
```

10.77.2.32 operator<()

```
bool operator< (
    const gcstring & str ) const
```

10.77.2.33 operator=()

```
gcstring& operator= (
    const gcstring & str )
```

10.77.2.34 operator==(1/2)

```
bool operator== (
    const gcstring & str ) const
```

10.77.2.35 operator==(2/2)

```
bool operator== (
    const char * pc ) const
```

10.77.2.36 operator>()

```
bool operator> (
    const gcstring & str ) const
```

10.77.2.37 resize()

```
virtual void resize (
    size_t n ) [virtual]
```

10.77.2.38 size()

```
virtual size_t size (
    void ) const [virtual]
```

10.77.2.39 substr()

```
virtual gcstring substr (
    size_t offset = 0,
    size_t count = GCSTRING_NPOS ) const [virtual]
```

10.77.2.40 swap()

```
virtual void swap (
    gcstring & Right ) [virtual]
```

10.77.3 Friends And Related Function Documentation**10.77.3.1 operator+ [1/3]**

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const gcstring & right ) [friend]
```

10.77.3.2 operator+ [2/3]

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const char * right ) [friend]
```

10.77.3.3 operator+ [3/3]

```
SPINNAKER_API friend gcstring operator+ (
    const char * left,
    const gcstring & right ) [friend]
```

10.77.4 Member Data Documentation

10.77.4.1 npos

```
const size_t npos [static]
```

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

- include/SpinGenApi/[GCString.h](#)

10.78 GVCP_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t [ChunkID](#)
- uint32_t [ChunkLength](#)

10.78.1 Detailed Description

header of a GVCP request packet

10.78.2 Member Data Documentation

10.78.2.1 ChunkID

```
uint32_t ChunkID
```

10.78.2.2 ChunkLength

```
uint32_t ChunkLength
```

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

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

10.79 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`

10.79.1 Detailed Description

layout of a GVCP event item (Extended ID flag not set)

10.79.2 Member Data Documentation

10.79.2.1 BlockId

```
uint16_t BlockId
```

10.79.2.2 EventId

```
uint16_t EventId
```

10.79.2.3 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

10.79.2.4 StreamChannelId

```
uint16_t StreamChannelId
```

10.79.2.5 TimestampHigh

```
uint32_t TimestampHigh
```

10.79.2.6 TimestampLow

```
uint32_t TimestampLow
```

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.80 GVCP_EVENT_ITEM_BASIC Struct Reference

layout of a GVCP event item (common to all types)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)

10.80.1 Detailed Description

layout of a GVCP event item (common to all types)

10.80.2 Member Data Documentation

10.80.2.1 EventId

```
uint16_t EventId
```

10.80.2.2 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.81 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`

10.81.1 Detailed Description

layout of a GVCP event item (Extended ID flag set)

10.81.2 Member Data Documentation

10.81.2.1 BlockId

```
uint16_t BlockId
```

10.81.2.2 BlockId64High

```
uint32_t BlockId64High
```

10.81.2.3 BlockId64Low

```
uint32_t BlockId64Low
```

10.81.2.4 EventId

```
uint16_t EventId
```

10.81.2.5 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

10.81.2.6 StreamChannelId

```
uint16_t StreamChannelId
```

10.81.2.7 TimestampHigh

```
uint32_t TimestampHigh
```

10.81.2.8 TimestampLow

```
uint32_t TimestampLow
```

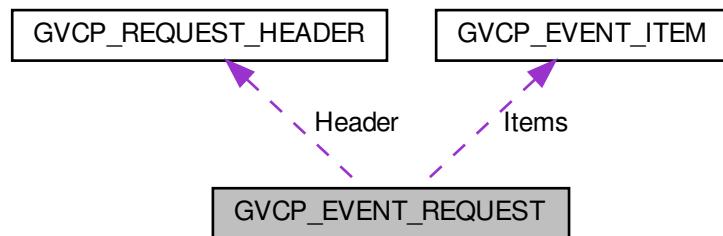
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.82 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]

10.82.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag not set)

10.82.2 Member Data Documentation

10.82.2.1 Header

[GVCP_REQUEST_HEADER](#) Header

10.82.2.2 Items

[GVCP_EVENT_ITEM](#) Items [1]

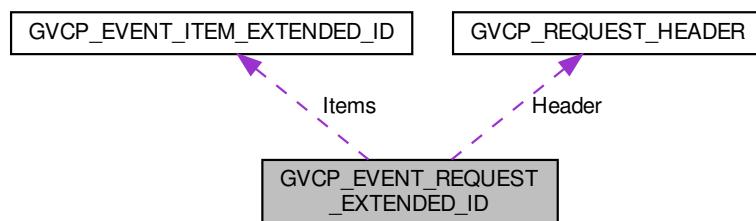
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.83 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]

10.83.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag set)

10.83.2 Member Data Documentation

10.83.2.1 Header

`GVCP_REQUEST_HEADER` Header

10.83.2.2 Items

`GVCP_EVENT_ITEM_EXTENDED_ID` Items [1]

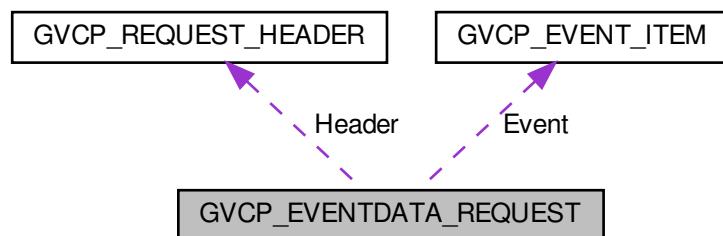
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.84 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]`

10.84.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag not set)

10.84.2 Member Data Documentation

10.84.2.1 Data

```
uint32_t Data[1]
```

10.84.2.2 Event

```
GVCP_EVENT_ITEM Event
```

10.84.2.3 Header

```
GVCP_REQUEST_HEADER Header
```

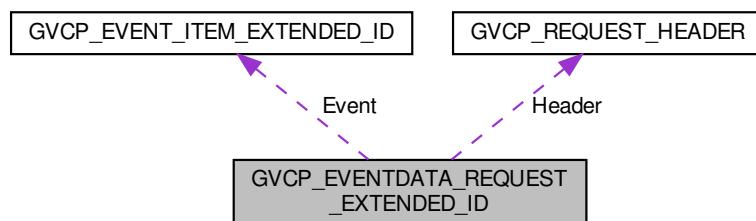
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.85 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]`

10.85.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag set)

10.85.2 Member Data Documentation

10.85.2.1 Data

`uint32_t Data [1]`

10.85.2.2 Event

`GVCP_EVENT_ITEM_EXTENDED_ID` Event

10.85.2.3 Header

`GVCP_REQUEST_HEADER` Header

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.86 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`

10.86.1 Detailed Description

header of a GVCP request packet

10.86.2 Member Data Documentation

10.86.2.1 Command

```
uint16_t Command
```

10.86.2.2 Flags

```
uint8_t Flags
```

10.86.2.3 Length

```
uint16_t Length
```

10.86.2.4 Magic

```
uint8_t Magic
```

10.86.2.5 ReqId

```
uint16_t ReqId
```

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

10.87 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.

10.87.1 Detailed Description

Options for saving H264 files.

10.87.2 Constructor & Destructor Documentation

10.87.2.1 [H264Option\(\)](#)

[H264Option \(\)](#) [inline]

10.87.3 Member Data Documentation

10.87.3.1 [bitrate](#)

`unsigned int bitrate`

Bit-rate to encode at.

10.87.3.2 frameRate

```
float frameRate
```

Frame rate of the stream.

10.87.3.3 height

```
unsigned int height
```

Height of source image.

10.87.3.4 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

10.87.3.5 width

```
unsigned int width
```

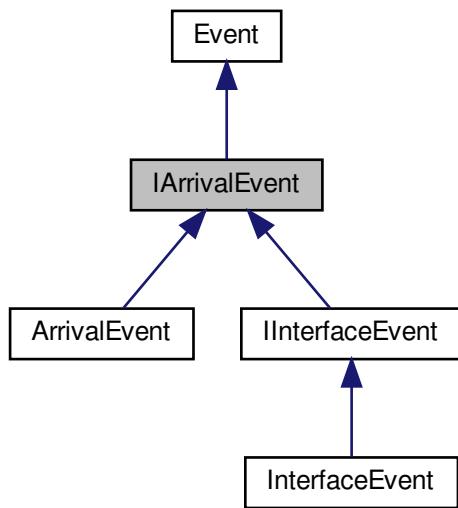
Width of source image.

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

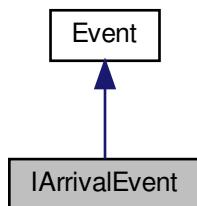
- [include/SpinVideoDefs.h](#)

10.88 IArrivalEvent Class Reference

Inheritance diagram for IArrivalEvent:



Collaboration diagram for IArrivalEvent:



Public Member Functions

- virtual `~IArrivalEvent ()`
- virtual void `OnDeviceArrival (uint64_t serialNumber)=0`

Protected Member Functions

- `IArrivalEvent ()`
- `IArrivalEvent (const IArrivalEvent &)`
- `IArrivalEvent & operator= (const IArrivalEvent &)`

Additional Inherited Members

10.88.1 Constructor & Destructor Documentation

10.88.1.1 ~IArrivalEvent()

```
virtual ~IArrivalEvent ( ) [inline], [virtual]
```

10.88.1.2 IArrivalEvent() [1/2]

```
IArrivalEvent ( ) [inline], [protected]
```

10.88.1.3 IArrivalEvent() [2/2]

```
IArrivalEvent (
    const IArrivalEvent & ) [inline], [protected]
```

10.88.2 Member Function Documentation

10.88.2.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Implemented in [ArrivalEvent](#), [InterfaceEvent](#), and [IInterfaceEvent](#).

10.88.2.2 operator=()

```
IArrivalEvent& operator= (
    const IArrivalEvent & ) [protected]
```

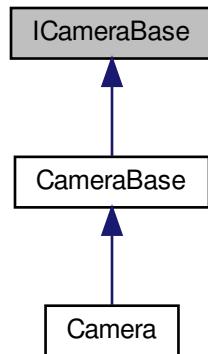
The documentation for this class was generated from the following file:

- include/Interface/IArrivalEvent.h

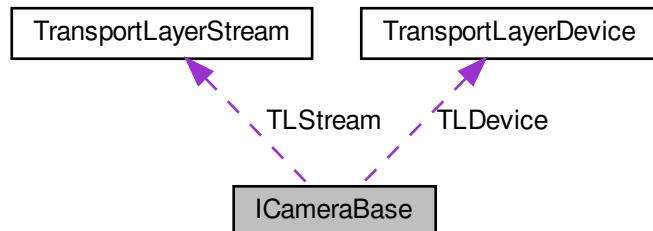
10.89 ICameraBase Class Reference

The interface file for base class for the camera object.

Inheritance diagram for ICameraBase:



Collaboration diagram for ICameraBase:



Public Member Functions

- virtual `~ICameraBase ()`
- virtual void `Init ()=0`
- virtual void `DeInit ()=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 [RegisterEvent](#) (Event &evtToRegister)=0
- virtual void [RegisterEvent](#) (Event &evtToRegister, const [GenICam::gcstring](#) &eventName)=0
- virtual void [UnregisterEvent](#) (Event &evtToUnregister)=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 [CameralInternal](#)
- class [InterfacelImpl](#)

10.89.1 Detailed Description

The interface file for base class for the camera object.

10.89.2 Constructor & Destructor Documentation

10.89.2.1 ~ICameraBase()

```
virtual ~ICameraBase (
    void ) [inline], [virtual]
```

10.89.2.2 ICameraBase() [1/2]

```
ICameraBase ( ) [inline], [protected]
```

10.89.2.3 ICameraBase() [2/2]

```
ICameraBase (
    const ICameraBase & ) [inline], [protected]
```

10.89.3 Member Function Documentation

10.89.3.1 BeginAcquisition()

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

Implemented in [CameraBase](#).

10.89.3.2 DeInit()

```
virtual void DeInit ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.3 DiscoverMaxPacketSize()

```
virtual unsigned int DiscoverMaxPacketSize ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.4 EndAcquisition()

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

Implemented in [CameraBase](#).

10.89.3.5 ForceIP()

```
virtual void ForceIP ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.6 GetAccessMode()

```
virtual GenApi::EAccessMode GetAccessMode ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.7 GetBufferOwnership()

```
virtual BufferOwnership GetBufferOwnership ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.8 GetGuiXml()

```
virtual GenICam::gcstring GetGuiXml ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.9 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT\_TIMEOUT\_INFINITE,
    uint64_t streamID = 0 ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.10 GetNodeMap()

```
virtual GenApi::INodeMap& GetNodeMap () const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.11 GetNumDataStreams()

```
virtual unsigned int GetNumDataStreams () [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.12 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse () [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.13 GetTLDeviceNodeMap()

```
virtual GenApi::INodeMap& GetTLDeviceNodeMap () const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.14 GetTLStreamNodeMap()

```
virtual GenApi::INodeMap& GetTLStreamNodeMap () const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.15 GetUniqueId()

```
virtual GenICam::gcstring GetUniqueId () [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.16 GetUserBufferCount()

```
virtual uint64_t GetUserBufferCount () const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.17 GetUserBufferSize()

```
virtual uint64_t GetUserBufferSize () const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.18 GetUserBufferTotalSize()

```
virtual uint64_t GetUserBufferTotalSize () const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.19 Init()

```
virtual void Init () [pure virtual]
```

Implemented in [CameraBase](#), and [Camera](#).

10.89.3.20 IsInitialized()

```
virtual bool IsInitialized () [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.21 IsStreaming()

```
virtual bool IsStreaming ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.22 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.23 operator=()

```
ICameraBase& operator= (
    const ICameraBase & ) [protected]
```

10.89.3.24 ReadPort()

```
virtual void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.25 RegisterEvent() [1/2]

```
virtual void RegisterEvent (
    Event & evtToRegister ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.26 RegisterEvent() [2/2]

```
virtual void RegisterEvent (
    Event & evtToRegister,
    const GenICam::gcstring & eventName ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.27 SetBufferOwnership()

```
virtual void SetBufferOwnership (
    const BufferOwnership mode ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.28 SetUserBuffers() [1/2]

```
virtual void SetUserBuffers (
    void *const pMemBuffers,
    uint64_t totalSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.29 SetUserBuffers() [2/2]

```
virtual void SetUserBuffers (
    void **const ppMemBuffers,
    const uint64_t bufferCount,
    const uint64_t bufferSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.30 UnregisterEvent()

```
virtual void UnregisterEvent (
    Event & evtToUnregister ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.3.31 WritePort()

```
virtual void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.89.4 Friends And Related Function Documentation

10.89.4.1 CameraInternal

```
friend class CameraInternal [friend]
```

10.89.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

10.89.5 Member Data Documentation

10.89.5.1 m_pCameraBaseData

```
CameraBaseData* m_pCameraBaseData [protected]
```

10.89.5.2 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.

10.89.5.3 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.

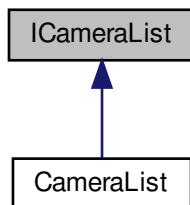
The documentation for this class was generated from the following file:

- include/Interface/[ICameraBase.h](#)

10.90 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 void `Clear ()=0`
- virtual void `RemoveBySerial (std::string serialNumber)=0`
- virtual void `RemoveByIndex (unsigned int index)=0`
- virtual void `Append (CameraList &otherList)=0`

Protected Member Functions

- `ICameraList ()`
- `ICameraList (const ICameraList &)`
- `ICameraList & operator= (const ICameraList &)`

Protected Attributes

- `CameraListData * m_pCameraListData`

Friends

- class `Interfacelmpl`
- class `CameraListImpl`

10.90.1 Detailed Description

Used to hold a list of camera objects.

10.90.2 Constructor & Destructor Documentation

10.90.2.1 ~ICameraList()

```
virtual ~ICameraList ( ) [inline], [virtual]
```

10.90.2.2 ICameraList() [1/2]

```
ICameraList ( ) [inline], [protected]
```

10.90.2.3 ICameraList() [2/2]

```
ICameraList (
    const ICameraList & ) [inline], [protected]
```

10.90.3 Member Function Documentation

10.90.3.1 Append()

```
virtual void Append (
    CameraList & otherList ) [pure virtual]
```

Implemented in [CameraList](#).

10.90.3.2 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [CameraList](#).

10.90.3.3 GetByIndex()

```
virtual CameraPtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [CameraList](#).

10.90.3.4 GetBySerial()

```
virtual CameraPtr GetBySerial (
    std::string serialNumber ) const [pure virtual]
```

Implemented in [CameraList](#).

10.90.3.5 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [CameraList](#).

10.90.3.6 operator=()

```
ICameraList& operator= (
    const ICameraList & ) [protected]
```

10.90.3.7 operator[]()

```
virtual CameraPtr operator[ ] (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

10.90.3.8 RemoveByIndex()

```
virtual void RemoveByIndex (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

10.90.3.9 RemoveBySerial()

```
virtual void RemoveBySerial (
    std::string serialNumber ) [pure virtual]
```

Implemented in [CameraList](#).

10.90.4 Friends And Related Function Documentation

10.90.4.1 CameraListImpl

```
friend class CameraListImpl [friend]
```

10.90.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

10.90.5 Member Data Documentation

10.90.5.1 m_pCameraListData

```
CameraListData* m_pCameraListData [protected]
```

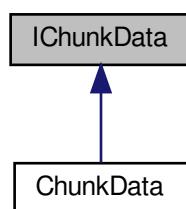
The documentation for this class was generated from the following file:

- include/Interface/[ICameraList.h](#)

10.91 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 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 GetInferenceResult () const =0
- virtual float64_t GetInferenceConfidence () const =0
- virtual InferenceBoundingBoxResult GetInferenceBoundingBoxResult () const =0

Protected Member Functions

- IChunkData ()

10.91.1 Detailed Description

The [Interface](#) file for [ChunkData](#).

10.91.2 Constructor & Destructor Documentation

10.91.2.1 ~IChunkData()

```
virtual ~IChunkData ( ) [inline], [virtual]
```

10.91.2.2 IChunkData()

```
IChunkData ( ) [inline], [protected]
```

10.91.3 Member Function Documentation

10.91.3.1 GetBlackLevel()

```
virtual float64_t GetBlackLevel ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.2 GetCounterValue()

```
virtual int64_t GetCounterValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.3 GetCRC()

```
virtual int64_t GetCRC ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.4 GetEncoderValue()

```
virtual int64_t GetEncoderValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.5 GetExposureEndLineStatusAll()

```
virtual int64_t GetExposureEndLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.6 GetExposureTime()

```
virtual float64_t GetExposureTime ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.7 GetFrameID()

```
virtual int64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.8 GetGain()

```
virtual float64_t GetGain ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.9 GetHeight()

```
virtual int64_t GetHeight ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.10 GetImage()

```
virtual int64_t GetImage ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.11 GetInferenceBoundingBoxResult()

```
virtual InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.12 GetInferenceConfidence()

```
virtual float64\_t GetInferenceConfidence ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.13 GetInferenceResult()

```
virtual int64_t GetInferenceResult ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.14 GetLinePitch()

```
virtual int64_t GetLinePitch ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.15 GetLineStatusAll()

```
virtual int64_t GetLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.16 GetOffsetX()

```
virtual int64_t GetOffsetX ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.17 GetOffsetY()

```
virtual int64_t GetOffsetY ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.18 GetPartSelector()

```
virtual int64_t GetPartSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.19 GetPixelDynamicRangeMax()

```
virtual int64_t GetPixelDynamicRangeMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.20 GetPixelDynamicRangeMin()

```
virtual int64_t GetPixelDynamicRangeMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.21 GetScan3dAxisMax()

```
virtual float64_t GetScan3dAxisMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.22 GetScan3dAxisMin()

```
virtual float64_t GetScan3dAxisMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.23 GetScan3dCoordinateOffset()

```
virtual float64_t GetScan3dCoordinateOffset ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.24 GetScan3dCoordinateReferenceValue()

```
virtual float64_t GetScan3dCoordinateReferenceValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.25 GetScan3dCoordinateScale()

```
virtual float64_t GetScan3dCoordinateScale ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.26 GetScan3dInvalidDataValue()

```
virtual float64_t GetScan3dInvalidDataValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.27 GetScan3dTransformValue()

```
virtual float64_t GetScan3dTransformValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.28 GetScanLineSelector()

```
virtual int64_t GetScanLineSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.29 GetSequencerSetActive()

```
virtual int64_t GetSequencerSetActive ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.30 GetSerialDataLength()

```
virtual int64_t GetSerialDataLength ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.31 GetStreamChannelID()

```
virtual int64_t GetStreamChannelID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.32 GetTimerValue()

```
virtual float64_t GetTimerValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.33 GetTimestamp()

```
virtual int64_t GetTimestamp ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.34 GetTimestampLatchValue()

```
virtual int64_t GetTimestampLatchValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.35 GetTransferBlockID()

```
virtual int64_t GetTransferBlockID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.36 GetTransferQueueCurrentBlockCount()

```
virtual int64_t GetTransferQueueCurrentBlockCount ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.37 GetWidth()

```
virtual int64_t GetWidth ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.91.3.38 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](#)

10.92 IDataStream Class Reference

Public Member Functions

- virtual ~IDataStream ()
- virtual StreamTypeEnum [GetStreamType \(\) const](#) =0
- virtual void [AnnounceImage \(size_t size\)](#)=0
- virtual void [AnnounceImage \(size_t size, void *pPrivate\)](#)=0
- virtual void [AnnounceImage \(size_t size, void *pData, void *pPrivate\)](#)=0
- virtual void [RevokeImages \(\)](#)=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 void * [GetBufferInfoPtrType \(GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD i↔ InfoCmd\)](#)=0
- virtual size_t [GetBufferInfoSizeType \(GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD i↔ InfoCmd\)](#)=0
- virtual uint64_t [GetBufferInfoUInt64Type \(GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD i↔ InfoCmd\)](#)=0
- virtual bool [GetBufferInfoBool8Type \(GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD i↔ InfoCmd\)](#)=0
- virtual void [RegisterImageEvent \(IImageEvent &imageEvent, EventPollingOptions pollingOption\)](#)=0
- virtual void [UnregisterImageEvent \(IImageEvent &imageEvent\)](#)=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 \(\)](#)

10.92.1 Constructor & Destructor Documentation

10.92.1.1 ~IDataStream()

```
virtual ~IDataStream( ) [inline], [virtual]
```

10.92.1.2 IDataStream()

```
IDataStream( ) [inline], [protected]
```

10.92.2 Member Function Documentation**10.92.2.1 AnnounceImage() [1/3]**

```
virtual void AnnounceImage (
    size_t size ) [pure virtual]
```

10.92.2.2 AnnounceImage() [2/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pPrivate ) [pure virtual]
```

10.92.2.3 AnnounceImage() [3/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pData,
    void * pPrivate ) [pure virtual]
```

10.92.2.4 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    GenApi::SingleChunkData_t * ChunkData,
    int64_t NumChunks ) [pure virtual]
```

10.92.2.5 CleanupChunkAdapter()

```
virtual void CleanupChunkAdapter ( ) [pure virtual]
```

10.92.2.6 FlushQueueAllDiscard()

```
virtual void FlushQueueAllDiscard ( ) [pure virtual]
```

10.92.2.7 GetBufferChunkData()

```
virtual GenTL::GC_ERROR GetBufferChunkData (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::SINGLE_CHUNK_DATA * pChunkData,
    size_t * piNumChunks ) [pure virtual]
```

10.92.2.8 GetBufferInfoBool8Type()

```
virtual bool GetBufferInfoBool8Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

10.92.2.9 GetBufferInfoPtrType()

```
virtual void* GetBufferInfoPtrType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

10.92.2.10 GetBufferInfoSizeType()

```
virtual size_t GetBufferInfoSizeType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

10.92.2.11 GetBufferInfoUInt64Type()

```
virtual uint64_t GetBufferInfoUInt64Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

10.92.2.12 GetDeviceNodeMap()

```
virtual GenApi::INodeMap* GetDeviceNodeMap ( ) const [pure virtual]
```

10.92.2.13 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout ) [pure virtual]
```

10.92.2.14 GetNextImageInternal()

```
virtual ImagePtr GetNextImageInternal (
    void ** ppPrivate,
    uint64_t grabTimeout ) [pure virtual]
```

10.92.2.15 GetNodeMap()

```
virtual GenApi::INodeMap& GetNodeMap ( ) const [pure virtual]
```

10.92.2.16 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) const [pure virtual]
```

10.92.2.17 GetPort()

```
virtual Port& GetPort ( ) const [pure virtual]
```

10.92.2.18 GetStreamType()

```
virtual StreamTypeEnum GetStreamType ( ) const [pure virtual]
```

10.92.2.19 InitChunkAdapter()

```
virtual void InitChunkAdapter (
    GenApi::INodeMap & nodemap ) [pure virtual]
```

10.92.2.20 IsCRCCheckEnabled()

```
virtual bool IsCRCCheckEnabled ( ) const [pure virtual]
```

10.92.2.21 IsImageInUse()

```
virtual bool IsImageInUse (
    const uint64_t imageID ) [pure virtual]
```

10.92.2.22 IsStreaming()

```
virtual bool IsStreaming ( ) [pure virtual]
```

10.92.2.23 KillBufferEvent()

```
virtual void KillBufferEvent ( ) [pure virtual]
```

10.92.2.24 RegisterImageEvent()

```
virtual void RegisterImageEvent (
    IImageEvent & imageEvent,
    EventPollingOptions pollingOption ) [pure virtual]
```

10.92.2.25 ReleaseImage()

```
virtual void ReleaseImage (
    const uint64_t imageID ) [pure virtual]
```

10.92.2.26 RevokeImages()

```
virtual void RevokeImages ( ) [pure virtual]
```

10.92.2.27 StartStream()

```
virtual void StartStream (
    const unsigned int stream_index = 0 ) [pure virtual]
```

10.92.2.28 StopStream()

```
virtual void StopStream ( ) [pure virtual]
```

10.92.2.29 TransportLayerStreamInfo()

```
virtual const TransportLayerStream& TransportLayerStreamInfo ( ) const [pure virtual]
```

10.92.2.30 UnregisterImageEvent()

```
virtual void UnregisterImageEvent (
    IImageEvent & imageEvent ) [pure virtual]
```

10.92.2.31 WaitOnImageEvent()

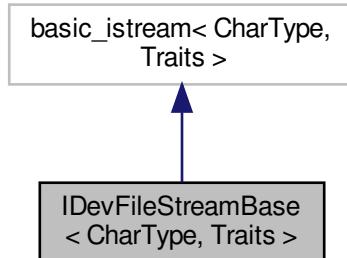
```
virtual void WaitOnImageEvent (
    uint64_t timeout ) [pure virtual]
```

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

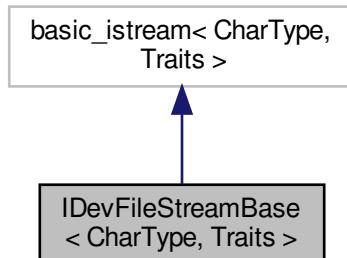
- include/Interface/IStream.h

10.93 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.

10.93.1 Member Typedef Documentation

10.93.1.1 filebuf_type

```
typedef IDevFileStreamBuf<CharType, Traits> filebuf_type
```

10.93.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

10.93.1.3 istream_type

```
typedef std::basic_istream<CharType, Traits> istream_type
```

10.93.2 Member Function Documentation

10.93.2.1 close()

```
void close () [inline]
```

Close the file on the device.

10.93.2.2 is_open()

```
bool is_open () const [inline]
```

10.93.2.3 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

10.93.2.4 rdbuf()

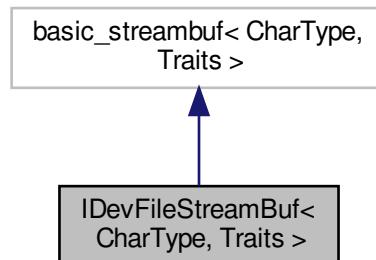
```
filebuf\_type* rdbuf ( ) const [inline]
```

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

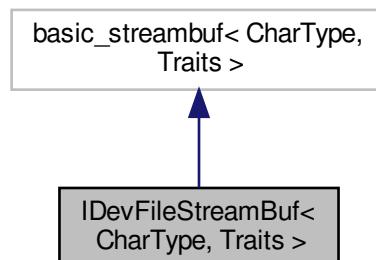
- include/SpinGenApi/[Filestream.h](#)

10.94 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)`

10.94.1 Constructor & Destructor Documentation

10.94.1.1 `IDevFileStreamBuf()`

```
IDevFileStreamBuf () [inline]
```

10.94.1.2 `~IDevFileStreamBuf()`

```
~IDevFileStreamBuf () [inline]
```

10.94.2 Member Function Documentation

10.94.2.1 `close()`

```
filebuf_type* close () [inline]
```

10.94.2.2 `is_open()`

```
bool is_open () const [inline]
```

10.94.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

10.94.2.4 pbackfail()

```
int_type pbackfail (
    int_type c ) [inline], [protected]
```

10.94.2.5 underflow()

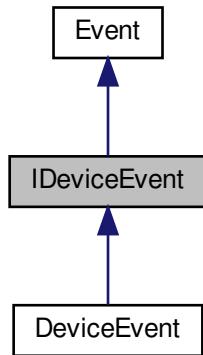
```
int_type underflow ( ) [inline], [protected]
```

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

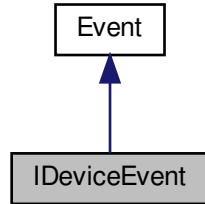
- include/SpinGenApi/Filestream.h

10.95 IDeviceEvent Class Reference

Inheritance diagram for IDeviceEvent:



Collaboration diagram for IDeviceEvent:



Public Member Functions

- virtual `~IDeviceEvent ()`
- virtual void `OnDeviceEvent (Spinnaker::GenICam::gcstring eventName)=0`
- virtual uint64_t `GetDeviceEventId () const =0`
- virtual `GenICam::gcstring GetDeviceEventName () const =0`

Protected Member Functions

- `IDeviceEvent ()`
- `IDeviceEvent (const IDeviceEvent &)`
- `IDeviceEvent & operator= (const IDeviceEvent &)`

Additional Inherited Members

10.95.1 Constructor & Destructor Documentation

10.95.1.1 `~IDeviceEvent()`

```
virtual ~IDeviceEvent ( ) [inline], [virtual]
```

10.95.1.2 `IDeviceEvent()` [1/2]

```
IDeviceEvent ( ) [inline], [protected]
```

10.95.1.3 IDeviceEvent() [2/2]

```
DeviceEvent (
    const DeviceEvent & ) [inline], [protected]
```

10.95.2 Member Function Documentation

10.95.2.1 GetDeviceEventId()

```
virtual uint64_t GetDeviceEventId () const [pure virtual]
```

Implemented in [DeviceEvent](#).

10.95.2.2 GetDeviceEventName()

```
virtual GenICam::gcstring GetDeviceEventName () const [pure virtual]
```

Implemented in [DeviceEvent](#).

10.95.2.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (
    Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Implemented in [DeviceEvent](#).

10.95.2.4 operator=()

```
DeviceEvent& operator= (
    const DeviceEvent & ) [protected]
```

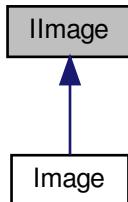
The documentation for this class was generated from the following file:

- include/Interface/IDeviceEvent.h

10.96 IImage Class Reference

The interface file for [Image](#).

Inheritance diagram for IImage:



Public Member Functions

- virtual [`~IImage \(\)`](#)
- 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`](#)
- [`DEPRECATED_FUNC \("This function has been deprecated. Please use more specific functions defined in the " "ImageUtilityPolarization class.", virtual ImagePtr ExtractPolarization\(const PolarizationAlgorithm polarizationAlogrithm, const PolarizationResolution resolution\) 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 [`Release \(\)=0`](#)
- virtual uint64_t [`GetID \(\) const =0`](#)
- virtual void * [`GetData \(\) const =0`](#)
- virtual void * [`GetPrivateData \(\) const =0`](#)
- virtual double [`GetDataAbsoluteMax \(\) const =0`](#)
- virtual double [`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 [`YPadding \(\) 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 [GenICam::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](#))=0
- virtual void [Save](#) (const char *pFilename, [PNGOption](#) &pOption)=0
- virtual void [Save](#) (const char *pFilename, [PPMOption](#) &pOption)=0
- virtual void [Save](#) (const char *pFilename, [PGMOption](#) &pOption)=0
- virtual void [Save](#) (const char *pFilename, [TIFFOption](#) &pOption)=0
- virtual void [Save](#) (const char *pFilename, [JPEGOption](#) &pOption)=0
- virtual void [Save](#) (const char *pFilename, [JPG2Option](#) &pOption)=0
- virtual void [Save](#) (const char *pFilename, [BMPOption](#) &pOption)=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
- **DEPRECATED_FUNC** ("This function has been deprecated. Polarization images created through the [ImageUtilityPolarization](#) class " "now use an appropriate pixel format to hold the raw polarization values.", virtual float *GetPolarizationValues() const)=0
- **DEPRECATED_FUNC** ("This function has been deprecated. Polarization algorithms are applied through specific " "functions defined in the [ImageUtilityPolarization](#) class.", virtual [PolarizationAlgorithm](#) GetPolarizationAlgorithm() const)=0

Protected Member Functions

- [IImage](#) ()
- virtual [ImageData](#) * [GetImageData](#) () const =0

Friends

- class [Stream](#)

10.96.1 Detailed Description

The interface file for [Image](#).

10.96.2 Constructor & Destructor Documentation

10.96.2.1 ~IlImage()

```
virtual ~IlImage( ) [inline], [virtual]
```

10.96.2.2 IlImage()

```
IlImage( ) [inline], [protected]
```

10.96.3 Member Function Documentation

10.96.3.1 CalculateStatistics()

```
virtual void CalculateStatistics(   
    ImageStatistics & pStatistics ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.2 CheckCRC()

```
virtual bool CheckCRC( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.3 Convert() [1/2]

```
virtual ImagePtr Convert(   
    PixelFormatEnums format,   
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.4 Convert() [2/2]

```
virtual void Convert(   
    ImagePtr destinationImage,   
    PixelFormatEnums format,   
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.5 DeepCopy()

```
virtual void DeepCopy (
    const ImagePtr pSrcImage ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.6 DEPRECATED_FUNC() [1/3]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use more specific functions defined
in the \" ImageUtilityPolarization class." ,
    virtual ImagePtr ExtractPolarization(const PolarizationAlgorithm polarization<-
Algorithm, const PolarizationResolution resolution) const ) [pure virtual]
```

10.96.3.7 DEPRECATED_FUNC() [2/3]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Polarization images created through the
ImageUtilityPolarization class " "now use an appropriate pixel format to hold the raw polarization
values." ,
    virtual float *GetPolarizationValues() const ) [pure virtual]
```

10.96.3.8 DEPRECATED_FUNC() [3/3]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Polarization algorithms are applied through
specific " "functions defined in the ImageUtilityPolarization class." ,
    virtual PolarizationAlgorithm GetPolarizationAlgorithm() const ) [pure virtual]
```

10.96.3.9 GetBitsPerPixel()

```
virtual size_t GetBitsPerPixel () const [pure virtual]
```

Implemented in [Image](#).

10.96.3.10 GetBufferSize()

```
virtual size_t GetBufferSize ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.11 GetChunkData()

```
virtual const ChunkData& GetChunkData ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.12 GetChunkLayoutId()

```
virtual uint64_t GetChunkLayoutId ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.13 GetColorProcessing()

```
virtual ColorProcessingAlgorithm GetColorProcessing ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.14 GetData()

```
virtual void* GetData ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.15 GetDataAbsoluteMax()

```
virtual double GetDataAbsoluteMax ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.16 GetDataAbsoluteMin()

```
virtual double GetDataAbsoluteMin ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.17 GetFrameID()

```
virtual uint64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.18 GetHeight()

```
virtual size_t GetHeight ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.19 GetID()

```
virtual uint64_t GetID ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.20 GetImageData()

```
virtual ImageData* GetImageData ( ) const [protected], [pure virtual]
```

Implemented in [Image](#).

10.96.3.21 GetImageSize()

```
virtual size_t GetImageSize ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.22 GetImageStatus()

```
virtual ImageStatus GetImageStatus ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.23 GetNumChannels()

```
virtual size_t GetNumChannels ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.24 GetPayloadType()

```
virtual size_t GetPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.25 GetPixelFormat()

```
virtual PixelFormatEnums GetPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.26 GetPixelFormatIntType()

```
virtual PixelFormatIntType GetPixelFormatIntType ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.27 GetPixelFormatName()

```
virtual GenICam::gcstring GetPixelFormatName ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.28 GetPrivateData()

```
virtual void* GetPrivateData ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.29 GetStride()

```
virtual size_t GetStride ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.30 GetTimeStamp()

```
virtual uint64_t GetTimeStamp ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.31 GetTLPayloadType()

```
virtual PayloadTypeIDs GetTLPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.32 GetTLPixelFormat()

```
virtual uint64_t GetTLPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.33 GetTLPixelFormatNamespace()

```
virtual PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.34 GetValidPayloadSize()

```
virtual size_t GetValidPayloadSize ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.35 GetWidth()

```
virtual size_t GetWidth ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.36 GetXOffset()

```
virtual size_t GetXOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.37 GetXPadding()

```
virtual size_t GetXPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.38 GetYOffset()

```
virtual size_t GetYOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.39 GetYPadding()

```
virtual size_t GetYPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.40 HasCRC()

```
virtual bool HasCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.41 IsIncomplete()

```
virtual bool IsIncomplete ( ) const [pure virtual]
```

Implemented in [Image](#).

10.96.3.42 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.43 Release()

```
virtual void Release ( ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.44 ResetImage() [1/2]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.45 ResetImage() [2/2]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.46 Save() [1/8]

```
virtual void Save (
    const char * pFilename,
    ImageFileFormat format = FROM_FILE_EXT ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.47 Save() [2/8]

```
virtual void Save (
    const char * pFilename,
    PNGOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.48 Save() [3/8]

```
virtual void Save (
    const char * pFilename,
    PPMOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.49 Save() [4/8]

```
virtual void Save (
    const char * pFilename,
    PGMOOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.50 Save() [5/8]

```
virtual void Save (
    const char * pFilename,
    TIFFOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.51 Save() [6/8]

```
virtual void Save (
    const char * pFilename,
    JPEGOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.52 Save() [7/8]

```
virtual void Save (
    const char * pFilename,
    JPG2Option & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.96.3.53 Save() [8/8]

```
virtual void Save (
    const char * pFilename,
    BMPOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.96.4 Friends And Related Function Documentation

10.96.4.1 Stream

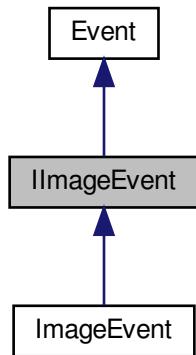
```
friend class Stream [friend]
```

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

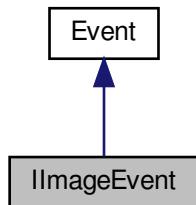
- include/Interface/IImage.h

10.97 IImageEvent Class Reference

Inheritance diagram for IImageEvent:



Collaboration diagram for IImageEvent:



Public Member Functions

- virtual ~IImageEvent ()
- virtual void OnImageEvent (ImagePtr image)=0

Protected Member Functions

- IImageEvent ()
- IImageEvent (const IImageEvent &)
- IImageEvent & operator= (const IImageEvent &)

Additional Inherited Members

10.97.1 Constructor & Destructor Documentation

10.97.1.1 ~IImageEvent()

```
virtual ~IImageEvent ( ) [inline], [virtual]
```

10.97.1.2 IImageEvent() [1/2]

```
IImageEvent ( ) [inline], [protected]
```

10.97.1.3 IImageEvent() [2/2]

```
IImageEvent (
    const IImageEvent & ) [inline], [protected]
```

10.97.2 Member Function Documentation

10.97.2.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

Implemented in [ImageEvent](#).

10.97.2.2 operator=()

```
IImageEvent& operator= (
    const IImageEvent & ) [protected]
```

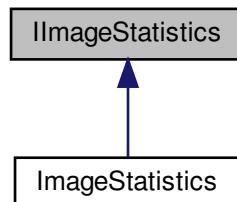
The documentation for this class was generated from the following file:

- include/Interface/IImageEvent.h

10.98 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 \(const IImageStatistics &\)](#)

10.98.1 Detailed Description

The interface file for image statistics.

10.98.2 Constructor & Destructor Documentation

10.98.2.1 ~IImageStatistics()

```
virtual ~IImageStatistics ( ) [inline], [virtual]
```

10.98.2.2 IImageStatistics() [1/2]

```
IImageStatistics ( ) [inline], [protected]
```

10.98.2.3 IImageStatistics() [2/2]

```
IImageStatistics (   
     const IImageStatistics & ) [inline], [protected]
```

10.98.3 Member Function Documentation

10.98.3.1 DisableAll()

```
virtual void DisableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.2 EnableAll()

```
virtual void EnableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
    StatisticsChannel channel,
    bool * pEnabled ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.7 GetHistogram()

```
virtual void GetHistogram (
    StatisticsChannel channel,
    int ** ppHistogram ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.8 GetMean()

```
virtual void GetMean (
    StatisticsChannel channel,
    float * pPixelValueMean ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
    StatisticsChannel channel,
    unsigned int * pNumPixelValues ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.10 GetPixelValueRange()

```
virtual void GetPixelValueRange (
    StatisticsChannel channel,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.11 GetRange()

```
virtual void GetRange (
    StatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.98.3.12 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](#).

10.98.3.13 SetChannelStatus()

```
virtual void SetChannelStatus (
    StatisticsChannel channel,
    bool enabled ) [pure virtual]
```

Implemented in [ImageStatistics](#).

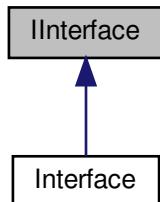
The documentation for this class was generated from the following file:

- include/Interface/[IImageStatistics.h](#)

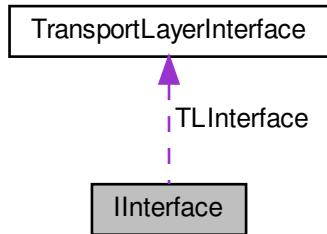
10.99 IInterface Class Reference

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 `RegisterEvent (Event &evtToRegister)=0`
- virtual void `UnregisterEvent (Event &evtToUnregister)=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 `SystemImpl`

10.99.1 Detailed Description

The interface file for `Interface`.

10.99.2 Constructor & Destructor Documentation

10.99.2.1 ~IInterface()

```
virtual ~IInterface ( ) [inline], [virtual]
```

10.99.2.2 [IInterface\(\)](#) [1/2]

```
IInterface () [inline], [protected]
```

10.99.2.3 [IInterface\(\)](#) [2/2]

```
IInterface (
    const IInterface & ) [inline], [protected]
```

10.99.3 Member Function Documentation**10.99.3.1 [GetCameras\(\)](#)**

```
virtual CameraList GetCameras (
    bool updateCameras = true ) const [pure virtual]
```

Implemented in [Interface](#).

10.99.3.2 [GetTLNodeMap\(\)](#)

```
virtual GenApi::INodeMap& GetTLNodeMap () const [pure virtual]
```

Implemented in [Interface](#).

10.99.3.3 [IsInUse\(\)](#)

```
virtual bool IsInUse () const [pure virtual]
```

Implemented in [Interface](#).

10.99.3.4 [IsValid\(\)](#)

```
virtual bool IsValid () [pure virtual]
```

Implemented in [Interface](#).

10.99.3.5 operator=()

```
IInterface& operator= (
    const IInterface & ) [protected]
```

10.99.3.6 RegisterEvent()

```
virtual void RegisterEvent (
    Event & evtToRegister ) [pure virtual]
```

Implemented in [Interface](#).

10.99.3.7 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](#).

10.99.3.8 UnregisterEvent()

```
virtual void UnregisterEvent (
    Event & evtToUnregister ) [pure virtual]
```

Implemented in [Interface](#).

10.99.3.9 UpdateCameras()

```
virtual bool UpdateCameras ( ) [pure virtual]
```

Implemented in [Interface](#).

10.99.4 Friends And Related Function Documentation

10.99.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

10.99.4.2 SystemImpl

```
friend class SystemImpl [friend]
```

10.99.5 Member Data Documentation

10.99.5.1 m_pInterfaceData

```
InterfaceData* m_pInterfaceData [protected]
```

10.99.5.2 TLInterface

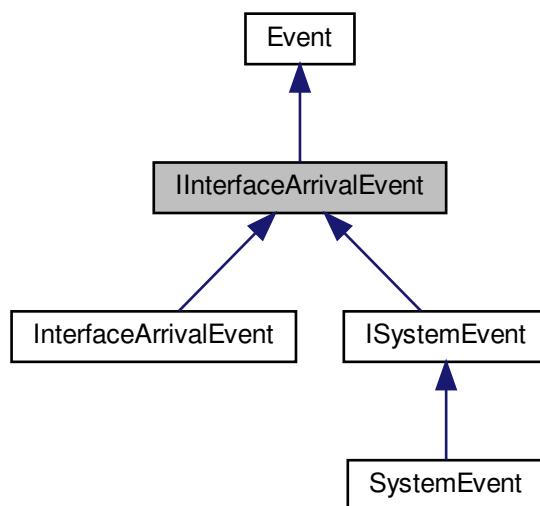
`TransportLayerInterface` `TLInterface`

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

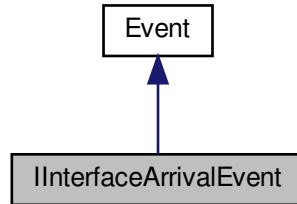
- include/Interface/IInterface.h

10.100 IInterfaceArrivalEvent Class Reference

Inheritance diagram for IInterfaceArrivalEvent:



Collaboration diagram for IIInterfaceArrivalEvent:



Public Member Functions

- virtual `~IIInterfaceArrivalEvent ()`
- virtual void `OnInterfaceArrival (std::string interfaceID)=0`

Protected Member Functions

- `IIInterfaceArrivalEvent ()`
- `IIInterfaceArrivalEvent (const IIInterfaceArrivalEvent &)`
- `IIInterfaceArrivalEvent & operator= (const IIInterfaceArrivalEvent &)`

Additional Inherited Members

10.100.1 Constructor & Destructor Documentation

10.100.1.1 `~IIInterfaceArrivalEvent()`

```
virtual ~IIInterfaceArrivalEvent ( ) [inline], [virtual]
```

10.100.1.2 `IIInterfaceArrivalEvent()` [1/2]

```
IIInterfaceArrivalEvent ( ) [inline], [protected]
```

10.100.1.3 `IInterfaceArrivalEvent()` [2/2]

```
IInterfaceArrivalEvent (
    const IInterfaceArrivalEvent & ) [inline], [protected]
```

10.100.2 Member Function Documentation

10.100.2.1 `OnInterfaceArrival()`

```
virtual void OnInterfaceArrival (
    std::string interfaceID ) [pure virtual]
```

Implemented in [InterfaceArrivalEvent](#), [SystemEvent](#), and [ISystemEvent](#).

10.100.2.2 `operator=()`

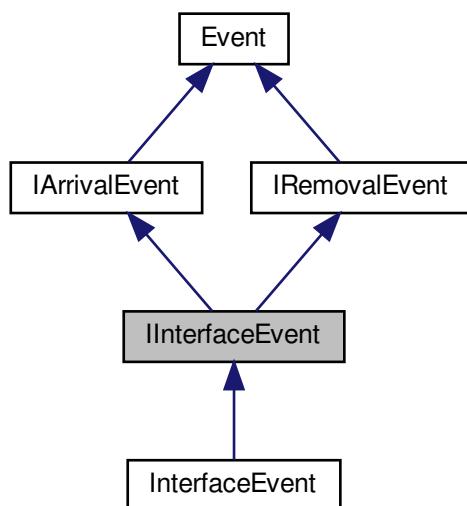
```
IInterfaceArrivalEvent& operator= (
    const IInterfaceArrivalEvent & ) [protected]
```

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

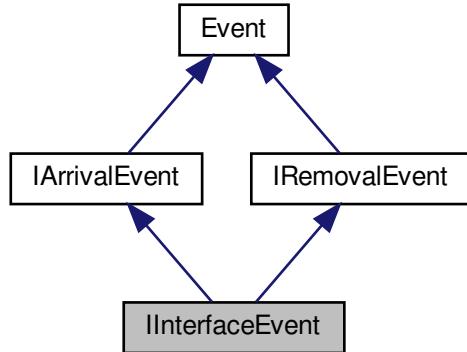
- [include/Interface/IInterfaceArrivalEvent.h](#)

10.101 IInterfaceEvent Class Reference

Inheritance diagram for IInterfaceEvent:



Collaboration diagram for IIInterfaceEvent:



Public Member Functions

- virtual `~IIInterfaceEvent ()`
- virtual void `OnDeviceArrival (uint64_t serialNumber)=0`
- virtual void `OnDeviceRemoval (uint64_t serialNumber)=0`

Protected Member Functions

- `IIInterfaceEvent ()`
- `IIInterfaceEvent (const IIInterfaceEvent &)`
- `IIInterfaceEvent & operator= (const IIInterfaceEvent &)`

Additional Inherited Members

10.101.1 Constructor & Destructor Documentation

10.101.1.1 `~IIInterfaceEvent()`

```
virtual ~IIInterfaceEvent ( ) [inline], [virtual]
```

10.101.1.2 `IIInterfaceEvent()` [1/2]

```
IIInterfaceEvent ( ) [inline], [protected]
```

10.101.1.3 [IInterfaceEvent\(\)](#) [2/2]

```
IInterfaceEvent (
    const IInterfaceEvent & ) [inline], [protected]
```

10.101.2 Member Function Documentation**10.101.2.1 [OnDeviceArrival\(\)](#)**

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IArrivalEvent](#).

Implemented in [InterfaceEvent](#).

10.101.2.2 [OnDeviceRemoval\(\)](#)

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IRemovalEvent](#).

Implemented in [InterfaceEvent](#).

10.101.2.3 [operator=\(\)](#)

```
IInterfaceEvent& operator= (
    const IInterfaceEvent & ) [protected]
```

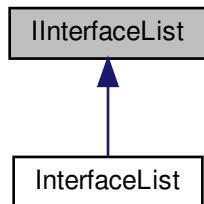
The documentation for this class was generated from the following file:

- include/Interface/IInterfaceEvent.h

10.102 IIInterfaceList Class Reference

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

Inheritance diagram for IIInterfaceList:



Public Member Functions

- virtual [~IIInterfaceList](#) (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

Protected Member Functions

- [IIInterfaceList](#) (void)
- [IIInterfaceList](#) (const [IIInterfaceList](#) &)
- [IIInterfaceList](#) & [operator=](#) (const [IIInterfaceList](#) &)

Protected Attributes

- [InterfaceListData](#) * [m_pInterfaceListData](#)

10.102.1 Detailed Description

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

10.102.2 Constructor & Destructor Documentation

10.102.2.1 ~IInterfaceList()

```
virtual ~IInterfaceList (
    void ) [inline], [virtual]
```

10.102.2.2 IInterfaceList() [1/2]

```
IInterfaceList (
    void ) [inline], [protected]
```

10.102.2.3 IInterfaceList() [2/2]

```
IInterfaceList (
    const IInterfaceList & ) [inline], [protected]
```

10.102.3 Member Function Documentation**10.102.3.1 Clear()**

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [InterfaceList](#).

10.102.3.2 GetByIndex()

```
virtual InterfacePtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [InterfaceList](#).

10.102.3.3 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [InterfaceList](#).

10.102.3.4 operator=()

```
IIInterfaceList& operator= (
    const IIInterfaceList & ) [protected]
```

10.102.3.5 operator[]()

```
virtual InterfacePtr operator[ ] (
    unsigned int index ) [pure virtual]
```

Implemented in [InterfaceList](#).

10.102.4 Member Data Documentation

10.102.4.1 m_pInterfaceListData

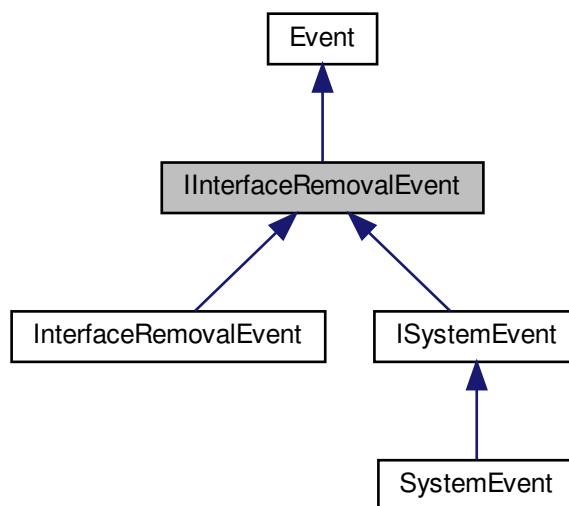
```
InterfaceListData* m_pInterfaceListData [protected]
```

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

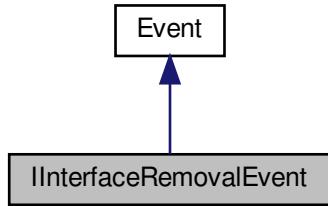
- include/Interface/IIInterfaceList.h

10.103 IIInterfaceRemovalEvent Class Reference

Inheritance diagram for IIInterfaceRemovalEvent:



Collaboration diagram for IIInterfaceRemovalEvent:



Public Member Functions

- virtual ~IIInterfaceRemovalEvent ()
- virtual void OnInterfaceRemoval (std::string interfaceID)=0

Protected Member Functions

- IIInterfaceRemovalEvent ()
- IIInterfaceRemovalEvent (const IIInterfaceRemovalEvent &)
- IIInterfaceRemovalEvent & operator= (const IIInterfaceRemovalEvent &)

Additional Inherited Members

10.103.1 Constructor & Destructor Documentation

10.103.1.1 ~IIInterfaceRemovalEvent()

```
virtual ~IIInterfaceRemovalEvent ( ) [inline], [virtual]
```

10.103.1.2 IIInterfaceRemovalEvent() [1/2]

```
IIInterfaceRemovalEvent ( ) [inline], [protected]
```

10.103.1.3 IIInterfaceRemovalEvent() [2/2]

```
IIInterfaceRemovalEvent (
    const IIInterfaceRemovalEvent & ) [inline], [protected]
```

10.103.2 Member Function Documentation

10.103.2.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

Implemented in [SystemEvent](#), [InterfaceRemovalEvent](#), and [ISystemEvent](#).

10.103.2.2 operator=(*)

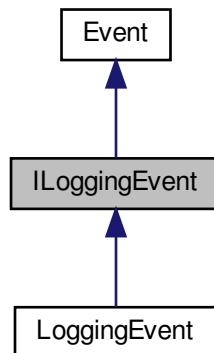
```
IIInterfaceRemovalEvent& operator= (
    const IIInterfaceRemovalEvent & ) [protected]
```

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

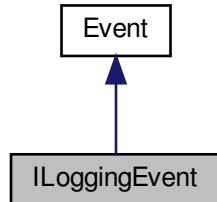
- include/Interface/IIInterfaceRemovalEvent.h

10.104 ILoggingEvent Class Reference

Inheritance diagram for ILoggingEvent:



Collaboration diagram for ILoggingEvent:



Public Member Functions

- virtual [~ILoggingEvent \(\)](#)
- virtual void [OnLogEvent \(LoggingEventDataPtr eventPtr\)=0](#)

Protected Member Functions

- [ILoggingEvent \(\)](#)
- [ILoggingEvent \(const ILoggingEvent &\)](#)
- [ILoggingEvent & operator= \(const ILoggingEvent &\)](#)

Additional Inherited Members

10.104.1 Constructor & Destructor Documentation

10.104.1.1 [~ILoggingEvent\(\)](#)

```
virtual ~ILoggingEvent ( ) [inline], [virtual]
```

10.104.1.2 [ILoggingEvent\(\)](#) [1/2]

```
ILoggingEvent ( ) [inline], [protected]
```

10.104.1.3 ILoggingEvent() [2/2]

```
ILoggingEvent (
    const ILoggingEvent & ) [inline], [protected]
```

10.104.2 Member Function Documentation

10.104.2.1 OnLogEvent()

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

Implemented in [LoggingEvent](#).

10.104.2.2 operator=()

```
ILoggingEvent& operator= (
    const ILoggingEvent & ) [protected]
```

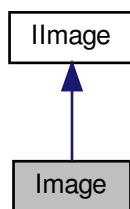
The documentation for this class was generated from the following file:

- [include/Interface/ILoggingEvent.h](#)

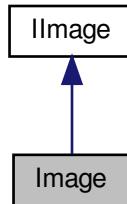
10.105 Image Class Reference

The image object class.

Inheritance diagram for Image:



Collaboration diagram for Image:



Public Member Functions

- virtual `~Image ()`
Virtual destructor.
- `DEPRECATED_FUNC` ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static `ImagePtr CreateHeatMap(const ImagePtr &srcImage)`)
Computes a heatmap image.
- `DEPRECATED_FUNC` ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static void `CreateHeatMap(const ImagePtr &srcImage, ImagePtr &destImage)`)
Computes a heatmap image.
- `DEPRECATED_FUNC` ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", 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`.
- `DEPRECATED_FUNC` ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static void `GetHeatMapColorGradient(HeatMapColor ¤tLowColor, HeatMapColor ¤tHighColor)`)
Returns the current heatmap gradient color range.
- `DEPRECATED_FUNC` ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", 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.
- `DEPRECATED_FUNC` ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static void `GetHeatMapRange(unsigned int ¤tLowValue, unsigned int ¤tHighValue)`)
Returns the current high and low values used in heatmap representations.
- `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 [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.
- double [GetDataAbsoluteMax](#) () const
Get the value for which no image data will exceed.
- double [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 [YPadding](#) () 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)**
Saves the image to the specified file name with the file format specified.
- **void Save (const char *pFilename, PNGOption &pOption)**
Saves the image to the specified file name with the options specified.
- **void Save (const char *pFilename, PPMOption &pOption)**
Saves the image to the specified file name with the options specified.
- **void Save (const char *pFilename, PGMOption &pOption)**
Saves the image to the specified file name with the options specified.
- **void Save (const char *pFilename, TIFFOption &pOption)**
Saves the image to the specified file name with the options specified.
- **void Save (const char *pFilename, JPEGOption &pOption)**
Saves the image to the specified file name with the options specified.
- **void Save (const char *pFilename, JPG2Option &pOption)**
Saves the image to the specified file name with the options specified.
- **void Save (const char *pFilename, BMPOption &pOption)**
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()
- **ImagePtr ExtractPolarization (const PolarizationAlgorithm polarizationAlgorithm, const PolarizationResolution resolution) const**
Extracts an image from a monochrome-polarized sensor.
- **float * GetPolarizationValues () const**
Returns the polarization values associated with an extracted polarization image.
- **PolarizationAlgorithm GetPolarizationAlgorithm () const**
Returns the polarization algorithm used to extract a polarization image.
- **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 \(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void *pData\)](#)
Create an image object with the specified parameters.
- static void [SetColorProcessing \(ColorProcessingAlgorithm colorAlgorithm\)](#)
Sets the default color processing algorithm.
- static [ColorProcessingAlgorithm GetColorProcessing \(\)](#)
Gets the default color processing algorithm.
- 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 \(size_t width, size_t height, size_t offsetX, size_t offsetY, PixelFormatEnums pixelFormat, void *pData\)](#)
- [ImagePtr CreateShared \(\) const](#)
- void [DeepCopy \(const Image &pSrcImage\)](#)
- void [Convert \(PixelFormatEnums format, Image &pDestImage, ColorProcessingAlgorithm colorAlgorithm=DEFAULT\) const](#)

Friends

- class [IDataStream](#)
- class [Stream](#)
- class [ImageConverter](#)
- class [ImageFiler](#)
- class [ImageStatsCalculator](#)
- class [ImageUtilityImpl](#)
- class [ImageUtilityPolarizationImpl](#)

10.105.1 Detailed Description

The image object class.

10.105.2 Constructor & Destructor Documentation

10.105.2.1 ~Image()

```
virtual ~Image ( ) [virtual]
```

Virtual destructor.

10.105.2.2 Image() [1/3]

```
Image ( ) [protected]
```

10.105.2.3 Image() [2/3]

```
Image (
    const ImagePtr image ) [protected]
```

10.105.2.4 Image() [3/3]

```
Image (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData ) [protected]
```

10.105.3 Member Function Documentation**10.105.3.1 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](#).

10.105.3.2 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](#).

10.105.3.3 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.

See also

[PixelFormatEnums](#)

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](#).

10.105.3.4 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.

See also

[Create\(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void* pData\)](#)

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](#).

10.105.3.5 Convert() [3/3]

```
void Convert (
    PixelFormatEnums format,
    Image & pDestImage,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [protected]
```

10.105.3.6 Create() [1/3]

```
static ImagePtr Create () [static]
```

Create an image object.

10.105.3.7 Create() [2/3]

```
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
--------------	-------------------------

10.105.3.8 Create() [3/3]

```
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

10.105.3.9 CreateShared()

```
ImagePtr CreateShared ( ) const [protected]
```

10.105.3.10 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](#).

10.105.3.11 DeepCopy() [2/2]

```
void DeepCopy (
    const Image & pSrcImage ) [protected]
```

10.105.3.12 DEPRECATED_FUNC() [1/6]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
ImageUtilityHeatmap class." ,
    static ImagePtr CreateHeatMapconst ImagePtr &srcImage )
```

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

10.105.3.13 DEPRECATED_FUNC() [2/6]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
ImageUtilityHeatmap class." ,
    static void CreateHeatMapconst ImagePtr &srcImage, ImagePtr &destImage )
```

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\(\)](#)

10.105.3.14 DEPRECATED_FUNC() [3/6]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
ImageUtilityHeatmap class." ,
    static void SetHeatMapColorGradientconst HeatMapColor newLowColor, const Heat←
MapColor newHighColor )
```

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.

10.105.3.15 DEPRECATED_FUNC() [4/6]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
ImageUtilityHeatmap class." ,
    static void GetHeatMapColorGradientHeatMapColor &currentLowColor, HeatMapColor
&currentHighColor )
```

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\(\)](#)

10.105.3.16 DEPRECATED_FUNC() [5/6]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
ImageUtilityHeatmap class." ,
    static void SetHeatMapRangeconst 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.

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.

10.105.3.17 DEPRECATED_FUNC() [6/6]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
ImageUtilityHeatmap class." ,
    static void GetHeatMapRangeunsigned int &currentLowValue, unsigned int &current←
HighValue )
```

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\(\)](#)

10.105.3.18 ExtractPolarization()

```
ImagePtr ExtractPolarization (
    const PolarizationAlgorithm polarizationAlgorithm,
    const PolarizationResolution resolution ) const
```

Extracts an image from a monochrome-polarized sensor.

The extracted image will be returned as Mono8 or BGRA8 for heatmap images.

Parameters

<i>polarizationAlgorithm</i>	Desired polarization algorithm to use.
<i>resolution</i>	Desired resolution of output image.

Returns

The converted image.

10.105.3.19 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](#).

10.105.3.20 GetBufferSize()

```
size_t GetBufferSize ( ) const [virtual]
```

Gets the size of the buffer associated with the image in bytes.

Returns

The size of the buffer, in bytes.

Implements [IImage](#).

10.105.3.21 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](#).

10.105.3.22 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](#).

10.105.3.23 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](#).

10.105.3.24 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](#).

10.105.3.25 GetDataAbsoluteMax()

```
double GetDataAbsoluteMax ( ) const [virtual]
```

Get the value for which no image data will exceed.

Returns

the maximum theoretical image data value

Implements [IImage](#).

10.105.3.26 GetDataAbsoluteMin()

```
double GetDataAbsoluteMin ( ) const [virtual]
```

Get the value for which no image data will be less than.

Returns

the minimum theoretical image data value

Implements [IImage](#).

10.105.3.27 GetDefaultColorProcessing()

```
static ColorProcessingAlgorithm GetDefaultColorProcessing ( ) [static]
```

Gets the default color processing algorithm.

See also

[SetDefaultColorProcessing\(\)](#)

Returns

The default color processing algorithm.

10.105.3.28 GetFrameID()

```
uint64_t GetFrameID ( ) const [virtual]
```

Gets the frame ID for this image.

Returns

The frame ID.

Implements [IImage](#).

10.105.3.29 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](#).

10.105.3.30 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](#).

10.105.3.31 GetImageData()

```
ImageData* GetImageData ( ) const [protected], [virtual]
```

Implements [IImage](#).

10.105.3.32 GetImageSize()

```
size_t GetImageSize ( ) const [virtual]
```

Returns the size of the image.

Returns

The image size in bytes.

Implements [IImage](#).

10.105.3.33 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](#).

10.105.3.34 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.

10.105.3.35 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](#).

10.105.3.36 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](#).

10.105.3.37 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](#).

10.105.3.38 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](#).

10.105.3.39 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](#).

10.105.3.40 GetPolarizationAlgorithm()

```
PolarizationAlgorithm GetPolarizationAlgorithm ( ) const
```

Returns the polarization algorithm used to extract a polarization image.

Returns

The polarization algorithm used to extract the polarization image.

10.105.3.41 GetPolarizationValues()

```
float* GetPolarizationValues ( ) const
```

Returns the polarization values associated with an extracted polarization image.

Note that standard quadrants (QUADRANT_I0_GRAYSCALE - QUADRANT_I135_GRAYSCALE) do not provide polarization values.

Returns

The polarization values associated with a polarization image.

10.105.3.42 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](#).

10.105.3.43 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 [IImage](#).

10.105.3.44 GetTimeStamp()

```
uint64_t GetTimeStamp ( ) const [virtual]
```

Gets the time stamp for the image in nanoseconds.

Returns

The time stamp of the image.

Implements [IImage](#).

10.105.3.45 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 [IImage](#).

10.105.3.46 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 [IImage](#).

10.105.3.47 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 [IImage](#).

10.105.3.48 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. [GetBufferSize\(\)](#) returns the total size of bytes allocated for the image.

See also

[GetBufferSize\(\)](#)

Returns

`size_t` value representing valid payload.

Implements [IImage](#).

10.105.3.49 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](#).

10.105.3.50 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](#).

10.105.3.51 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 [IImage](#).

10.105.3.52 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 [IImage](#).

10.105.3.53 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 [IImage](#).

10.105.3.54 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](#).

10.105.3.55 IsCompressed()

```
bool IsCompressed () const
```

Returns a boolean value indicating whether this image is compressed.

Returns

Returns true if image is compressed, false otherwise.

10.105.3.56 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](#).

10.105.3.57 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](#).

10.105.3.58 Release()

```
void Release ( ) [virtual]
```

Implements [IImage](#).

10.105.3.59 ResetImage() [1/2]

```
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](#).

10.105.3.60 ResetImage() [2/2]

```
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 [IImage](#).

10.105.3.61 Save() [1/8]

```
void Save (
    const char * pFilename,
    ImageFileFormat format = FROM_FILE_EXT ) [virtual]
```

Saves the image to the specified file name with the file format specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>format</i>	File format to save in.

Implements [IImage](#).

10.105.3.62 Save() [2/8]

```
void Save (
    const char * pFilename,
    PNGOption & pOption ) [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](#).

10.105.3.63 Save() [3/8]

```
void Save (
    const char * pFilename,
    PPMOption & pOption ) [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](#).

10.105.3.64 Save() [4/8]

```
void Save (
    const char * pFilename,
    PGMOption & pOption ) [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](#).

10.105.3.65 Save() [5/8]

```
void Save (
    const char * pFilename,
    TIFFOption & pOption ) [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](#).

10.105.3.66 Save() [6/8]

```
void Save (
    const char * pFilename,
    JPEGOption & pOption ) [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](#).

10.105.3.67 Save() [7/8]

```
void Save (
    const char * pFilename,
    JPG2Option & pOption ) [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](#).

10.105.3.68 Save() [8/8]

```
void Save (
    const char * pFilename,
    BMPOption & pOption ) [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](#).

10.105.3.69 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\(\)](#)

10.105.4 Friends And Related Function Documentation

10.105.4.1 IDataStream

```
friend class IDataStream [friend]
```

10.105.4.2 ImageConverter

```
friend class ImageConverter [friend]
```

10.105.4.3 ImageFiler

```
friend class ImageFiler [friend]
```

10.105.4.4 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

10.105.4.5 ImageUtilityImpl

```
friend class ImageUtilityImpl [friend]
```

10.105.4.6 ImageUtilityPolarizationImpl

```
friend class ImageUtilityPolarizationImpl [friend]
```

10.105.4.7 Stream

```
friend class Stream [friend]
```

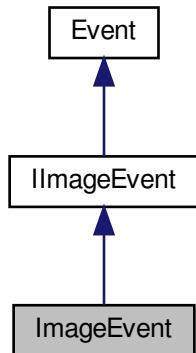
The documentation for this class was generated from the following file:

- [include/Image.h](#)

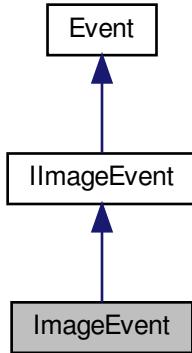
10.106 ImageEvent Class Reference

A handler for capturing image arrival events.

Inheritance diagram for ImageEvent:



Collaboration diagram for ImageEvent:



Public Member Functions

- [ImageEvent \(\)](#)
Default Constructor.
- [virtual ~ImageEvent \(\)](#)
Virtual Destructor.
- [virtual void OnImageEvent \(ImagePtr image\)=0](#)
Image event callback.

Protected Member Functions

- [ImageEvent & operator= \(const ImageEvent &\)](#)
Assignment operator.

Additional Inherited Members

10.106.1 Detailed Description

A handler for capturing image arrival events.

10.106.2 Constructor & Destructor Documentation

10.106.2.1 `ImageEvent()`

```
ImageEvent ( )
```

Default Constructor.

10.106.2.2 `~ImageEvent()`

```
virtual ~ImageEvent ( ) [virtual]
```

Virtual Destructor.

10.106.3 Member Function Documentation

10.106.3.1 `OnImageEvent()`

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

`Image` event callback.

Parameters

<code>image</code>	The <code>ImagePtr</code> object
--------------------	----------------------------------

Implements `IImageEvent`.

10.106.3.2 `operator=()`

```
ImageEvent& operator=
    const ImageEvent & ) [protected]
```

Assignment operator.

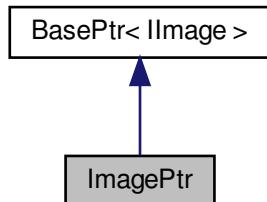
The documentation for this class was generated from the following file:

- `include/ImageEvent.h`

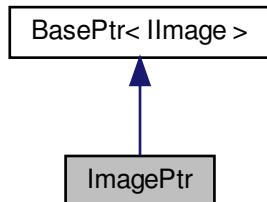
10.107 ImagePtr Class Reference

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

10.107.1 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.

10.107.2 Constructor & Destructor Documentation

10.107.2.1 `ImagePtr()` [1/4]

```
ImagePtr ()
```

Default constructor.

10.107.2.2 `ImagePtr()` [2/4]

```
ImagePtr (
    const int )
```

Default constructor with argument.

10.107.2.3 `ImagePtr()` [3/4]

```
ImagePtr (
    const long )
```

Default constructor with argument.

10.107.2.4 `ImagePtr()` [4/4]

```
ImagePtr (
    const std::nullptr_t )
```

Default constructor with argument.

10.107.2.5 ~ImagePtr()

```
virtual ~ImagePtr (
    void ) [virtual]
```

Virtual destructor.

10.107.3 Member Function Documentation

10.107.3.1 operator=()

```
virtual ImagePtr& operator=
    const ImagePtr & ) [virtual]
```

Assignment operator.

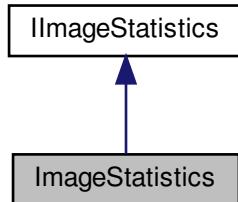
The documentation for this class was generated from the following file:

- [include/ImagePtr.h](#)

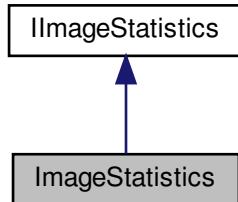
10.108 ImageStatistics Class Reference

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

10.108.1 Detailed Description

Represents image statistics for an image.

10.108.2 Constructor & Destructor Documentation

10.108.2.1 `ImageStatistics()` [1/2]

```
ImageStatistics ( )
```

Default constructor.

10.108.2.2 `~ImageStatistics()`

```
virtual ~ImageStatistics ( ) [virtual]
```

Default destructor.

10.108.2.3 `ImageStatistics()` [2/2]

```
ImageStatistics (   
    const ImageStatistics & other )
```

Copy constructor.

10.108.3 Member Function Documentation

10.108.3.1 `DisableAll()`

```
virtual void DisableAll ( ) [virtual]
```

Disable all channels.

Implements [IImageStatistics](#).

10.108.3.2 `EnableAll()`

```
virtual void EnableAll ( ) [virtual]
```

Enable all channels.

Implements [IImageStatistics](#).

10.108.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [virtual]
```

Enable only the grey channel.

Implements [IImageStatistics](#).

10.108.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [virtual]
```

Enable only the HSL channels.

Implements [IImageStatistics](#).

10.108.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [virtual]
```

Enable only the RGB channels.

Implements [IImageStatistics](#).

10.108.3.6 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](#).

10.108.3.7 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](#).

10.108.3.8 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](#).

10.108.3.9 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](#).

10.108.3.10 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](#).

10.108.3.11 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](#).

10.108.3.12 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 [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](#).

10.108.3.13 operator=()

```
IImageStatistics& operator= (
    const IImageStatistics & other )
```

Assignment operator.

Parameters

<i>other</i>	The IImageStatistics object to copy from.
--------------	---

10.108.3.14 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](#).

10.108.4 Friends And Related Function Documentation

10.108.4.1 ImageStatsCalculator

friend class ImageStatsCalculator [friend]

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

- [include/IImageStatistics.h](#)

10.109 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](#) &srclImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static void [CreateScaled](#) (const [ImagePtr](#) &srclImage, [ImagePtr](#) &destImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static [ImagePtr CreateNormalized](#) (const [ImagePtr](#) &srclImage, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=**IMAGE_DATA_RANGE**)
Computes a normalized image.
- static [ImagePtr CreateNormalized](#) (const [ImagePtr](#) &srclImage, const double min, const double max, [SourceDataRange](#) srcDataRange=**IMAGE_DATA_RANGE**)
Computes a normalized image.
- static [ImagePtr CreateNormalized](#) (const [ImagePtr](#) &srclImage, const double min, const double max, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=**IMAGE_DATA_RANGE**)
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srclImage, [ImagePtr](#) &destImage, [SourceDataRange](#) srcDataRange=**IMAGE_DATA_RANGE**)
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srclImage, [ImagePtr](#) &destImage, const double min, const double max, [SourceDataRange](#) srcDataRange=**IMAGE_DATA_RANGE**)
Computes a normalized image.

10.109.1 Detailed Description

Static helper functions for the image object class.

10.109.2 Member Enumeration Documentation

10.109.2.1 ImageScalingAlgorithm

enum [ImageScalingAlgorithm](#)

[Image](#) scaling algorithms.

Enumerator

NEAREST_NEIGHBOR	
------------------	--

10.109.2.2 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.

10.109.3 Member Function Documentation

10.109.3.1 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

10.109.3.2 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

10.109.3.3 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

10.109.3.4 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

10.109.3.5 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

10.109.3.6 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

10.109.3.7 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

10.110 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 &srclImage)

Computes a heatmap image.
- static void CreateHeatmap (const ImagePtr &srclImage, ImagePtr &destImage)

Computes a heatmap image.
- static void SetHeatmapColorGradient (const HeatmapColor newLowColor, const HeatmapColor newHighColor)

- static void [GetHeatmapColorGradient](#) ([HeatmapColor](#) ¤tLowColor, [HeatmapColor](#) ¤tHighColor)
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 ¤tLowValue, unsigned int ¤tHighValue)
Returns the current high and low values used in heatmap representations.

10.110.1 Detailed Description

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

10.110.2 Member Enumeration Documentation

10.110.2.1 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	

10.110.3 Member Function Documentation

10.110.3.1 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

<code>srcImage</code>	The source image from which to create the heatmap
-----------------------	---

See also

[SetHeatmapRange\(\)](#)
[SetHeatmapColorGradient\(\)](#)

Returns

The heatmap image

10.110.3.2 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

<code>srcImage</code>	The source image from which to create the heatmap
<code>destImage</code>	The destination image in which to store the created heatmap

See also

[SetHeatmapRange\(\)](#)
[SetHeatmapColorGradient\(\)](#)

10.110.3.3 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\(\)](#)

10.110.3.4 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\(\)](#)

10.110.3.5 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.

10.110.3.6 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

<code>newLowValue</code>	New value at which to begin color representation.
<code>newHighValue</code>	New value at which to end color representation.

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

- [include/ImageUtilityHeatmap.h](#)

10.111 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` &`srclImage`, const `PolarizationQuadrant` `desiredQuadrant`)

Extracts all pixels of a specified degree of linear polarization into a new image object.
- static void `ExtractPolarQuadrant` (const `ImagePtr` &`srclImage`, `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` &`srclImage`)

*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` &`srclImage`, `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` &`srclImage`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)

*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 `CreateStokesS0` (const `ImagePtr` &srclImage, `ImagePtr` &destStokesS0Image, const `ColorProcessingAlgorithm` colorProcessingAlg=`DEFAULT`)

Computes an image representing the overall intensity of light from a polarized image.
- static `ImagePtr` `CreateStokesS1` (const `ImagePtr` &srclImage, const `ColorProcessingAlgorithm` colorProcessingAlg=`DEFAULT`)

Computes an image representing the overall intensity of light from a polarized image.
- static void `CreateStokesS1` (const `ImagePtr` &srclImage, `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` &srclImage, 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 `CreateStokesS2` (const `ImagePtr` &srclImage, `ImagePtr` &destStokesS2Image, const `ColorProcessingAlgorithm` colorProcessingAlg=`DEFAULT`)

Computes an image representing the difference in intensity accepted through the polarizers.
- static `ImagePtr` `CreateDolp` (const `ImagePtr` &srclImage, const `ColorProcessingAlgorithm` colorProcessingAlg=`DEFAULT`)

Computes an image representing the fraction of incident light intensity in the linear polarization states.
- static void `CreateDolp` (const `ImagePtr` &srclImage, `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` &srclImage, 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` &srclImage, `ImagePtr` &destAolpImg, const `ColorProcessingAlgorithm` colorProcessingAlg=`DEFAULT`)

Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

10.111.1 Detailed Description

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

10.111.2 Member Enumeration Documentation

10.111.2.1 PolarizationQuadrant

```
enum PolarizationQuadrant
```

Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

Enumerator

<code>QUADRANT_I0</code>	The 0 degree of polarization.
<code>QUADRANT_I45</code>	The 45 degree of polarization.
<code>QUADRANT_I90</code>	The 90 degree of polarization.
<code>QUADRANT_I135</code>	The 135 degree of polarization.

10.111.3 Member Function Documentation

10.111.3.1 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

10.111.3.2 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

10.111.3.3 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

10.111.3.4 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

10.111.3.5 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

10.111.3.6 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

10.111.3.7 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

10.111.3.8 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

10.111.3.9 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

10.111.3.10 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

10.111.3.11 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

10.111.3.12 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

10.111.3.13 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

10.111.3.14 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](#)

10.112 InferenceBoundingBox Struct Reference

Inference Bounding Boxes data structure.

Public Attributes

- [InferenceBoxType boxType](#)
- [int16_t classId](#)
- [float32_t confidence](#)
- [InferenceBoxRect rect](#)
- [InferenceBoxCircle circle](#)
- [InferenceBoxRotatedRect rotatedRect](#)

10.112.1 Detailed Description

Inference Bounding Boxes data structure.

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

- [include/ChunkDataInference.h](#)

10.113 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.

10.113.1 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](#)

10.114 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](#)

10.115 InferenceBoxRect Struct Reference

Inference Bounding Box Type Data Structures.

Public Attributes

- `int16_t topLeftXCoord`
- `int16_t topLeftYCoord`
- `int16_t bottomRightXCoord`
- `int16_t bottomRightYCoord`

10.115.1 Detailed Description

Inference Bounding Box Type Data Structures.

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

- [include/ChunkDataInference.h](#)

10.116 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

10.117 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`

10.117.1 Detailed Description

Vector of integers with reference counting.

10.117.2 Constructor & Destructor Documentation

10.117.2.1 int64_autovector_t() [1/3]

```
int64_autovector_t ()
```

10.117.2.2 int64_autovector_t() [2/3]

```
int64_autovector_t (
    const int64_autovector_t & obj )
```

10.117.2.3 int64_autovector_t() [3/3]

```
int64_autovector_t (
    size_t n ) [explicit]
```

10.117.2.4 ~int64_autovector_t()

```
virtual ~int64_autovector_t (
    void ) [virtual]
```

10.117.3 Member Function Documentation**10.117.3.1 operator delete()**

```
void operator delete (
    void * pWhere )
```

10.117.3.2 operator new()

```
void* operator new (
    size_t uiSize )
```

10.117.3.3 operator=()

```
int64_autovector_t& operator= (
    const int64_autovector_t & obj )
```

10.117.3.4 operator[]() [1/2]

```
int64_t& operator[ ] (
    size_t uiIndex )
```

10.117.3.5 operator[]() [2/2]

```
const int64_t& operator[ ] (
    size_t uiIndex ) const
```

10.117.3.6 size()

```
size_t size ( ) const
```

10.117.4 Member Data Documentation**10.117.4.1 _pCount**

```
ATOMIC_VARIABLE* _pCount [protected]
```

10.117.4.2 _pv

```
std::vector<int64_t>* _pv [protected]
```

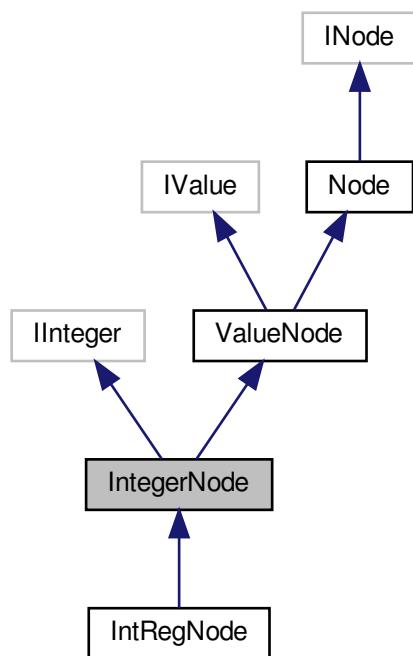
The documentation for this class was generated from the following file:

- include/SpinGenApi/Autovector.h

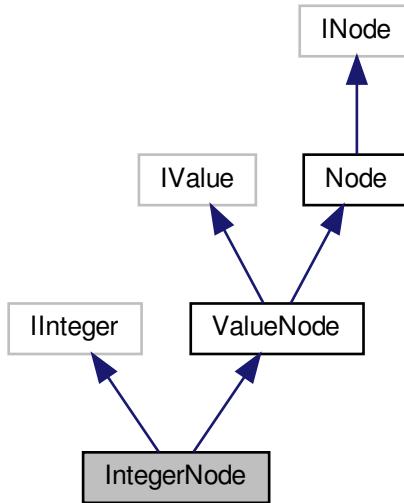
10.118 IntegerNode Class Reference

[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 IIInteger & 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

10.118.1 Detailed Description

`Interface` for string properties.

10.118.2 Constructor & Destructor Documentation

10.118.2.1 `IntegerNode()` [1/2]

`IntegerNode ()`

10.118.2.2 `IntegerNode()` [2/2]

```
IntegerNode (
    std::shared_ptr< Node::NodeImpl > pInteger )
```

10.118.2.3 `~IntegerNode()`

`virtual ~IntegerNode () [virtual]`

10.118.3 Member Function Documentation

10.118.3.1 GetFloatAlias()

```
virtual IFloat* GetFloatAlias ( ) [virtual]
```

gets the interface of an alias node.

10.118.3.2 GetInc()

```
virtual int64_t GetInc ( ) [virtual]
```

Get increment.

10.118.3.3 GetIncMode()

```
virtual EIncMode GetIncMode ( ) [virtual]
```

Get increment mode.

10.118.3.4 GetListOfValidValues()

```
virtual int64_autovector_t GetListOfValidValues ( bool bounded = true ) [virtual]
```

Get list of valid value.

10.118.3.5 GetMax()

```
virtual int64_t GetMax ( ) [virtual]
```

Get maximum value allowed.

10.118.3.6 GetMin()

```
virtual int64_t GetMin ( ) [virtual]
```

Get minimum value allowed.

10.118.3.7 GetRepresentation()

```
virtual ERepresentation GetRepresentation() [virtual]
```

Get recommended representation.

10.118.3.8 GetUnit()

```
virtual GenICam::gcstring GetUnit() [virtual]
```

Get the physical unit name.

10.118.3.9 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

10.118.3.10 ImposeMax()

```
virtual void ImposeMax(
    int64_t Value) [virtual]
```

Restrict maximum value.

10.118.3.11 ImposeMin()

```
virtual void ImposeMin(
    int64_t Value) [virtual]
```

Restrict minimum value.

10.118.3.12 operator()

```
virtual int64_t operator() ( ) [virtual]
```

Get node value.

10.118.3.13 operator*()

```
virtual int64_t operator* ( ) [virtual]
```

Get node value.

10.118.3.14 operator=()

```
virtual IInteger& operator= (
    int64_t Value ) [virtual]
```

Set node value.

10.118.3.15 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Integer

Reimplemented from [ValueNode](#).

Reimplemented in [IntRegNode](#).

10.118.3.16 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)

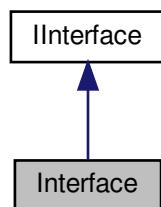
The documentation for this class was generated from the following file:

- include/SpinGenApi/[IntegerNode.h](#)

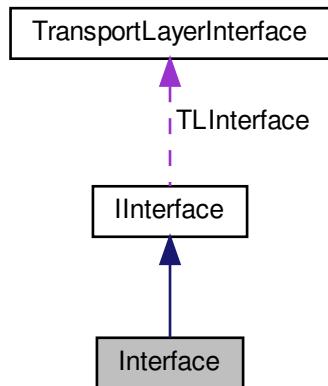
10.119 Interface Class Reference

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

- **bool UpdateCameras ()**
Returns a list of cameras available on this interface.
- **GenApi::INodeMap & GetTLNodeMap () const**
Updates the list of cameras on this interface.
- **Gets a nodeMap that is generated from a [GenICam XML file](#) for the GenTL interface Module.**
- **void RegisterEvent (Event &evtToRegister)**
Registers an event for the interface.
- **void UnregisterEvent (Event &evtToUnregister)**
Unregisters an event 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

10.119.1 Detailed Description

An interface object which holds a list of cameras.

10.119.2 Constructor & Destructor Documentation

10.119.2.1 ~Interface()

```
virtual ~Interface (
    void ) [virtual]
```

Virtual Destructor.

10.119.3 Member Function Documentation

10.119.3.1 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

<code>updateCameras</code>	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](#).

10.119.3.2 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](#).

10.119.3.3 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](#).

10.119.3.4 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](#).

10.119.3.5 RegisterEvent()

```
void RegisterEvent (
    Event & evtToRegister ) [virtual]
```

Registers an event for the interface.

Parameters

<code>evtToRegister</code>	The event to register for the interface
----------------------------	---

Implements [IInterface](#).

10.119.3.6 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

<code>deviceKey</code>	The Action Command's device key
<code>groupKey</code>	The Action Command's group key
<code>groupMask</code>	The Action Command's group mask
<code>actionTime</code>	(Optional) Time when to assert a future action. Zero means immediate action.
<code>pResultSize</code>	(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.

Parameters

<code>results</code>	(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](#).

10.119.3.7 UnregisterEvent()

```
void UnregisterEvent (
    Event & evtToUnregister ) [virtual]
```

Unregisters an event for the interface.

Parameters

<code>evtToUnregister</code>	The event to unregister from the interface
------------------------------	--

Implements [IInterface](#).

10.119.3.8 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 [IInterface](#).

10.119.4 Friends And Related Function Documentation

10.119.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

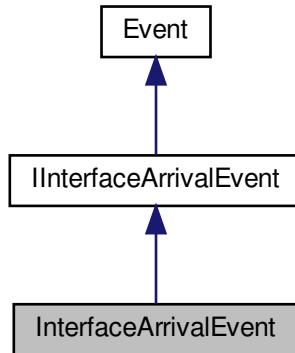
The documentation for this class was generated from the following file:

- [include/Interface.h](#)

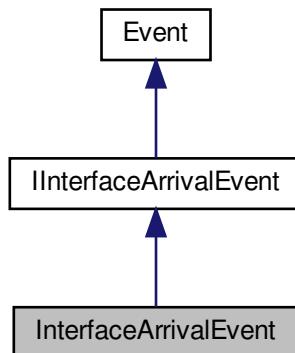
10.120 InterfaceArrivalEvent Class Reference

An event handler for capturing the interface arrival event.

Inheritance diagram for InterfaceArrivalEvent:



Collaboration diagram for InterfaceArrivalEvent:



Public Member Functions

- `InterfaceArrivalEvent ()`
Default constructor.
- `virtual ~InterfaceArrivalEvent ()`
Virtual destructor.
- `virtual void OnInterfaceArrival (std::string interfaceID)=0`
Interface arrival event callback.

Protected Member Functions

- `InterfaceArrivalEvent & operator= (const InterfaceArrivalEvent &)`
Assignment operator.

Additional Inherited Members

10.120.1 Detailed Description

An event handler for capturing the interface arrival event.

10.120.2 Constructor & Destructor Documentation

10.120.2.1 `InterfaceArrivalEvent()`

`InterfaceArrivalEvent ()`

Default constructor.

10.120.2.2 `~InterfaceArrivalEvent()`

`virtual ~InterfaceArrivalEvent () [virtual]`

Virtual destructor.

10.120.3 Member Function Documentation

10.120.3.1 `OnInterfaceArrival()`

`virtual void OnInterfaceArrival (`
 `std::string interfaceID) [pure virtual]`

Interface arrival event callback.

Parameters

<i>interfaceID</i>	The ID of the interface that arrived
--------------------	--------------------------------------

Implements [IInterfaceArrivalEvent](#).

10.120.3.2 operator=()

```
InterfaceArrivalEvent& operator= (
    const InterfaceArrivalEvent & ) [protected]
```

Assignment operator.

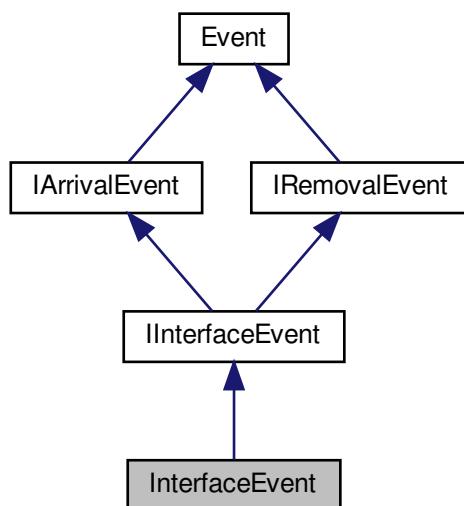
The documentation for this class was generated from the following file:

- [include/InterfaceArrivalEvent.h](#)

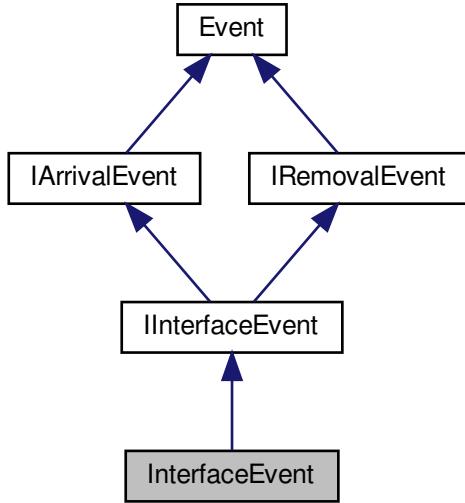
10.121 InterfaceEvent Class Reference

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

Inheritance diagram for InterfaceEvent:



Collaboration diagram for InterfaceEvent:



Public Member Functions

- `InterfaceEvent ()`
Default constructor.
- `virtual ~InterfaceEvent ()`
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

- `InterfaceEvent & operator= (const InterfaceEvent &)`
Assignment operator.

Additional Inherited Members

10.121.1 Detailed Description

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

10.121.2 Constructor & Destructor Documentation

10.121.2.1 InterfaceEvent()

```
InterfaceEvent ( )
```

Default constructor.

10.121.2.2 ~InterfaceEvent()

```
virtual ~InterfaceEvent ( ) [virtual]
```

Virtual destructor.

10.121.3 Member Function Documentation

10.121.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Device arrival event callback.

Implements [IInterfaceEvent](#).

10.121.3.2 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device removal event.

Parameters

<code>serialNumber</code>	The serial number of the removed device
---------------------------	---

Implements [IInterfaceEvent](#).

10.121.3.3 operator=()

```
InterfaceEvent& operator= (
    const InterfaceEvent & ) [protected]
```

Assignment operator.

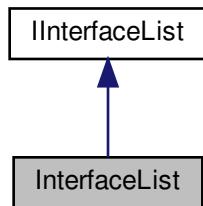
The documentation for this class was generated from the following file:

- [include/InterfaceEvent.h](#)

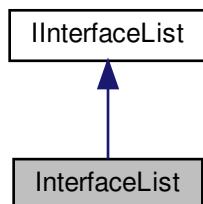
10.122 InterfaceList Class Reference

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.

Friends

- class `SystemImpl`

Additional Inherited Members

10.122.1 Detailed Description

A list of the available interfaces on the system.

10.122.2 Constructor & Destructor Documentation

10.122.2.1 `InterfaceList()` [1/2]

```
InterfaceList (
    void )
```

10.122.2.2 `~InterfaceList()`

```
virtual ~InterfaceList (
    void ) [virtual]
```

10.122.2.3 `InterfaceList()` [2/2]

```
InterfaceList (
    const InterfaceList & iface )
```

10.122.3 Member Function Documentation

10.122.3.1 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 [IInterfaceList](#).

10.122.3.2 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](#).

10.122.3.3 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](#).

10.122.3.4 operator=()

```
InterfaceList& operator= (
    const InterfaceList & iface )
```

Assignment operator.

10.122.3.5 operator[]()

```
InterfacePtr operator[ ] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [IInterfaceList](#).

10.122.4 Friends And Related Function Documentation

10.122.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

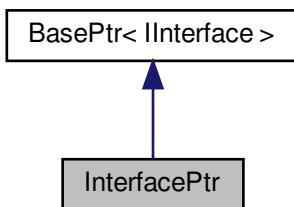
The documentation for this class was generated from the following file:

- [include/IInterfaceList.h](#)

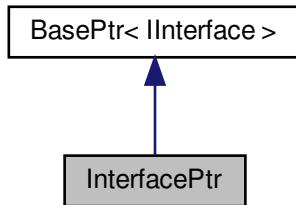
10.123 InterfacePtr Class Reference

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

10.123.1 Detailed Description

A reference tracked pointer to the interface object.

10.123.2 Constructor & Destructor Documentation

10.123.2.1 [InterfacePtr\(\)](#) [1/4]

`InterfacePtr () throw () [inline]`

Default Constructor.

10.123.2.2 [InterfacePtr\(\)](#) [2/4]

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

Default Constructor with argument.

10.123.2.3 InterfacePtr() [3/4]

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

10.123.2.4 InterfacePtr() [4/4]

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

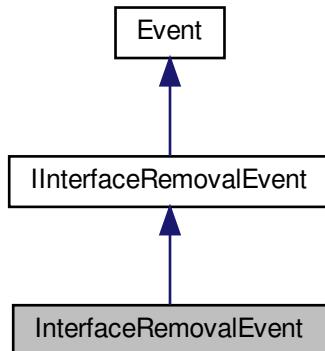
The documentation for this class was generated from the following file:

- [include/InterfacePtr.h](#)

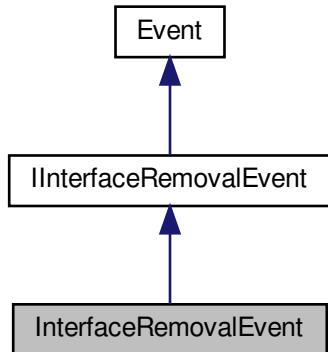
10.124 InterfaceRemovalEvent Class Reference

An event handler for capturing the interface removal event.

Inheritance diagram for InterfaceRemovalEvent:



Collaboration diagram for InterfaceRemovalEvent:



Public Member Functions

- [InterfaceRemovalEvent \(\)](#)
Default Constructor.
- [virtual ~InterfaceRemovalEvent \(\)](#)
Virtual Destructor.
- [virtual void OnInterfaceRemoval \(std::string interfaceID\)=0](#)
Interface removal event callback.

Protected Member Functions

- [InterfaceRemovalEvent & operator= \(const InterfaceRemovalEvent &\)](#)
Assignment operator.

Additional Inherited Members

10.124.1 Detailed Description

An event handler for capturing the interface removal event.

10.124.2 Constructor & Destructor Documentation

10.124.2.1 InterfaceRemovalEvent()

```
InterfaceRemovalEvent ( )
```

Default Constructor.

10.124.2.2 ~InterfaceRemovalEvent()

```
virtual ~InterfaceRemovalEvent ( ) [virtual]
```

Virtual Destructor.

10.124.3 Member Function Documentation

10.124.3.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

Interface removal event callback.

Parameters

<i>interfaceID</i>	The ID of the interface removed
--------------------	---------------------------------

Implements [IInterfaceRemovalEvent](#).

10.124.3.2 operator=()

```
InterfaceRemovalEvent& operator= (
    const InterfaceRemovalEvent & ) [protected]
```

Assignment operator.

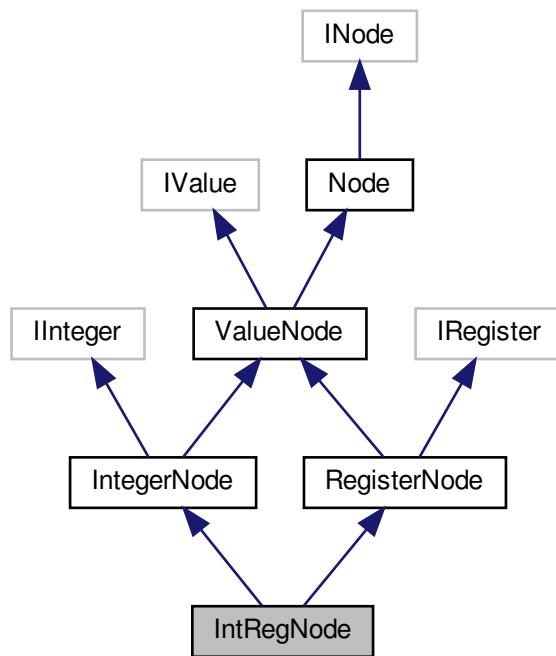
The documentation for this class was generated from the following file:

- [include/InterfaceRemovalEvent.h](#)

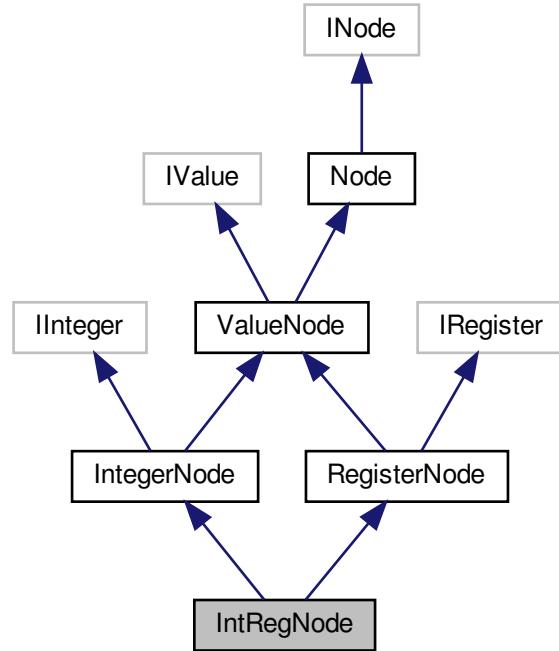
10.125 IntRegNode Class Reference

[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

10.125.1 Detailed Description

[Interface](#) for string properties.

10.125.2 Constructor & Destructor Documentation

10.125.2.1 IntRegNode() [1/2]

```
IntRegNode ( )
```

10.125.2.2 IntRegNode() [2/2]

```
IntRegNode ( std::shared_ptr< Node::NodeImpl > pInteger )
```

10.125.2.3 ~IntRegNode()

```
virtual ~IntRegNode ( ) [virtual]
```

10.125.3 Member Function Documentation

10.125.3.1 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](#)

10.126 IplInfo Struct Reference

Public Member Functions

- [IplInfo \(\)](#)

Public Attributes

- std::string [ipAddress](#)
- std::string [gateway](#)
- std::string [subnetMask](#)
- unsigned int [subnetLength](#)

10.126.1 Constructor & Destructor Documentation

10.126.1.1 IpInfo()

```
IpInfo () [inline]
```

10.126.2 Member Data Documentation

10.126.2.1 gateway

```
std::string gateway
```

10.126.2.2 ipAddress

```
std::string ipAddress
```

10.126.2.3 subnetLength

```
unsigned int subnetLength
```

10.126.2.4 subnetMask

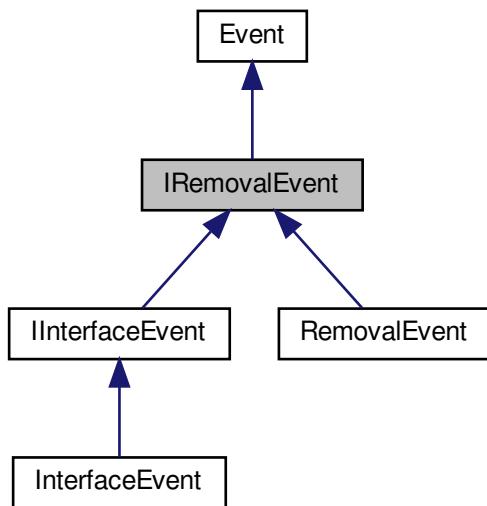
```
std::string subnetMask
```

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

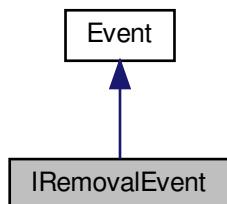
- include/AdapterConfig.h

10.127 IRemovalEvent Class Reference

Inheritance diagram for IRemovalEvent:



Collaboration diagram for IRemovalEvent:



Public Member Functions

- virtual `~IRemovalEvent ()`
- virtual void `OnDeviceRemoval (uint64_t serialNumber)=0`

Protected Member Functions

- `IRemovalEvent ()`
- `IRemovalEvent (const IRemovalEvent &)`
- `IRemovalEvent & operator= (const IRemovalEvent &)`

Additional Inherited Members

10.127.1 Constructor & Destructor Documentation

10.127.1.1 ~IRemovalEvent()

```
virtual ~IRemovalEvent ( ) [inline], [virtual]
```

10.127.1.2 IRemovalEvent() [1/2]

```
IRemovalEvent ( ) [inline], [protected]
```

10.127.1.3 IRemovalEvent() [2/2]

```
IRemovalEvent (
    const IRemovalEvent & ) [inline], [protected]
```

10.127.2 Member Function Documentation

10.127.2.1 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implemented in [InterfaceEvent](#), [RemovalEvent](#), and [IInterfaceEvent](#).

10.127.2.2 operator=()

```
IRemovalEvent& operator= (
    const IRemovalEvent & ) [protected]
```

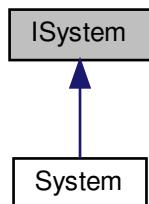
The documentation for this class was generated from the following file:

- include/Interface/IRemovalEvent.h

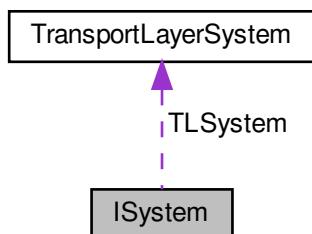
10.128 ISystem Class Reference

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 [CameraList GetCameras \(bool updateInterfaces=true, bool updateCameras=true\)=0](#)
- virtual bool [UpdateCameras \(bool updateInterfaces=true\)=0](#)
- virtual void [UpdateInterfaceList \(\)=0](#)
- virtual void [RegisterEvent \(Event &evtToRegister\)=0](#)
- virtual void [UnregisterEvent \(Event &evtToUnregister\)=0](#)
- virtual void [RegisterInterfaceEvent \(Event &evtToRegister, bool updateInterface=true\)=0](#)
- virtual void [UnregisterInterfaceEvent \(Event &evtToUnregister\)=0](#)
- virtual void [RegisterLoggingEvent \(LoggingEvent &handler\)=0](#)
- virtual void [UnregisterAllLoggingEvent \(\)=0](#)
- virtual void [UnregisterLoggingEvent \(LoggingEvent &handler\)=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`

10.128.1 Detailed Description

The interface file for `System`.

10.128.2 Constructor & Destructor Documentation

10.128.2.1 ~ISystem()

```
virtual ~ISystem ( ) [inline], [virtual]
```

10.128.2.2 ISystem() [1/2]

```
ISystem ( ) [inline], [protected]
```

10.128.2.3 ISystem() [2/2]

```
ISystem (
    const ISystem & ) [inline], [protected]
```

10.128.3 Member Function Documentation

10.128.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateInterfaces = true,
    bool updateCameras = true ) [pure virtual]
```

Implemented in [System](#).

10.128.3.2 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

10.128.3.3 GetLibraryVersion()

```
virtual const LibraryVersion GetLibraryVersion () [pure virtual]
```

Implemented in [System](#).

10.128.3.4 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel () [pure virtual]
```

Implemented in [System](#).

10.128.3.5 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap () const [pure virtual]
```

Implemented in [System](#).

10.128.3.6 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [System](#).

10.128.3.7 operator=()

```
ISystem& operator= (
    const ISystem & ) [protected]
```

10.128.3.8 RegisterEvent()

```
virtual void RegisterEvent (
    Event & evtToRegister ) [pure virtual]
```

Implemented in [System](#).

10.128.3.9 RegisterInterfaceEvent()

```
virtual void RegisterInterfaceEvent (
    Event & evtToRegister,
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

10.128.3.10 RegisterLoggingEvent()

```
virtual void RegisterLoggingEvent (
    LoggingEvent & handler ) [pure virtual]
```

Implemented in [System](#).

10.128.3.11 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [pure virtual]
```

Implemented in [System](#).

10.128.3.12 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](#).

10.128.3.13 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [pure virtual]
```

Implemented in [System](#).

10.128.3.14 UnregisterAllLoggingEvent()

```
virtual void UnregisterAllLoggingEvent ( ) [pure virtual]
```

Implemented in [System](#).

10.128.3.15 UnregisterEvent()

```
virtual void UnregisterEvent (
    Event & evtToUnregister ) [pure virtual]
```

Implemented in [System](#).

10.128.3.16 UnregisterInterfaceEvent()

```
virtual void UnregisterInterfaceEvent (
    Event & evtToUnregister ) [pure virtual]
```

Implemented in [System](#).

10.128.3.17 UnregisterLoggingEvent()

```
virtual void UnregisterLoggingEvent (
    LoggingEvent & handler ) [pure virtual]
```

Implemented in [System](#).

10.128.3.18 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [pure virtual]
```

Implemented in [System](#).

10.128.3.19 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [pure virtual]
```

Implemented in [System](#).

10.128.4 Friends And Related Function Documentation**10.128.4.1 SystemPtrInternal**

```
friend class SystemPtrInternal [friend]
```

10.128.5 Member Data Documentation**10.128.5.1 TLSYSTEM**

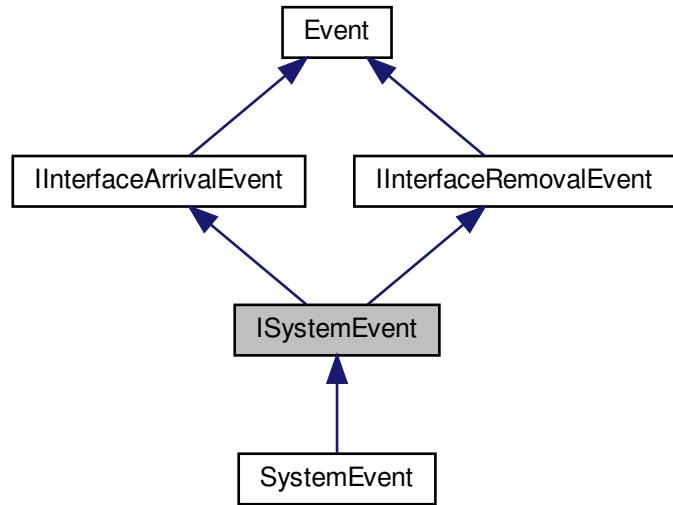
[TransportLayerSystem](#) TLSYSTEM

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

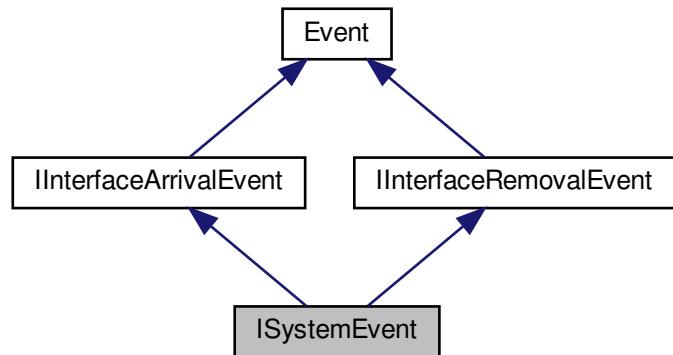
- include/Interface/[ISystem.h](#)

10.129 ISystemEvent Class Reference

Inheritance diagram for ISystemEvent:



Collaboration diagram for ISystemEvent:



Public Member Functions

- virtual `~ISystemEvent ()`
- virtual void `OnInterfaceArrival (std::string interfaceID)=0`
- virtual void `OnInterfaceRemoval (std::string interfaceID)=0`

Protected Member Functions

- [ISystemEvent \(\)](#)
- [ISystemEvent \(const ISystemEvent &\)](#)
- [ISystemEvent & operator= \(const ISystemEvent &\)](#)

Additional Inherited Members

10.129.1 Constructor & Destructor Documentation

10.129.1.1 ~ISystemEvent()

```
virtual ~ISystemEvent ( ) [inline], [virtual]
```

10.129.1.2 ISystemEvent() [1/2]

```
ISystemEvent ( ) [inline], [protected]
```

10.129.1.3 ISystemEvent() [2/2]

```
ISystemEvent (
    const ISystemEvent & ) [inline], [protected]
```

10.129.2 Member Function Documentation

10.129.2.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    std::string interfaceID ) [pure virtual]
```

Implements [IInterfaceArrivalEvent](#).

Implemented in [SystemEvent](#).

10.129.2.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

Implements [IInterfaceRemovalEvent](#).

Implemented in [SystemEvent](#).

10.129.2.3 operator=()

```
ISystemEvent& operator= (
    const ISystemEvent & ) [protected]
```

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

- include/Interface/[ISystemEvent.h](#)

10.130 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.

10.130.1 Detailed Description

Options for saving JPEG image.

10.130.2 Constructor & Destructor Documentation

10.130.2.1 JPEGOption()

```
JPEGOption ( ) [inline]
```

10.130.3 Member Data Documentation

10.130.3.1 progressive

```
bool progressive
```

Whether to save as a progressive JPEG file.

10.130.3.2 quality

```
unsigned int quality
```

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

10.130.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

10.131 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.

10.131.1 Detailed Description

Options for saving JPEG2000 image.

10.131.2 Constructor & Destructor Documentation

10.131.2.1 [JPG2Option\(\)](#)

`JPG2Option ()` [inline]

10.131.3 Member Data Documentation

10.131.3.1 `quality`

`unsigned int quality`

JPEG saving quality in range (1-512).

10.131.3.2 `reserved`

`unsigned int reserved[16]`

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

10.132 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.

10.132.1 Detailed Description

Provides easier access to the current version of [Spinnaker](#).

10.132.2 Member Data Documentation

10.132.2.1 build

`unsigned int build`

Build number of the library.

10.132.2.2 major

`unsigned int major`

Major version of the library.

10.132.2.3 minor

`unsigned int minor`

Minor version of the library.

10.132.2.4 type

```
unsigned int type
```

Version type of the library.

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

- [include/SpinnakerDefs.h](#)

10.133 LockableObject< Object >::Lock Class Reference

A scopelevel [Lock](#) class.

Public Member Functions

- [Lock \(const LockableObject< Object > &obj\)](#)
- [~Lock \(\)](#)

10.133.1 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.

10.133.2 Constructor & Destructor Documentation

10.133.2.1 Lock()

```
Lock (
    const LockableObject< Object > & obj ) [inline]
```

10.133.2.2 ~Lock()

```
~Lock ( ) [inline]
```

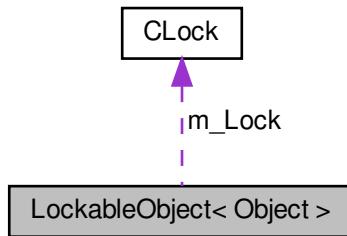
The documentation for this class was generated from the following file:

- [include/SpinGenApi/GCSynch.h](#)

10.134 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](#)

10.134.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >
```

Instance-Lock for an object.

10.134.2 Member Function Documentation

10.134.2.1 GetLock()

```
Lock GetLock ( ) const [inline]
```

Get a new lock.

10.134.3 Friends And Related Function Documentation

10.134.3.1 Lock

```
friend class Lock [friend]
```

10.134.4 Member Data Documentation

10.134.4.1 m_Lock

```
Clock m_Lock [mutable]
```

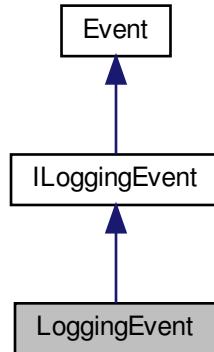
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

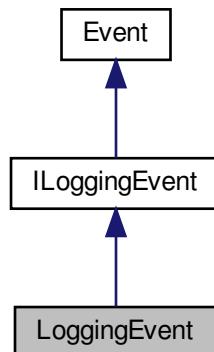
10.135 LoggingEvent Class Reference

An event handler for capturing the device logging event.

Inheritance diagram for LoggingEvent:



Collaboration diagram for LoggingEvent:



Public Member Functions

- [LoggingEvent \(\)](#)
Default constructor.
- [~LoggingEvent \(\)](#)
Virtual destructor.
- [virtual void OnLogEvent \(LoggingEventDataPtr eventPtr\)=0](#)
The callback for the log event.

Protected Member Functions

- `LoggingEvent & operator= (const LoggingEvent &)`
Assignment operator.

Additional Inherited Members

10.135.1 Detailed Description

An event handler for capturing the device logging event.

10.135.2 Constructor & Destructor Documentation

10.135.2.1 `LoggingEvent()`

`LoggingEvent ()`

Default constructor.

10.135.2.2 `~LoggingEvent()`

`~LoggingEvent ()`

Virtual destructor.

10.135.3 Member Function Documentation

10.135.3.1 `OnLogEvent()`

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

The callback for the log event.

Parameters

<code>eventPtr</code>	The logging event pointer
-----------------------	---------------------------

Implements [ILoggingEvent](#).

10.135.3.2 operator=()

```
LoggingEvent& operator= (
    const LoggingEvent & ) [protected]
```

Assignment operator.

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

- [include/LoggingEvent.h](#)

10.136 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](#)

10.136.1 Detailed Description

The [LoggingEventData](#) object.

10.136.2 Constructor & Destructor Documentation

10.136.2.1 ~LoggingEventData()

```
~LoggingEventData ( )
```

Default Destructor.

10.136.2.2 LoggingEventData()

```
LoggingEventData (
    void * data ) [protected]
```

Default Constructor.

10.136.3 Member Function Documentation

10.136.3.1 GetCategoryName()

```
const char* GetCategoryName ( )
```

Gets the logging event category name.

Returns

The category name

10.136.3.2 GetLogMessage()

```
const char* GetLogMessage ( )
```

Gets the logging event message.

Returns

The log message

10.136.3.3 GetNDC()

```
const char* GetNDC ( )
```

Gets the logging event's Nested Diagnostic Context (NDC).

Returns

The log event's NDC

10.136.3.4 GetPriority()

```
const int GetPriority ( )
```

Gets the logging event priority.

Returns

The log priority

10.136.3.5 GetPriorityName()

```
const char* GetPriorityName ( )
```

Gets the logging event priority name.

Returns

The priority name of the log

10.136.3.6 GetThreadName()

```
const char* GetThreadName ( )
```

Gets the logging event thread name.

Returns

The thread name

10.136.3.7 GetTimestamp()

```
const char* GetTimestamp( )
```

Gets the logging event time stamp.

Returns

The time stamp of the log

10.136.4 Friends And Related Function Documentation

10.136.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

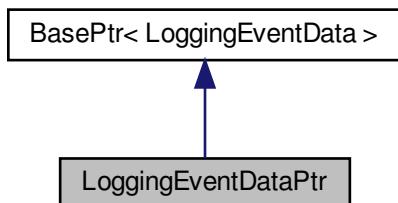
The documentation for this class was generated from the following file:

- [include/LoggingEventData.h](#)

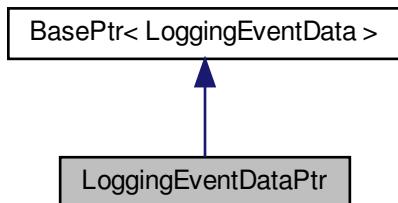
10.137 LoggingEventDataPtr Class Reference

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

10.137.1 Detailed Description

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

10.137.2 Constructor & Destructor Documentation

10.137.2.1 LoggingEventDataPtr() [1/4]

```
LoggingEventDataPtr ( ) throw () [inline]
```

Default Constructor.

10.137.2.2 LoggingEventDataPtr() [2/4]

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

Default Constructor with argument.

10.137.2.3 LoggingEventDataPtr() [3/4]

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

Default Constructor with argument.

10.137.2.4 LoggingEventDataPtr() [4/4]

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

Default Constructor with argument.

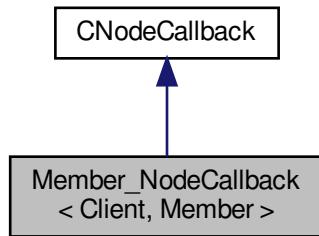
The documentation for this class was generated from the following file:

- [include/LoggingEventDataPtr.h](#)

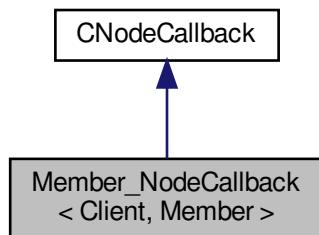
10.138 Member_NodeCallback< Client, Member > Class Template Reference

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

10.138.1 Detailed Description

```
template<class Client, class Member>
class Spinnaker::GenApi::Member_NodeCallback< Client, Member >
```

Container for a member function pointer.

10.138.2 Member Typedef Documentation

10.138.2.1 PMEMBERFUNC

```
typedef void(Client::* PMEMBERFUNC) (INode *)
```

Member function type.

10.138.3 Constructor & Destructor Documentation

10.138.3.1 Member_NodeCallback()

```
Member_NodeCallback (
    INode * pNode,
    Client & client,
    Member member,
    ECallbackType CallbackType ) [inline]
```

Constructor.

10.138.4 Member Function Documentation

10.138.4.1 Destroy()

```
virtual void Destroy ( ) [inline], [virtual]
```

destroys the object

Implements [CNodeCallback](#).

10.138.4.2 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](#)

10.139 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]

10.139.1 Detailed Description

Options for saving MJPG files.

10.139.2 Constructor & Destructor Documentation

10.139.2.1 MJPGOption()

```
MJPGOption ( ) [inline]
```

10.139.3 Member Data Documentation

10.139.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

10.139.3.2 quality

```
unsigned int quality
```

[Image](#) quality (1-100)

10.139.3.3 reserved

```
unsigned int reserved[256]
```

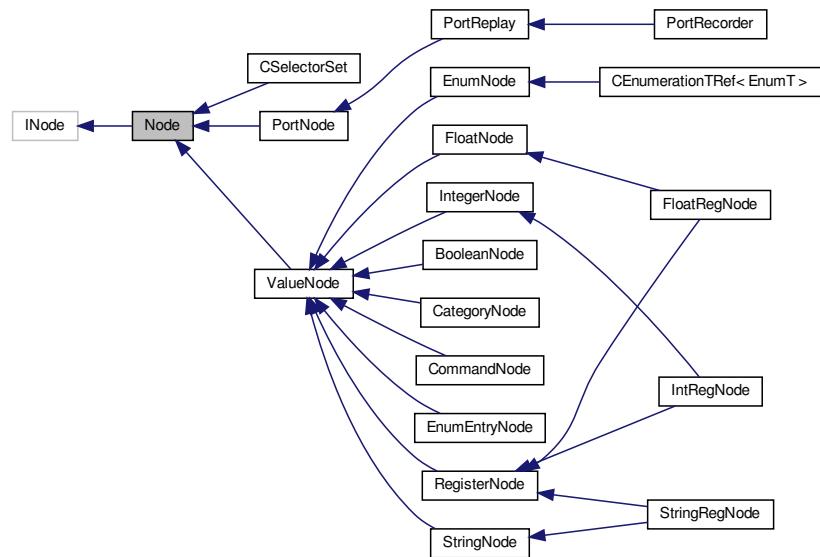
The documentation for this struct was generated from the following file:

- [include/SpinVideoDefs.h](#)

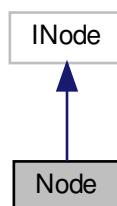
10.140 Node Class Reference

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 `IsCachable` () 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 `lingroup 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`

10.140.1 Detailed Description

class common to all nodes

10.140.2 Constructor & Destructor Documentation

10.140.2.1 `Node()` [1/2]

`Node ()`

Constructor.

10.140.2.2 Node() [2/2]

```
Node ( std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Constructor.

10.140.2.3 ~Node()

```
~Node ( )
```

Destructor.

10.140.3 Member Function Documentation

10.140.3.1 DeregisterCallback()

```
virtual bool DeregisterCallback ( CallbackHandleType hCallback ) [virtual]
```

De register change callback Destroys [CNodeCallback](#) object.

Returns

true if the callback handle was valid

10.140.3.2 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented in [PortRecorder](#).

10.140.3.3 GetAlias()

```
virtual INode* GetAlias ( ) const [virtual]
```

Retrieves the a node which describes the same feature in a different way.

10.140.3.4 GetCachingMode()

```
virtual ECachingMode GetCachingMode ( ) const [virtual]
```

Get Caching Mode.

10.140.3.5 GetCastAlias()

```
virtual INode* GetCastAlias ( ) const [virtual]
```

Retrieves the a node which describes the same feature so that it can be casted.

10.140.3.6 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

10.140.3.7 GetDescription()

```
virtual GenICam::gcstring GetDescription ( ) const [virtual]
```

Get a long description of the node.

10.140.3.8 GetDeviceName()

```
virtual GenICam::gcstring GetDeviceName ( ) const [virtual]
```

Get a name of the device.

10.140.3.9 GetDisplayName()

```
virtual GenICam::gcstring GetDisplayName( ) const [virtual]
```

Get a name string for display.

10.140.3.10 GetDocuURL()

```
virtual GenICam::gcstring GetDocuURL( ) const [virtual]
```

Gets a URL pointing to the documentation of that feature.

10.140.3.11 GetEventID()

```
virtual GenICam::gcstring GetEventID( ) const [virtual]
```

Get the EventId of the node.

10.140.3.12 GetName()

```
virtual GenICam::gcstring GetName(   
    bool FullQualified = false ) const [virtual]
```

Get node name.

10.140.3.13 GetNameSpace()

```
virtual GenApi::ENameSpace GetNameSpace( ) const [virtual]
```

Get name space.

10.140.3.14 GetNodeHandle()

```
std::shared_ptr<Node::NodeImpl> GetNodeHandle( ) const
```

Get **Node** handle.

10.140.3.15 GetNodeMap()

```
virtual INodeMap* GetNodeMap( ) const [virtual]
```

Retrieves the central node map.

10.140.3.16 GetParents()

```
virtual void GetParents(   
    GenApi::NodeList_t & Parents ) const [virtual]
```

Gets all nodes this node is directly depending on.

Parameters

<code>out</code>	<code>Parents</code>	List of parent nodes
------------------	----------------------	----------------------

10.140.3.17 GetPollingTime()

```
virtual int64_t GetPollingTime ( ) const [virtual]
```

recommended polling time (for not cacheable nodes)

10.140.3.18 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

10.140.3.19 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.

10.140.3.20 GetPropertyNames()

```
virtual void GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [virtual]
```

Returns a list of the names all properties set during initialization.

10.140.3.21 GetSelectedFeatures()

```
virtual void GetSelectedFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of selected features

10.140.3.22 GetSelectingFeatures()

```
virtual void GetSelectingFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of features selecting this node

10.140.3.23 GetToolTip()

```
virtual GenICam::gcstring GetToolTip ( ) const [virtual]
```

Get a short description of the node.

10.140.3.24 GetVisibility()

```
virtual EVisibility GetVisibility ( ) const [virtual]
```

Get the recommended visibility of the node.

10.140.3.25 ImposeAccessMode()

```
virtual void ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [virtual]
```

Imposes an access mode to the natural access mode of the node.

10.140.3.26 ImposeVisibility()

```
virtual void ImposeVisibility (
    EVisibility ImposedVisibility ) [virtual]
```

Imposes a visibility to the natural visibility of the node.

10.140.3.27 InvalidateNode()

```
virtual void InvalidateNode ( ) [virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

10.140.3.28 IsAccessModeCacheable()

```
virtual EYesNo IsAccessModeCacheable() const [virtual]
```

True if the AccessMode can be cached.

10.140.3.29 IsCachable()

```
virtual bool IsCachable() const [virtual]
```

Is the node value cacheable.

10.140.3.30 IsDeprecated()

```
virtual bool IsDeprecated() const [virtual]
```

True if the node should not be used any more.

10.140.3.31 IsFeature()

```
virtual bool IsFeature() const [virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

10.140.3.32 IsSelector()

```
virtual bool IsSelector() const [virtual]
```

Selector interface overrides.

true if this feature selects a group of features

10.140.3.33 IsStreamable()

```
virtual bool IsStreamable() const [virtual]
```

True if the node is streamable.

10.140.3.34 operator"!=()

```
virtual bool operator!= (
    int nullPtr ) const [virtual]
```

10.140.3.35 operator==()

```
virtual bool operator== (
    int nullPtr ) const [virtual]
```

10.140.3.36 RegisterCallback()

```
virtual CallbackHandleType RegisterCallback (
    CNodeCallback * pCallback ) [virtual]
```

Register change callback Takes ownership of the [CNodeCallback](#) object.

10.140.3.37 SetNodeHandle()

```
void SetNodeHandle (
    std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Set [Node](#) handle.

10.140.3.38 SetNodeMap()

```
void SetNodeMap (
    INodeMap * pNodeMap )
```

10.140.3.39 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](#).

10.140.3.40 SetReference() [2/2]

```
virtual void SetReference (
    ISelector * pBase ) [virtual]
```

10.140.4 Member Data Documentation

10.140.4.1 m_Callbacks

`std::list<CallbackHandleType_t*> m_Callbacks [protected]`

List of callbacks.

10.140.4.2 m_pNodeData

`std::shared_ptr<Node::NodeImpl> m_pNodeData [protected]`

10.140.4.3 m_pNodeMap

`INodeMap* m_pNodeMap [protected]`

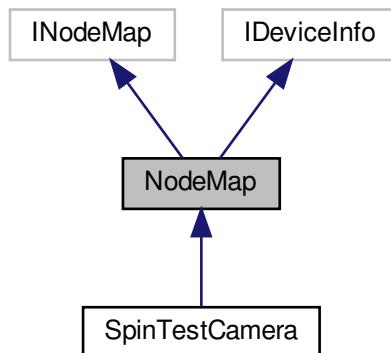
The documentation for this class was generated from the following file:

- include/SpinGenApi/Node.h

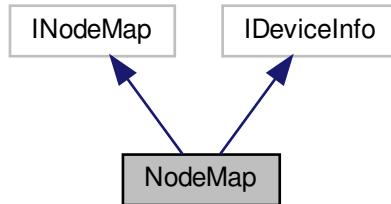
10.141 NodeMap Class Reference

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](#).

10.141.1 Detailed Description

Smart pointer template for NodeMaps with create function.

Parameters

TCameraParams	The camera specific parameter class (auto generated from camera xml file)
-------------------------------	---

10.141.2 Constructor & Destructor Documentation

10.141.2.1 NodeMap()

```
NodeMap (   
    GenICam::gcstring DeviceName = "Device" )
```

Constructor.

10.141.2.2 ~NodeMap()

```
virtual ~NodeMap ( ) [virtual]
```

Destructor.

10.141.3 Member Function Documentation**10.141.3.1 ClearXMLCache()**

```
static bool ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

10.141.3.2 Connect() [1/2]

```
virtual bool Connect (   
    IPort * pPort,   
    const GenICam::gcstring & PortName ) const [virtual]
```

Connects a port to a port node with given name.

10.141.3.3 Connect() [2/2]

```
virtual bool Connect (   
    IPort * pPort ) const [virtual]
```

Connects a port to the standard port "Device".

10.141.3.4 Destroy()

```
void Destroy ( )
```

Destroys the node map.

10.141.3.5 GetDeviceName()

```
virtual GenICam::gcstring GetDeviceName ( ) [virtual]
```

Get device name.

10.141.3.6 GetDeviceVersion()

```
virtual void GetDeviceVersion ( GenICam::Version\_t & Version ) [virtual]
```

Get the version of the device description file.

10.141.3.7 GetGenApiVersion()

```
virtual void GetGenApiVersion ( GenICam::Version\_t & Version,
                               uint16_t & Build ) [virtual]
```

Get the version of the DLL's [GenApi](#) implementation.

10.141.3.8 GetLock()

```
virtual CLock& GetLock ( ) const [virtual]
```

Returns the lock which guards the node map.

10.141.3.9 GetModelName()

```
virtual GenICam::gcstring GetModelName ( ) [virtual]
```

Get the model name.

10.141.3.10 GetNode()

```
virtual INode* GetNode (
    const GenICam::gcstring & key ) const [virtual]
```

Retrieves the node from the central map by name.

10.141.3.11 GetNodeMapHandle()

```
void* GetNodeMapHandle ( ) const
```

10.141.3.12 GetNodes()

```
virtual void GetNodes (
    NodeList_t & Nodes ) const [virtual]
```

Retrieves all nodes in the node map.

10.141.3.13 GetNumNodes()

```
virtual uint64_t GetNumNodes ( ) const [virtual]
```

Get the number of nodes in the map.

10.141.3.14 GetProductGuid()

```
virtual GenICam::gcstring GetProductGuid ( ) [virtual]
```

Get the GUID describing the product.

10.141.3.15 GetSchemaVersion()

```
virtual void GetSchemaVersion (
    GenICam::Version_t & Version ) [virtual]
```

Get the schema version number.

10.141.3.16 GetStandardNameSpace()

```
virtual GenICam::gcstring GetStandardNameSpace ( ) [virtual]
```

Get the standard name space.

10.141.3.17 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}" were {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

10.141.3.18 GetToolTip()

```
virtual GenICam::gcstring GetToolTip ( ) [virtual]
```

Get tool tip.

10.141.3.19 GetVendorName()

```
virtual GenICam::gcstring GetVendorName ( ) [virtual]
```

Get the vendor name.

10.141.3.20 GetVersionGuid()

```
virtual GenICam::gcstring GetVersionGuid ( ) [virtual]
```

Get the GUID describing the product version.

10.141.3.21 InvalidateNodes()

```
virtual void InvalidateNodes ( ) const [virtual]
```

Invalidates all nodes.

10.141.3.22 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*);

10.141.3.23 LoadXMLFromFileInject()

```
void LoadXMLFromFileInject ( GenICam::gcstring TargetFileName,  
                            GenICam::gcstring InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

10.141.3.24 LoadXMLFromString()

```
void LoadXMLFromString ( const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

10.141.3.25 LoadXMLFromStringInject()

```
void LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

10.141.3.26 LoadXMLFromZIPData()

```
void LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

10.141.3.27 LoadXMLFromZIPFile()

```
void LoadXMLFromZIPFile (
    GenICam::gcstring ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

10.141.3.28 Poll()

```
virtual void Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

10.141.4 Member Data Documentation

10.141.4.1 _Ptr

```
INodeMap* _Ptr
```

Pointer to the [NodeMap](#).

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

- include/SpinGenApi/[NodeMap.h](#)

10.142 CNodeMapFactory::NodeStatistics_t Struct Reference

Public Attributes

- `uint32_t NumNodes`
- `uint32_t NumProperties`
- `uint32_t NumLinks`
- `uint32_t NumStrings`

10.142.1 Member Data Documentation

10.142.1.1 NumLinks

```
uint32_t NumLinks
```

10.142.1.2 NumNodes

```
uint32_t NumNodes
```

10.142.1.3 NumProperties

```
uint32_t NumProperties
```

10.142.1.4 NumStrings

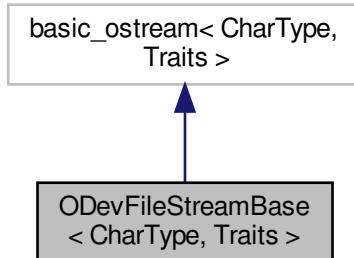
```
uint32_t NumStrings
```

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

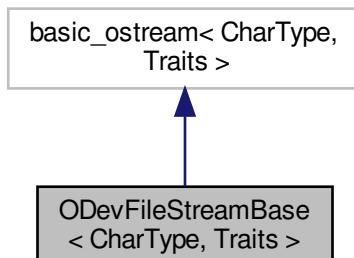
- `include/SpinGenApi/NodeMapFactory.h`

10.143 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`
Open file on device in write mode.
- `bool is_open () const`
- `void open (INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::out|std::ios_base::trunc)`
Close the file on device.
- `void close ()`

10.143.1 Member Typedef Documentation

10.143.1.1 filebuf_type

```
typedef ODevFileStreamBuf<CharType, Traits> filebuf_type
```

10.143.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

10.143.1.3 ostream_type

```
typedef std::basic_ostream<CharType, Traits> ostream_type
```

10.143.2 Member Function Documentation

10.143.2.1 close()

```
void close () [inline]
```

Close the file on device.

10.143.2.2 is_open()

```
bool is_open () const [inline]
```

10.143.2.3 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

10.143.2.4 rdbuf()

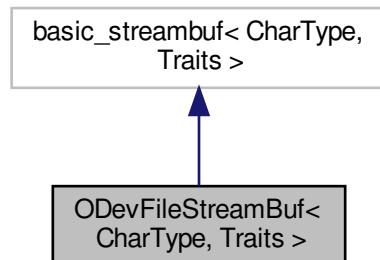
```
filebuf\_type* rdbuf ( ) const [inline]
```

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

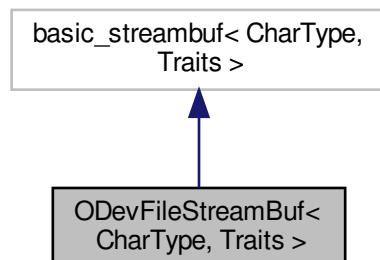
- include/SpinGenApi/[Filestream.h](#)

10.144 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 xputn (const char_type *s, std::streamsize n)`
- `int_type overflow (int_type c=traits_type::eof())`
- `int sync ()`

10.144.1 Constructor & Destructor Documentation

10.144.1.1 ODevFileStreamBuf()

```
ODevFileStreamBuf ( ) [inline]
```

10.144.1.2 ~ODevFileStreamBuf()

```
~ODevFileStreamBuf ( ) [inline]
```

10.144.2 Member Function Documentation

10.144.2.1 close()

```
filebuf_type* close ( ) [inline]
```

10.144.2.2 is_open()

```
bool is_open ( ) const [inline]
```

10.144.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode ) [inline]
```

10.144.2.4 overflow()

```
int_type overflow (
    int_type c = traits_type::eof() ) [inline], [protected]
```

10.144.2.5 sync()

```
int sync ( ) [inline], [protected]
```

10.144.2.6 xsputn()

```
std::streamsize xsputn (
    const char_type * s,
    std::streamsize n ) [inline], [protected]
```

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

- include/SpinGenApi/Filestream.h

10.145 PGMOOption Struct Reference

Options for saving PGM images.

Public Member Functions

- [PGMOOption \(\)](#)

Public Attributes

- bool [binaryFile](#)
Whether to save the PPM as a binary file.
- unsigned int [reserved \[16\]](#)
Reserved for future use.

10.145.1 Detailed Description

Options for saving PGM images.

10.145.2 Constructor & Destructor Documentation

10.145.2.1 PGMOption()

```
PGMOption ( ) [inline]
```

10.145.3 Member Data Documentation

10.145.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

10.145.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

10.146 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.

10.146.1 Detailed Description

Options for saving PNG images.

10.146.2 Constructor & Destructor Documentation

10.146.2.1 `PNGOption()`

`PNGOption ()` [inline]

10.146.3 Member Data Documentation

10.146.3.1 `compressionLevel`

`unsigned int compressionLevel`

Compression level (0-9).

0 is no compression, 9 is best compression.

10.146.3.2 `interlaced`

`bool interlaced`

Whether to save the PNG as interlaced.

10.146.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

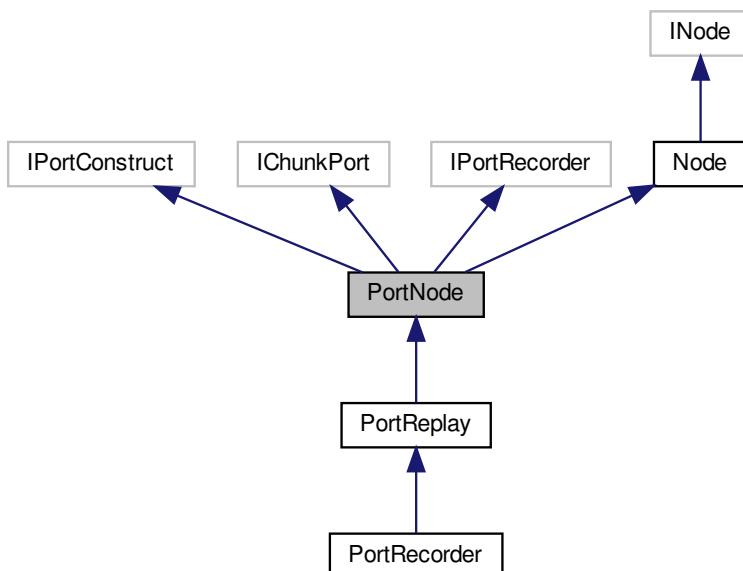
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

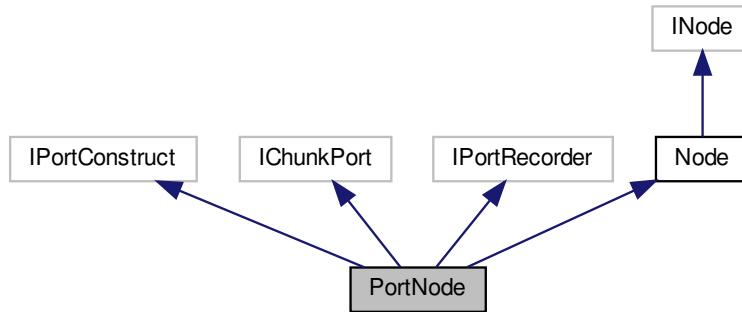
10.147 PortNode Class Reference

[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 GetSwapEndianess ()**
Determines if the port adapter must perform an endianess 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

10.147.1 Detailed Description

[Interface](#) for value properties.

10.147.2 Constructor & Destructor Documentation

10.147.2.1 PortNode() [1/2]

```
PortNode ( )
```

Constructor.

10.147.2.2 PortNode() [2/2]

```
PortNode (
    std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenICam IValue](#)

10.147.2.3 ~PortNode()

```
~PortNode ( )
```

Destructor.

10.147.3 Member Function Documentation

10.147.3.1 CacheChunkData()

```
virtual EYesNo CacheChunkData ( ) const [virtual]
```

Indicates if the chunk a adapter must hold a cached version of the chunk data.

10.147.3.2 GetChunkID()

```
virtual Spinnaker::GenICam::gcstring GetChunkID () const [virtual]
```

Get the Id of the chunk the port should be attached to.

10.147.3.3 GetPortHandle()

```
std::shared_ptr<Node::NodeImpl> GetPortHandle () [inline]
```

10.147.3.4 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess () [virtual]
```

Determines if the port adapter must perform an endianess swap.

10.147.3.5 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

10.147.3.6 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 [PortReplay](#).

10.147.3.7 SetPortImpl()

```
void SetPortImpl (
    IPort * pPort )
```

Sets pointer the real port implementation; this function may called only once.

10.147.3.8 SetReference() [1/3]

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [Node](#).

10.147.3.9 SetReference() [2/3]

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented in [PortRecorder](#), and [PortReplay](#).

10.147.3.10 SetReference() [3/3]

```
virtual void SetReference (
    IChunkPort * pBase ) [virtual]
```

overload SetReference for Value

10.147.3.11 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

Starts logging all WriteRegister commands to a list.

Reimplemented in [PortRecorder](#).

10.147.3.12 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

Stops recording.

Reimplemented in [PortRecorder](#).

10.147.3.13 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

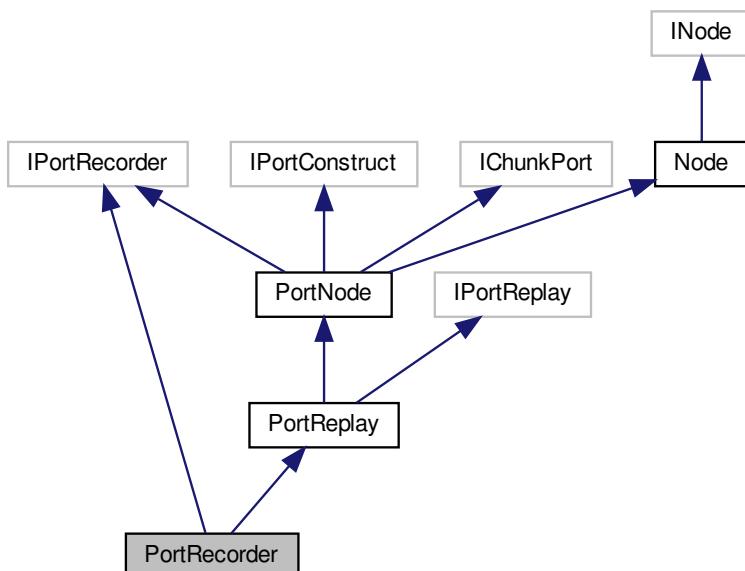
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortNode.h](#)

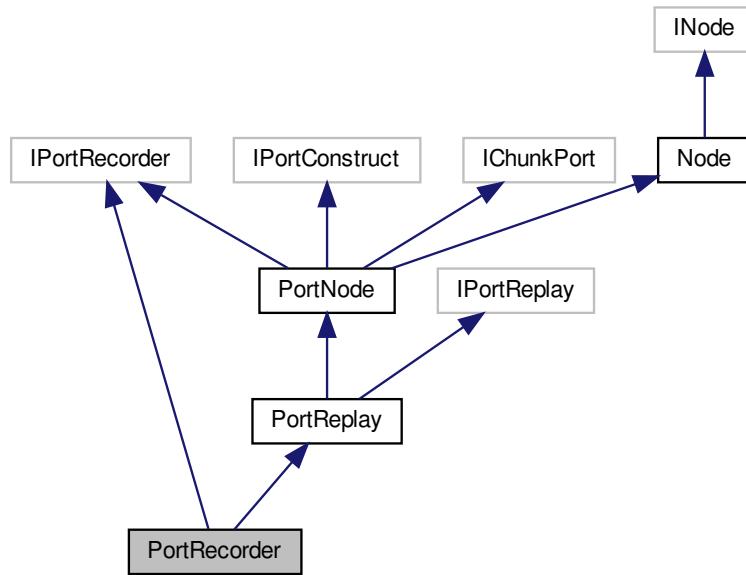
10.148 PortRecorder Class Reference

[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 `SetReference (IPort *pBase)`
overload SetReference for Value

Additional Inherited Members

10.148.1 Detailed Description

[Interface](#) for recording write commands on a port.

10.148.2 Constructor & Destructor Documentation

10.148.2.1 PortRecorder()

```
PortRecorder ( )
```

10.148.2.2 ~PortRecorder()

```
virtual ~PortRecorder ( ) [virtual]
```

10.148.3 Member Function Documentation

10.148.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

Reimplemented from [Node](#).

10.148.3.2 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortReply](#).

10.148.3.3 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

starts logging all WriteRegister commands to a list

Reimplemented from [PortNode](#).

10.148.3.4 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

stops recording

Reimplemented from [PortNode](#).

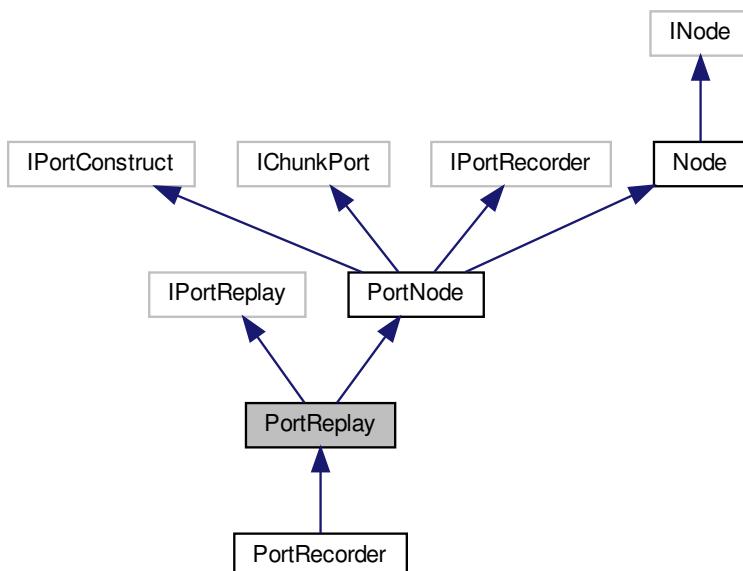
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortRecorder.h](#)

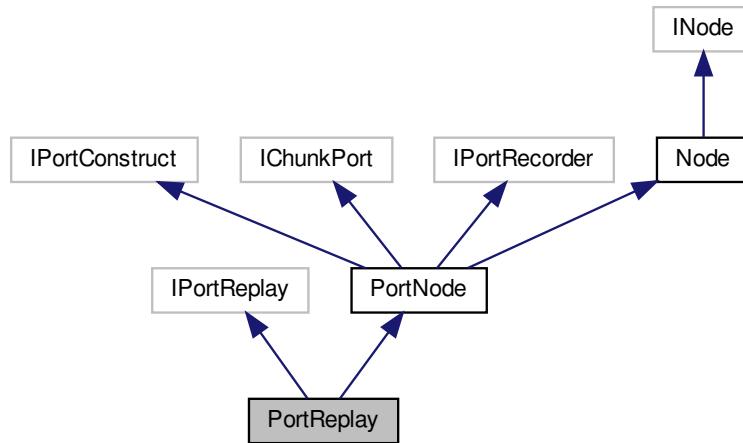
10.149 PortReplay Class Reference

[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 ()`

Additional Inherited Members

10.149.1 Detailed Description

[Interface](#) for replaying write commands on a port.

10.149.2 Constructor & Destructor Documentation

10.149.2.1 PortReplay()

`PortReplay ()`

10.149.2.2 ~PortReplay()

```
virtual ~PortReplay( ) [virtual]
```

10.149.3 Member Function Documentation

10.149.3.1 GetPortReplayHandle()

```
void* GetPortReplayHandle( )
```

10.149.3.2 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](#).

10.149.3.3 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

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

- include/SpinGenApi/[PortReplay.h](#)

10.150 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.

10.150.1 Detailed Description

Options for saving PPM images.

10.150.2 Constructor & Destructor Documentation

10.150.2.1 [PPMOption\(\)](#)

[PPMOption \(\)](#) [inline]

10.150.3 Member Data Documentation

10.150.3.1 [binaryFile](#)

bool [binaryFile](#)

Whether to save the PPM as a binary file.

10.150.3.2 [reserved](#)

unsigned int [reserved\[16\]](#)

Reserved for future use.

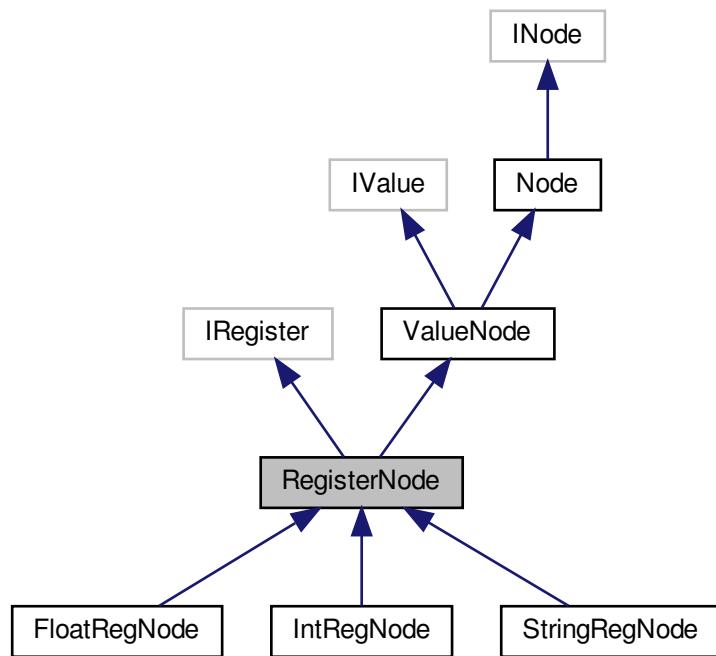
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

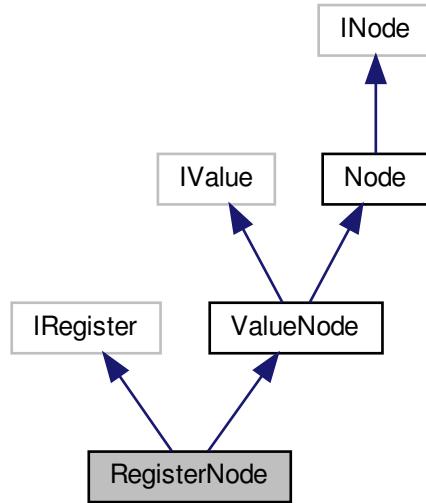
10.151 RegisterNode Class Reference

[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

10.151.1 Detailed Description

[Interface](#) for string properties.

10.151.2 Constructor & Destructor Documentation

10.151.2.1 RegisterNode() [1/2]

```
RegisterNode ( )
```

10.151.2.2 RegisterNode() [2/2]

```
RegisterNode (
    std::shared_ptr< Node::NodeImpl > pRegister )
```

10.151.2.3 ~RegisterNode()

```
virtual ~RegisterNode ( ) [virtual]
```

10.151.3 Member Function Documentation**10.151.3.1 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

10.151.3.2 GetAddress()

```
virtual int64_t GetAddress ( ) [virtual]
```

Retrieves the Address of the register.

10.151.3.3 GetLength()

```
virtual int64_t GetLength ( ) [virtual]
```

Retrieves the Length of the register [Bytes].

10.151.3.4 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)

10.151.3.5 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Register

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

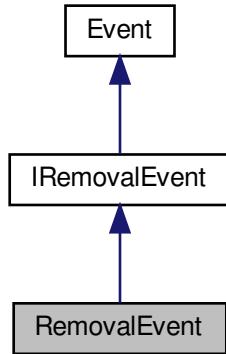
The documentation for this class was generated from the following file:

- include/SpinGenApi/[RegisterNode.h](#)

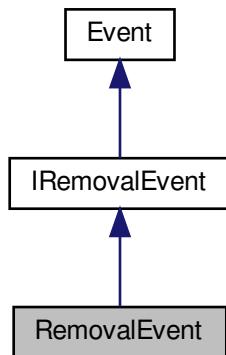
10.152 RemovalEvent Class Reference

An event handler for capturing the device removal event.

Inheritance diagram for RemovalEvent:



Collaboration diagram for RemovalEvent:



Public Member Functions

- [RemovalEvent \(\)](#)
Default Constructor.
- [virtual ~RemovalEvent \(\)](#)
Virtual Destructor.
- [virtual void OnDeviceRemoval \(uint64_t serialNumber\)=0](#)
Device removal event callback.

Protected Member Functions

- `RemovalEvent & operator= (const RemovalEvent &)`
Assignment operator.

Additional Inherited Members

10.152.1 Detailed Description

An event handler for capturing the device removal event.

10.152.2 Constructor & Destructor Documentation

10.152.2.1 RemovalEvent()

`RemovalEvent ()`

Default Constructor.

10.152.2.2 ~RemovalEvent()

`virtual ~RemovalEvent () [virtual]`

Virtual Destructor.

10.152.3 Member Function Documentation

10.152.3.1 OnDeviceRemoval()

`virtual void OnDeviceRemoval (`
`uint64_t serialNumber) [pure virtual]`

Device removal event callback.

Parameters

<code>serialNumber</code>	The serial number of the device removed
---------------------------	---

Implements [IRemovalEvent](#).

10.152.3.2 operator=()

```
RemovalEvent& operator= (
    const RemovalEvent & ) [protected]
```

Assignment operator.

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

- [include/RemovalEvent.h](#)

10.153 SingleChunkData_t Struct Reference

Public Attributes

- `uint64_t ChunkID`
- `ptrdiff_t ChunkOffset`
- `size_t ChunkLength`

10.153.1 Member Data Documentation

10.153.1.1 ChunkID

```
uint64_t ChunkID
```

10.153.1.2 ChunkLength

```
size_t ChunkLength
```

10.153.1.3 ChunkOffset

```
ptrdiff_t ChunkOffset
```

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

- [include/SpinGenApi/ChunkAdapterGeneric.h](#)

10.154 SingleChunkDataStr_t Struct Reference

Public Attributes

- GenICam::gcstring [ChunkID](#)
- ptrdiff_t [ChunkOffset](#)
- size_t [ChunkLength](#)

10.154.1 Member Data Documentation

10.154.1.1 [ChunkID](#)

GenICam::gcstring [ChunkID](#)

10.154.1.2 [ChunkLength](#)

size_t [ChunkLength](#)

10.154.1.3 [ChunkOffset](#)

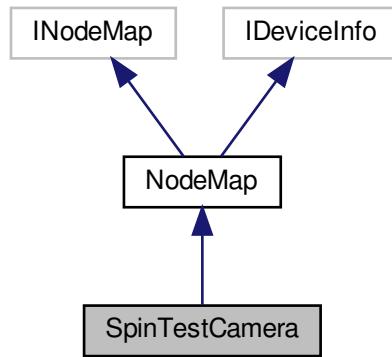
ptrdiff_t [ChunkOffset](#)

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

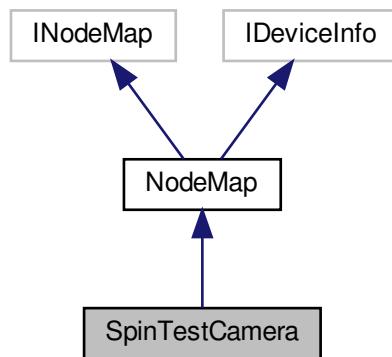
- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

10.155 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](#)

10.156 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 JPEG 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 plImage)`**
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.

10.156.1 Detailed Description

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

10.156.2 Constructor & Destructor Documentation

10.156.2.1 `SpinVideo()`

`SpinVideo ()`

Default constructor.

10.156.2.2 `~SpinVideo()`

`virtual ~SpinVideo () [virtual]`

Default destructor.

10.156.3 Member Function Documentation

10.156.3.1 `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.
---------------	----------------------

10.156.3.2 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\(ImagePtr pImage\)](#)

10.156.3.3 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\(\)](#)

10.156.3.4 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](#)

10.156.3.5 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](#)

10.156.3.6 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\(ImagePtr plImage\)](#)

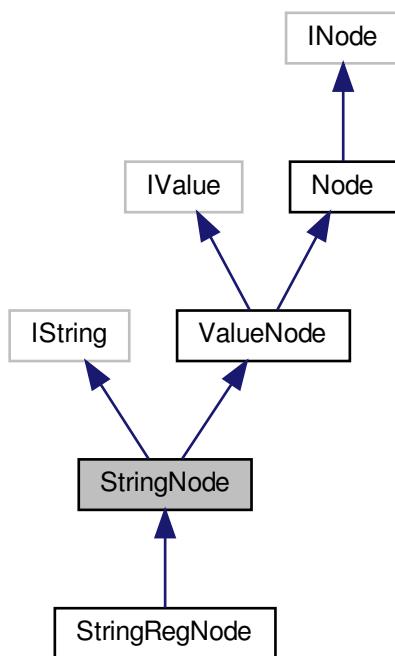
The documentation for this class was generated from the following file:

- [include/SpinVideo.h](#)

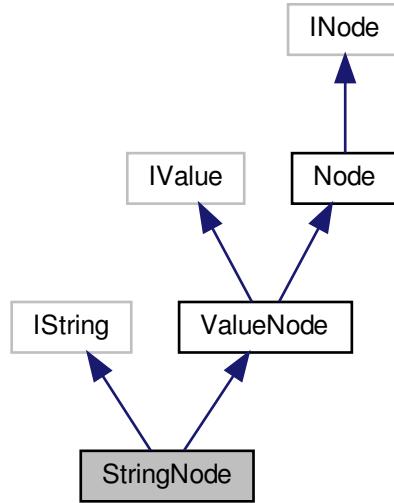
10.157 StringNode Class Reference

[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 GenICam::gcstring &Value, bool Verify=true)`
Set node value.
- `virtual IString & operator= (const GenICam::gcstring &Value)`
Set node value.
- `virtual GenICam::gcstring GetValue (bool Verify=false, bool IgnoreCache=false)`
Get node value.
- `virtual GenICam::gcstring operator() ()`
Get node value.
- `virtual GenICam::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

10.157.1 Detailed Description

[Interface](#) for string properties.

10.157.2 Constructor & Destructor Documentation

10.157.2.1 `StringNode()` [1/2]

```
StringNode ( )
```

10.157.2.2 `StringNode()` [2/2]

```
StringNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

10.157.2.3 `~StringNode()`

```
virtual ~StringNode ( ) [virtual]
```

10.157.3 Member Function Documentation

10.157.3.1 `GetMaxLength()`

```
virtual int64_t GetMaxLength ( ) [virtual]
```

Retrieves the maximum length of the string in bytes.

10.157.3.2 `GetValue()`

```
virtual GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<code>Verify</code>	Enables Range verification (default = false). The AccessMode is always checked
<code>IgnoreCache</code>	If true the value is read ignoring any caches (default = false)

Returns

The value read

10.157.3.3 operator()

```
virtual GenICam::gcstring operator() () [virtual]
```

Get node value.

10.157.3.4 operator*()

```
virtual GenICam::gcstring operator* () [virtual]
```

Get node value.

10.157.3.5 operator=()

```
virtual IString& operator= (
    const GenICam::gcstring & Value ) [virtual]
```

Set node value.

10.157.3.6 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#).

10.157.3.7 SetValue()

```
virtual void SetValue (
    const GenICam::gcstring & Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

Value	The value to set
Verify	Enables AccessMode and Range verification (default = true)

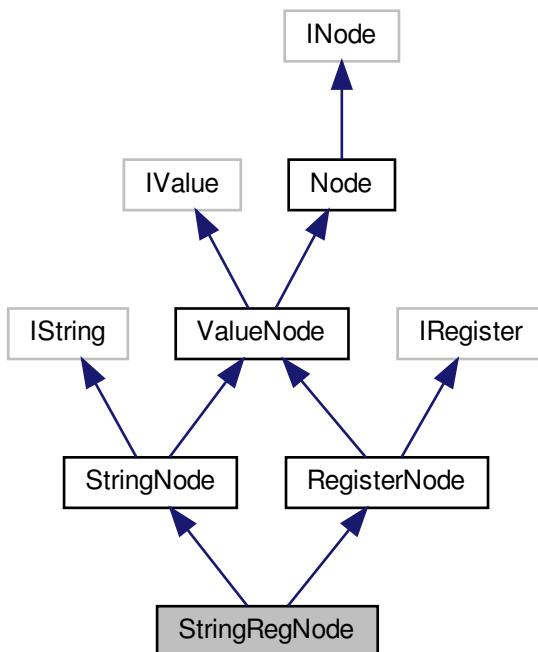
The documentation for this class was generated from the following file:

- include/SpinGenApi/StringNode.h

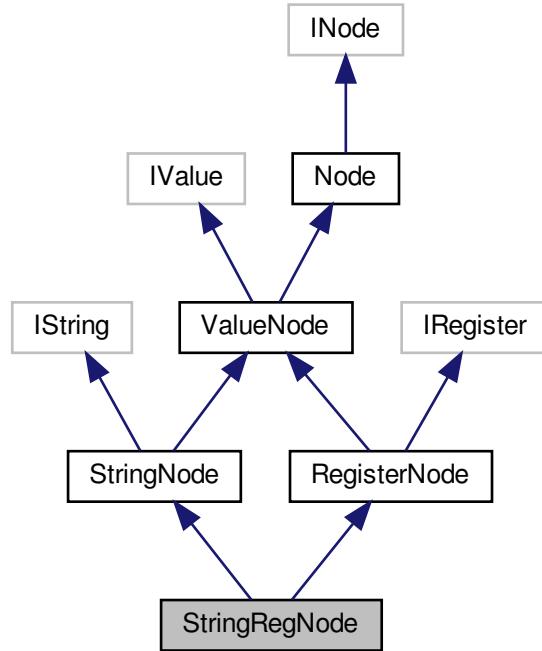
10.158 StringRegNode Class Reference

[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

10.158.1 Detailed Description

[Interface](#) for string properties.

10.158.2 Constructor & Destructor Documentation

10.158.2.1 StringRegNode() [1/2]

```
StringRegNode ( )
```

10.158.2.2 StringRegNode() [2/2]

```
StringRegNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

10.158.2.3 ~StringRegNode()

```
virtual ~StringRegNode ( ) [virtual]
```

10.158.3 Member Function Documentation**10.158.3.1 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [RegisterNode](#).

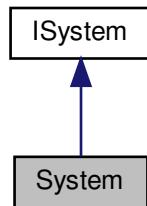
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringRegNode.h](#)

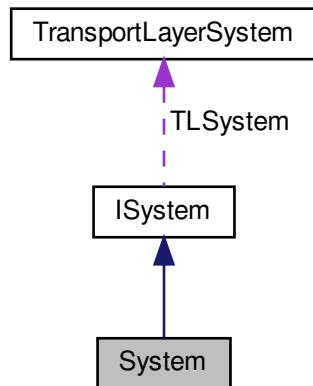
10.159 System Class Reference

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 `ReleaselInstance ()`
This call releases the instance of the `System` Singleton for this process.
- virtual `InterfaceList GetInterfaces (bool updateInterface=true)`
Returns a list of interfaces available 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.
- virtual void [UpdateInterfaceList](#) ()
Updates the list of interfaces on the system.
- void [RegisterEvent](#) (Event &evtToRegister)
Registers an event for the system.
- void [UnregisterEvent](#) (Event &evtToUnregister)
Unregisters an event for the system.
- virtual void [RegisterInterfaceEvent](#) (Event &evtToRegister, bool updateInterface=true)
Registers events for all available interfaces that are found on the system.
- void [UnregisterInterfaceEvent](#) (Event &evtToUnregister)
Unregisters events for all available interfaces that are found on the system.
- virtual void [RegisterLoggingEvent](#) (LoggingEvent &handler)
Registers a logging event.
- virtual void [UnregisterAllLoggingEvent](#) ()
Unregisters all previously registered logging events.
- virtual void [UnregisterLoggingEvent](#) (LoggingEvent &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

10.159.1 Detailed Description

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

10.159.2 Constructor & Destructor Documentation

10.159.2.1 ~System()

```
virtual ~System ( ) [virtual]
```

Default destructor.

10.159.2.2 System()

```
System ( ) [protected]
```

Default constructor.

10.159.3 Member Function Documentation

10.159.3.1 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](#).

10.159.3.2 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.

10.159.3.3 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

<code>updateInterface</code>	Determines whether or not UpdateInterfaceList() is called before getting available interfaces
------------------------------	---

Returns

An [InterfaceList](#) object that contains a list of all interfaces.

Implements [ISystem](#).

10.159.3.4 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](#).

10.159.3.5 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](#).

10.159.3.6 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](#).

10.159.3.7 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](#).

10.159.3.8 RegisterEvent()

```
void RegisterEvent (
    Event & evtToRegister ) [virtual]
```

Registers an event for the system.

Parameters

<code>evtToRegister</code>	The event to register for the system
----------------------------	--------------------------------------

Implements [ISystem](#).

10.159.3.9 RegisterInterfaceEvent()

```
virtual void RegisterInterfaceEvent (
    Event & evtToRegister,
    bool updateInterface = true ) [virtual]
```

Registers events for all available interfaces that are found on the system.

Parameters

<code>evtToRegister</code>	The event to register for the available interfaces
<code>updateInterface</code>	Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system

Implements [ISystem](#).

10.159.3.10 RegisterLoggingEvent()

```
virtual void RegisterLoggingEvent (
    LoggingEvent & handler ) [virtual]
```

Registers a logging event.

Parameters

<i>handler</i>	The logging event handler to register
----------------	---------------------------------------

Implements [ISystem](#).

10.159.3.11 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [virtual]
```

This call releases the instance of the [System](#) Singleton for this process.

After successfully releasing the [System](#) instance the pointer returned by [GetInstance\(\)](#) will be invalid. Calling [ReleaseInstance](#) while a camera reference is still held will throw an error of type SPINNAKER_ERR_RESOU↔RCE_IN_USE.

See also

Error
GetInstance()

Implements [ISystem](#).

10.159.3.12 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.

Parameters

<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](#).

10.159.3.13 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](#).

10.159.3.14 UnregisterAllLoggingEvent()

```
virtual void UnregisterAllLoggingEvent ( ) [virtual]
```

Unregisters all previously registered logging events.

Implements [ISystem](#).

10.159.3.15 UnregisterEvent()

```
void UnregisterEvent (
    Event & evtToUnregister ) [virtual]
```

Unregisters an event for the system.

Parameters

<code>evtToUnregister</code>	The event to unregister from the system
------------------------------	---

Implements [ISystem](#).

10.159.3.16 UnregisterInterfaceEvent()

```
void UnregisterInterfaceEvent (
    Event & evtToUnregister ) [virtual]
```

Unregisters events for all available interfaces that are found on the system.

Parameters

<code>evtToUnregister</code>	The event to unregister from the available interfaces
------------------------------	---

Implements [ISystem](#).

10.159.3.17 UnregisterLoggingEvent()

```
virtual void UnregisterLoggingEvent (
    LoggingEvent & handler ) [virtual]
```

Unregisters a logging event.

Parameters

<i>handler</i>	The logging event handler to unregister
----------------	---

Implements [ISystem](#).

10.159.3.18 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](#).

10.159.3.19 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](#).

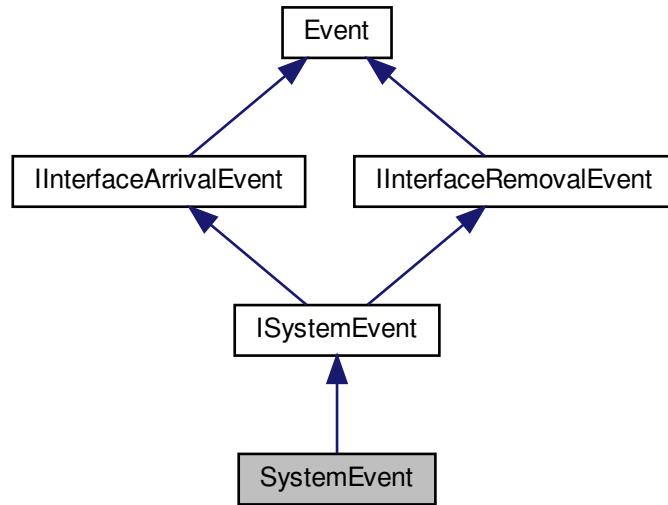
The documentation for this class was generated from the following file:

- [include/System.h](#)

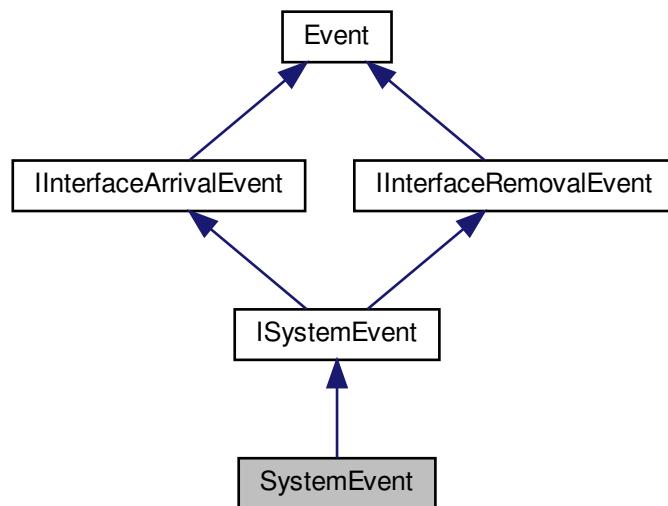
10.160 SystemEvent Class Reference

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

Inheritance diagram for SystemEvent:



Collaboration diagram for SystemEvent:



Public Member Functions

- `SystemEvent ()`
Default constructor.
- `virtual ~SystemEvent ()`
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

- `SystemEvent & operator= (const SystemEvent &)`
Assignment operator.

Additional Inherited Members

10.160.1 Detailed Description

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

10.160.2 Constructor & Destructor Documentation

10.160.2.1 SystemEvent()

`SystemEvent ()`

Default constructor.

10.160.2.2 ~SystemEvent()

`virtual ~SystemEvent () [virtual]`

Virtual destructor.

10.160.3 Member Function Documentation

10.160.3.1 OnInterfaceArrival()

`virtual void OnInterfaceArrival (`
`std::string interfaceID) [pure virtual]`

Interface arrival event callback.

Parameters

<i>interfaceID</i>	The ID of the arrived interface
--------------------	---------------------------------

Implements [ISystemEvent](#).

10.160.3.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

[Interface](#) removal event callback.

Parameters

<i>interfaceID</i>	The ID of the removed interface
--------------------	---------------------------------

Implements [ISystemEvent](#).

10.160.3.3 operator=()

```
SystemEvent& operator=
    const SystemEvent & ) [protected]
```

Assignment operator.

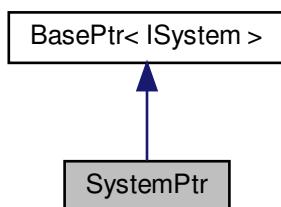
The documentation for this class was generated from the following file:

- [include/SystemEvent.h](#)

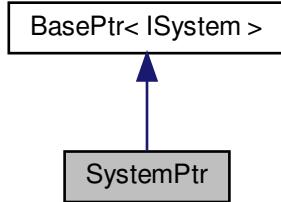
10.161 SystemPtr Class Reference

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

10.161.1 Detailed Description

A reference tracked pointer to a system object.

10.161.2 Constructor & Destructor Documentation

10.161.2.1 [SystemPtr\(\)](#) [1/4]

[SystemPtr \(\)](#)

Default constructor.

10.161.2.2 SystemPtr() [2/4]

```
SystemPtr (
    const int )
```

Copy constructor.

10.161.2.3 SystemPtr() [3/4]

```
SystemPtr (
    const long )
```

Copy constructor.

10.161.2.4 SystemPtr() [4/4]

```
SystemPtr (
    const std::nullptr_t )
```

Copy constructor.

10.161.2.5 ~SystemPtr()

```
virtual ~SystemPtr (
    void ) [virtual]
```

Virtual destructor.

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

- [include/SystemPtr.h](#)

10.162 TIFFOption Struct Reference

Options for saving TIFF images.

Public Types

- enum [CompressionMethod](#) {
 NONE = 1,
 PACKBITS,
 DEFLATE,
 ADOBE_DEFLATE,
 CCITTFAZ3,
 CCITTFAZ4,
 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.

10.162.1 Detailed Description

Options for saving TIFF images.

10.162.2 Member Enumeration Documentation

10.162.2.1 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.
CCITTFAEX3	Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
CCITTFAEX4	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.

10.162.3 Constructor & Destructor Documentation

10.162.3.1 TIFFOption()

[TIFFOption \(\)](#) [inline]

10.162.4 Member Data Documentation

10.162.4.1 compression

```
CompressionMethod compression
```

Compression method to use for encoding TIFF images.

10.162.4.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

10.163 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](#)

- **GenApi::IString & DeviceVersion**
Description: Gets the access status the transport layer Producer has on the device.
- **GenApi::IString & DeviceUserID**
Description: Version of the device.
- **GenApi::IString & DeviceName**
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::IInteger & GevDeviceIPAddress**
Description: Current IP address of the GVCP interface of the selected remote device.
- **GenApi::IInteger & GevDeviceSubnetMask**
Description: Current subnet mask of the GVCP interface of the selected remote device.
- **GenApi::IInteger & GevDeviceMACAddress**
Description: 48-bit MAC address of the GVCP interface of the selected remote device.
- **GenApi::IInteger & GevDeviceGateway**
Description: Current gateway IP address of the GVCP interface of the remote device.
- **GenApi::IInteger & DeviceLinkSpeed**
Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.
- **GenApi::IInteger & GevVersionMajor**
Description: Major version of the specification.
- **GenApi::IInteger & GevVersionMinor**
Description: Minor version of the specification.
- **GenApi::IBoolean & GevDeviceModelsBigEndian**
Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).
- **GenApi::IInteger & GevDeviceReadAndWriteTimeout**
Description: The timeout in us for read/write operations to the camera.
- **GenApi::IInteger & GevDeviceMaximumRetryCount**
Description: Maximum number of times to retry a read/write operation.
- **GenApi::IInteger & GevDevicePort**
Description: Current IP port of the GVCP interface of the selected remote device.
- **GenApi::ICommand & GevDeviceDiscoverMaximumPacketSize**
Description: Discovers and updates the maximum packet size that can be safely used by the device on the current interface.
- **GenApi::IInteger & GevDeviceMaximumPacketSize**
Description: The maximum packet size that can be safely used by the device on the current interface.
- **GenApi::IBoolean & GevDeviceIsWrongSubnet**
Description: Indicates whether the device is on the wrong subnet.
- **GenApi::ICommand & GevDeviceForceIP**
Description: Forces the camera to be on the same subnet as its corresponding interface.

- [GenApi::ICommand & GevDeviceForceIPEx](#)
Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDeviceForceGateway) in the Device using ForceIP command.
- [GenApi::IInteger & GevDeviceForceIPAddress](#)
Description: Static IP address to set for the GVCP interface of the remote device.
- [GenApi::IInteger & GevDeviceForceSubnetMask](#)
Description: Static subnet mask to set for GVCP interface of the remote device.
- [GenApi::IInteger & GevDeviceForceGateway](#)
Description: Static gateway IP address to set for the GVCP interface of the remote device.
- [GenApi::IBoolean & DeviceMulticastMonitorMode](#)
Description: Controls and indicates if the device is operating in as a Multicast Monitor.
- [GenApi::IEnumerationT< DeviceEndianessMechanismEnum > & DeviceEndianessMechanism](#)
Description: Identifies the endianness handling mode.
- [GenApi::IString & DeviceInstanceId](#)
Description: Visibility: Invisible.
- [GenApi::IString & DeviceLocation](#)
Description: Device Location.
- [GenApi::IEnumerationT< DeviceCurrentSpeedEnum > & DeviceCurrentSpeed](#)
Description: The USB Speed that the device is currently operating at.
- [GenApi::IBoolean & DeviceU3VProtocol](#)
Description: Indicates whether the device is communicating in U3V Protocol.

Protected Member Functions

- [TransportLayerDevice \(\)](#)

Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameralInternal](#)

10.163.1 Detailed Description

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

10.163.2 Constructor & Destructor Documentation

10.163.2.1 TransportLayerDevice() [1/2]

```
TransportLayerDevice (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.163.2.2 ~TransportLayerDevice()

```
~TransportLayerDevice ( )
```

10.163.2.3 TransportLayerDevice() [2/2]

```
TransportLayerDevice ( ) [protected]
```

10.163.3 Friends And Related Function Documentation**10.163.3.1 CameraBase**

```
friend class CameraBase [friend]
```

10.163.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

10.163.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

10.163.4 Member Data Documentation**10.163.4.1 DeviceAccessStatus**

```
GenApi::IEnumeration<DeviceAccessStatusEnum>& DeviceAccessStatus
```

Description: Gets the access status the transport layer Producer has on the device.

Visibility: Beginner

10.163.4.2 DeviceCurrentSpeed

```
GenApi::IEnumerationT<DeviceCurrentSpeedEnum>& DeviceCurrentSpeed
```

Description: The USB Speed that the device is currently operating at.

Visibility: Expert

10.163.4.3 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

10.163.4.4 DeviceDriverVersion

```
GenApi::IString& DeviceDriverVersion
```

Description: Version of the device driver.

Visibility: Expert

10.163.4.5 DeviceEndianessMechanism

```
GenApi::IEnumerationT<DeviceEndianessMechanismEnum>& DeviceEndianessMechanism
```

Description: Identifies the endianness handling mode.

Visibility: Expert

10.163.4.6 DeviceID

```
GenApi::IString& DeviceID
```

Description: Interface-wide unique identifier of this device.

Visibility: Expert

10.163.4.7 DeviceInstanceId

```
GenApi::IString& DeviceInstanceId
```

Description: Visibility: Invisible.

10.163.4.8 DeviceIsUpdater

`GenApi::IBoolean& DeviceIsUpdater`

Description: Indicates whether the device is in updater mode.

Visibility: Expert

10.163.4.9 DeviceLinkSpeed

`GenApi::IInteger& DeviceLinkSpeed`

Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.

Visibility: Expert

10.163.4.10 DeviceLocation

`GenApi::IString& DeviceLocation`

Description: Device Location.

Visibility: Expert

10.163.4.11 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the remote device model.

Visibility: Beginner

10.163.4.12 DeviceMulticastMonitorMode

`GenApi::IBoolean& DeviceMulticastMonitorMode`

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

Visibility: Expert

10.163.4.13 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description: Serial number of the remote device.

Visibility: Expert

10.163.4.14 DeviceType

```
GenApi::IEnumerationT<DeviceTypeEnum>& DeviceType
```

Description: Transport layer type of the device.

Visibility: Expert

10.163.4.15 DeviceU3VProtocol

```
GenApi::IBoolean& DeviceU3VProtocol
```

Description: Indicates whether the device is communicating in U3V Protocol.

Visibility: Expert

10.163.4.16 DeviceUserID

```
GenApi::IString& DeviceUserID
```

Description: User Defined Name.

Visibility: Expert

10.163.4.17 DeviceVendorName

```
GenApi::IString& DeviceVendorName
```

Description: Name of the remote device vendor.

Visibility: Beginner

10.163.4.18 DeviceVersion

```
GenApi::IString& DeviceVersion
```

Description: Version of the device.

Visibility: Expert

10.163.4.19 GenICamXMLLocation

```
GenApi::IEnumerationT<GenICamXMLLocationEnum>& GenICamXMLLocation
```

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

Visibility: Beginner

10.163.4.20 GenICamXMLPath

`GenApi::IString& GenICamXMLPath`

Description: [GenICam](#) XML Path.

Visibility: Beginner

10.163.4.21 GevCCP

`GenApi::IEnumerationT<GevCCPEnum>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility: Beginner

10.163.4.22 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

10.163.4.23 GevDeviceForceGateway

`GenApi::IInteger& GevDeviceForceGateway`

Description: Static gateway IP address to set for the GVCP interface of the remote device.

Visibility: Expert

10.163.4.24 GevDeviceForceIP

`GenApi:: ICommand& GevDeviceForceIP`

Description: Forces the camera to be on the same subnet as its corresponding interface.

Visibility: Expert

10.163.4.25 GevDeviceForceIPAddress

`GenApi::IInteger& GevDeviceForceIPAddress`

Description: Static IP address to set for the GVCP interface of the remote device.

Visibility: Expert

10.163.4.26 GevDeviceForceIPEx

`GenApi:: ICommand& GevDeviceForceIPEx`

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDeviceForceGateway) in the Device using ForceIP command.

Visibility: Expert

10.163.4.27 GevDeviceForceSubnetMask

`GenApi:: IInteger& GevDeviceForceSubnetMask`

Description: Static subnet mask to set for GVCP interface of the remote device.

Visibility: Expert

10.163.4.28 GevDeviceGateway

`GenApi:: IInteger& GevDeviceGateway`

Description: Current gateway IP address of the GVCP interface of the remote device.

Visibility: Expert

10.163.4.29 GevDeviceIPAddress

`GenApi:: IInteger& GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

10.163.4.30 GevDeviceIsWrongSubnet

`GenApi:: IBoolean& GevDeviceIsWrongSubnet`

Description: Indicates whether the device is on the wrong subnet.

Visibility: Expert

10.163.4.31 GevDeviceMACAddress

`GenApi:: IInteger& GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

10.163.4.32 GevDeviceMaximumPacketSize

```
GenApi::IInteger& GevDeviceMaximumPacketSize
```

Description: The maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

10.163.4.33 GevDeviceMaximumRetryCount

```
GenApi::IInteger& GevDeviceMaximumRetryCount
```

Description: Maximum number of times to retry a read/write operation.

Visibility: Expert

10.163.4.34 GevDeviceModelsBigEndian

```
GenApi::IBoolean& GevDeviceModeIsBigEndian
```

Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).

Visibility: Expert

10.163.4.35 GevDevicePort

```
GenApi::IInteger& GevDevicePort
```

Description: Current IP port of the GVCP interface of the selected remote device.

Visibility: Expert

10.163.4.36 GevDeviceReadAndWriteTimeout

```
GenApi::IInteger& GevDeviceReadAndWriteTimeout
```

Description: The timeout in us for read/write operations to the camera.

Visibility: Expert

10.163.4.37 GevDeviceSubnetMask

```
GenApi::IInteger& GevDeviceSubnetMask
```

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

10.163.4.38 GevVersionMajor

`GenApi::IInteger& GevVersionMajor`

Description: Major version of the specification.

Visibility: Expert

10.163.4.39 GevVersionMinor

`GenApi::IInteger& GevVersionMinor`

Description: Minor version of the specification.

Visibility: Expert

10.163.4.40 GUIXMLLocation

`GenApi::IEnumerationT<GUIXMLLocationEnum>& GUIXMLLocation`

Description: Sets the location to load GUI XML.

Visibility: Beginner

10.163.4.41 GUIXMLPath

`GenApi::IString& GUIXMLPath`

Description: GUI XML Path.

Visibility: Beginner

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

- `include/TransportLayerDevice.h`

10.164 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::IString & InterfaceType**
Description: Transport layer type of the interface.
- **GenApi::IInteger & GevInterfaceGateway**
Description: IP address of the selected gateway entry of this interface.
- **GenApi::IInteger & GevInterfaceMACAddress**
Description: 48-bit MAC address of this interface.
- **GenApi::IInteger & GevInterfaceIPAddress**
Description: IP address of the selected subnet of this interface.
- **GenApi::IInteger & GevInterfaceSubnetMask**
Description: Subnet mask of the selected subnet of this interface.
- **GenApi::IInteger & GevInterfaceTransmitLinkSpeed**
Description: Transmit link speed of this interface in bits per second.
- **GenApi::IInteger & GevInterfaceReceiveLinkSpeed**
Description: Receive link speed of this interface in bits per second.
- **GenApi::IInteger & GevInterfaceMTU**
Description: Maximum transmission unit of this interface.
- **GenApi::IEnumerationT< POEStatusEnum > & POEStatus**
Description: Reports and controls the interface's power over Ethernet status.
- **GenApi::IEnumerationT< FilterDriverStatusEnum > & FilterDriverStatus**
Description: Reports whether FLIR Light Weight Filter Driver is enabled or not.
- **GenApi::IInteger & GevActionDeviceKey**
Description: Key to authorize the action for the device.
- **GenApi::IInteger & GevActionGroupKey**
Description: Provides the key that the device will use to validate the action on reception of the action protocol message.
- **GenApi::IInteger & GevActionGroupMask**
Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.
- **GenApi::IInteger & 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::IInteger & DeviceCount**
Description: Number of compatible devices detected on current interface.
- **GenApi::IInteger & 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::IEnumerationT< DeviceAccessStatusEnum > & DeviceAccessStatus](#)
 Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".
 - [GenApi::IInteger & GevDeviceIPAddress](#)
 Description: Current IP address of the GVCP interface of the selected remote device.
 - [GenApi::IInteger & GevDeviceSubnetMask](#)
 Description: Current subnet mask of the GVCP interface of the selected remote device.
 - [GenApi::IInteger & GevDeviceMACAddress](#)
 Description: 48-bit MAC address of the GVCP interface of the selected remote device.
 - [GenApi::ICommand & AutoForceIP](#)
 Description: Automatically forces any cameras on interface to an IP Address on the same subnet as the interface.
 - [GenApi::IInteger & IncompatibleDeviceCount](#)
 Description: Number of incompatible devices detected on current interface.
 - [GenApi::IInteger & 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::IInteger & IncompatibleGevDeviceIPAddress](#)
 Description: Current IP address of the GVCP interface of the selected remote incompatible device.
 - [GenApi::IInteger & IncompatibleGevDeviceSubnetMask](#)
 Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.
 - [GenApi::IInteger & IncompatibleGevDeviceMACAddress](#)
 Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.
 - [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](#)

10.164.1 Detailed Description

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

10.164.2 Constructor & Destructor Documentation

10.164.2.1 TransportLayerInterface() [1/2]

```
TransportLayerInterface (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.164.2.2 ~TransportLayerInterface()

```
~TransportLayerInterface ( )
```

10.164.2.3 TransportLayerInterface() [2/2]

```
TransportLayerInterface ( ) [protected]
```

10.164.3 Friends And Related Function Documentation

10.164.3.1 IInterface

```
friend class IInterface [friend]
```

10.164.3.2 Interface

```
friend class Interface [friend]
```

10.164.3.3 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

10.164.4 Member Data Documentation

10.164.4.1 ActionCommand

`GenApi::ICommand& ActionCommand`

Description: Issues an Action Command to attached GEV devices on interface.

Visibility: Expert

10.164.4.2 AutoForceIP

`GenApi::ICommand& AutoForceIP`

Description: Automatically forces any cameras on interface to an IP Address on the same subnet as the interface.

Visibility: Expert

10.164.4.3 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

10.164.4.4 DeviceCount

`GenApi::IInteger& DeviceCount`

Description: Number of compatible devices detected on current interface.

Visibility: Expert

10.164.4.5 DeviceID

`GenApi::IString& DeviceID`

Description: **Interface** wide unique identifier of the selected device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.164.4.6 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.164.4.7 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

10.164.4.8 DeviceUnlock

`GenApi::IString& DeviceUnlock`

Description: Unlocks devices for internal use.

Visibility: Expert

10.164.4.9 DeviceUpdateList

`GenApi:: ICommand& DeviceUpdateList`

Description: Updates the internal device list.

Visibility: Expert

10.164.4.10 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.164.4.11 FilterDriverStatus

`GenApi::IEnumerationT<FilterDriverStatusEnum>& FilterDriverStatus`

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

Visibility: Expert

10.164.4.12 GevActionDeviceKey

`GenApi::IInteger& GevActionDeviceKey`

Description: Key to authorize the action for the device.

Visibility: Expert

10.164.4.13 GevActionGroupKey

`GenApi::IInteger& GevActionGroupKey`

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

10.164.4.14 GevActionGroupMask

`GenApi::IInteger& GevActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

10.164.4.15 GevActionTime

`GenApi::IInteger& GevActionTime`

Description: Provides the time in nanoseconds when the action is to be executed.

Visibility: Expert

10.164.4.16 GevDeviceIPAddress

`GenApi::IInteger& GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

10.164.4.17 GevDeviceMACAddress

`GenApi::IInteger& GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

10.164.4.18 GevDeviceSubnetMask

`GenApi::IInteger& GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

10.164.4.19 GevInterfaceGateway

`GenApi::IInteger& GevInterfaceGateway`

Description: IP address of the selected gateway entry of this interface.

Visibility: Expert

10.164.4.20 GevInterfaceIPAddress

`GenApi::IInteger& GevInterfaceIPAddress`

Description: IP address of the selected subnet of this interface.

Visibility: Expert

10.164.4.21 GevInterfaceMACAddress

`GenApi::IInteger& GevInterfaceMACAddress`

Description: 48-bit MAC address of this interface.

Visibility: Expert

10.164.4.22 GevInterfaceMTU

`GenApi::IInteger& GevInterfaceMTU`

Description: Maximum transmission unit of this interface.

Visibility: Expert

10.164.4.23 GevInterfaceReceiveLinkSpeed

`GenApi::IInteger& GevInterfaceReceiveLinkSpeed`

Description: Receive link speed of this interface in bits per second.

Visibility: Expert

10.164.4.24 GevInterfaceSubnetMask

`GenApi::IInteger& GevInterfaceSubnetMask`

Description: Subnet mask of the selected subnet of this interface.

Visibility: Expert

10.164.4.25 GevInterfaceTransmitLinkSpeed

`GenApi::IInteger& GevInterfaceTransmitLinkSpeed`

Description: Transmit link speed of this interface in bits per second.

Visibility: Expert

10.164.4.26 HostAdapterDriverVersion

`GenApi::IString& HostAdapterDriverVersion`

Description: Driver version of the interface's host adapter.

Visibility: Expert

10.164.4.27 HostAdapterName

`GenApi::IString& HostAdapterName`

Description: User readable name of the interface's host adapter.

Visibility: Expert

10.164.4.28 HostAdapterVendor

`GenApi::IString& HostAdapterVendor`

Description: User readable name of the host adapter's vendor.

Visibility: Expert

10.164.4.29 IncompatibleDeviceCount

`GenApi::IInteger& IncompatibleDeviceCount`

Description: Number of incompatible devices detected on current interface.

Visibility: Expert

10.164.4.30 IncompatibleDeviceID

`GenApi::IString& IncompatibleDeviceID`

Description: **Interface** wide unique identifier of the selected incompatible device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.164.4.31 IncompatibleDeviceModelName

`GenApi::IString& IncompatibleDeviceModelName`

Description: Name of the incompatible device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.164.4.32 IncompatibleDeviceSelector

`GenApi::IInteger& 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

10.164.4.33 IncompatibleDeviceVendorName

`GenApi::IString& IncompatibleDeviceVendorName`

Description: Name of the incompatible device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.164.4.34 IncompatibleGevDeviceIPAddress

`GenApi::IInteger& IncompatibleGevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

10.164.4.35 IncompatibleGevDeviceMACAddress

`GenApi::IInteger& IncompatibleGevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

10.164.4.36 IncompatibleGevDeviceSubnetMask

`GenApi::IInteger& IncompatibleGevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

10.164.4.37 InterfaceDisplayName

`GenApi::IString& InterfaceDisplayName`

Description: User readable name of the selected interface.

Visibility: Expert

10.164.4.38 InterfaceID

`GenApi::IString& InterfaceID`

Description: Transport layer Producer wide unique identifier of the selected interface.

Visibility: Expert

10.164.4.39 InterfaceType

`GenApi::IString& InterfaceType`

Description: Transport layer type of the interface.

Visibility: Expert

10.164.4.40 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`

10.165 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 & StreamTotalBufferCount](#)
Description: Counts the number of image buffers that arrived since stream started.
- [GenApi::IInteger & StreamDefaultBufferCount](#)
Description: DEPRECATED; Replaced by StreamBufferCountManual.
- [GenApi::IInteger & StreamDefaultBufferCountMax](#)
Description: DEPRECATED; Replaced by StreamBufferCountMax.
- [GenApi::IEnumerationT< StreamDefaultBufferCountModeEnum > & StreamDefaultBufferCountMode](#)
Description: DEPRECATED; Replaced by StreamBufferCountMode.
- [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::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::IInteger & GevMaximumNumberResendBuffers](#)
Description: This node is not used and has been deprecated.
- [GenApi::IInteger & GevTotalPacketCount](#)
Description: Displays number of packets received on this stream.
- [GenApi::IInteger & GevFailedPacketCount](#)
Description: Displays number of packets missed on this stream.
- [GenApi::IInteger & GevResendPacketCount](#)
Description: Displays number of packets received after retransmit request on this stream.
- [GenApi::IInteger & StreamFailedBufferCount](#)
Description: Displays number of incomplete images with missing leader/trailer information.
- [GenApi::IInteger & StreamBufferUnderrunCount](#)
Description: Displays number of dropped images caused by driver running out of buffers.
- [GenApi::IInteger & GevResendRequestCount](#)
Description: Displays number of packets requested to be retransmitted on this stream.
- [GenApi::IInteger & StreamBlockTransferSize](#)
Description: Controls the image breakup size that should be used on this stream.

Protected Member Functions

- [TransportLayerStream \(\)](#)

Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameralInternal](#)

10.165.1 Detailed Description

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

10.165.2 Constructor & Destructor Documentation

10.165.2.1 [TransportLayerStream\(\)](#) [1/2]

```
TransportLayerStream (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.165.2.2 [~TransportLayerStream\(\)](#)

```
~TransportLayerStream ( )
```

10.165.2.3 [TransportLayerStream\(\)](#) [2/2]

```
TransportLayerStream ( ) [protected]
```

10.165.3 Friends And Related Function Documentation

10.165.3.1 [CameraBase](#)

```
friend class CameraBase [friend]
```

10.165.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

10.165.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

10.165.4 Member Data Documentation

10.165.4.1 GevFailedPacketCount

`GenApi::IInteger& GevFailedPacketCount`

Description: Displays number of packets missed on this stream.

Visibility: Expert

10.165.4.2 GevMaximumNumberResendBuffers

`GenApi::IInteger& GevMaximumNumberResendBuffers`

Description: This node is not used and has been deprecated.

Visibility: Invisible

10.165.4.3 GevMaximumNumberResendRequests

`GenApi::IInteger& GevMaximumNumberResendRequests`

Description: Maximum number of resend requests per image.

Each resend request consists of a span of consecutive packet IDs. Visibility: Expert

10.165.4.4 GevPacketResendMode

`GenApi::IBoolean& GevPacketResendMode`

Description: Enables or disables the packet resend mechanism.

Visibility: Expert

10.165.4.5 GevPacketResendTimeout

```
GenApi::IInteger& GevPacketResendTimeout
```

Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.

Visibility: Expert

10.165.4.6 GevResendPacketCount

```
GenApi::IInteger& GevResendPacketCount
```

Description: Displays number of packets received after retransmit request on this stream.

Visibility: Expert

10.165.4.7 GevResendRequestCount

```
GenApi::IInteger& GevResendRequestCount
```

Description: Displays number of packets requested to be retransmitted on this stream.

Visibility: Expert

10.165.4.8 GevTotalPacketCount

```
GenApi::IInteger& GevTotalPacketCount
```

Description: Displays number of packets received on this stream.

Visibility: Expert

10.165.4.9 StreamBlockTransferSize

```
GenApi::IInteger& StreamBlockTransferSize
```

Description: Controls the image breakup size that should be used on this stream.

Visibility: Expert

10.165.4.10 StreamBufferCountManual

```
GenApi::IInteger& StreamBufferCountManual
```

Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.

Visibility: Expert

10.165.4.11 StreamBufferCountMax

`GenApi::IInteger& 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

10.165.4.12 StreamBufferCountMode

`GenApi::IEnumerationT<StreamBufferCountModeEnum>& StreamBufferCountMode`

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

Locked to Manual mode on 32-bit Windows due to memory constraints. Visibility: Expert

10.165.4.13 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

10.165.4.14 StreamBufferHandlingMode

`GenApi::IEnumerationT<StreamBufferHandlingModeEnum>& StreamBufferHandlingMode`

Description: Available buffer handling modes of this data stream: Visibility: Beginner.

10.165.4.15 StreamBufferUnderrunCount

`GenApi::IInteger& StreamBufferUnderrunCount`

Description: Displays number of dropped images caused by driver running out of buffers.

Visibility: Expert

10.165.4.16 StreamCRCCheckEnable

`GenApi::IBoolean& StreamCRCCheckEnable`

Description: Enables or disables CRC checks on received images.

Visibility: Expert

10.165.4.17 StreamDefaultBufferCount

`GenApi::IInteger& StreamDefaultBufferCount`

Description: DEPRECATED; Replaced by StreamBufferCountManual.

Controls the number of buffers to be used on this stream upon acquisition start when in manual mode. Visibility: Invisible

10.165.4.18 StreamDefaultBufferCountMax

`GenApi::IInteger& StreamDefaultBufferCountMax`

Description: DEPRECATED; Replaced by StreamBufferCountMax.

Controls the maximum number of buffers that should be used on this stream. This value is calculated based on the available system memory. Visibility: Invisible

10.165.4.19 StreamDefaultBufferCountMode

`GenApi::IEnumerationT<StreamDefaultBufferCountModeEnum>& StreamDefaultBufferCountMode`

Description: DEPRECATED; Replaced by StreamBufferCountMode.

Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints. Visibility: Invisible

10.165.4.20 StreamFailedBufferCount

`GenApi::IInteger& StreamFailedBufferCount`

Description: Displays number of incomplete images with missing leader/trailer information.

Visibility: Expert

10.165.4.21 StreamID

`GenApi::IString& StreamID`

Description: Device unique ID for the data stream, e.g.

a GUID. Visibility: Expert

10.165.4.22 StreamTotalBufferCount

`GenApi::IInteger& StreamTotalBufferCount`

Description: Counts the number of image buffers that arrived since stream started.

Visibility: Expert

10.165.4.23 StreamType

`GenApi::IEnumerationT<StreamTypeEnum>& StreamType`

Description: Stream type of the device.

Visibility: Expert

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

- `include/TransportLayerStream.h`

10.166 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::IBoolean & EnumerateGEVInterfaces`
Description: Enables or disables enumeration of GEV Interfaces.
- `GenApi::ICommand & AutoForceIP`
Description: Automatically forces any cameras on the system to an IP Address on the same subnet as the interfaces they are connected to.

Protected Member Functions

- `TransportLayerSystem ()`

Friends

- class `System`
- class `ISystem`
- class `SystemPtrInternal`

10.166.1 Detailed Description

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

10.166.2 Constructor & Destructor Documentation

10.166.2.1 TransportLayerSystem() [1/2]

```
TransportLayerSystem (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.166.2.2 ~TransportLayerSystem()

```
~TransportLayerSystem ( )
```

10.166.2.3 TransportLayerSystem() [2/2]

```
TransportLayerSystem ( ) [protected]
```

10.166.3 Friends And Related Function Documentation**10.166.3.1 ISystem**

```
friend class ISystem [friend]
```

10.166.3.2 System

```
friend class System [friend]
```

10.166.3.3 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

10.166.4 Member Data Documentation

10.166.4.1 AutoForceIP

`GenApi:: ICommand& AutoForceIP`

Description: Automatically forces any cameras on the system to an IP Address on the same subnet as the interfaces they are connected to.

Visibility: Expert

10.166.4.2 EnumerateGEVInterfaces

`GenApi:: IBoolean& EnumerateGEVInterfaces`

Description: Enables or disables enumeration of GEV Interfaces.

Visibility: Expert

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

- `include/TransportLayerSystem.h`

10.167 U3V_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- `uint32_t ChunkID`
- `uint32_t ChunkLength`

10.167.1 Detailed Description

header of a GVCP request packet

10.167.2 Member Data Documentation

10.167.2.1 ChunkID

`uint32_t ChunkID`

10.167.2.2 ChunkLength

```
uint32_t ChunkLength
```

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

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

10.168 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](#)

10.168.1 Detailed Description

U3V/GenCP command header.

10.168.2 Member Data Documentation

10.168.2.1 CommandId

```
uint16_t CommandId
```

10.168.2.2 Flags

```
uint16_t Flags
```

10.168.2.3 Length

```
uint16_t Length
```

10.168.2.4 Prefix

```
uint32_t Prefix
```

10.168.2.5 ReqId

```
uint16_t ReqId
```

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

- include/SpinGenApi/[EventAdapterU3V.h](#)

10.169 U3V_EVENT_DATA Struct Reference

U3V/GenCP EVENT_CMD specific command data.

Public Attributes

- uint16_t [Reserved](#)
- uint16_t [EventId](#)
- uint64_t [Timestamp](#)

10.169.1 Detailed Description

U3V/GenCP EVENT_CMD specific command data.

10.169.2 Member Data Documentation

10.169.2.1 EventId

```
uint16_t EventId
```

10.169.2.2 Reserved

```
uint16_t Reserved
```

10.169.2.3 Timestamp

```
uint64_t Timestamp
```

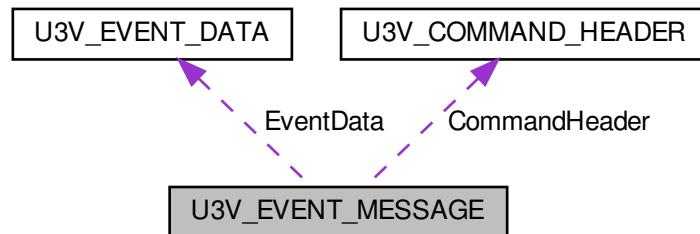
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

10.170 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](#)

10.170.1 Detailed Description

Entire event data message (without the variable-sized data field)

10.170.2 Member Data Documentation

10.170.2.1 CommandHeader

```
U3V\_COMMAND\_HEADER CommandHeader
```

10.170.2.2 EventData

`U3V_EVENT_DATA` EventData

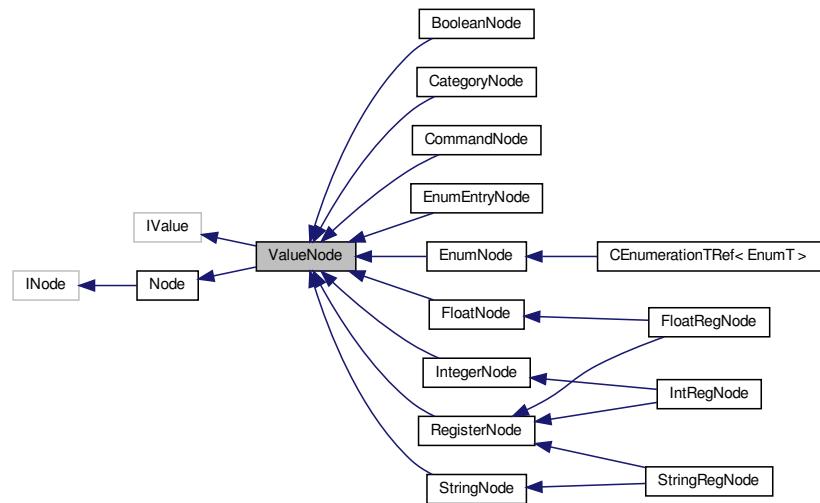
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

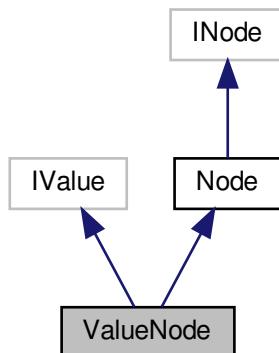
10.171 ValueNode Class Reference

[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

10.171.1 Detailed Description

[Interface](#) for value properties.

10.171.2 Constructor & Destructor Documentation

10.171.2.1 `ValueNode()` [1/2]

`ValueNode ()`

Constructor.

10.171.2.2 `ValueNode()` [2/2]

```
ValueNode (
    std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenICam IValue](#)

10.171.2.3 ~ValueNode()

```
~ValueNode( )
```

Destructor.

10.171.3 Member Function Documentation

10.171.3.1 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)

10.171.3.2 GetNode()

```
virtual INode* GetNode( ) [virtual]
```

10.171.3.3 IsValueCacheValid()

```
virtual bool IsValueCacheValid( ) const [virtual]
```

Checks if the value comes from cache or is requested from another node.

10.171.3.4 SetReference()

```
virtual void SetReference(
    INode * pBase) [virtual]
```

overload SetReference for Value

Reimplemented from [Node](#).

Reimplemented in [FloatNode](#), [IntegerNode](#), [EnumNode](#), [CEnumerationTRef< EnumT >](#), [StringNode](#), [Register< Node](#), [BooleanNode](#), [CommandNode](#), [EnumEntryNode](#), [CategoryNode](#), [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

10.171.3.5 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](#)

10.172 Version_t Struct Reference

Version.

Public Attributes

- uint16_t **Major**
- uint16_t **Minor**
a is incompatible with b if a != b
- uint16_t **SubMinor**
a is incompatible b a > b

10.172.1 Detailed Description

Version.

10.172.2 Member Data Documentation

10.172.2.1 Major

```
uint16_t Major
```

10.172.2.2 Minor

```
uint16_t Minor
```

a is incompatible with b if a != b

10.172.2.3 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 11

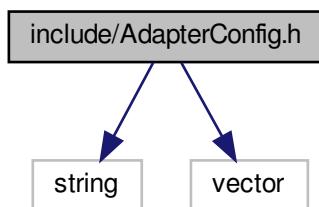
File Documentation

11.1 doc/Doxygen/spindocs/Licensing.dox File Reference

11.2 doc/Doxygen/spindocs/MainPage.dox File Reference

11.3 include/AdapterConfig.h File Reference

Include dependency graph for AdapterConfig.h:



Classes

- struct [IplInfo](#)
- struct [AdapterInfo](#)
- class [AdapterConfigException](#)

Namespaces

- [AdapterConfig](#)

Macros

- `#define ADAPTERCONFIG_API __declspec(dllexport)`

Enumerations

- `enum AdapterConfigErr {
 IP_ADDRESS_INVALID,
 IP_ADDRESS_IS_NOT_V4,
 IP_ADDRESS_TOO_LARGE,
 IP_ADDRESS_TOO_SMALL,
 HOST_ADDRESS_ZERO,
 GATEWAY_INVALID,
 SUBNET_MASK_INVALID,
 GATEWAY_SUBNET_SAME_IP,
 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)`
- `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 GetGatewayAddress (const std::string &ipAddrStr, unsigned int subnetMaskLength)`
- `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 ()`
- `ADAPTERCONFIG_API std::string GetAutoStartGateway ()`

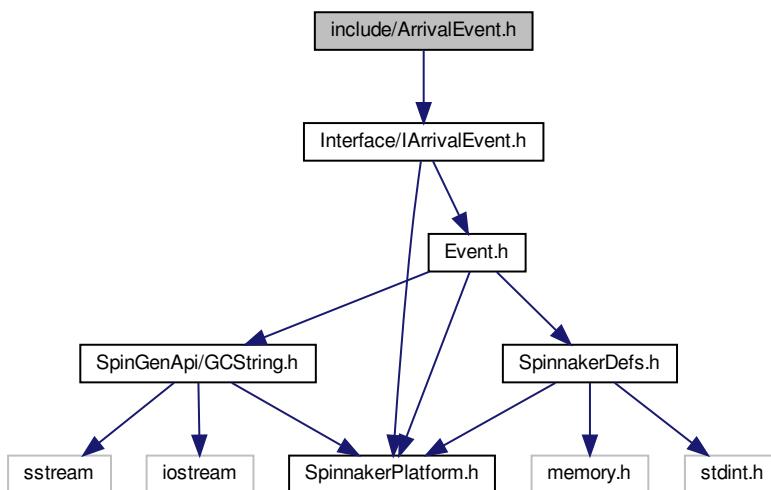
11.3.1 Macro Definition Documentation

11.3.1.1 ADAPTERCONFIG_API

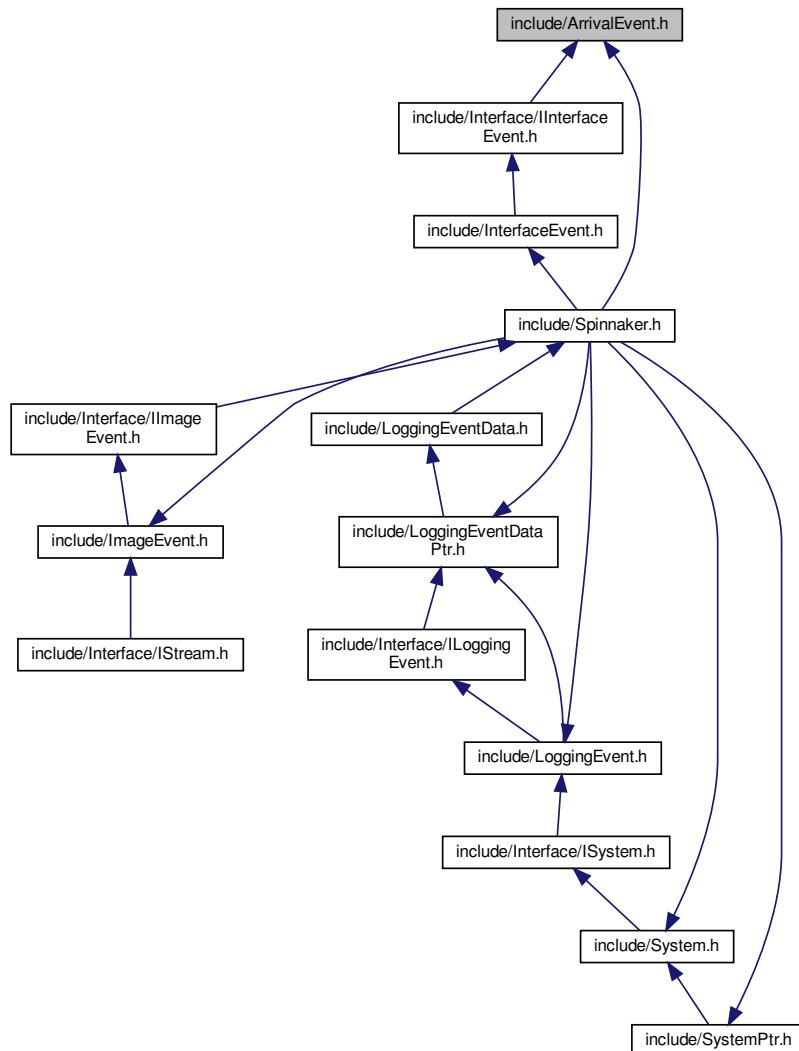
```
#define ADAPTERCONFIG_API __declspec(dllimport)
```

11.4 include/ArrivalEvent.h File Reference

Include dependency graph for ArrivalEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ArrivalEvent](#)

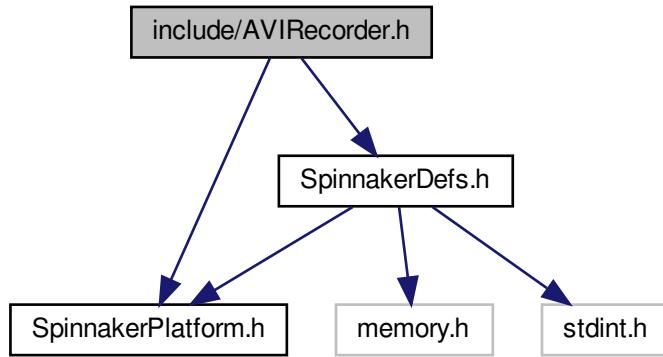
An event handler for capturing the device arrival event.

Namespaces

- [Spinnaker](#)

11.5 include/AVIRecorder.h File Reference

Include dependency graph for AVIRecorder.h:



Namespaces

- [Spinnaker](#)

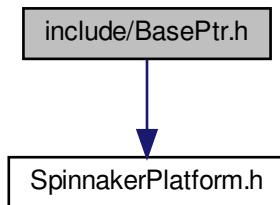
Functions

- class [DEPRECATED_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API A←
VIRecorder

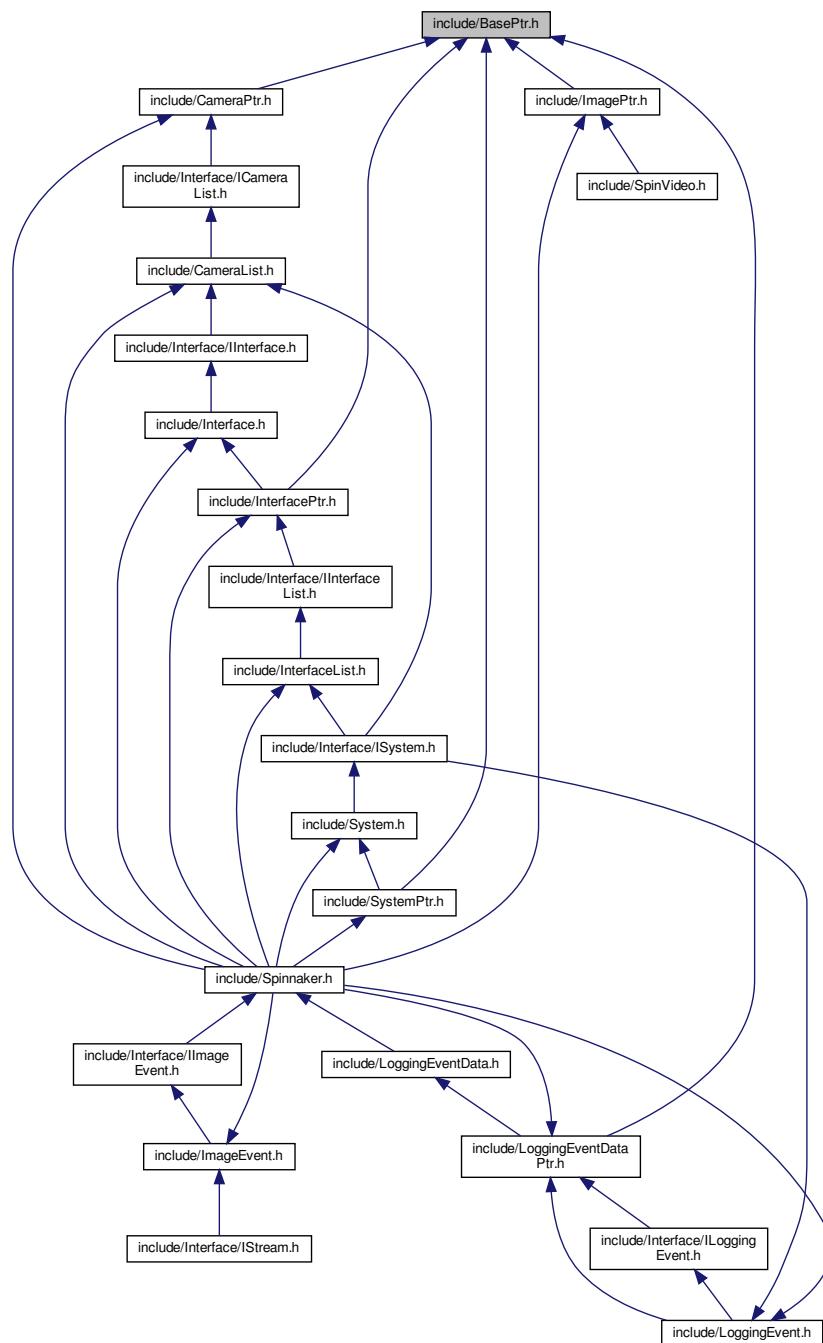
Provides the functionality for the user to record images to an AVI file.

11.6 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 >](#)

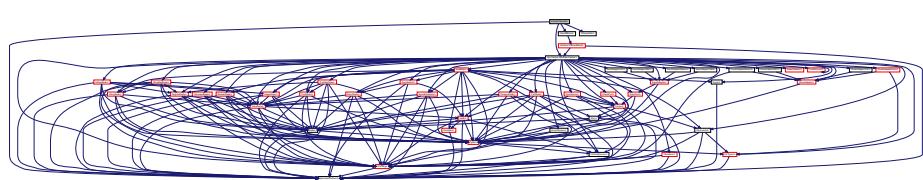
The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

Namespaces

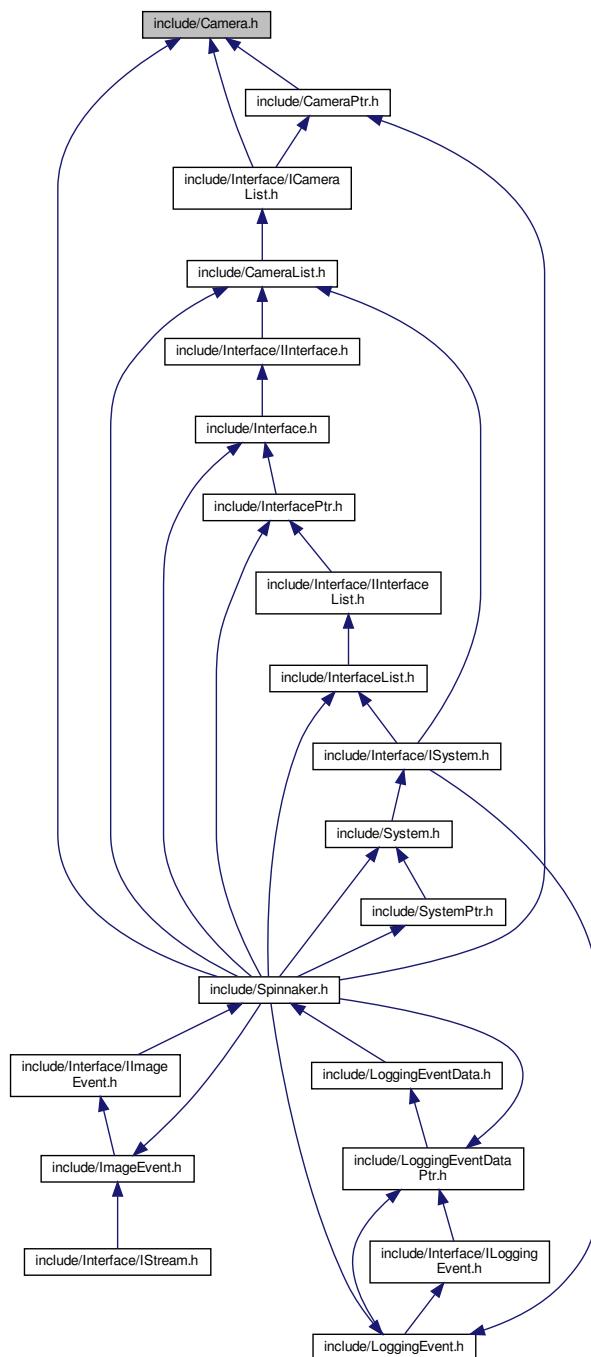
- [Spinnaker](#)

11.7 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](#)

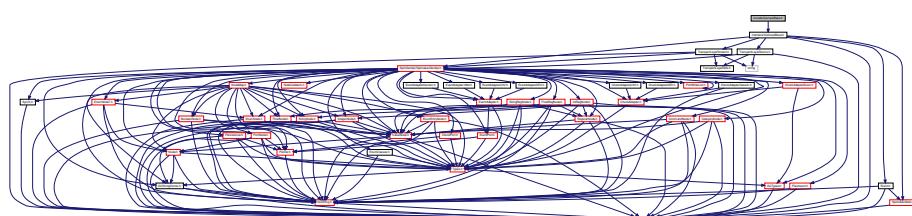
The camera object class.

Namespaces

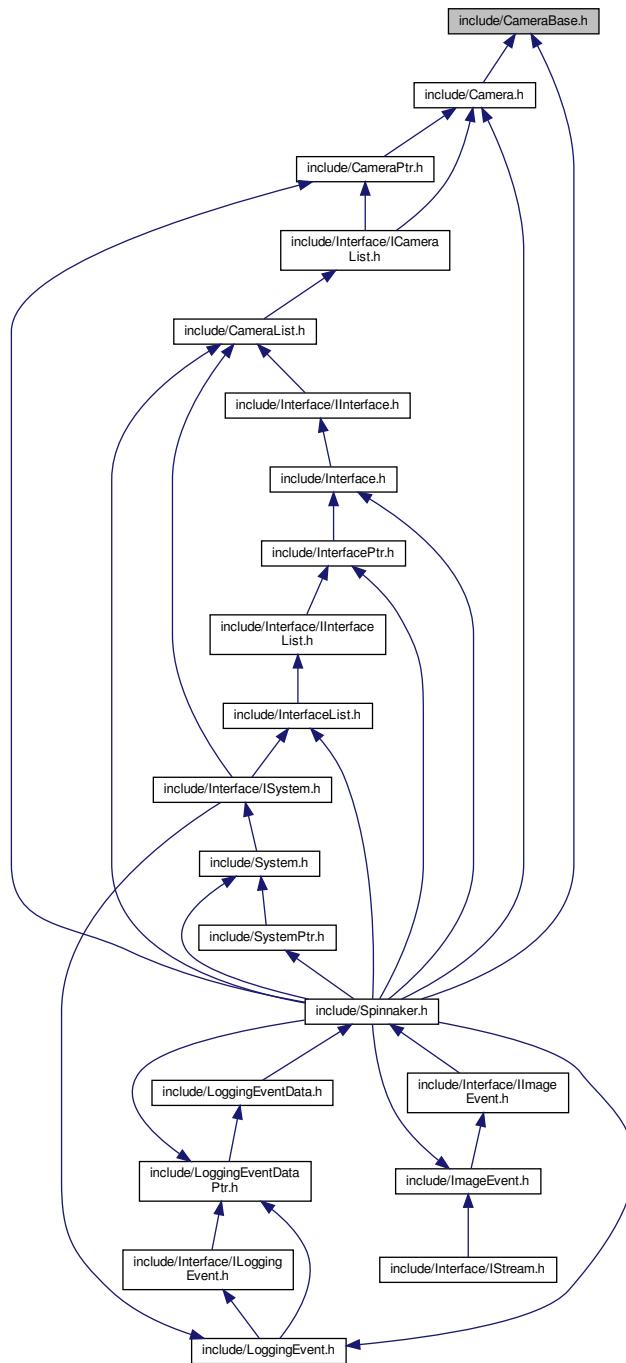
- [Spinnaker](#)

11.8 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](#)

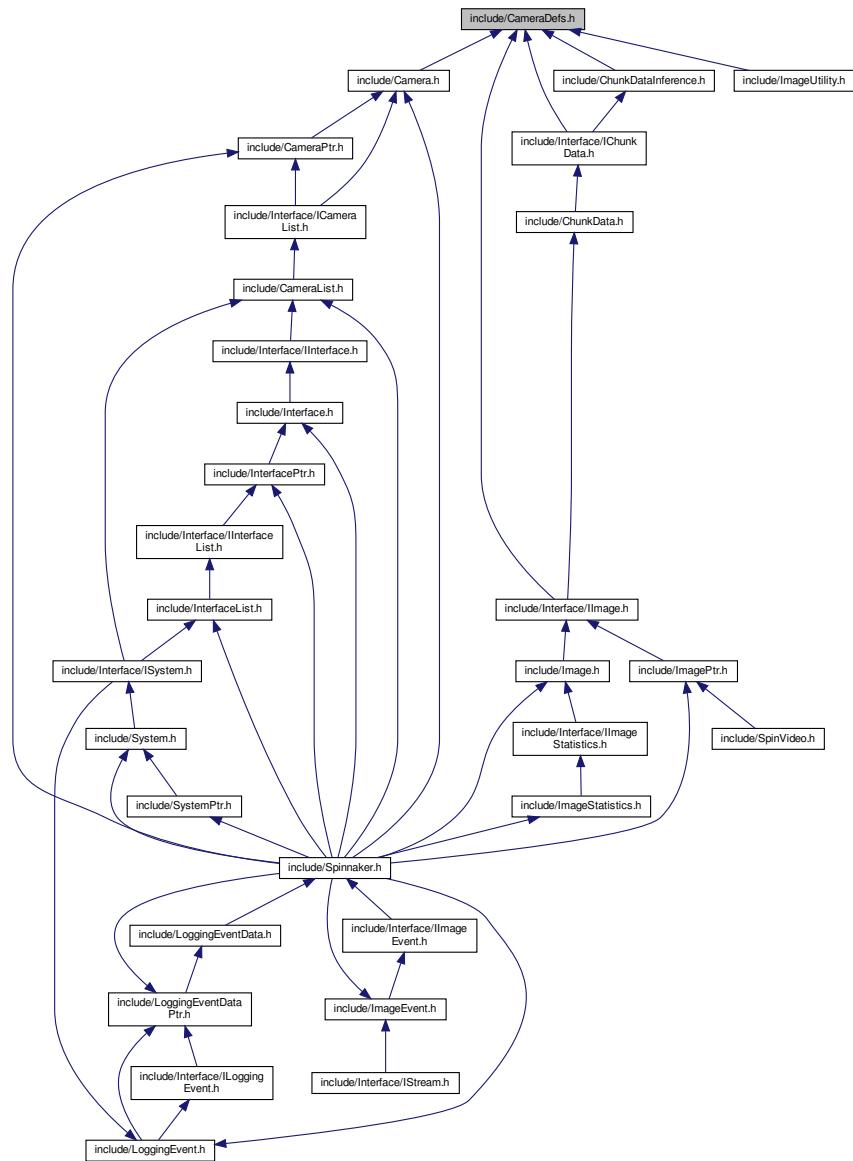
The base class for the camera object.

Namespaces

- [Spinnaker](#)

11.9 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_DEVICEREGISTERSENDIANCESS` }
- 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_DEVICETLTTYPE` }
- 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_AUTOEXPOSURECONTROL_PRIORITY` }

- 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_GEVSUPPORTEOPTIONSELECTOR }

• 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 FileModeEnums {
FileMode_Read,
FileMode_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 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_DEVICESTREAMCHANNELENDIANCESS }

• 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_SENSELDIGITIZATIONTAPS }

• 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_YYYYYY,
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_GEVGVSPEXTENDEDDIDMODE }

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

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

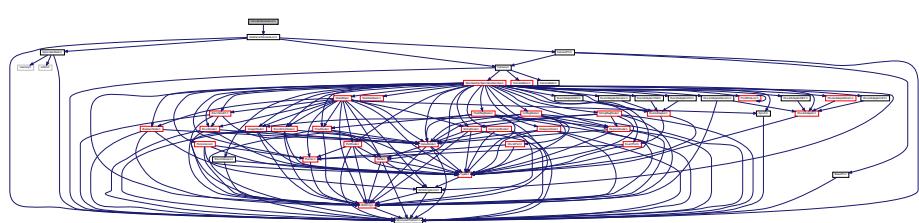
- enum [CxpLinkConfigurationStatusEnums](#) {
 CxpLinkConfigurationStatus_None,
 CxpLinkConfigurationStatus_Pending,
 CxpLinkConfigurationStatus_CXP1_X1,
 CxpLinkConfigurationStatus_CXP2_X1,
 CxpLinkConfigurationStatus_CXP3_X1,
 CxpLinkConfigurationStatus_CXP5_X1,
 CxpLinkConfigurationStatus_CXP6_X1,
 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 }

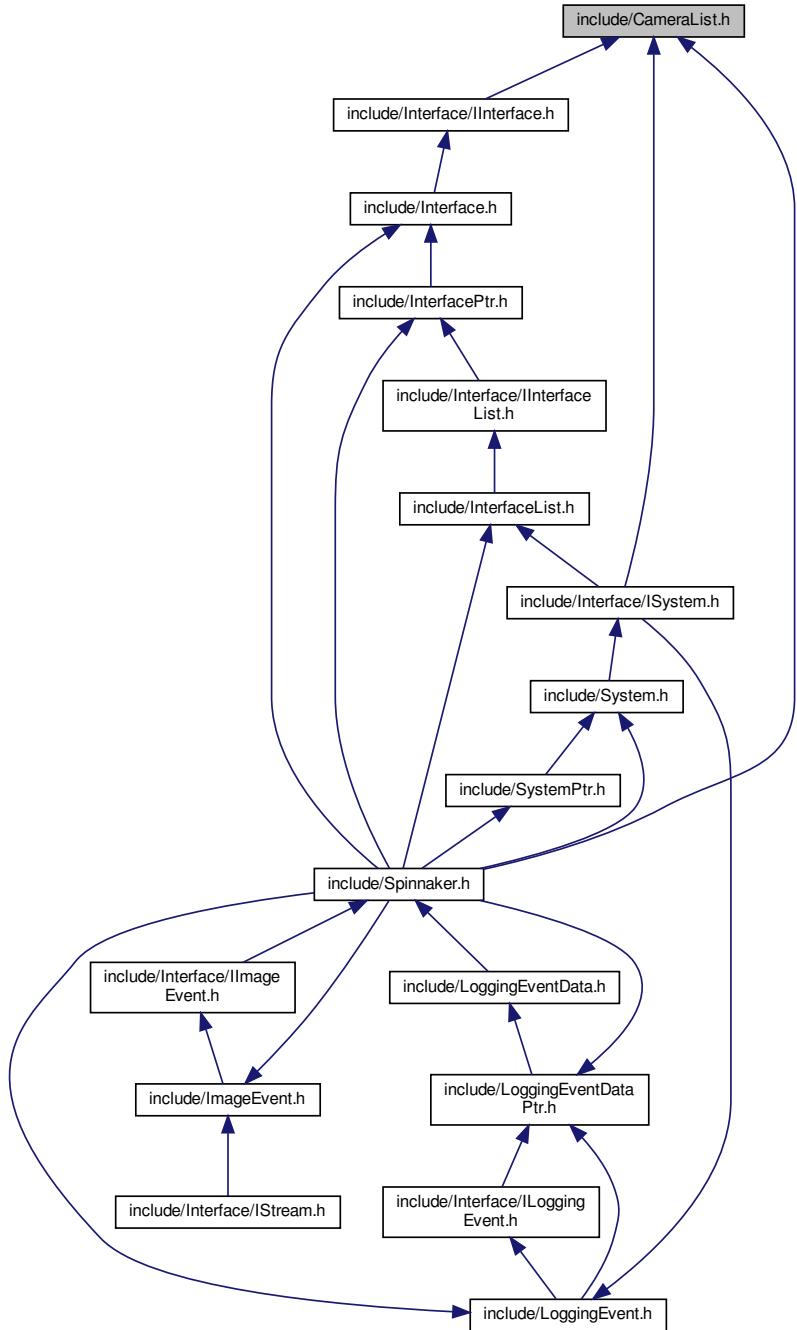
• enum CxpLinkConfigurationEnums {
 CxpLinkConfiguration_Auto,
 CxpLinkConfiguration_CXP1_X1,
 CxpLinkConfiguration_CXP2_X1,
 CxpLinkConfiguration_CXP3_X1,
 CxpLinkConfiguration_CXP5_X1,
 CxpLinkConfiguration_CXP6_X1,
 CxpLinkConfiguration_CXP1_X2,
 CxpLinkConfiguration_CXP2_X2,
 CxpLinkConfiguration_CXP3_X2,
 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_CXP CONNECTIONTESTMODE }
```
-
- ```
• enum CxpPoCxpStatusEnums {
 CxpPoCxpStatus_Auto,
 CxpPoCxpStatus_Off,
 CxpPoCxpStatus_Tripped,
 NUM_CXPOCXPSTATUS }
```

## 11.10 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

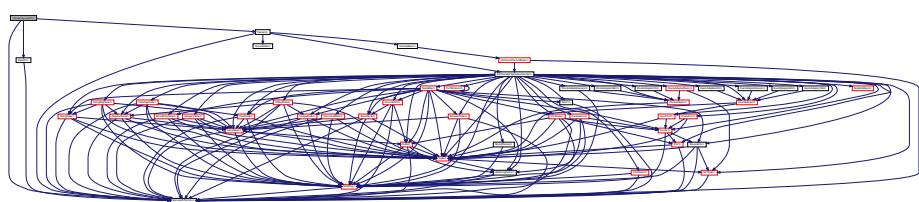
*Used to hold a list of camera objects.*

## Namespaces

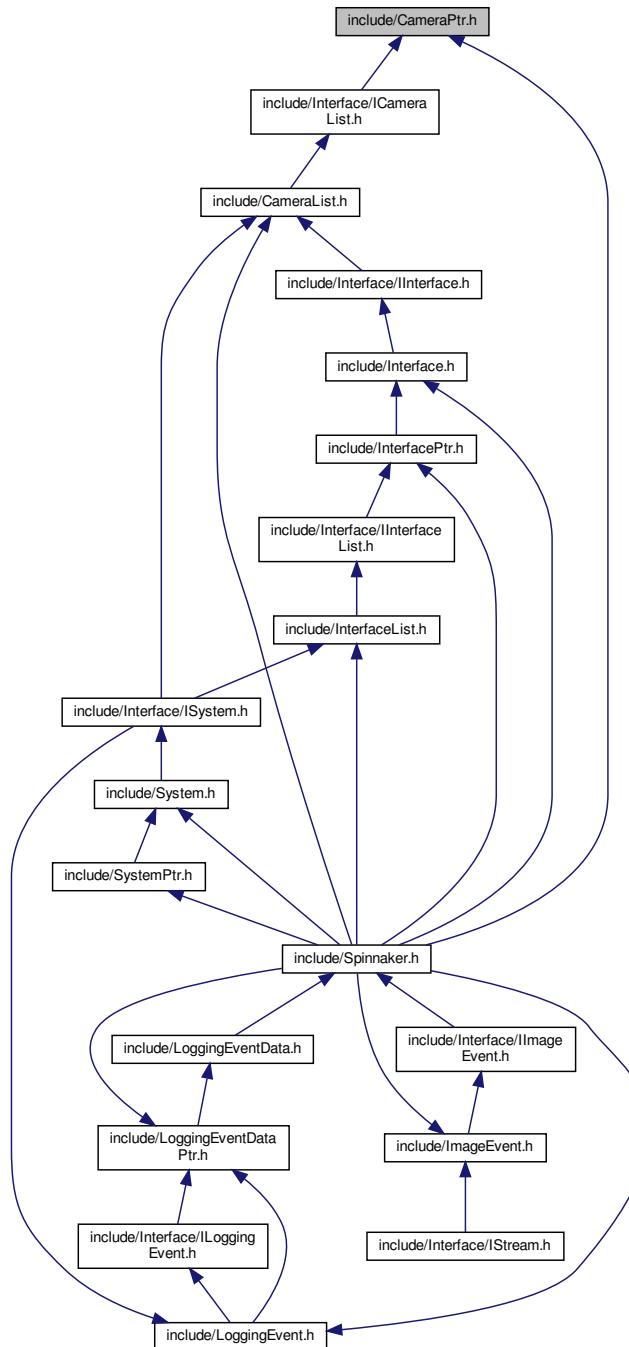
- Spinnaker

## 11.11 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](#)

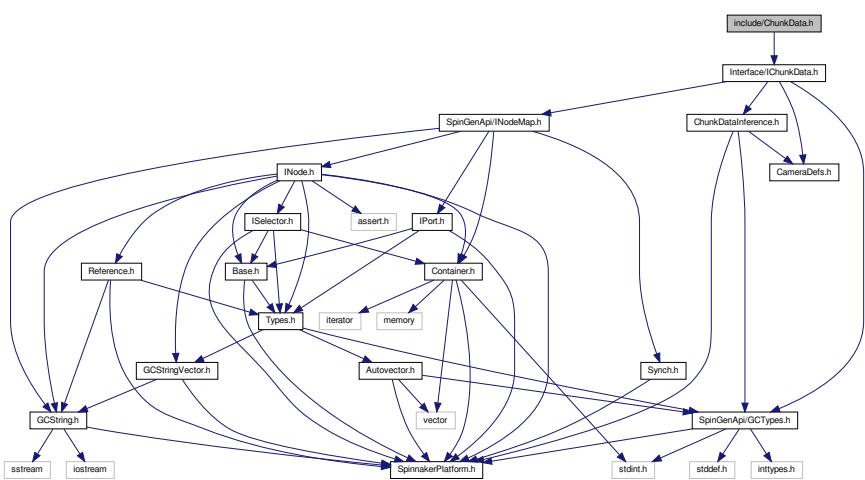
*A reference tracked pointer to a camera object.*

## Namespaces

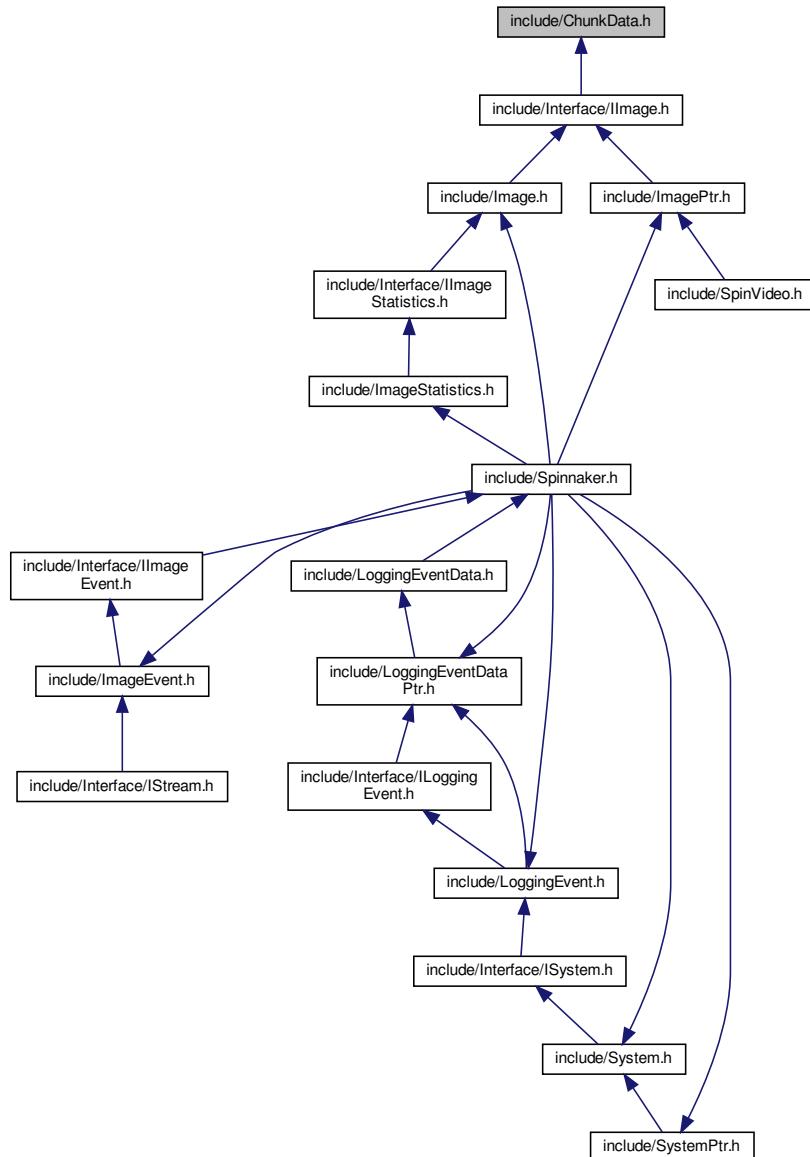
- [Spinnaker](#)

## 11.12 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](#)

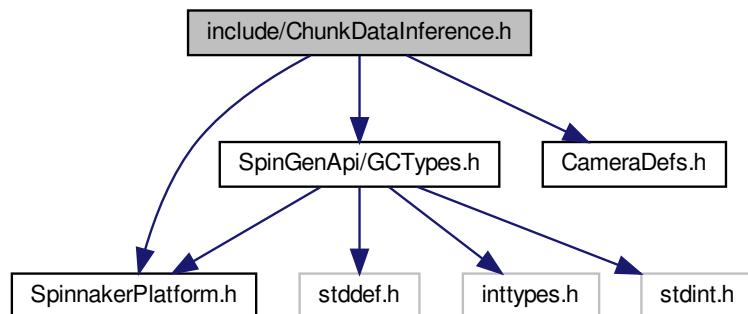
*The chunk data which contains additional information about an image.*

## Namespaces

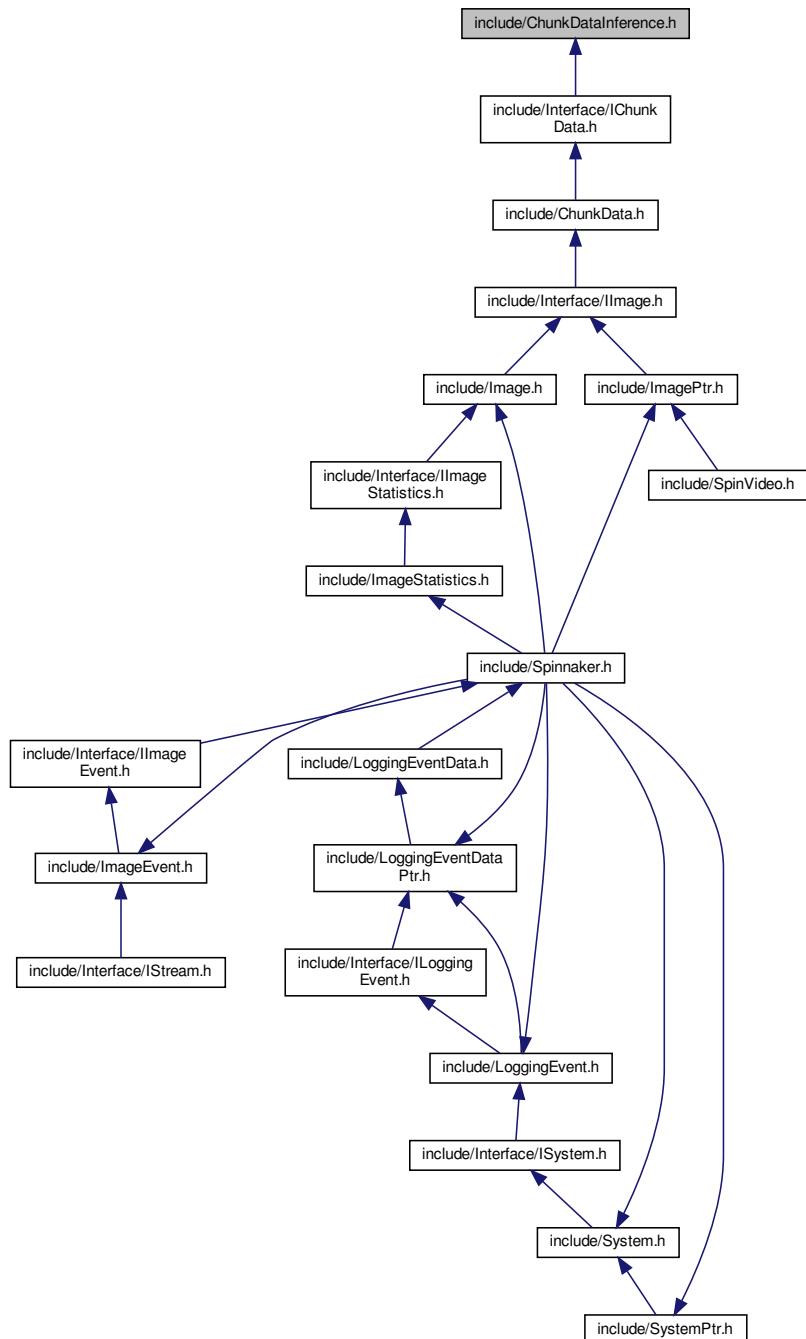
- [Spinnaker](#)

## 11.13 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](#)

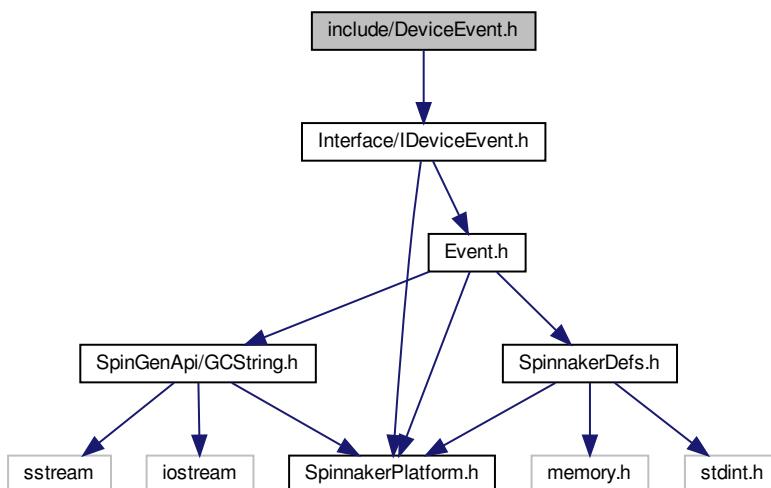
## Enumerations

- enum [InferenceBoxType](#) {
   
INFERENCE\_BOX\_TYPE\_RECTANGLE = 0,
   
INFERENCE\_BOX\_TYPE\_CIRCLE = 1,
   
INFERENCE\_BOX\_TYPE\_ROTATED\_RECTANGLE = 2 }

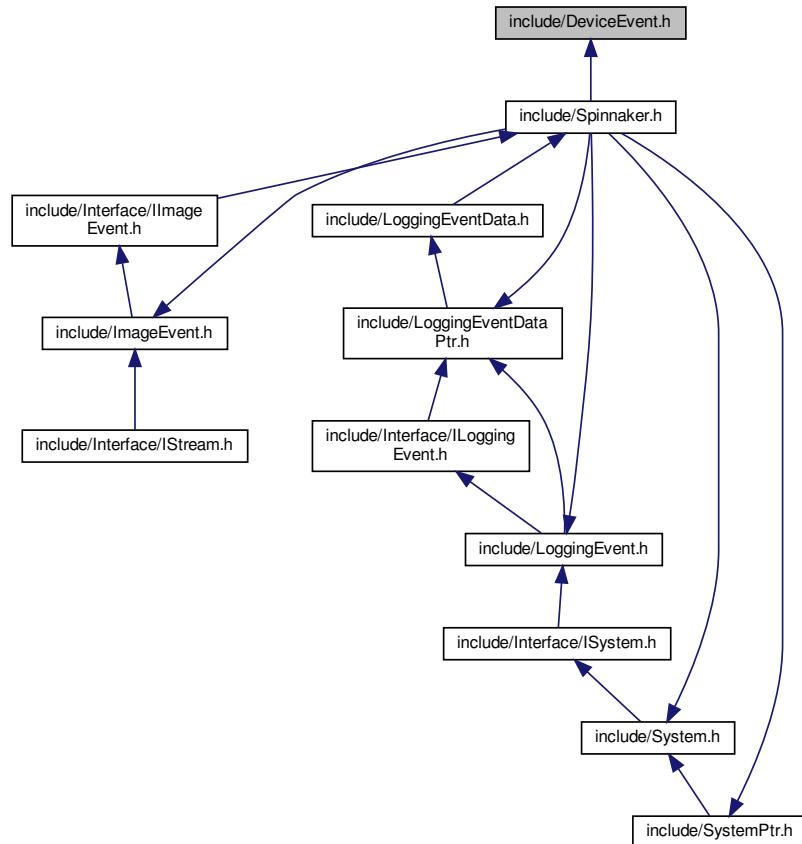
*Inference Bounding Box Type.*

## 11.14 include/DeviceEvent.h File Reference

Include dependency graph for DeviceEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [DeviceEvent](#)

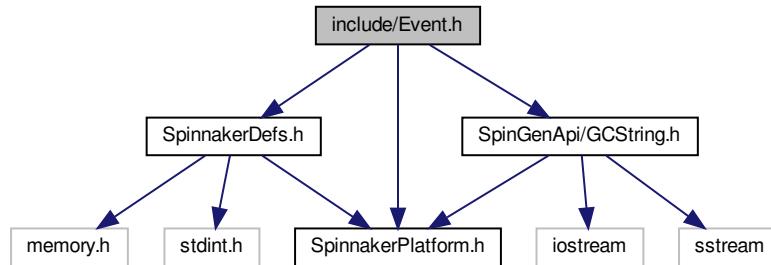
*A handler to device events.*

## Namespaces

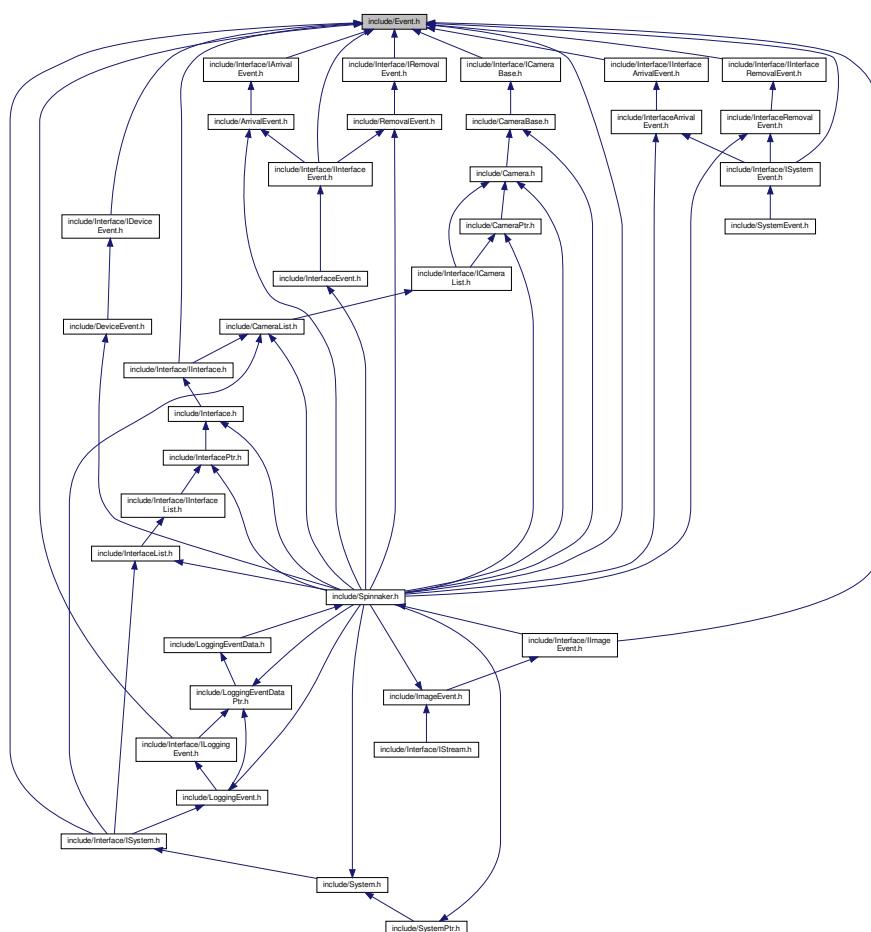
- [Spinnaker](#)

## 11.15 include/Event.h File Reference

Include dependency graph for Event.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Event](#)

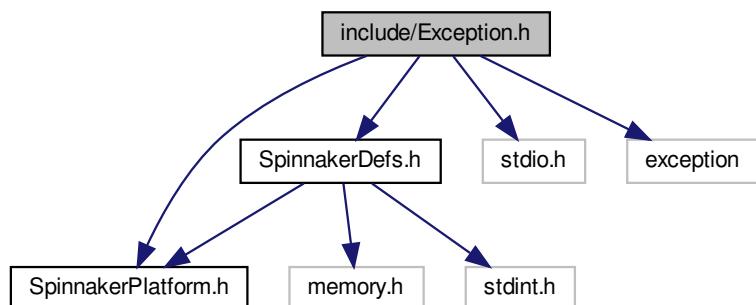
*The base class for all event types.*

## Namespaces

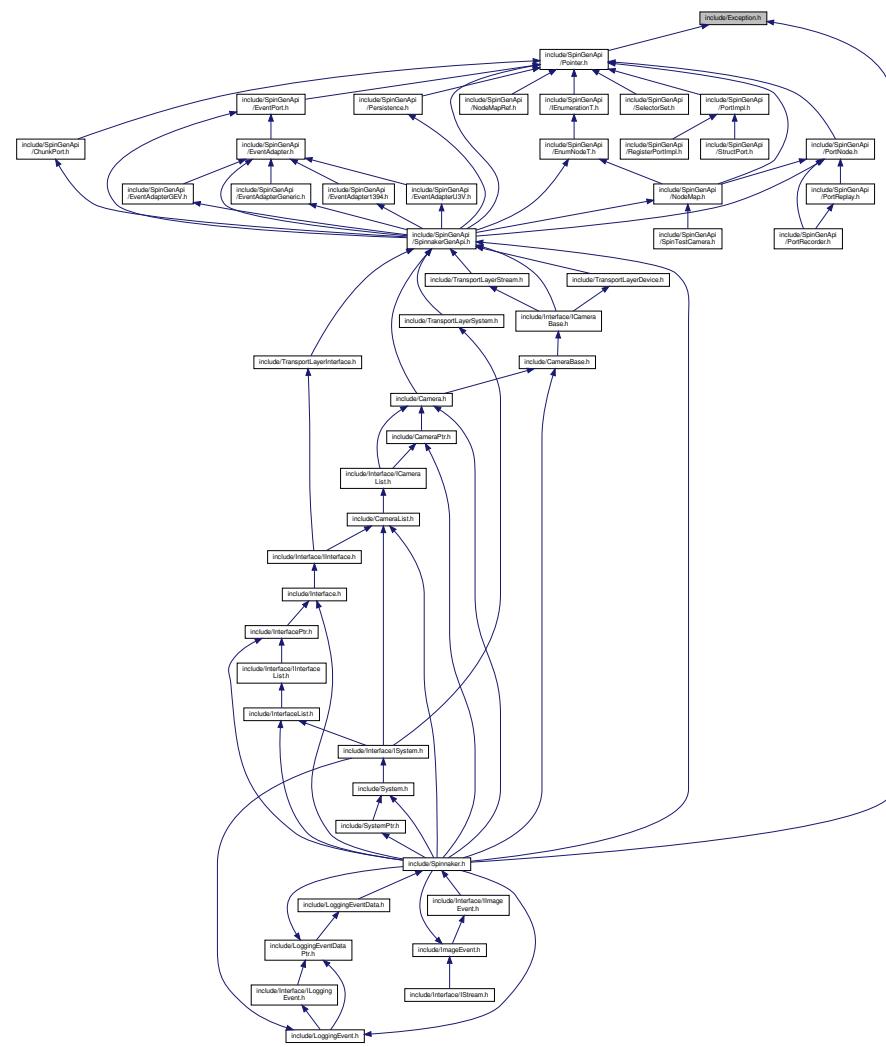
- [Spinnaker](#)

## 11.16 include/Exception.h File Reference

Include dependency graph for Exception.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `Exception`

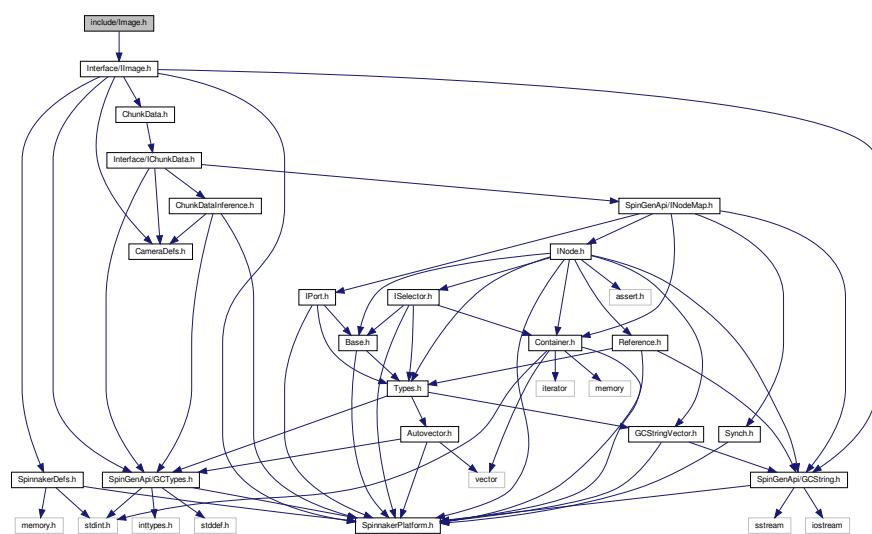
The [Exception](#) object represents an error that is returned from the library.

## Namespaces

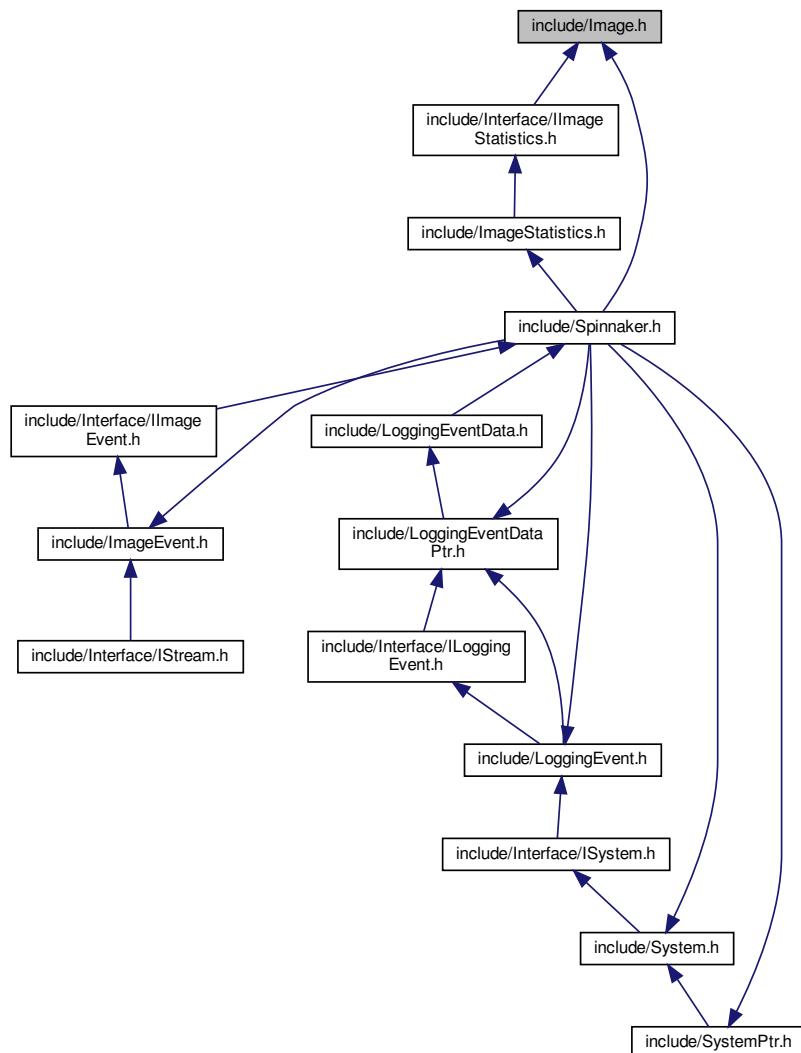
- Spinnaker

## 11.17 include/Image.h File Reference

Include dependency graph for Image.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Image](#)

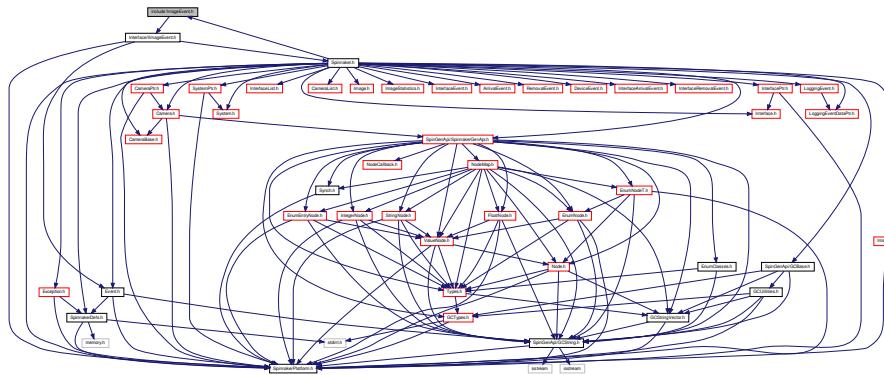
*The image object class.*

## Namespaces

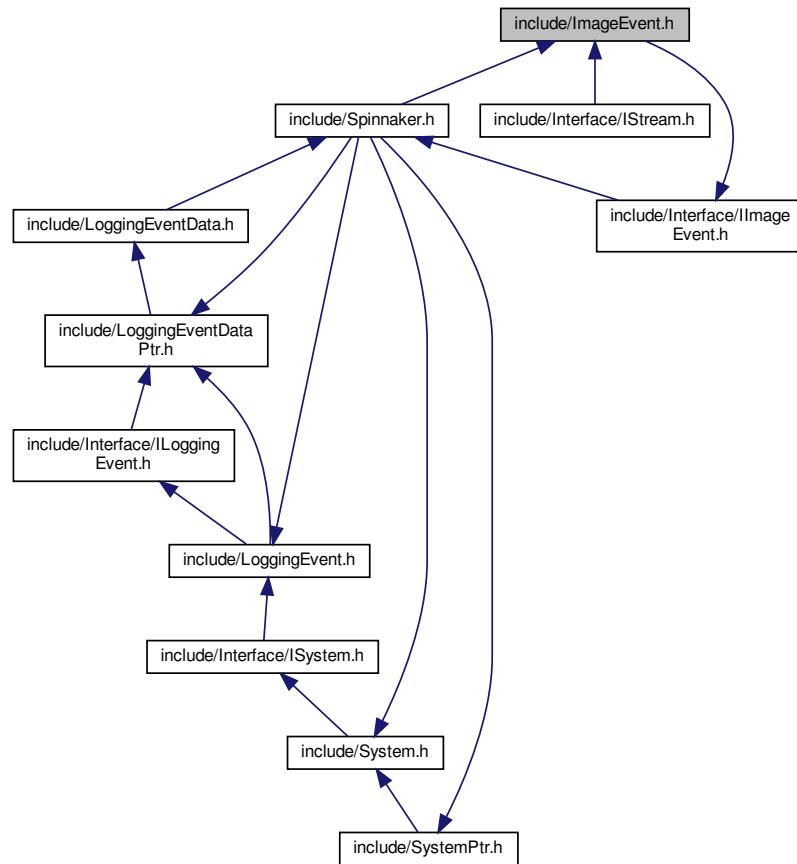
- [Spinnaker](#)

## 11.18 include/ImageEvent.h File Reference

Include dependency graph for ImageEvent.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [ImageEvent](#)

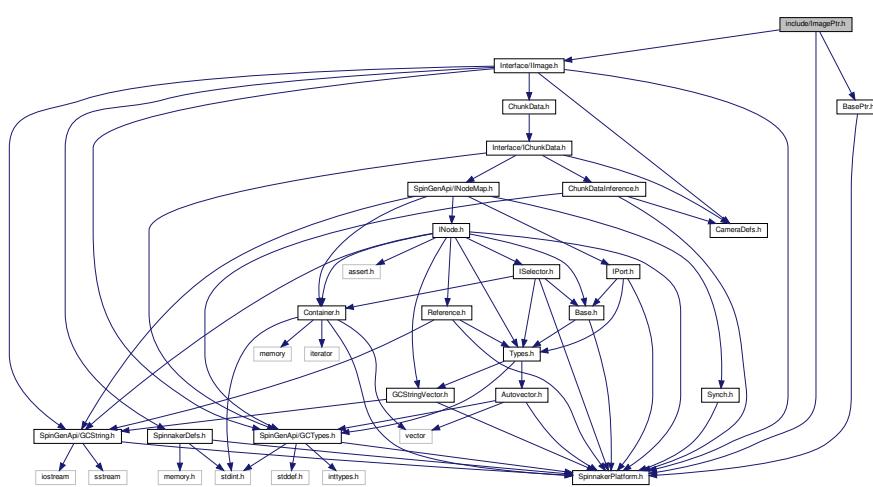
*A handler for capturing image arrival events.*

## Namespaces

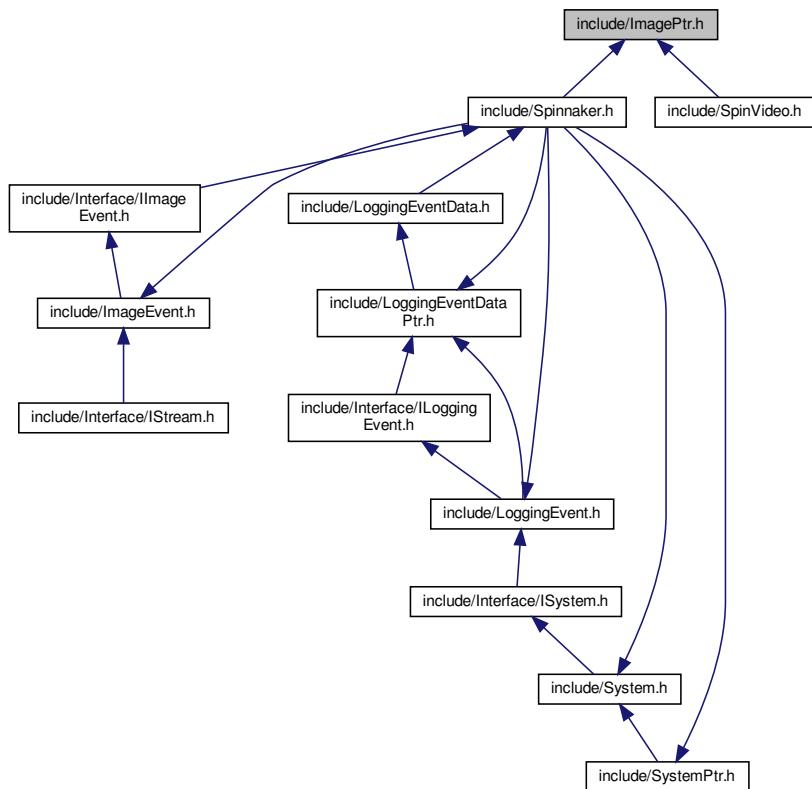
- Spinnaker

### 11.19 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](#)

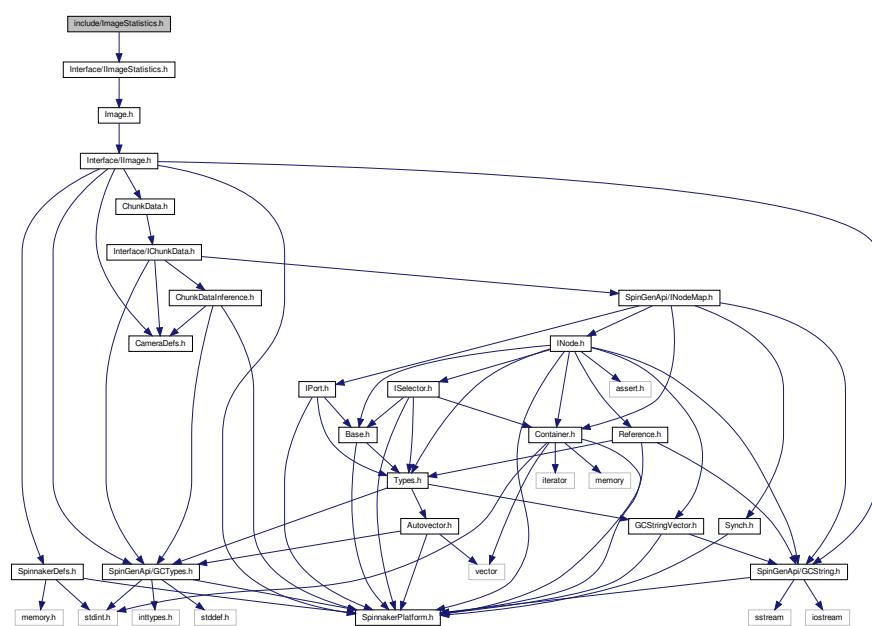
*A reference tracked pointer to an image object.*

## Namespaces

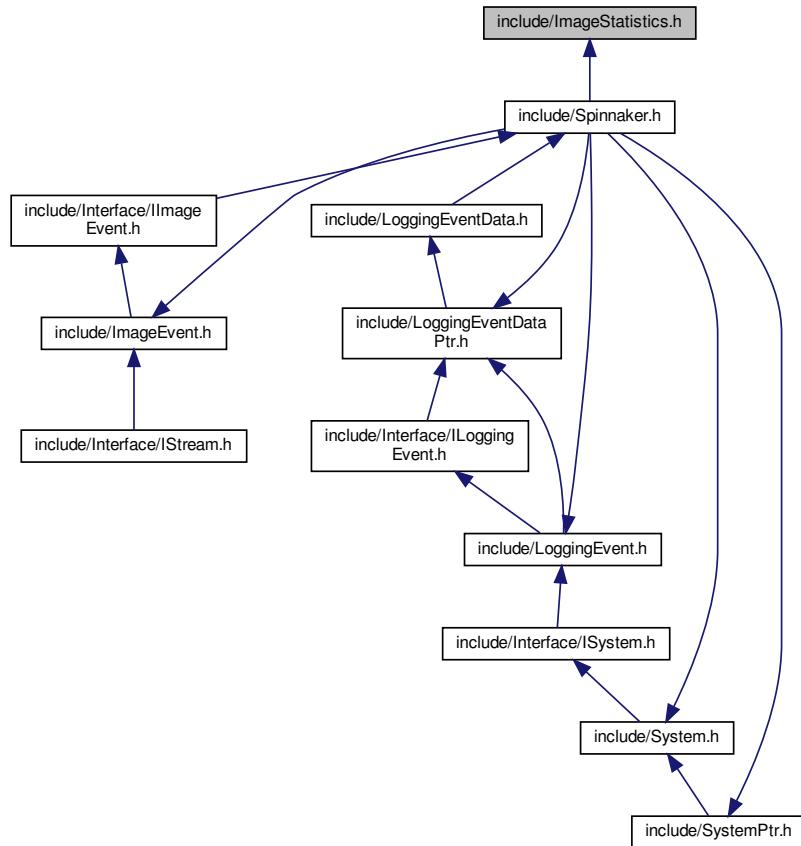
- [Spinnaker](#)

## 11.20 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](#)

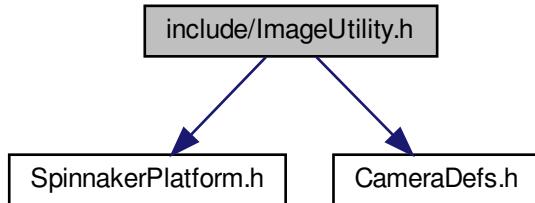
*Represents image statistics for an image.*

## Namespaces

- [Spinnaker](#)

## 11.21 include/ImageUtility.h File Reference

Include dependency graph for ImageUtility.h:



### Classes

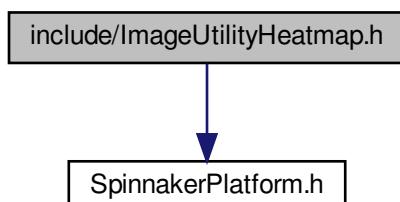
- class [ImageUtility](#)  
*Static helper functions for the image object class.*

### Namespaces

- [Spinnaker](#)

## 11.22 include/ImageUtilityHeatmap.h File Reference

Include dependency graph for ImageUtilityHeatmap.h:



### Classes

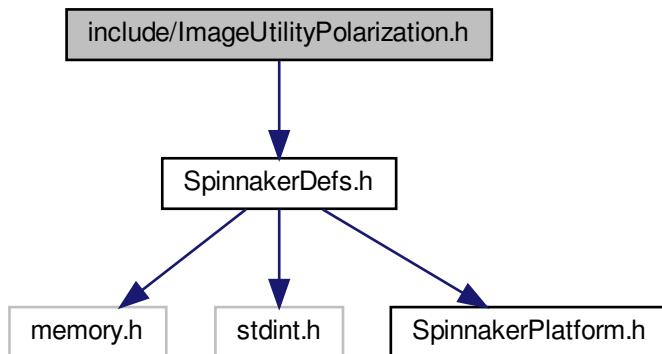
- class [ImageUtilityHeatmap](#)  
*Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.*

## Namespaces

- [Spinnaker](#)

## 11.23 include/ImageUtilityPolarization.h File Reference

Include dependency graph for ImageUtilityPolarization.h:



## Classes

- class [ImageUtilityPolarization](#)

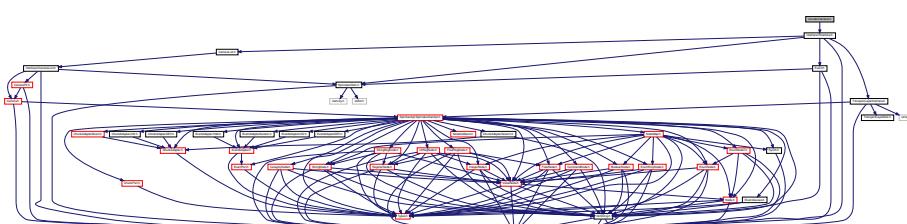
*Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.*

## Namespaces

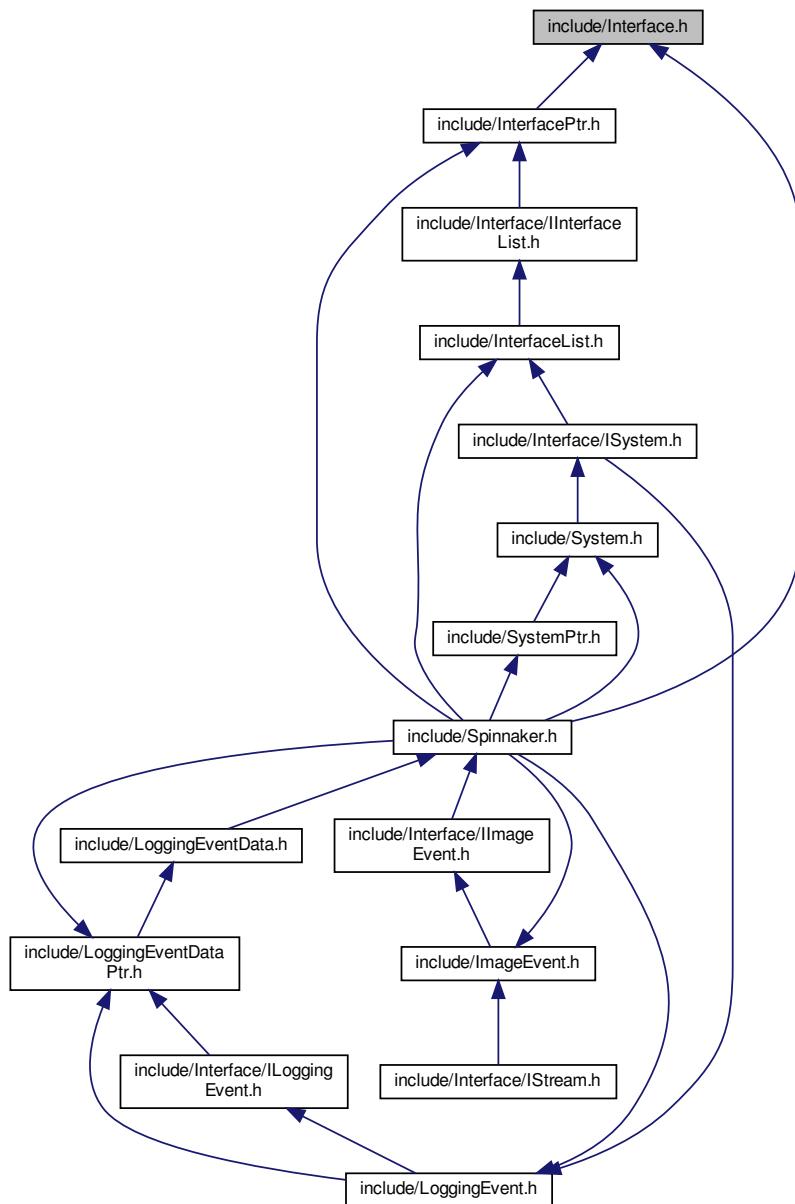
- [Spinnaker](#)

## 11.24 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](#)

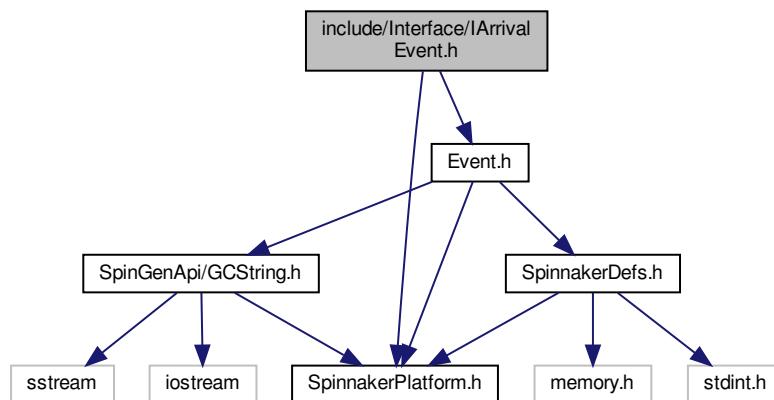
*An interface object which holds a list of cameras.*

## Namespaces

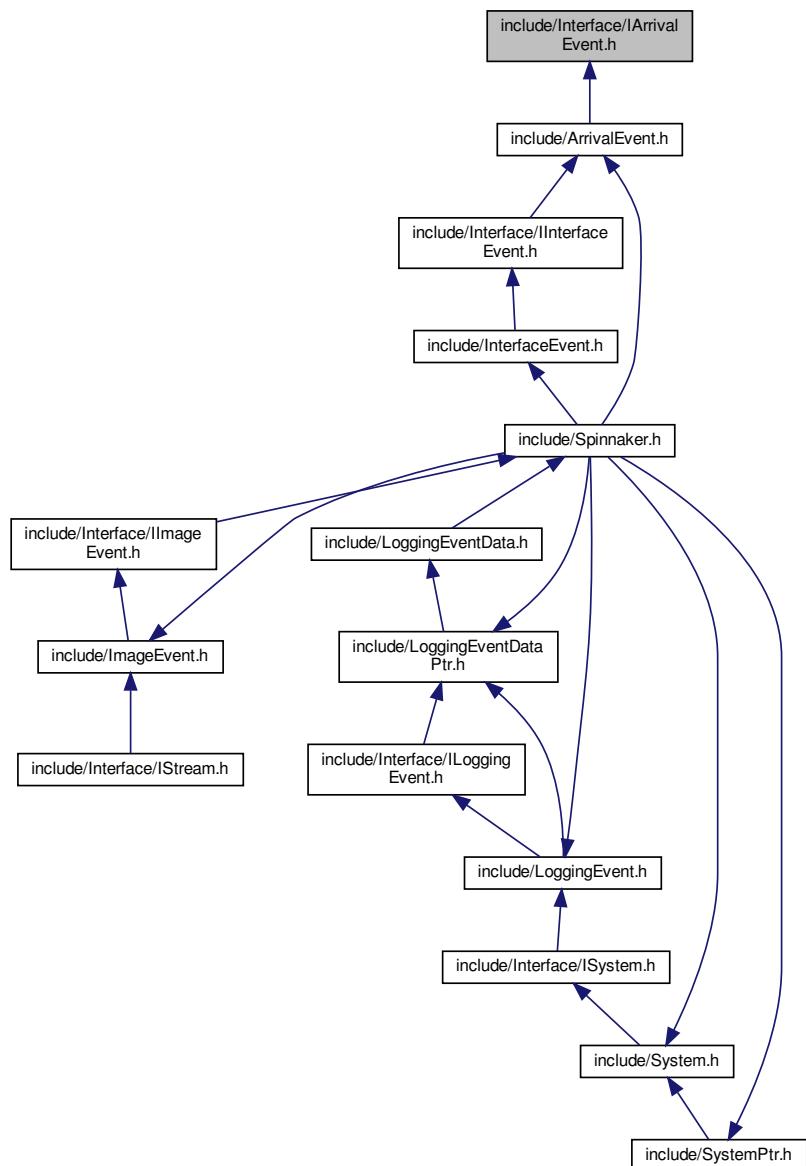
- [Spinnaker](#)

## 11.25 include/Interface/IArrivalEvent.h File Reference

Include dependency graph for IArrivalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

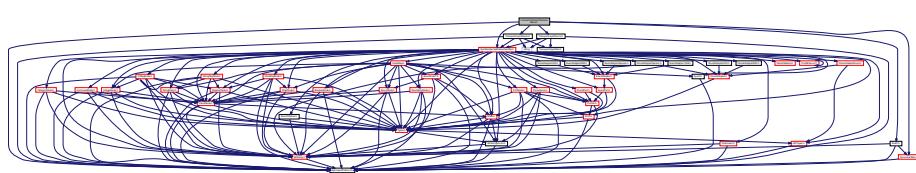
- class [IArrivalEvent](#)

## Namespaces

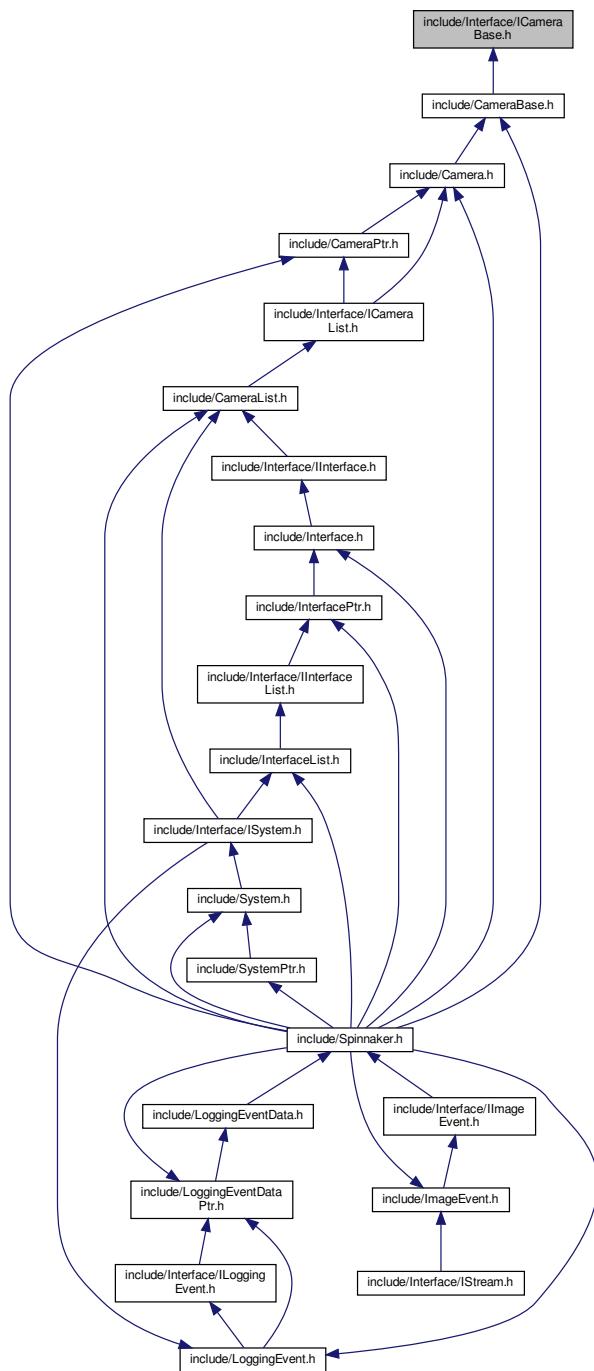
- [Spinnaker](#)

## 11.26 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](#)

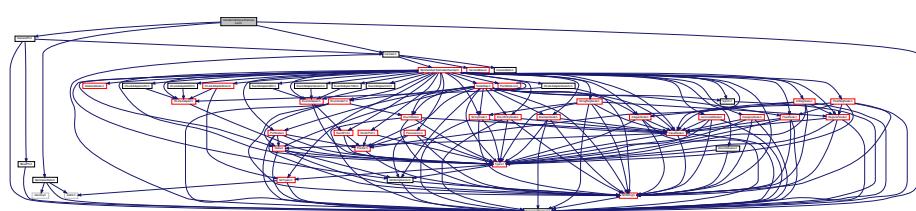
*The interface file for base class for the camera object.*

## Namespaces

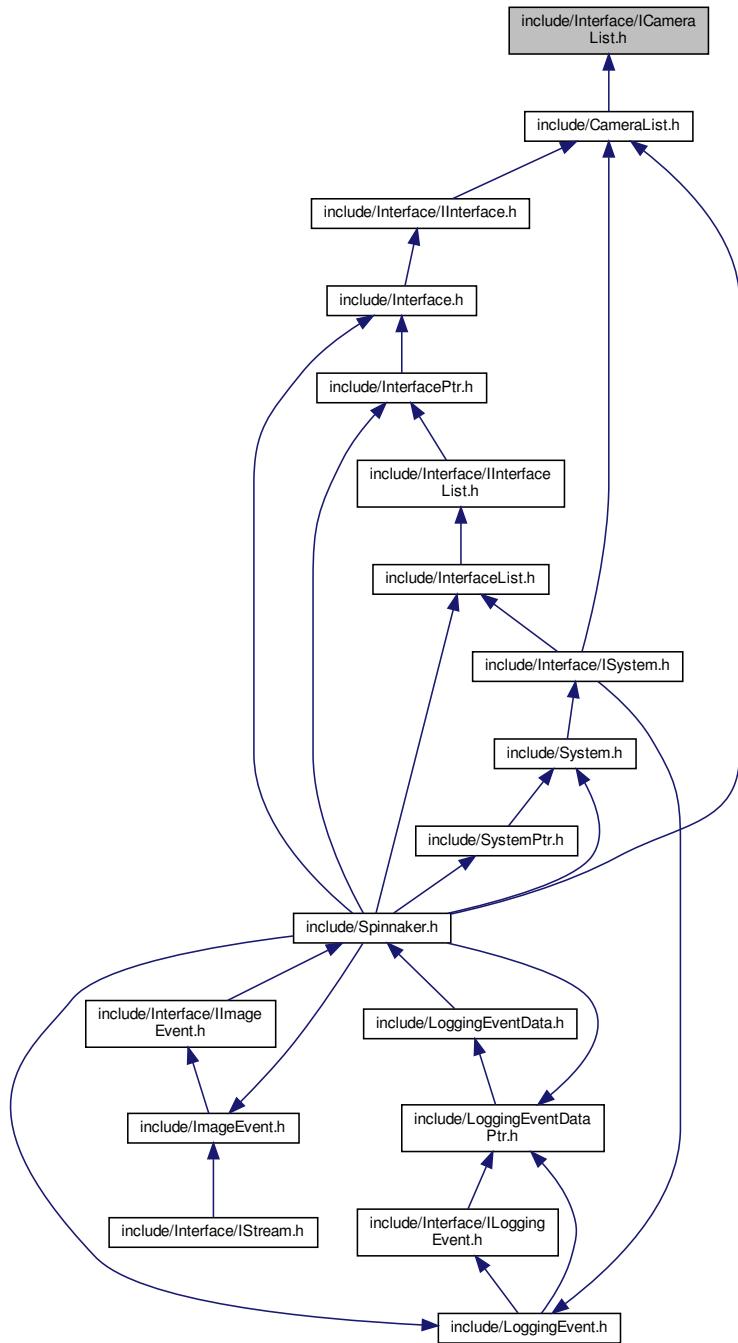
- [Spinnaker](#)

## 11.27 include/Interface/ICameraList.h File Reference

Include dependency graph for ICameraList.h:



This graph shows which files directly or indirectly include this file:



## Classes

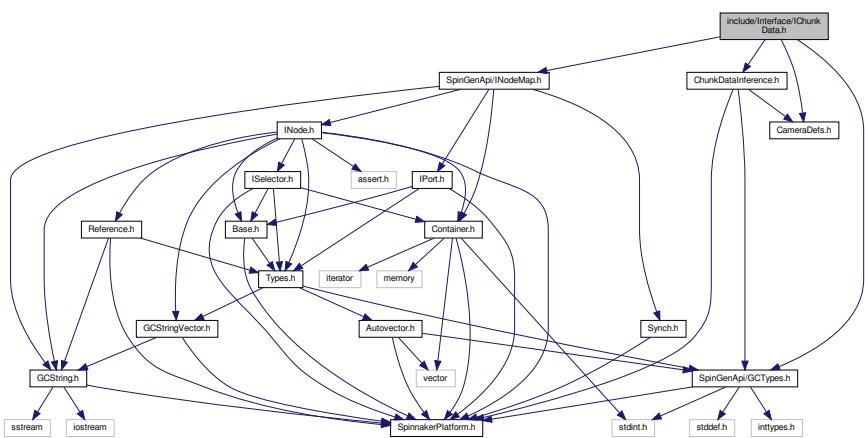
- class **ICameraList**  
*Used to hold a list of camera objects.*

## Namespaces

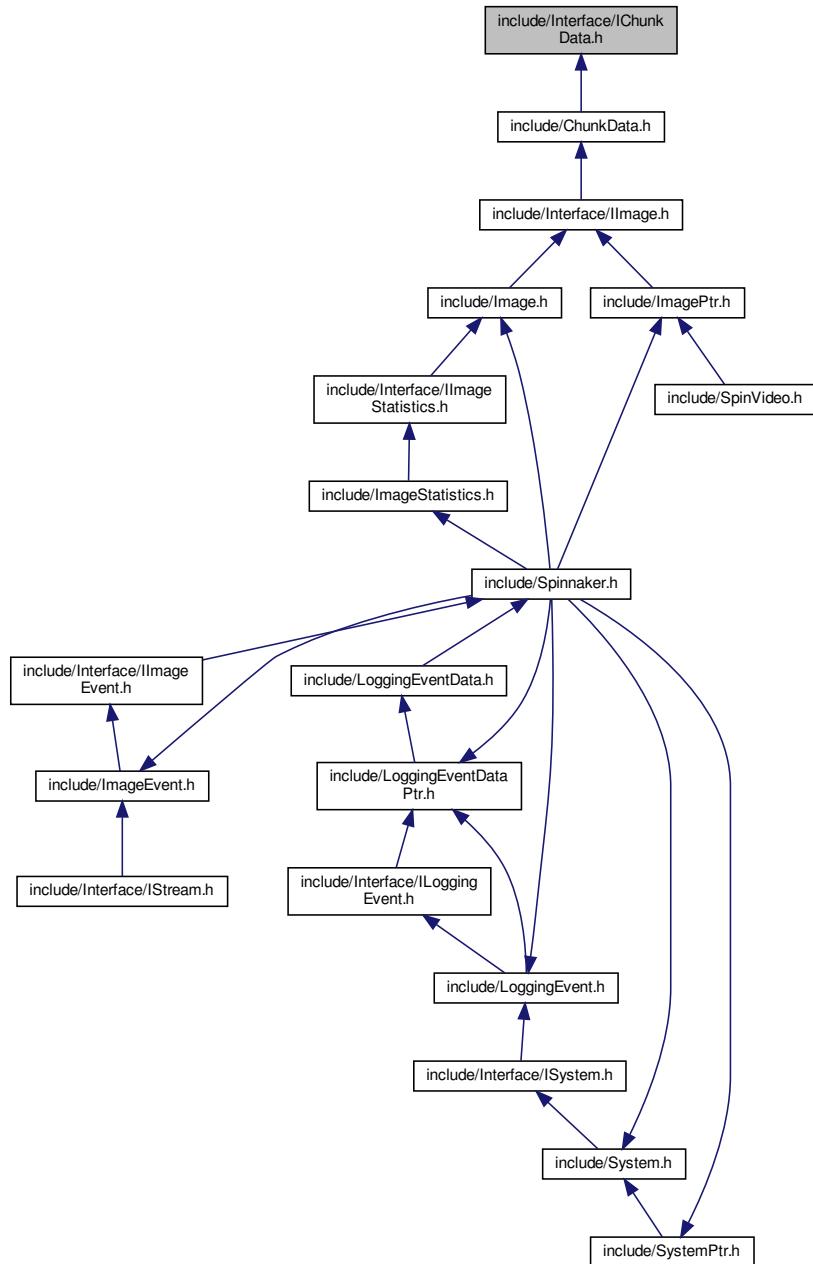
- Spinnaker

## 11.28 include/Interface/IChunkData.h File Reference

Include dependency graph for IChunkData.h:



This graph shows which files directly or indirectly include this file:



## Classes

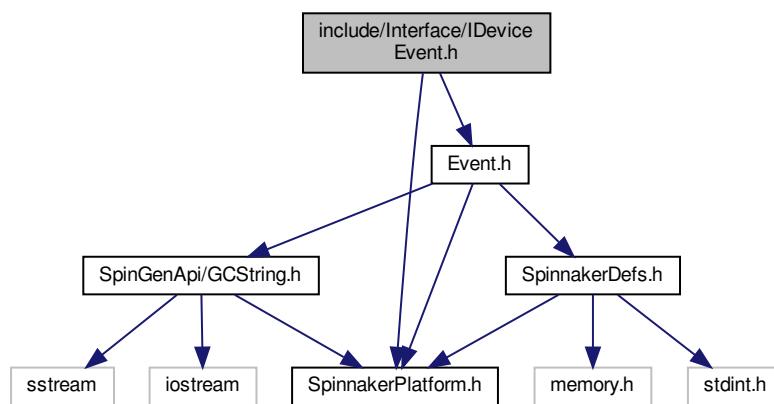
- class [IChunkData](#)  
*The Interface file for ChunkData.*

## Namespaces

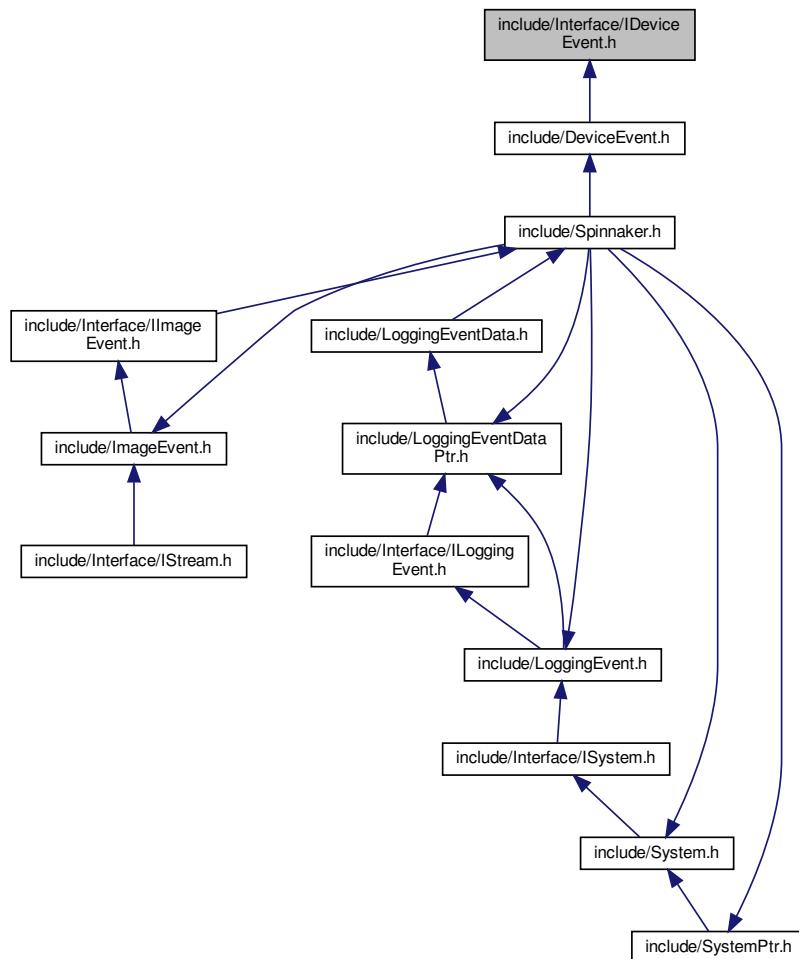
- [Spinnaker](#)

## 11.29 include/Interface/IDeviceEvent.h File Reference

Include dependency graph for IDeviceEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

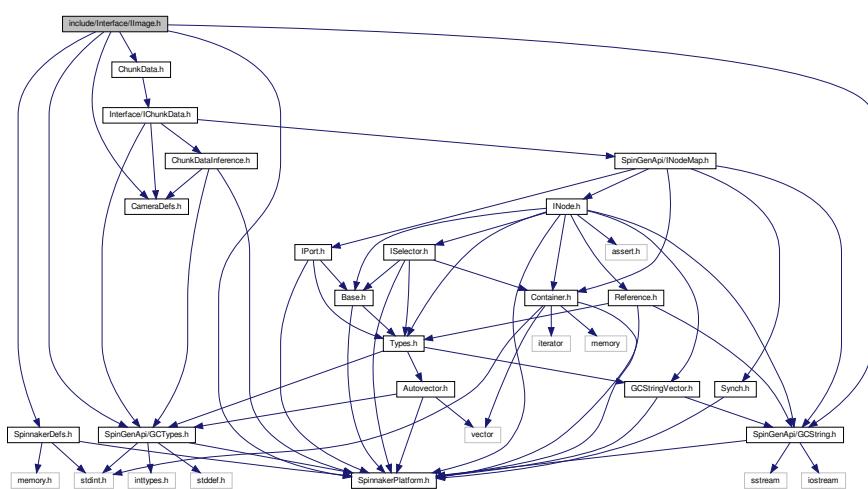
- class `IDeviceEvent`

## Namespaces

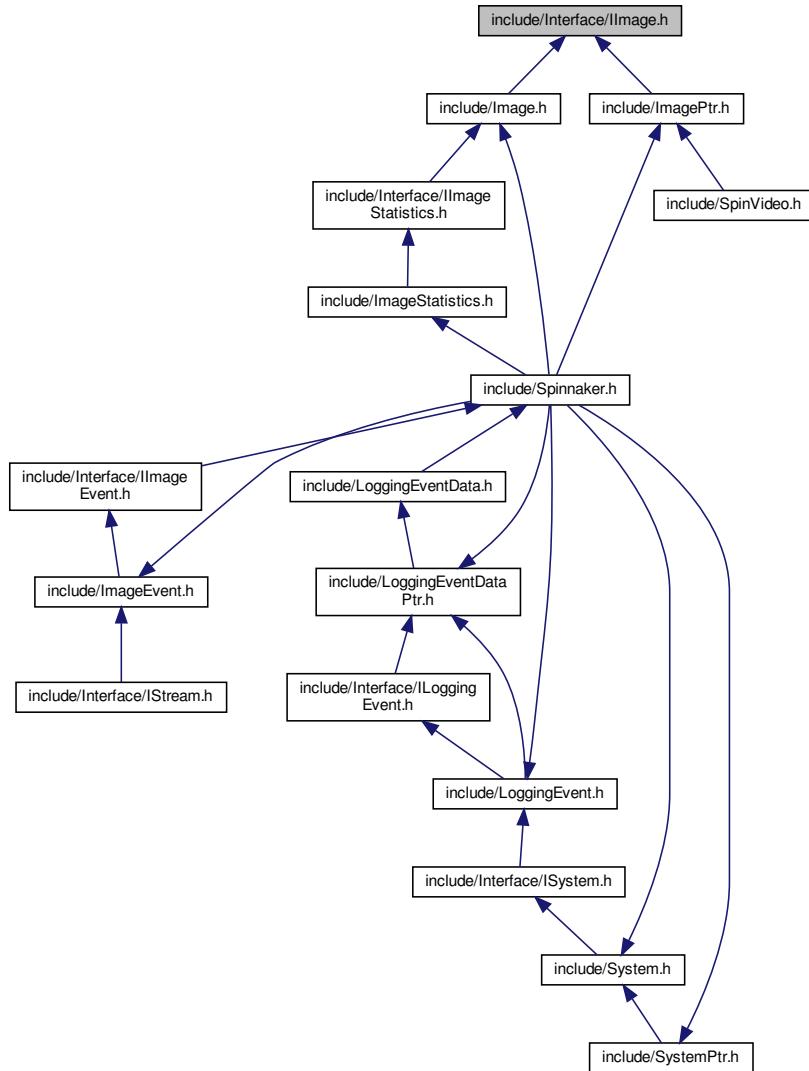
- `Spinnaker`

## 11.30 include/Interface/IImage.h File Reference

Include dependency graph for IImage.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [IImage](#)

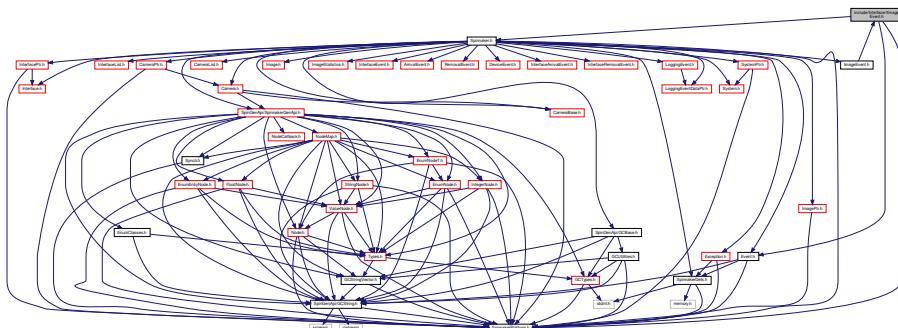
*The interface file for [Image](#).*

## Namespaces

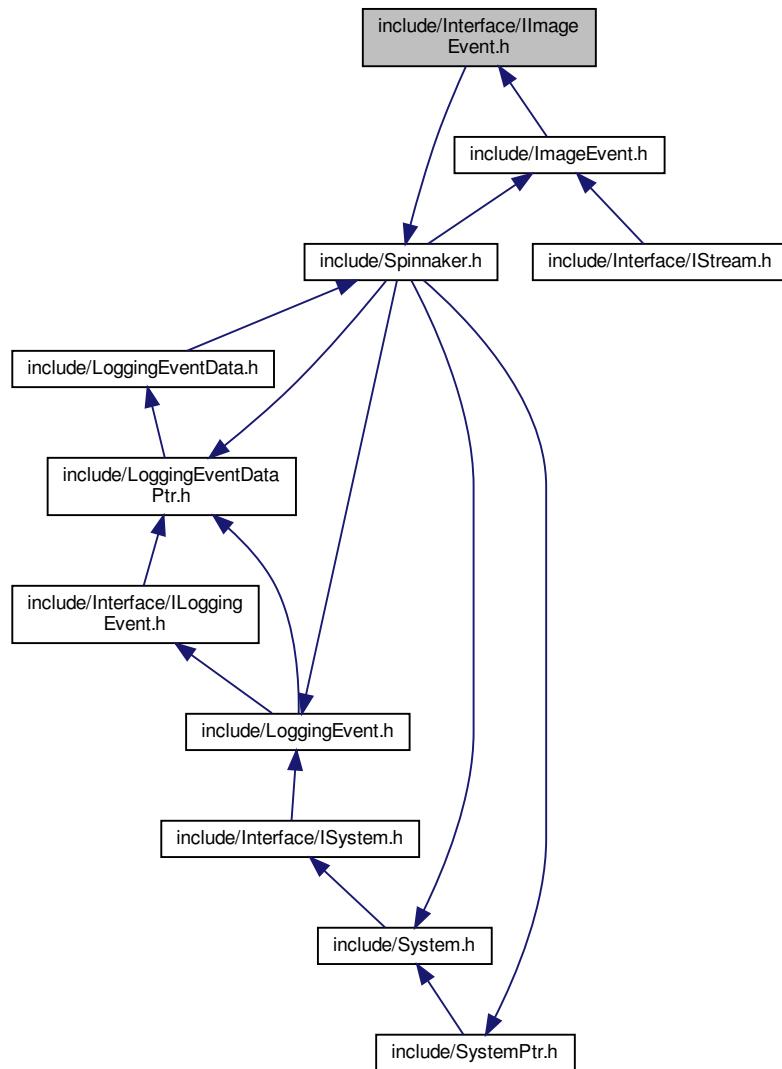
- [Spinnaker](#)

## 11.31 include/Interface/IImageEvent.h File Reference

Include dependency graph for IImageEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

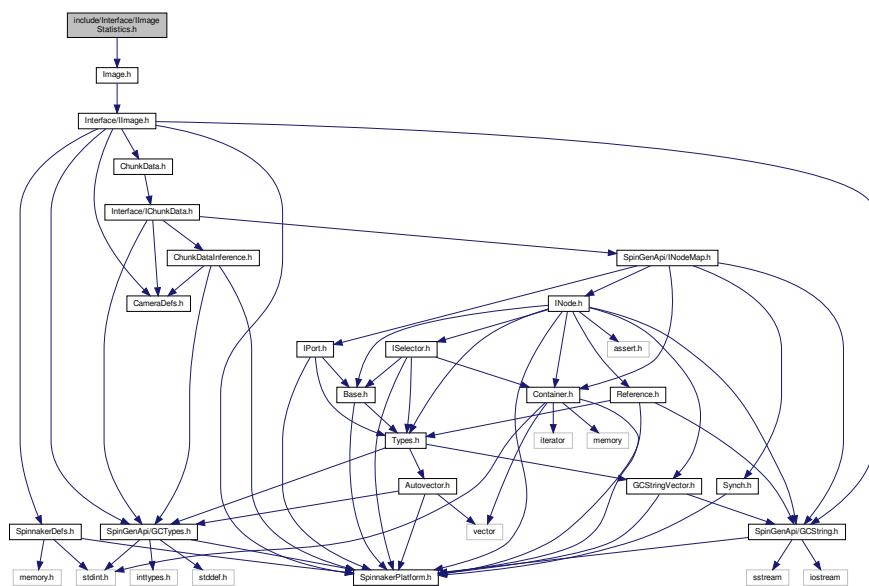
- class [IImageEvent](#)

## Namespaces

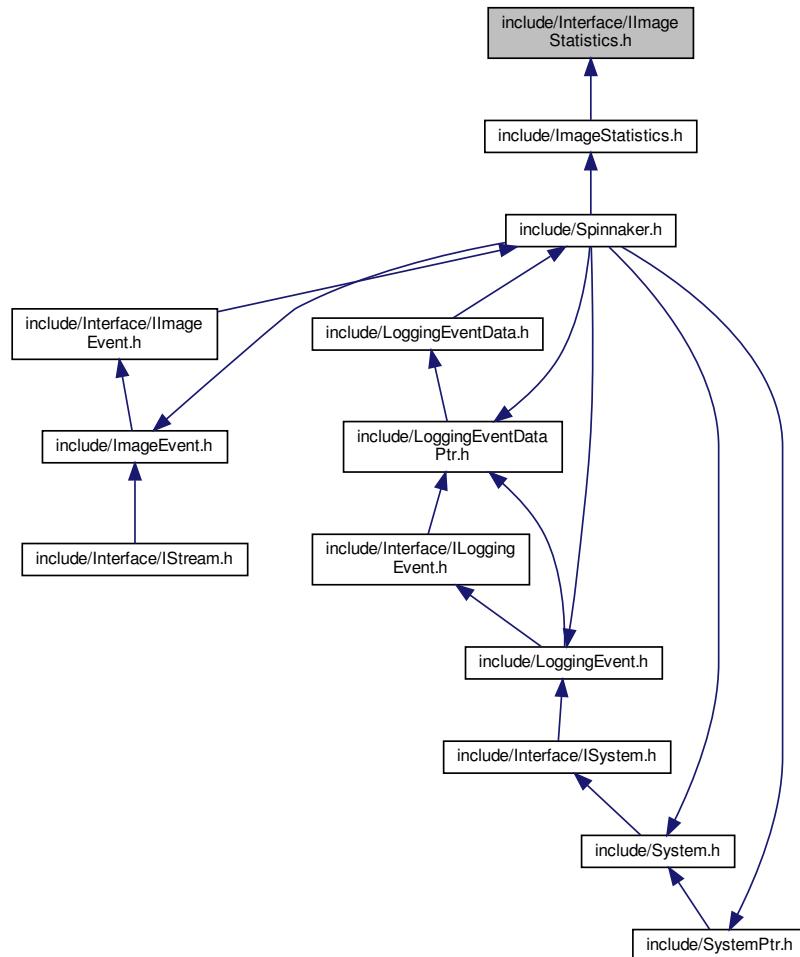
- [Spinnaker](#)

## 11.32 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](#)

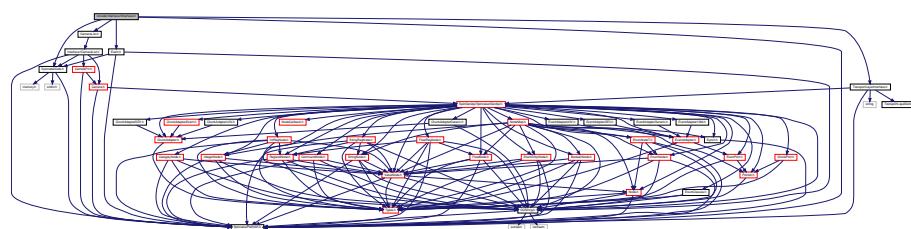
*The interface file for image statistics.*

## Namespaces

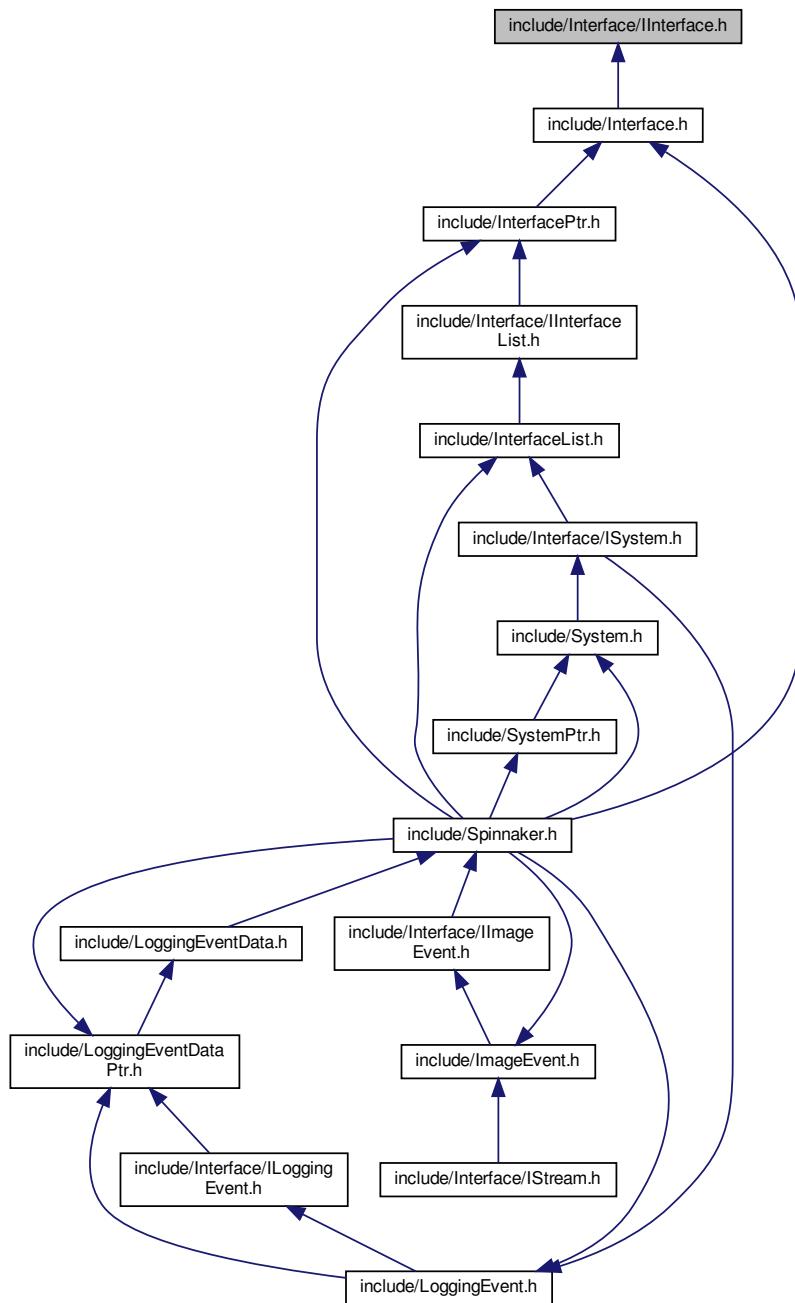
- [Spinnaker](#)

### 11.33 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](#)

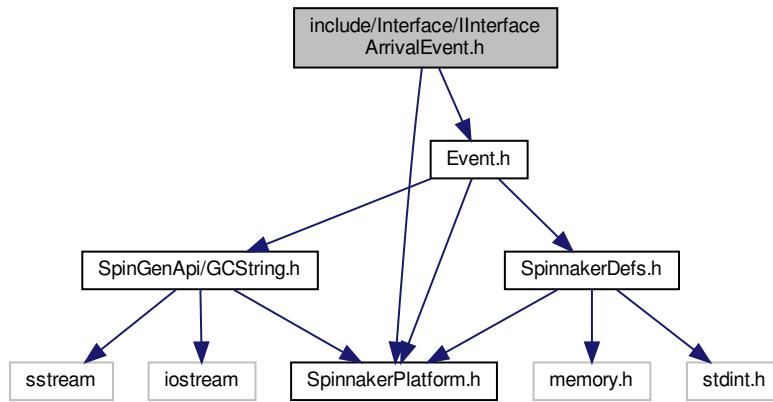
*The interface file for [Interface](#).*

## Namespaces

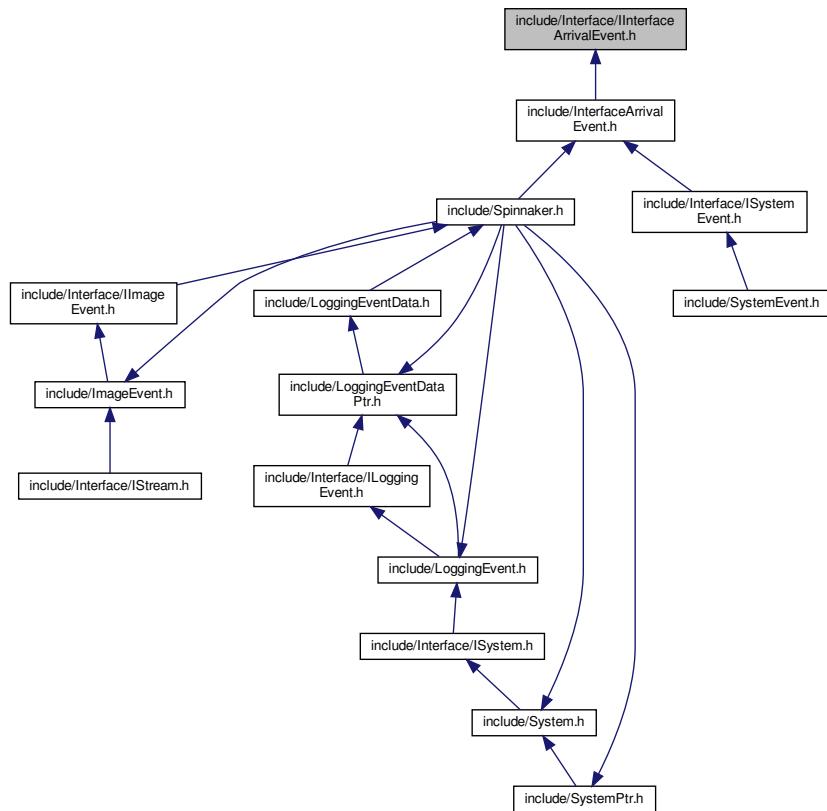
- [Spinnaker](#)

## 11.34 include/Interface/IInterfaceArrivalEvent.h File Reference

Include dependency graph for IInterfaceArrivalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

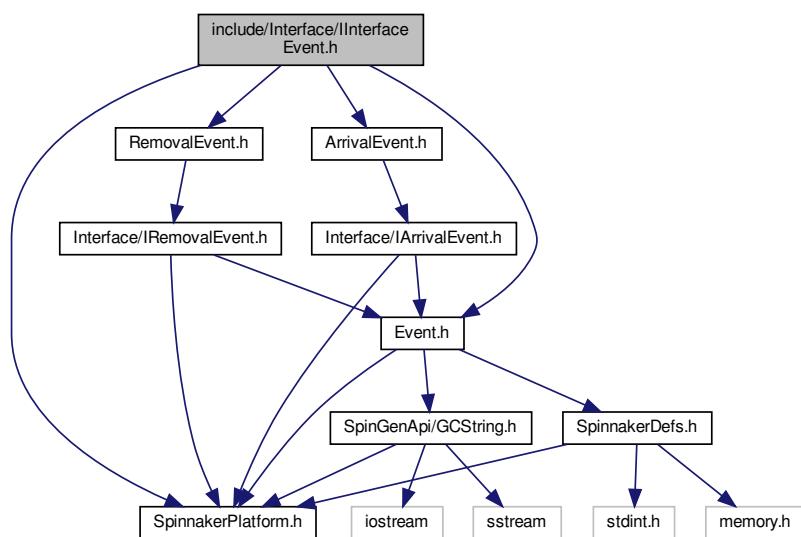
- class [IInterfaceArrivalEvent](#)

## Namespaces

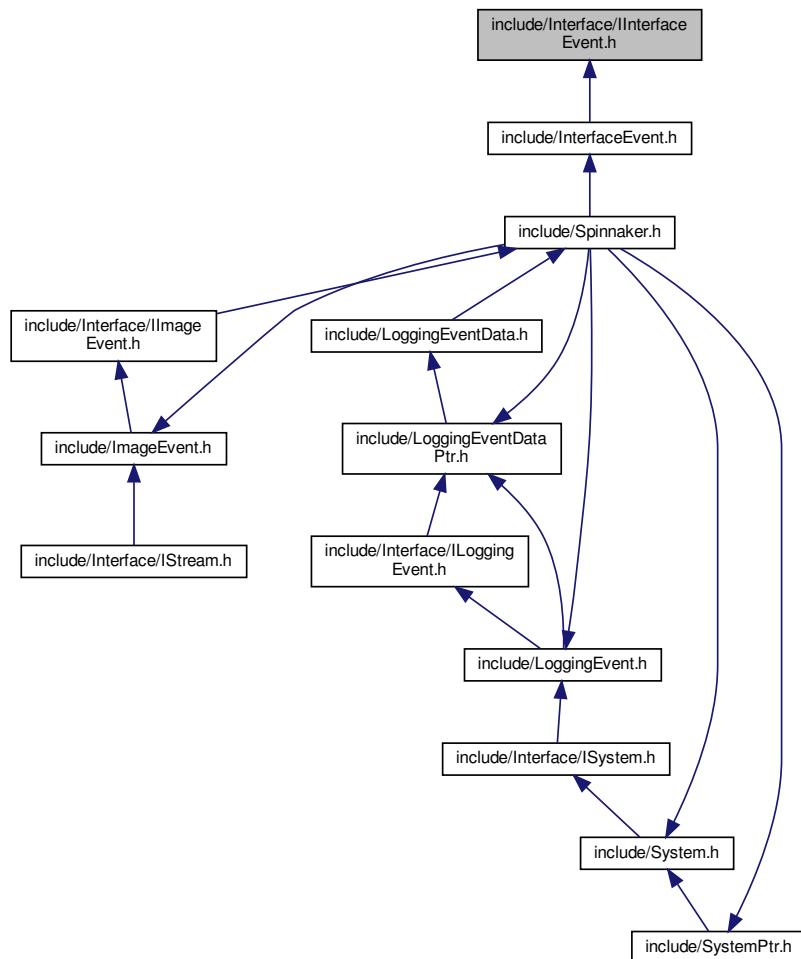
- [Spinnaker](#)

## 11.35 include/Interface/IInterfaceEvent.h File Reference

Include dependency graph for IInterfaceEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

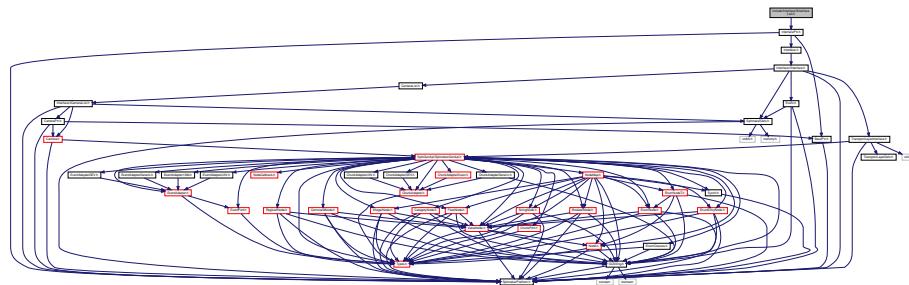
- class `IInterfaceEvent`

## Namespaces

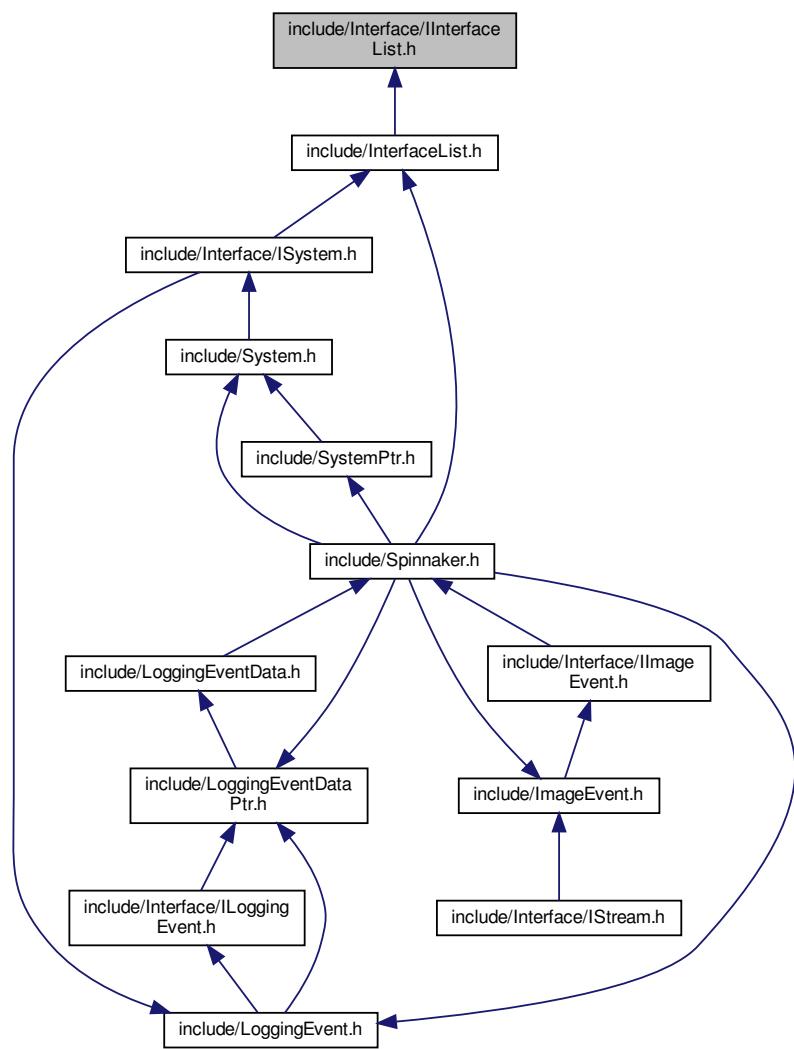
- `Spinnaker`

## 11.36 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 [IInterfaceList](#)

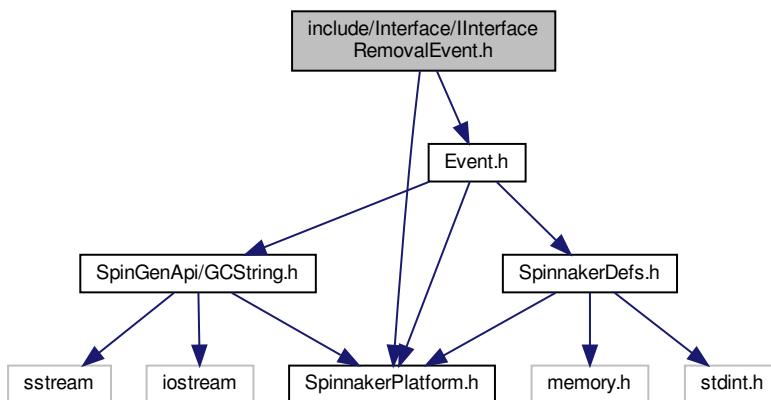
The interface file for [InterfaceList](#) class.

## Namespaces

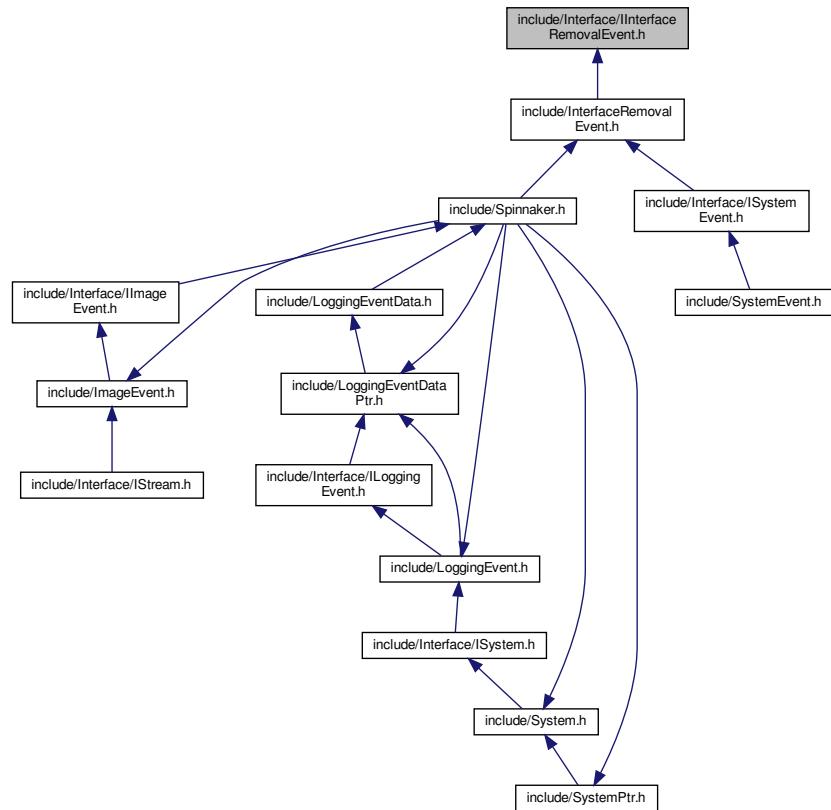
- [Spinnaker](#)

## 11.37 include/Interface/IInterfaceRemovalEvent.h File Reference

Include dependency graph for IInterfaceRemovalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

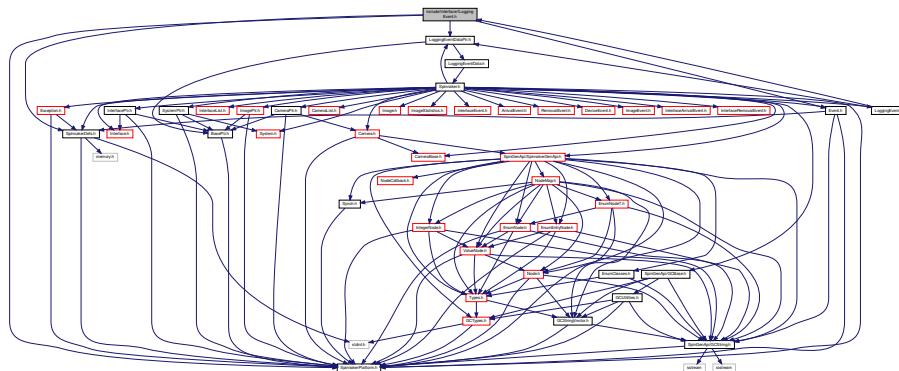
- class `IInterfaceRemovalEvent`

## Namespaces

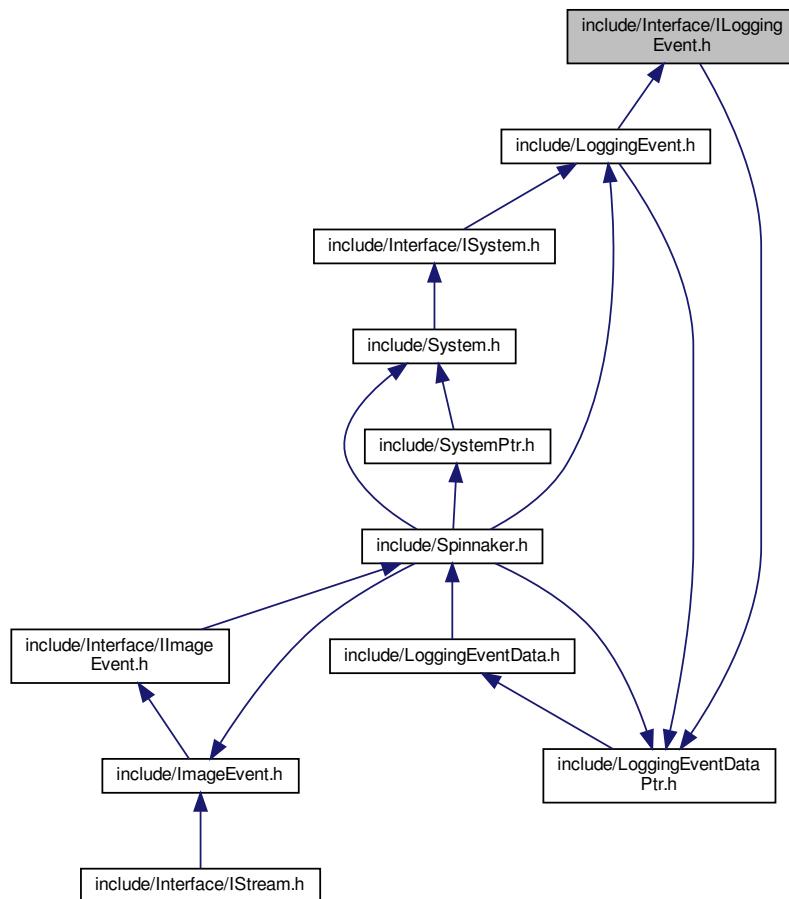
- `Spinnaker`

## 11.38 include/Interface/ILoggingEvent.h File Reference

Include dependency graph for ILoggingEvent.h:



This graph shows which files directly or indirectly include this file:



### Classes

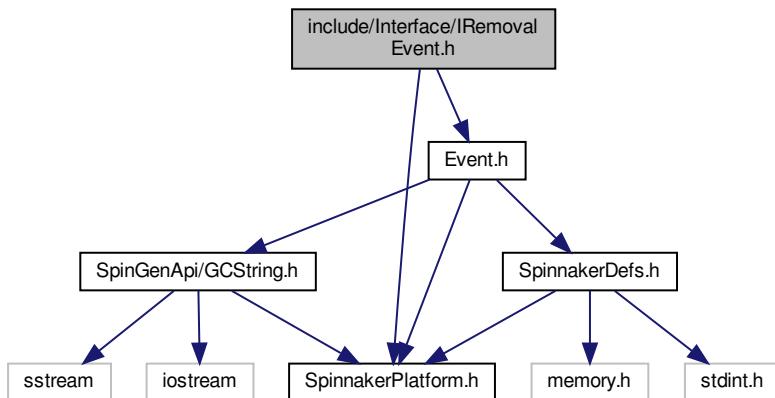
- class [ILoggingEvent](#)

## Namespaces

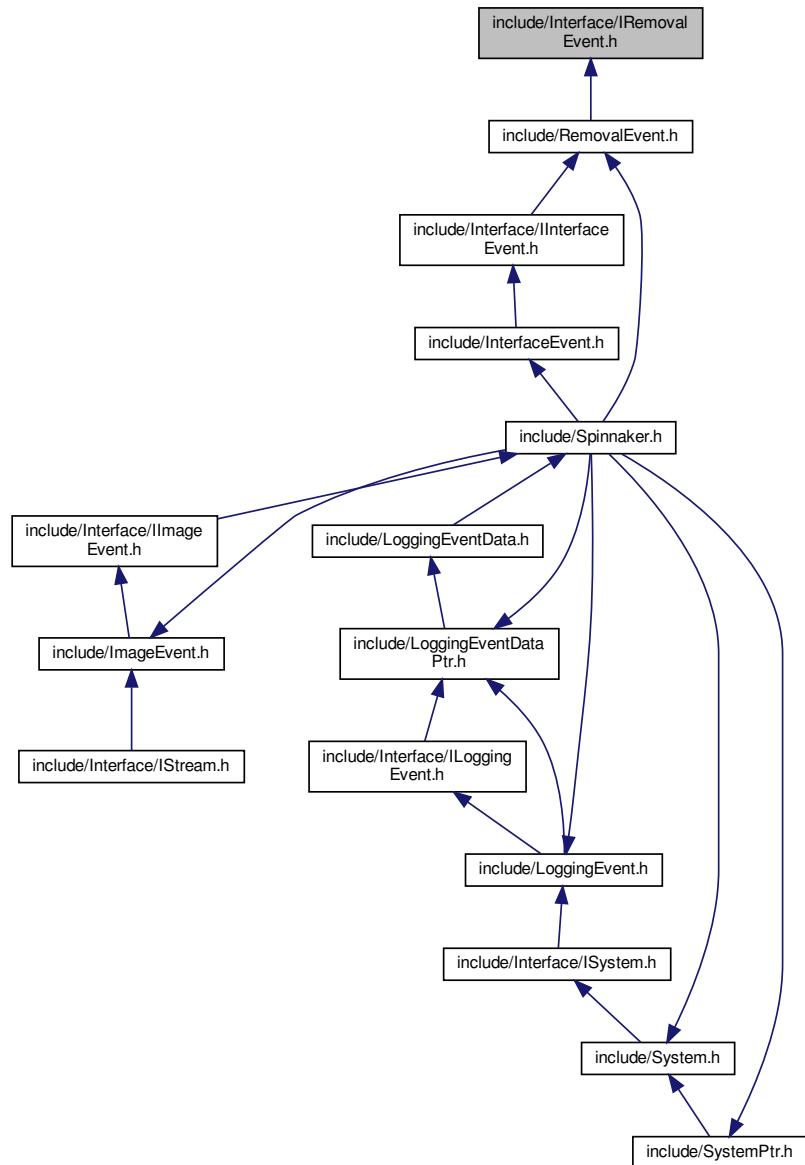
- [Spinnaker](#)

## 11.39 include/Interface/IRemovalEvent.h File Reference

Include dependency graph for IRemovalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

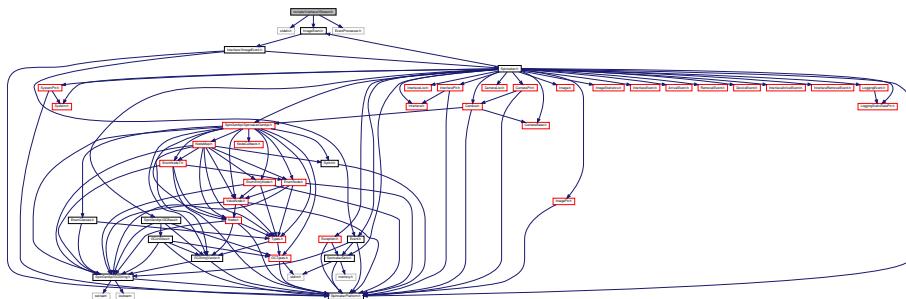
- class IRemovalEvent

## Namespaces

- Spinnaker

## 11.40 include/Interface/IStream.h File Reference

Include dependency graph for IStream.h:



### Classes

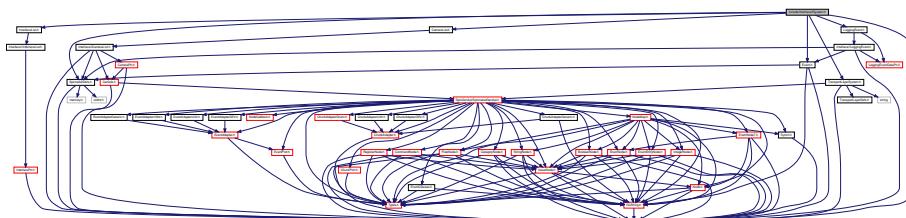
- class [IDataStream](#)

### Namespaces

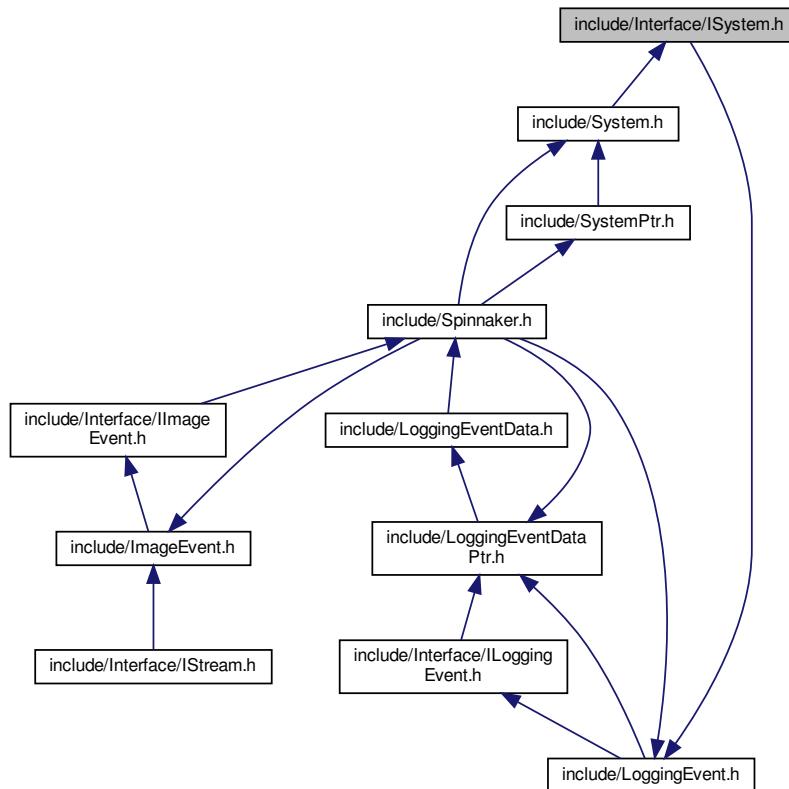
- [Spinnaker](#)

## 11.41 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](#)

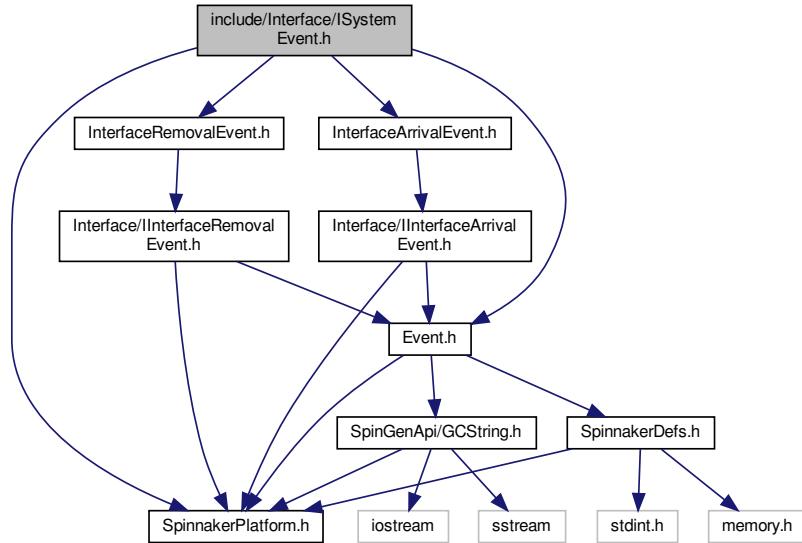
*The interface file for [System](#).*

## Namespaces

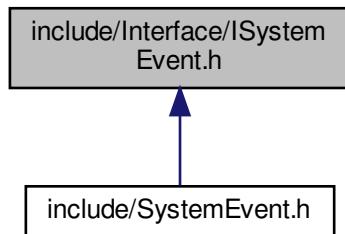
- [Spinnaker](#)

## 11.42 include/Interface/ISystemEvent.h File Reference

Include dependency graph for ISystemEvent.h:



This graph shows which files directly or indirectly include this file:



### Classes

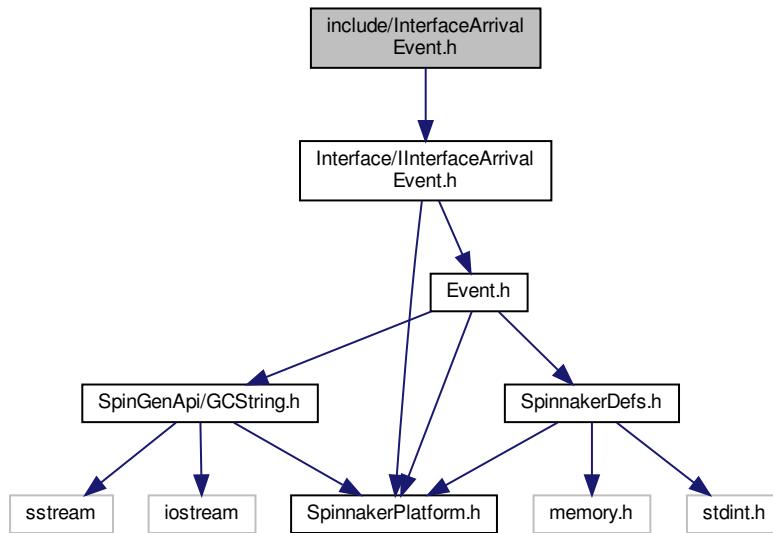
- class [ISystemEvent](#)

### Namespaces

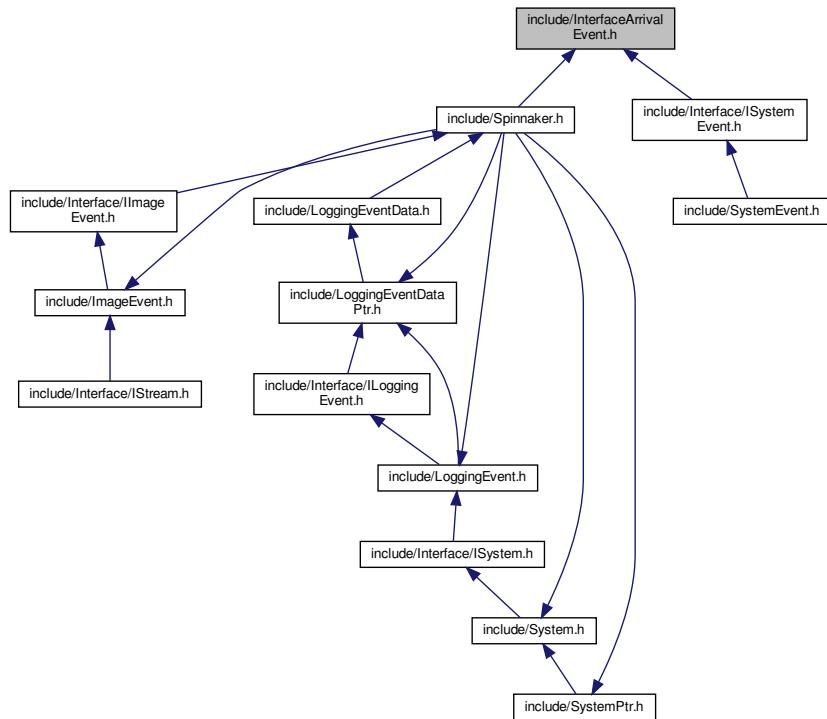
- [Spinnaker](#)

## 11.43 include/InterfaceArrivalEvent.h File Reference

Include dependency graph for InterfaceArrivalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [InterfaceArrivalEvent](#)

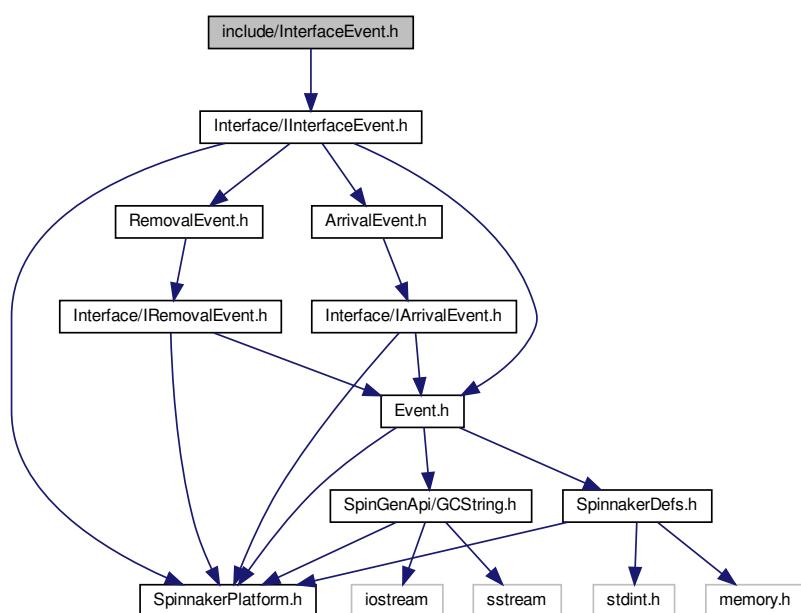
*An event handler for capturing the interface arrival event.*

## Namespaces

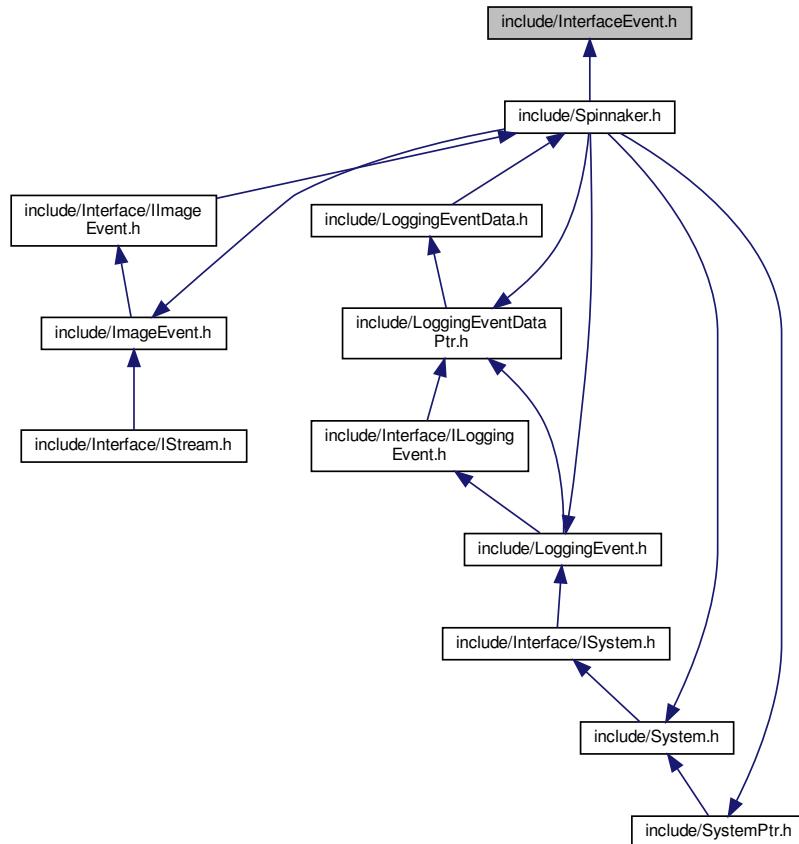
- [Spinnaker](#)

## 11.44 include/InterfaceEvent.h File Reference

Include dependency graph for InterfaceEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [InterfaceEvent](#)

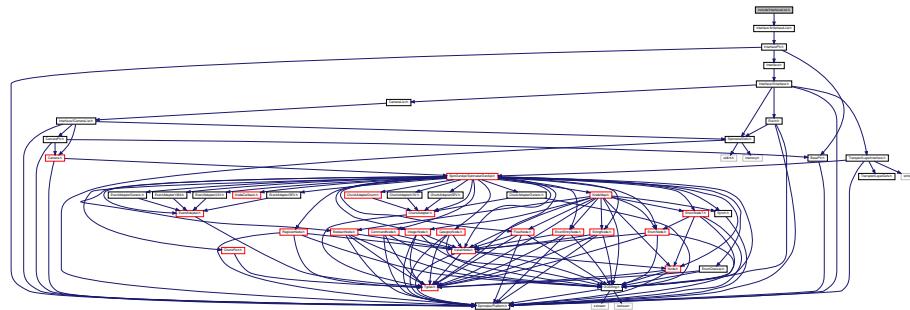
A handler to device arrival and removal events on all interfaces.

## Namespaces

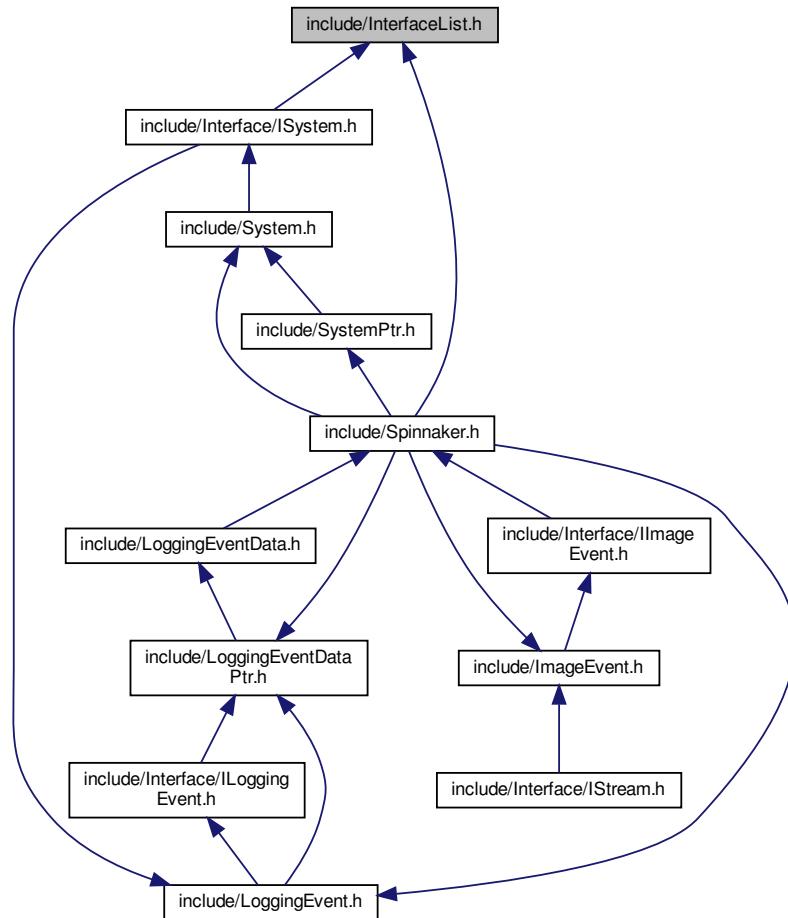
- [Spinnaker](#)

## 11.45 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](#)

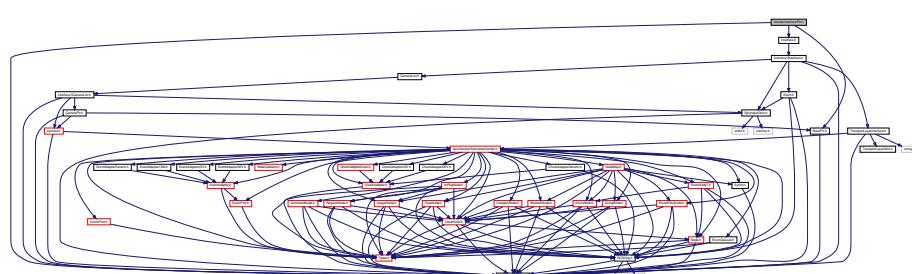
*A list of the available interfaces on the system.*

## Namespaces

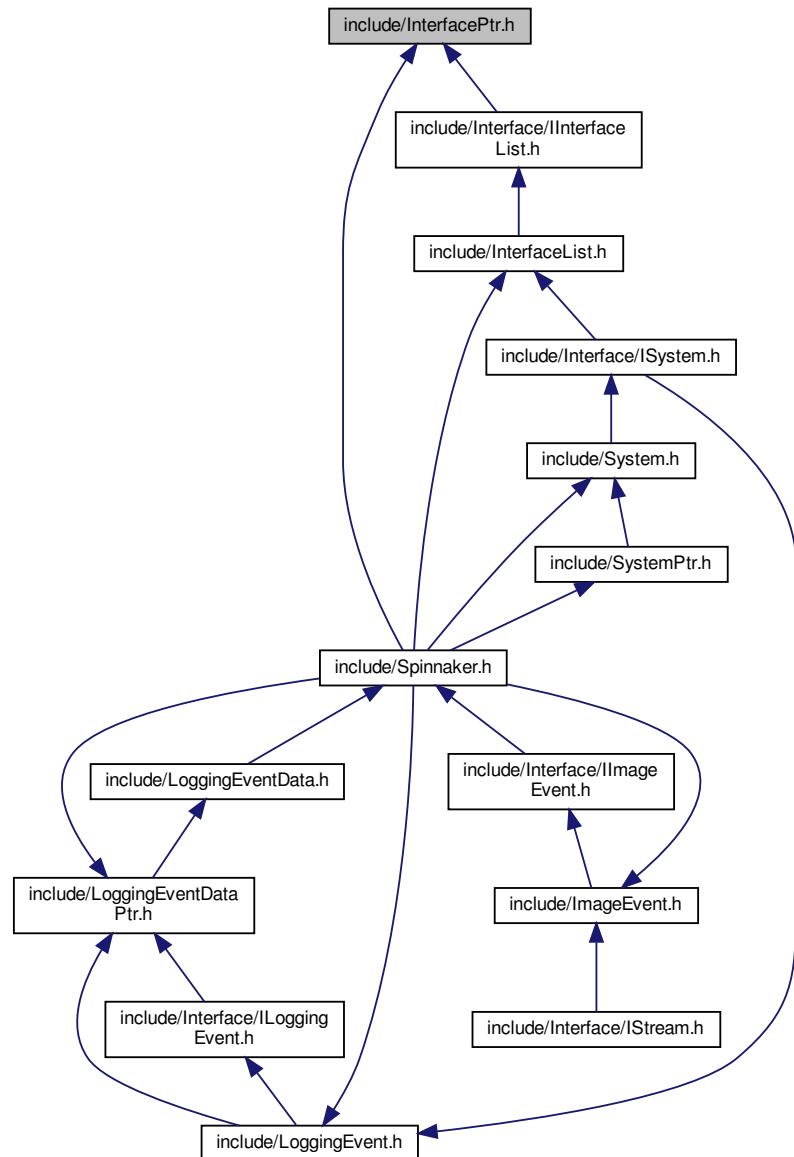
- [Spinnaker](#)

## 11.46 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](#)

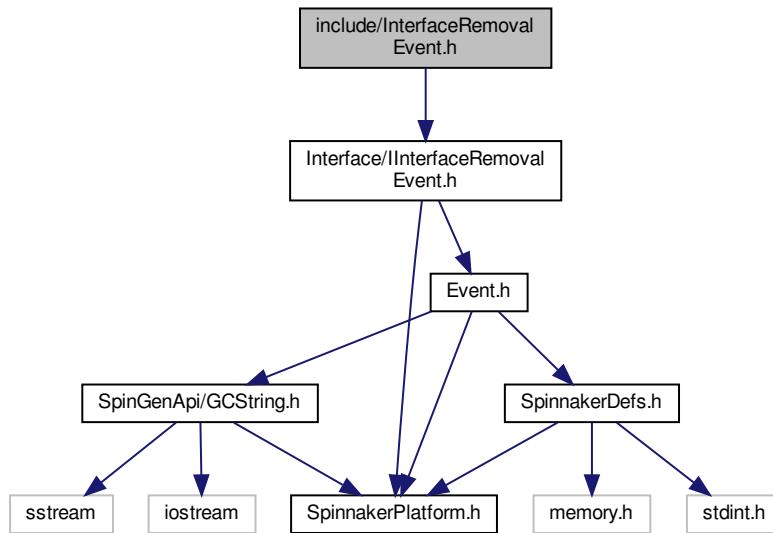
*A reference tracked pointer to the interface object.*

## Namespaces

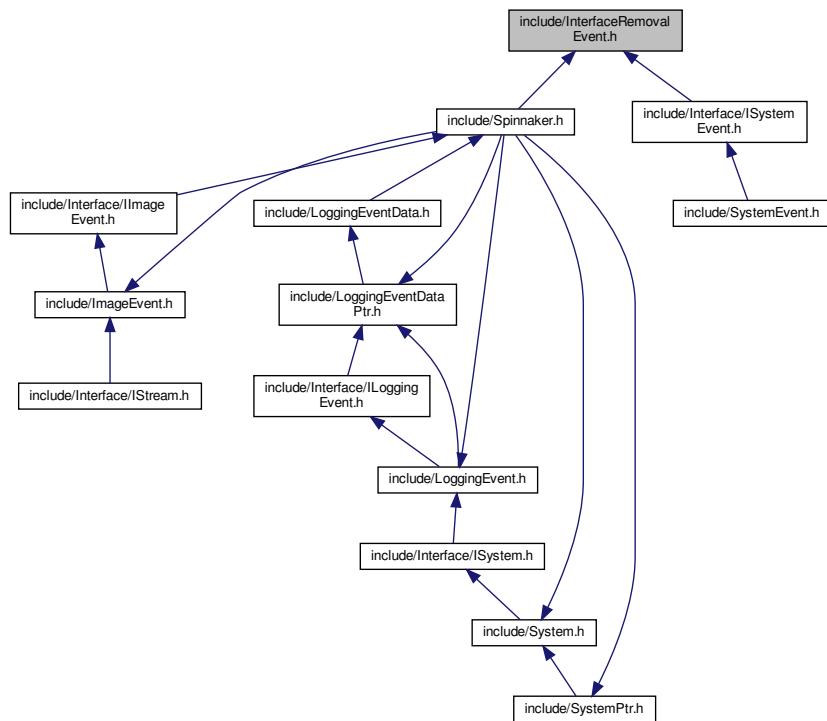
- [Spinnaker](#)

## 11.47 include/InterfaceRemovalEvent.h File Reference

Include dependency graph for InterfaceRemovalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `InterfaceRemovalEvent`

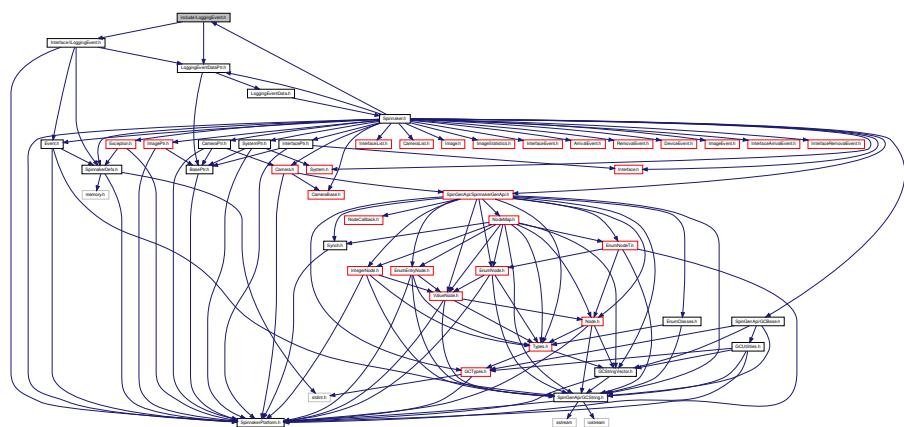
*An event handler for capturing the interface removal event.*

## Namespaces

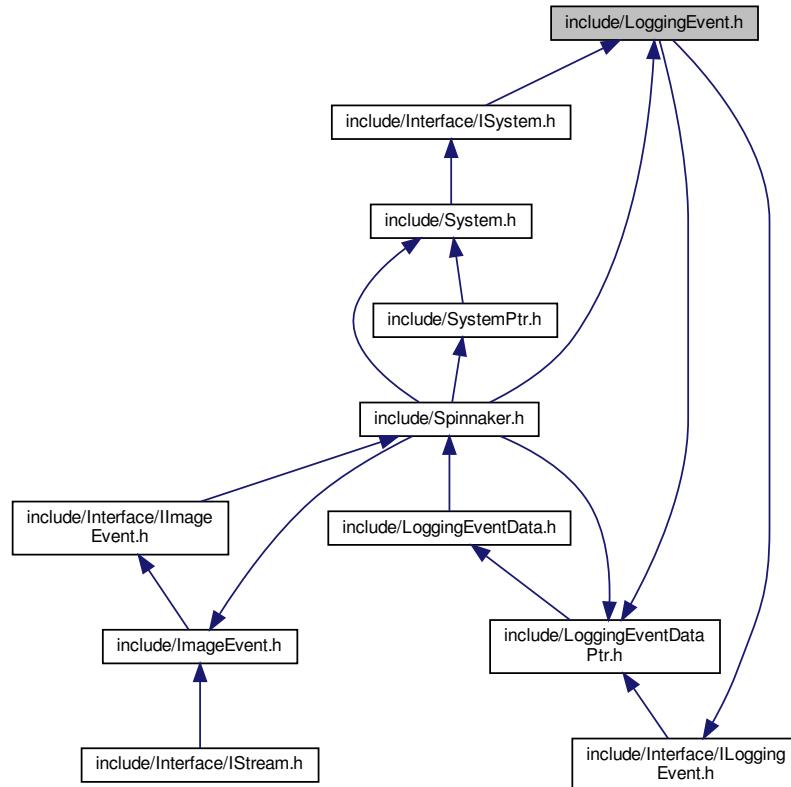
- Spinnaker

## 11.48 include/LoggingEvent.h File Reference

Include dependency graph for LoggingEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class LoggingEvent

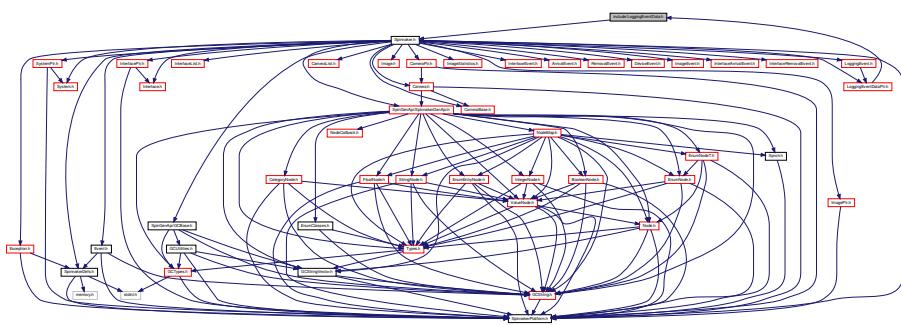
*An event handler for capturing the device logging event.*

## Namespaces

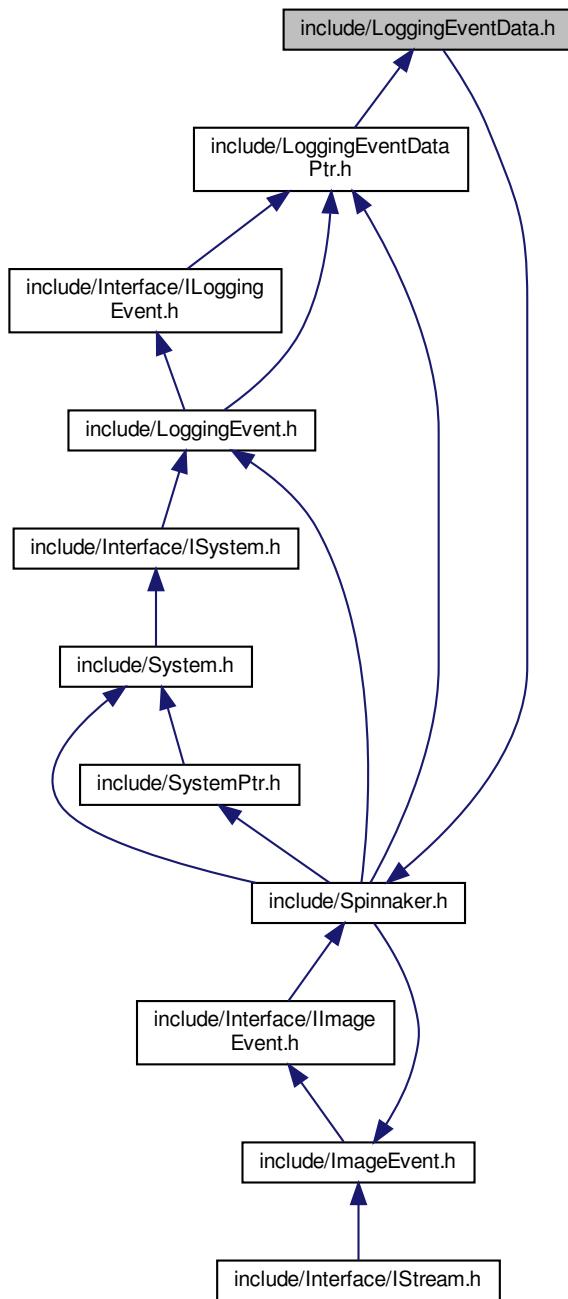
- Spinnaker

## 11.49 include/LoggingEventData.h File Reference

Include dependency graph for LoggingEventData.h:



This graph shows which files directly or indirectly include this file:



## Classes

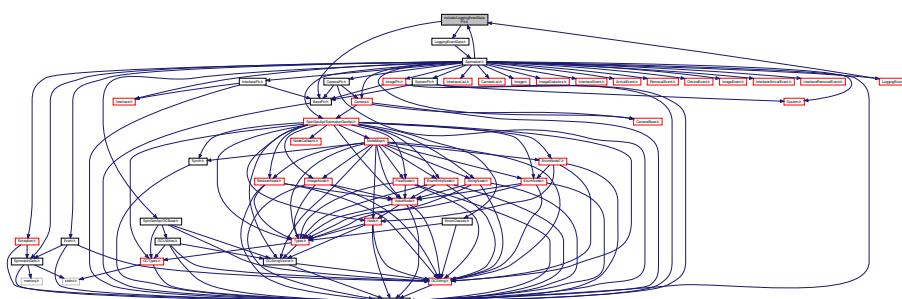
- class [LoggingEventData](#)  
*The `LoggingEventData` object.*

## Namespaces

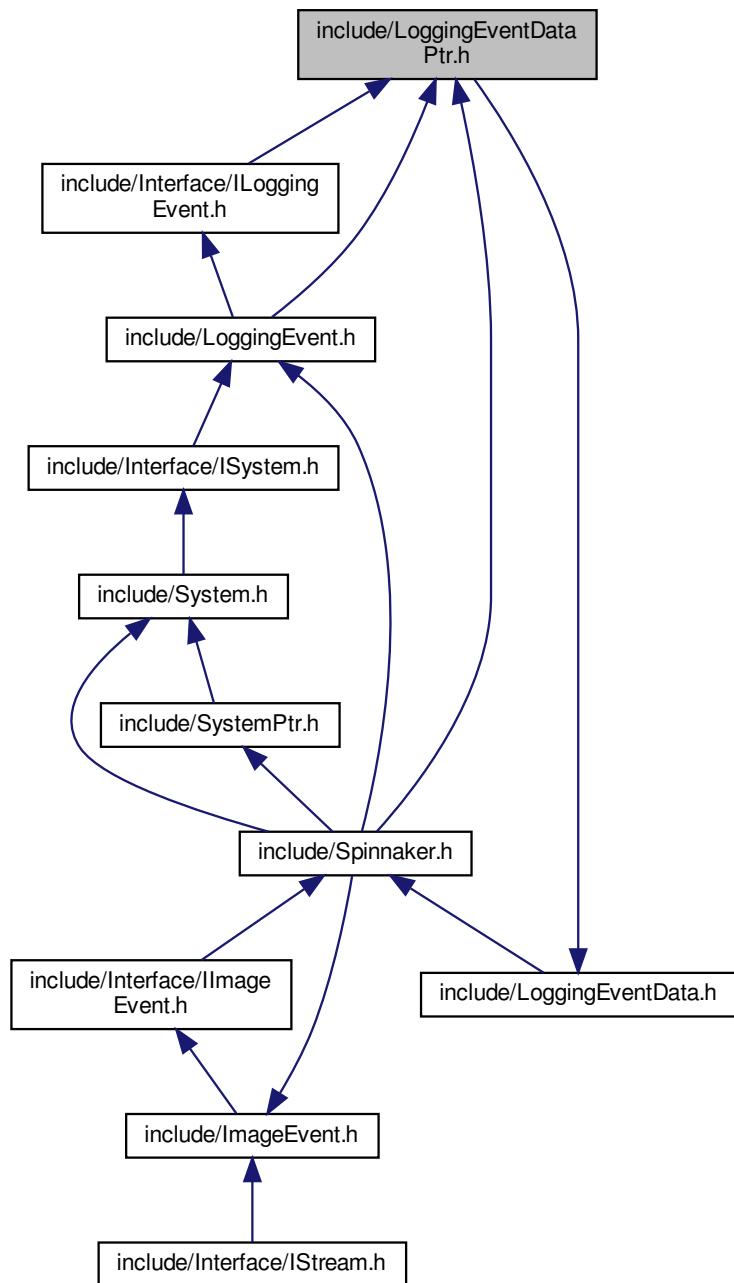
- [Spinnaker](#)

## 11.50 include/LoggingEventDataPtr.h File Reference

Include dependency graph for LoggingEventDataPtr.h:



This graph shows which files directly or indirectly include this file:



## Classes

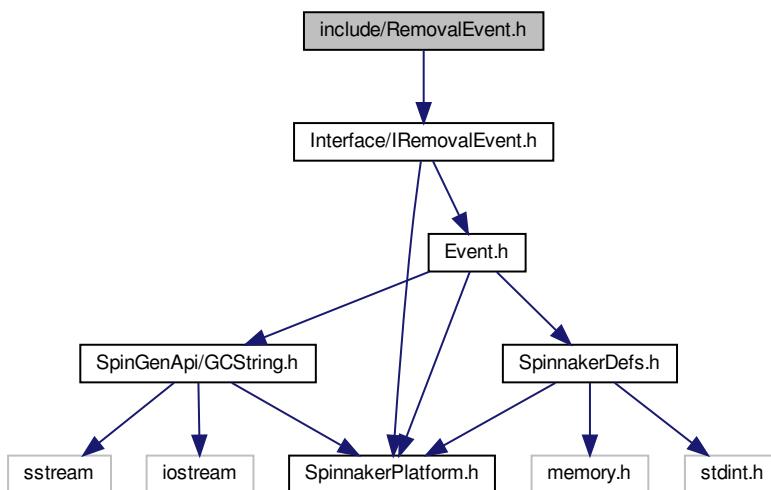
- class [LoggingEventDataPtr](#)  
*A reference tracked pointer to the `LoggingEvent` object.*

## Namespaces

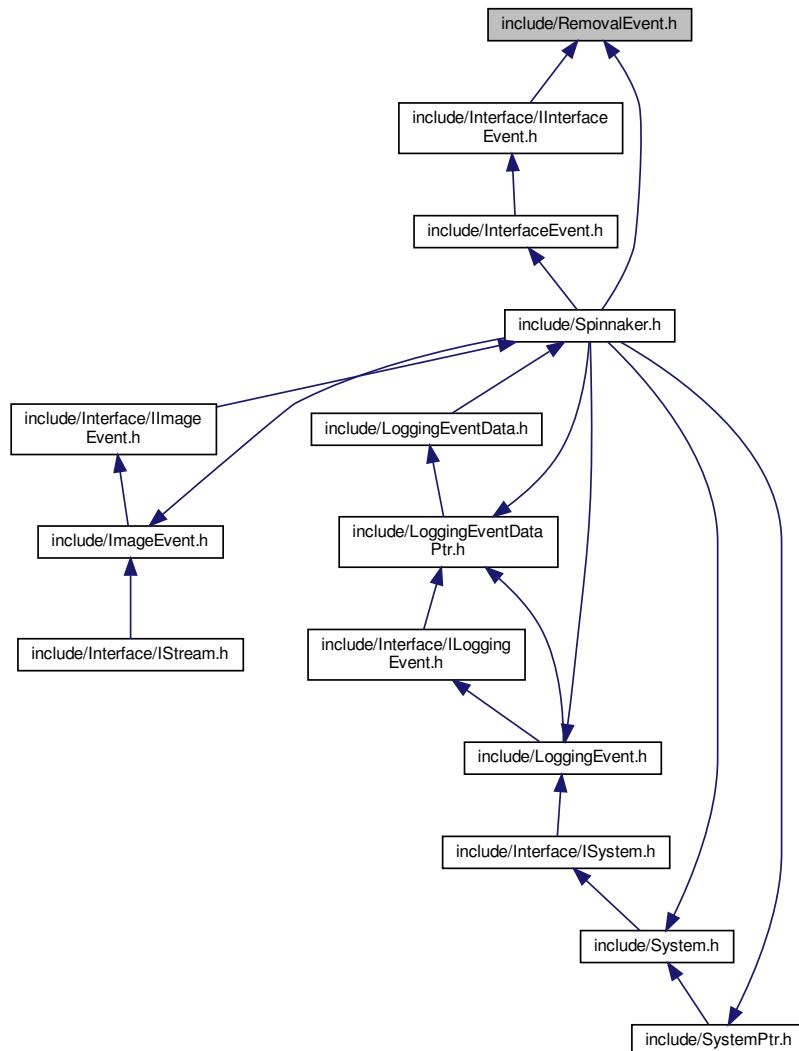
- [Spinnaker](#)

## 11.51 include/RemovalEvent.h File Reference

Include dependency graph for RemovalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [RemovalEvent](#)

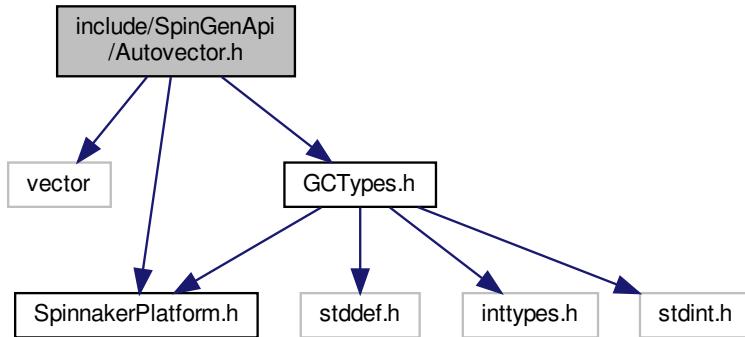
*An event handler for capturing the device removal event.*

## Namespaces

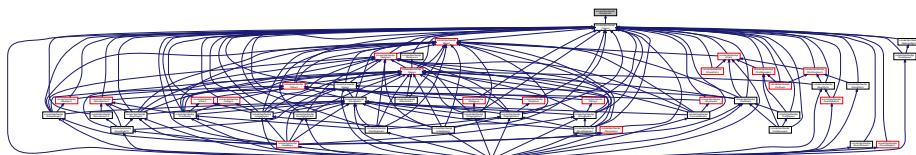
- [Spinnaker](#)

## 11.52 include/SpinGenApi/Autovector.h File Reference

Include dependency graph for Autovector.h:



This graph shows which files directly or indirectly include this file:



### Classes

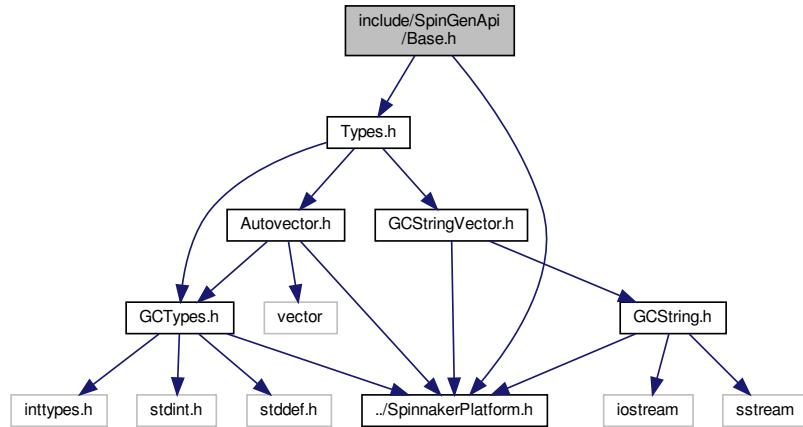
- class [int64\\_autovector\\_t](#)  
*Vector of integers with reference counting.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*

### Namespaces

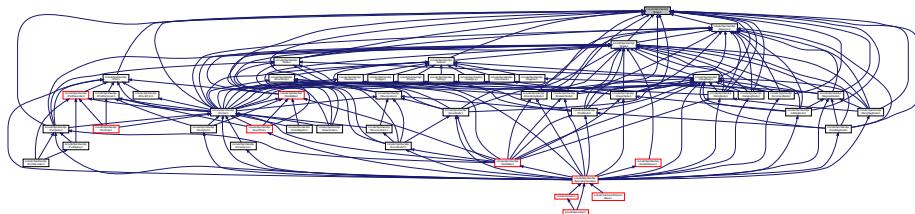
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.53 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

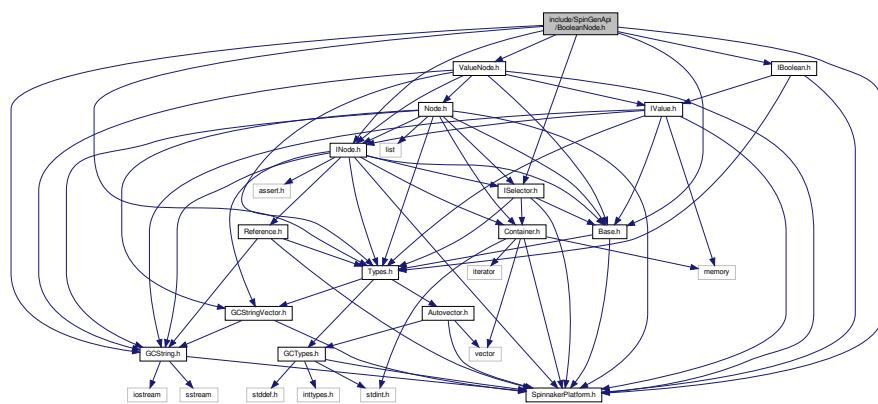
## Variables

- interface SPINNAKER\_API\_ABSTRACT IBase

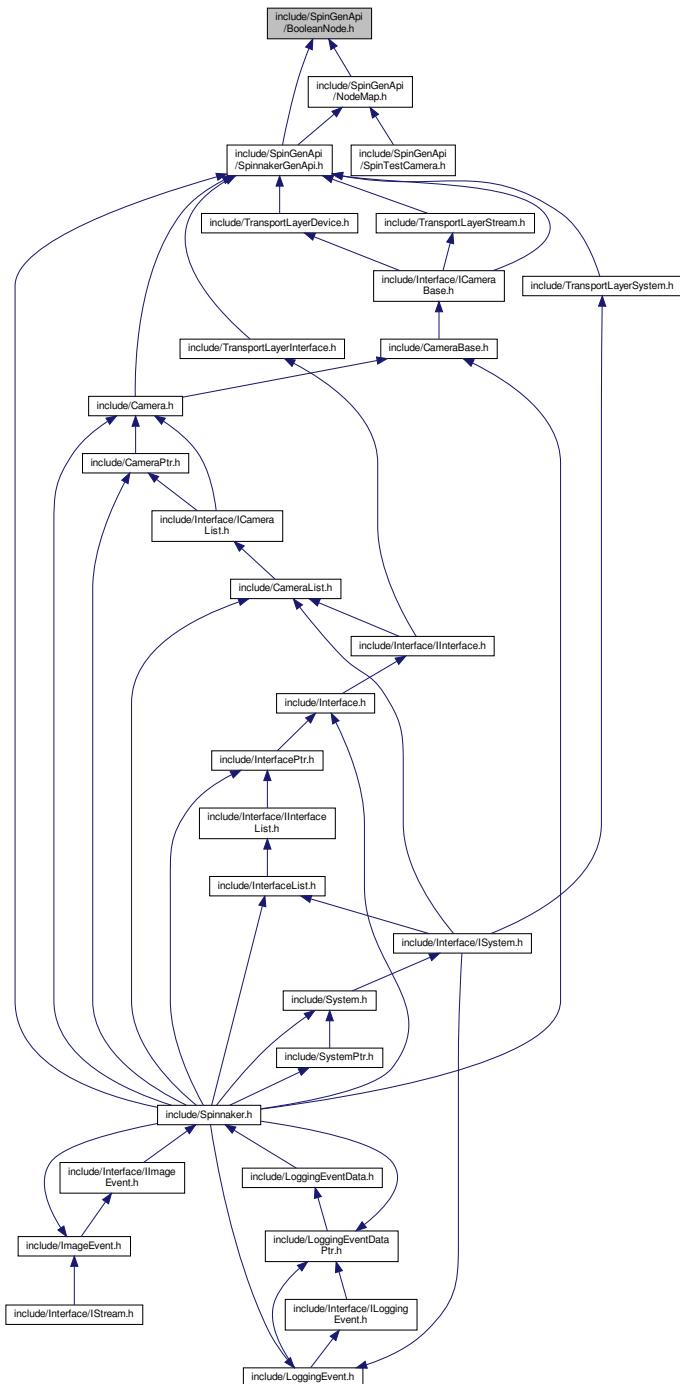
*Base interface common to all nodes.*

## 11.54 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

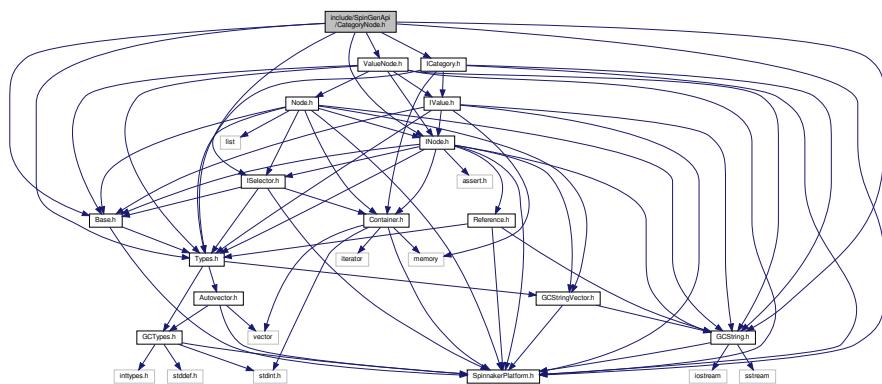
- Spinnaker
- Spinnaker::GenApi

## Typedefs

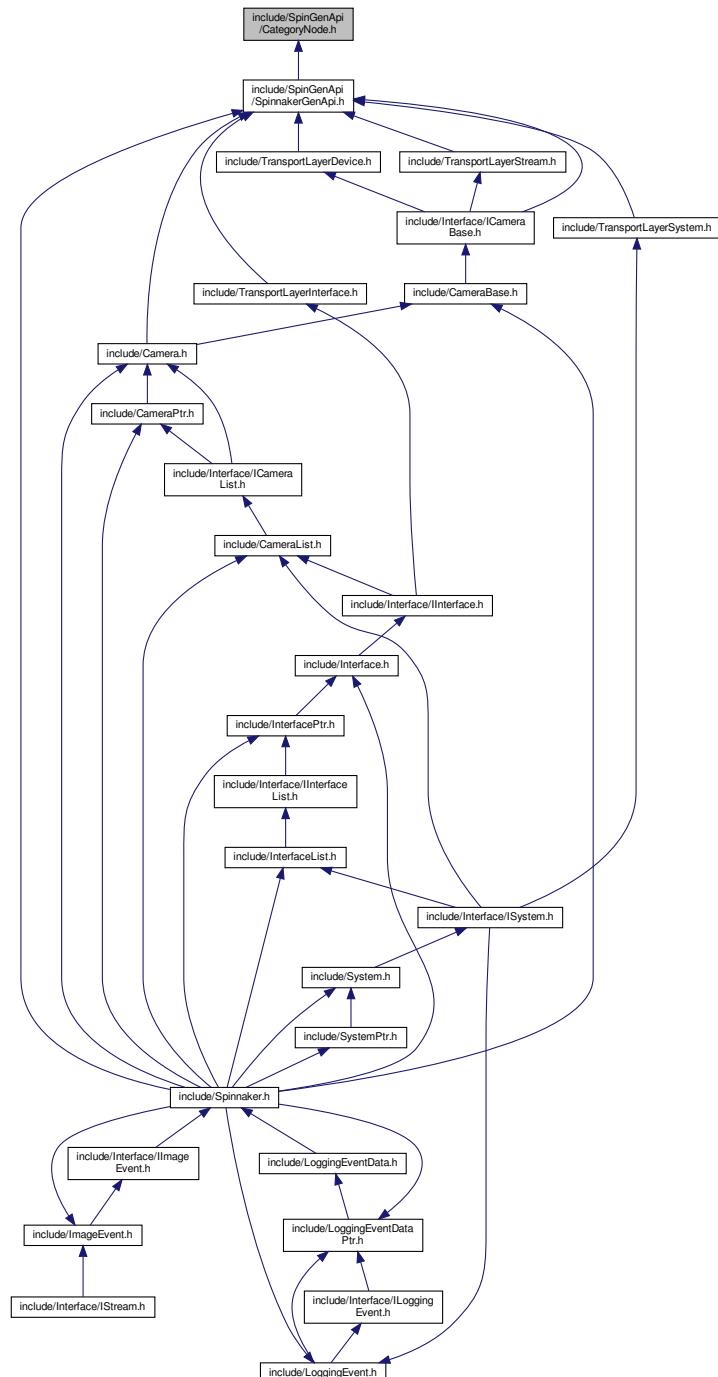
- typedef BooleanNode CBooleanRef

## 11.55 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

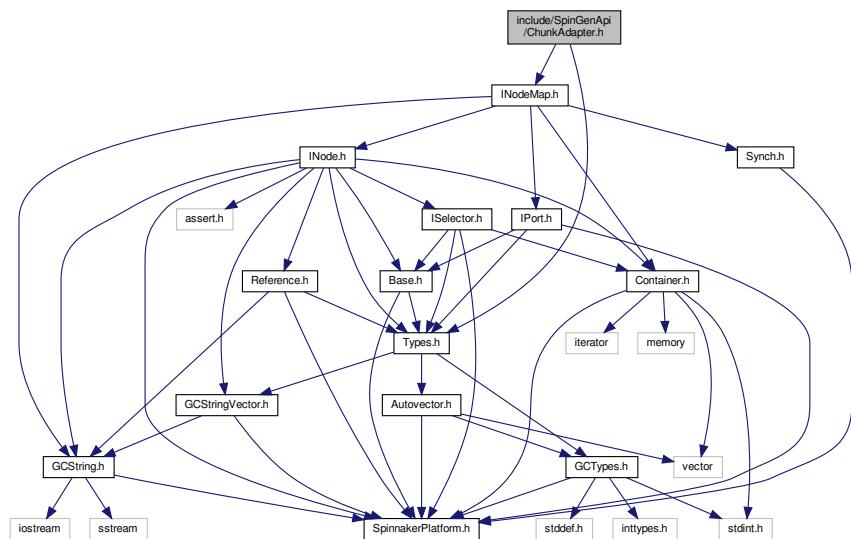
- Spinnaker
- Spinnaker::GenApi

## TypeDefs

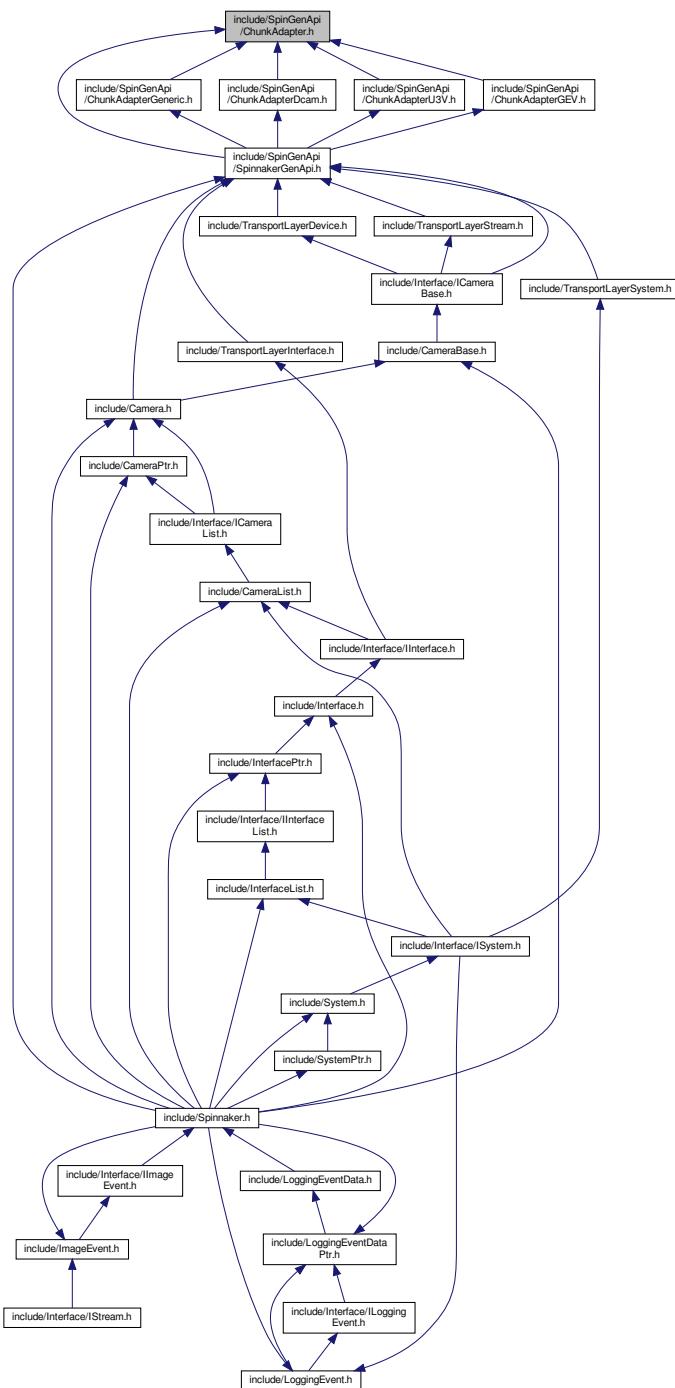
- typedef CategoryNode [CCategoryRef](#)

## 11.56 include/SpinGenApi/ChunkAdapter.h File Reference

Include dependency graph for ChunkAdapter.h:



This graph shows which files directly or indirectly include this file:



## Classes

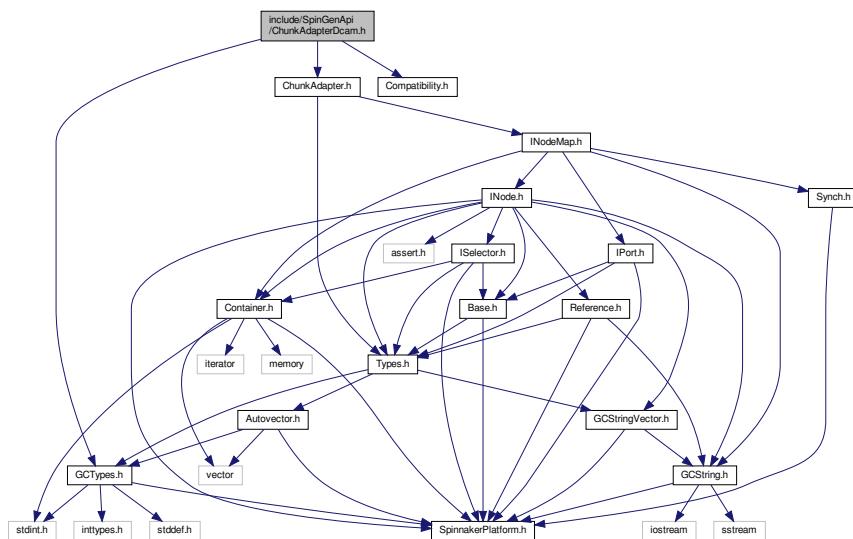
- struct [AttachStatistics\\_t](#)  
*Delivers information about the attached chunks and nodes.*
  - class [CChunkAdapter](#)  
*Connects a chunked buffer to a node map.*

## Namespaces

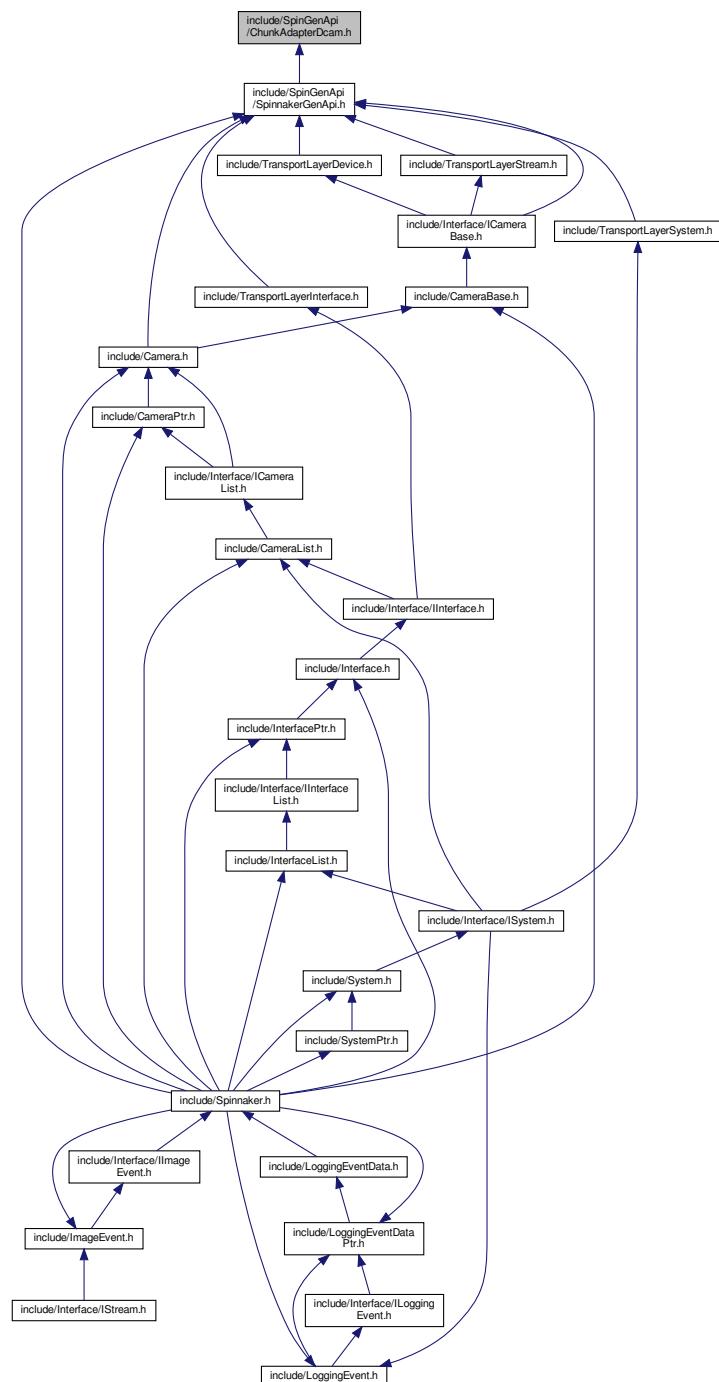
- Spinnaker
- Spinnaker::GenApi

## 11.57 include/SpinGenApi/ChunkAdapterDcam.h File Reference

Include dependency graph for ChunkAdapterDcam.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [DCAM\\_CHUNK\\_TRAILER](#)
- struct [DCAM\\_CHECKSUM](#)
- class [CChunkAdapterDcam](#)

*Connects a chunked DCAM buffer to a node map.*

## Namespaces

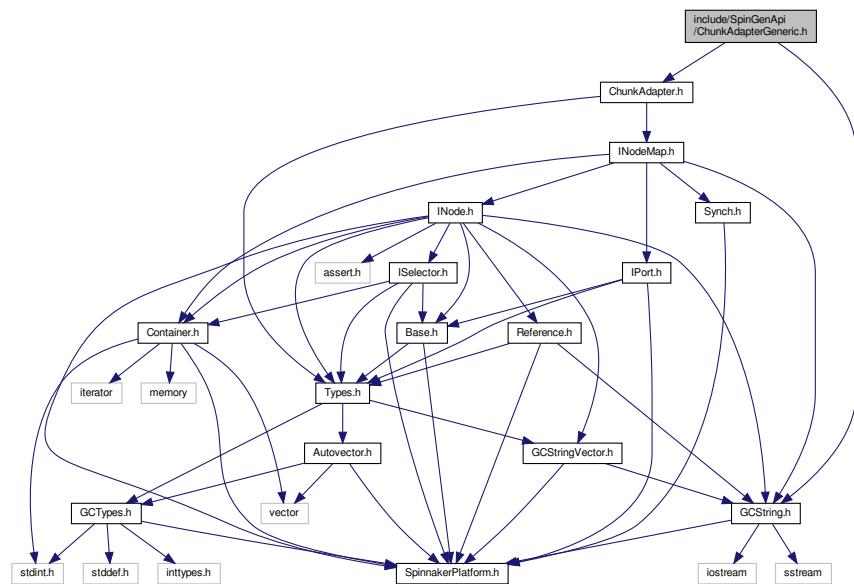
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

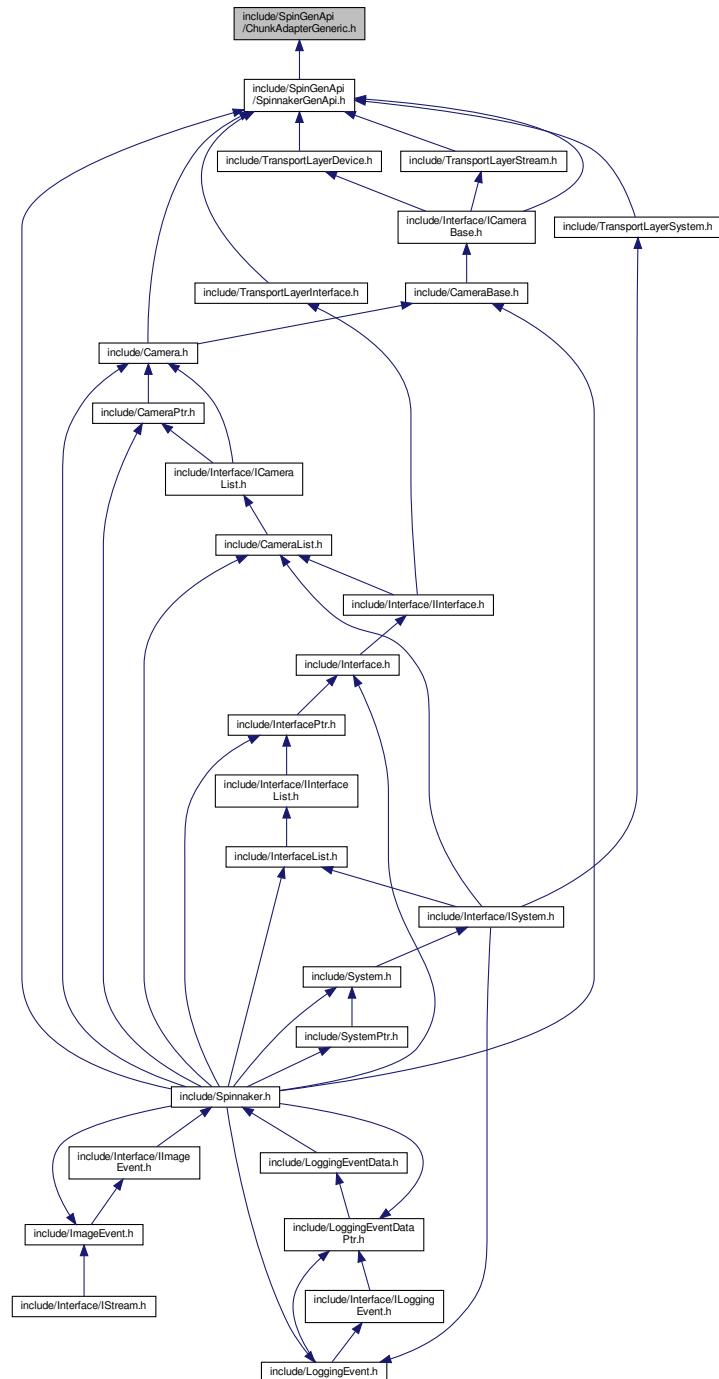
- void [SPINNAKER\\_API SET\\_GUID](#) (SPIN\_GUID &name, uint32\_t l, uint16\_t w1, uint16\_t w2, uint8\_t b1, uint8\_t b2, uint8\_t b3, uint8\_t b4, uint8\_t b5, uint8\_t b6, uint8\_t b7, uint8\_t b8)

## 11.58 include/SpinGenApi/ChunkAdapterGeneric.h File Reference

Include dependency graph for ChunkAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



## Classes

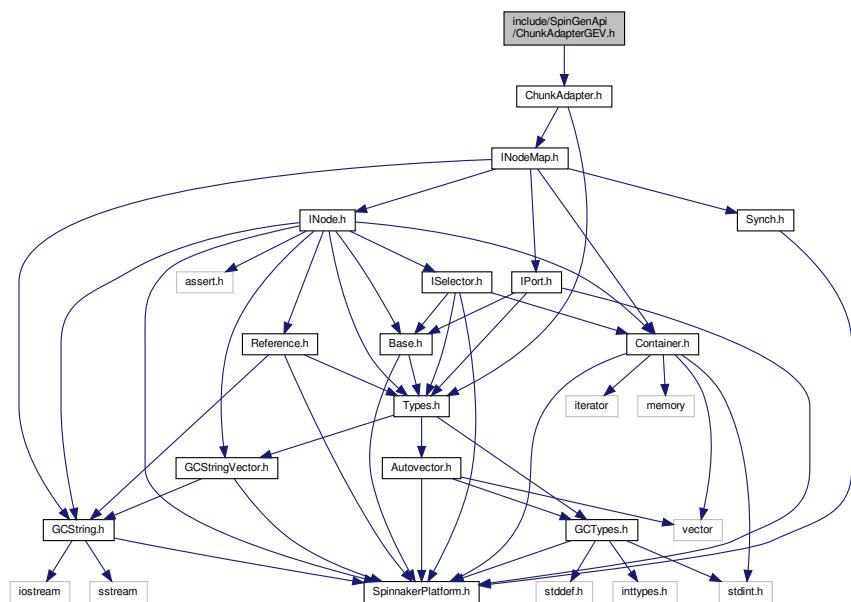
- struct [SingleChunkData\\_t](#)
- struct [SingleChunkDataStr\\_t](#)
- class [CChunkAdapterGeneric](#)

## Namespaces

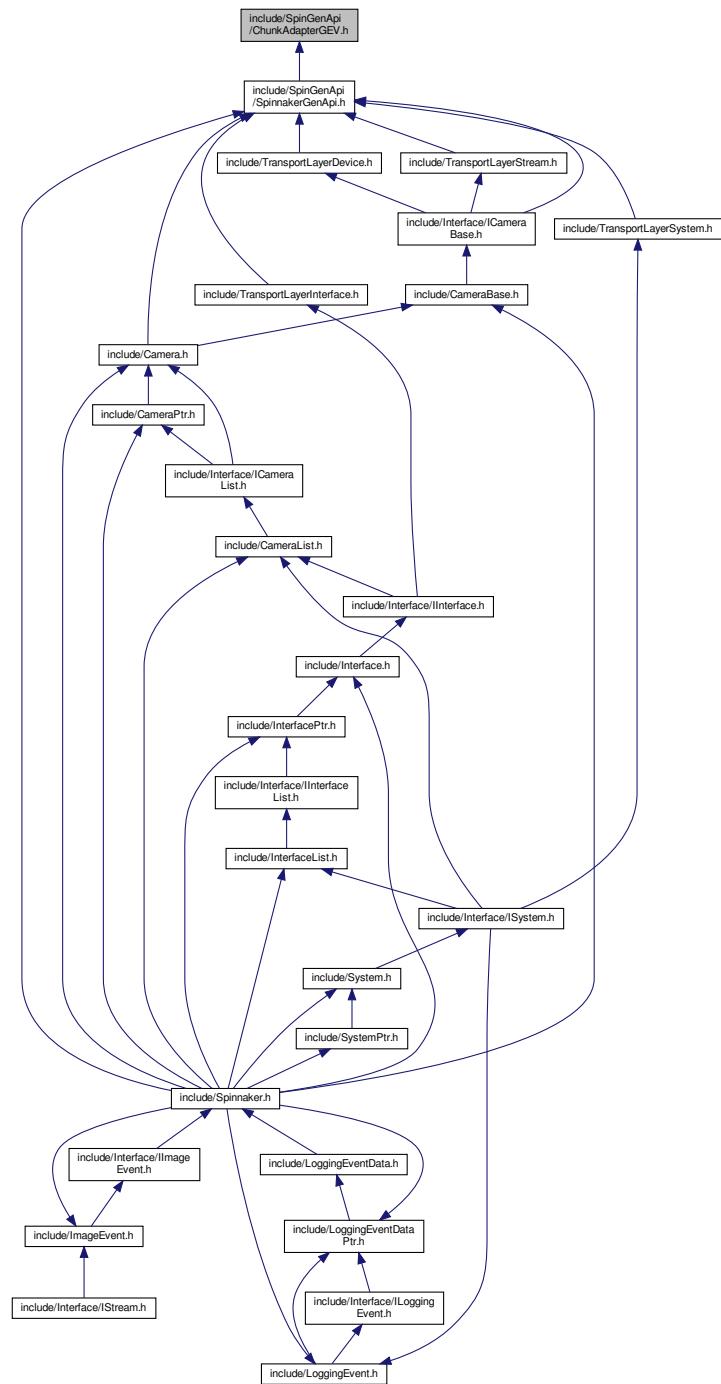
- Spinnaker
- Spinnaker::GenApi

## 11.59 include/SpinGenApi/ChunkAdapterGEV.h File Reference

Include dependency graph for ChunkAdapterGEV.h:



This graph shows which files directly or indirectly include this file:



## Classes

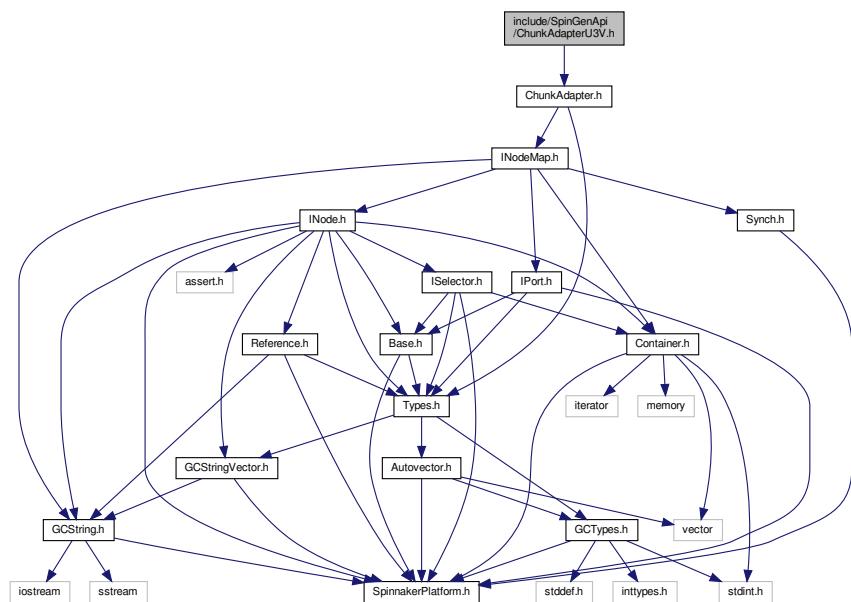
- struct **GVCP\_CHUNK\_TRAILER**  
*header of a GVCP request packet*
  - class **CChunkAdapterGEV**  
*Connects a chunked DCAM buffer to a node map.*

## Namespaces

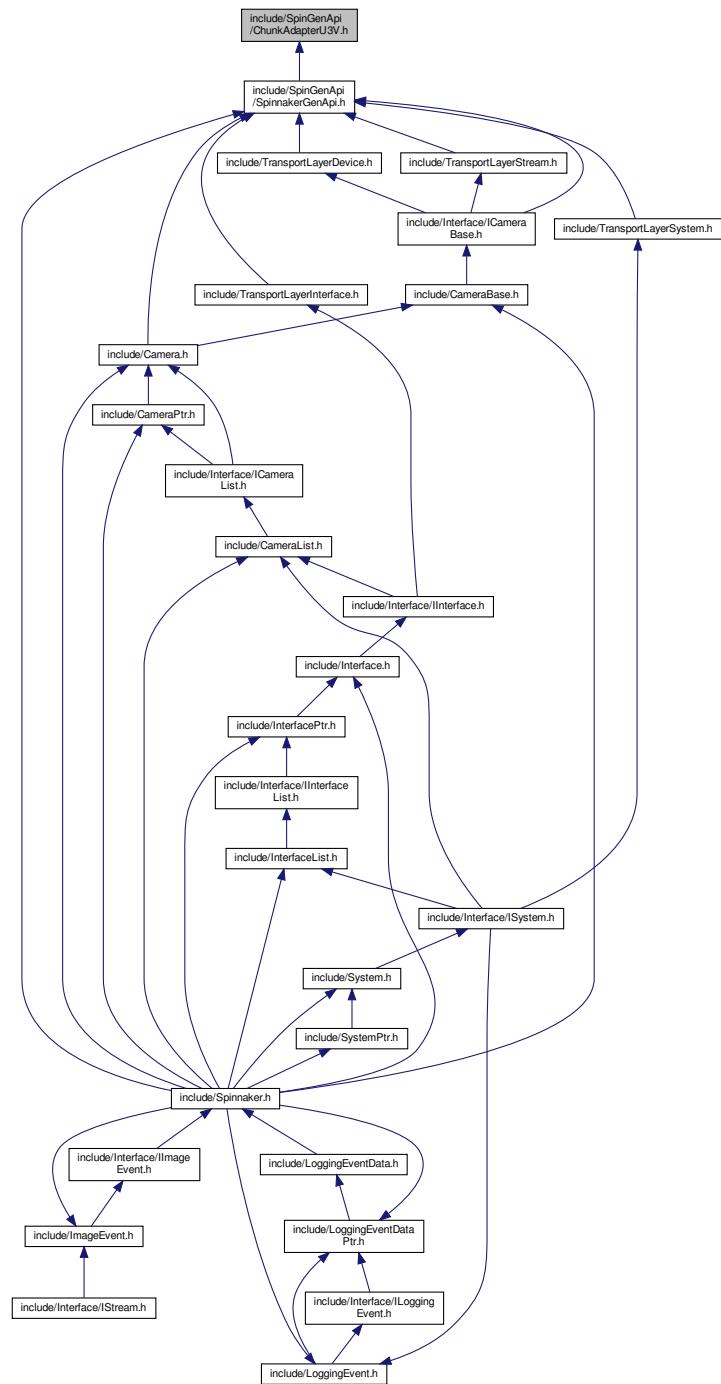
- Spinnaker
- Spinnaker::GenApi

## 11.60 include/SpinGenApi/ChunkAdapterU3V.h File Reference

Include dependency graph for ChunkAdapterU3V.h:



This graph shows which files directly or indirectly include this file:



## Classes

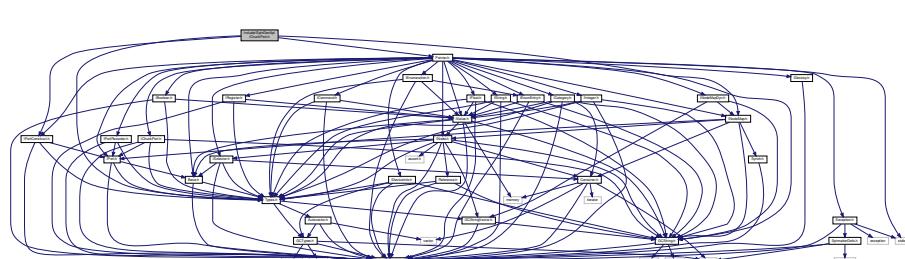
- struct **U3V\_CHUNK\_TRAILER**  
*header of a GVCP request packet*
  - class **CChunkAdapterU3V**  
*Connects a chunked U3V buffer to a node map.*

## Namespaces

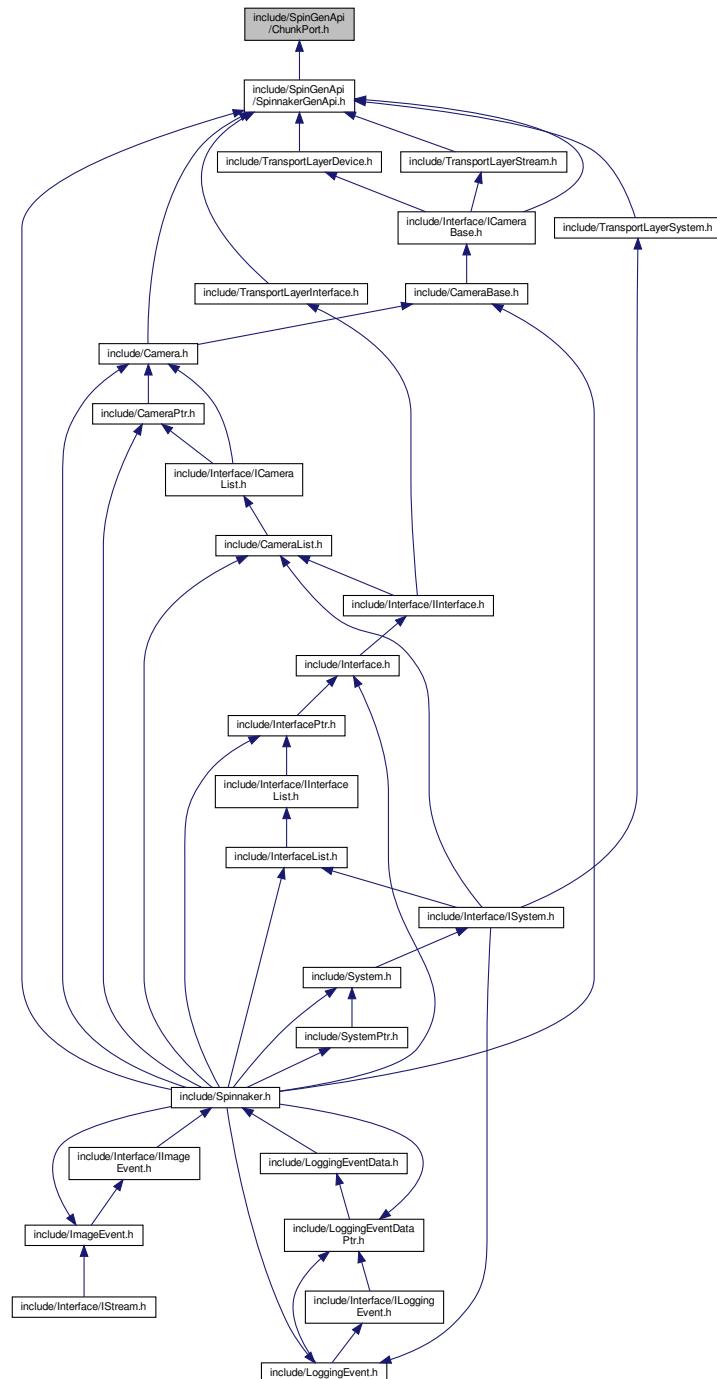
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.61 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

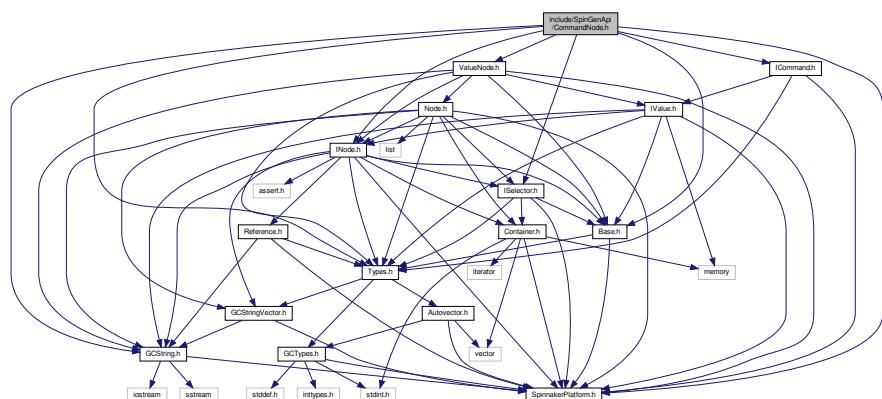
*Port attachable to a chunk in a buffer.*

## Namespaces

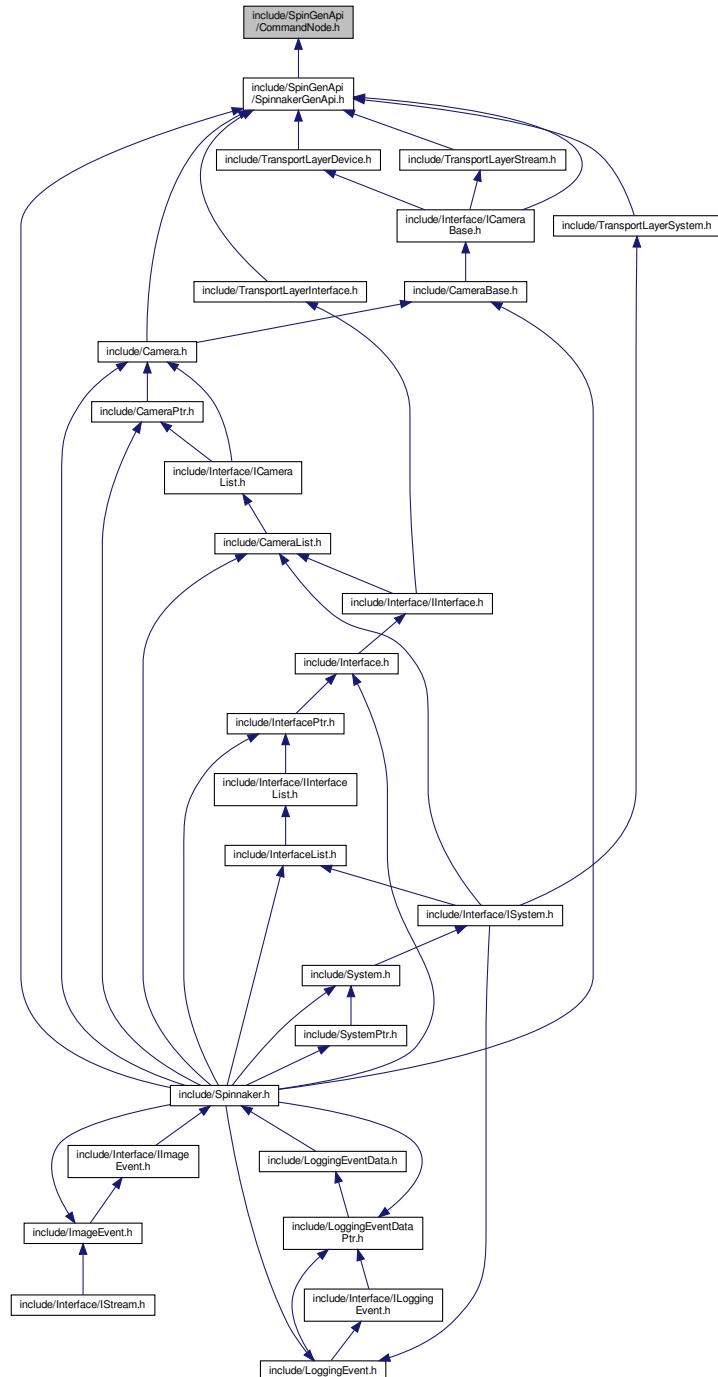
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.62 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

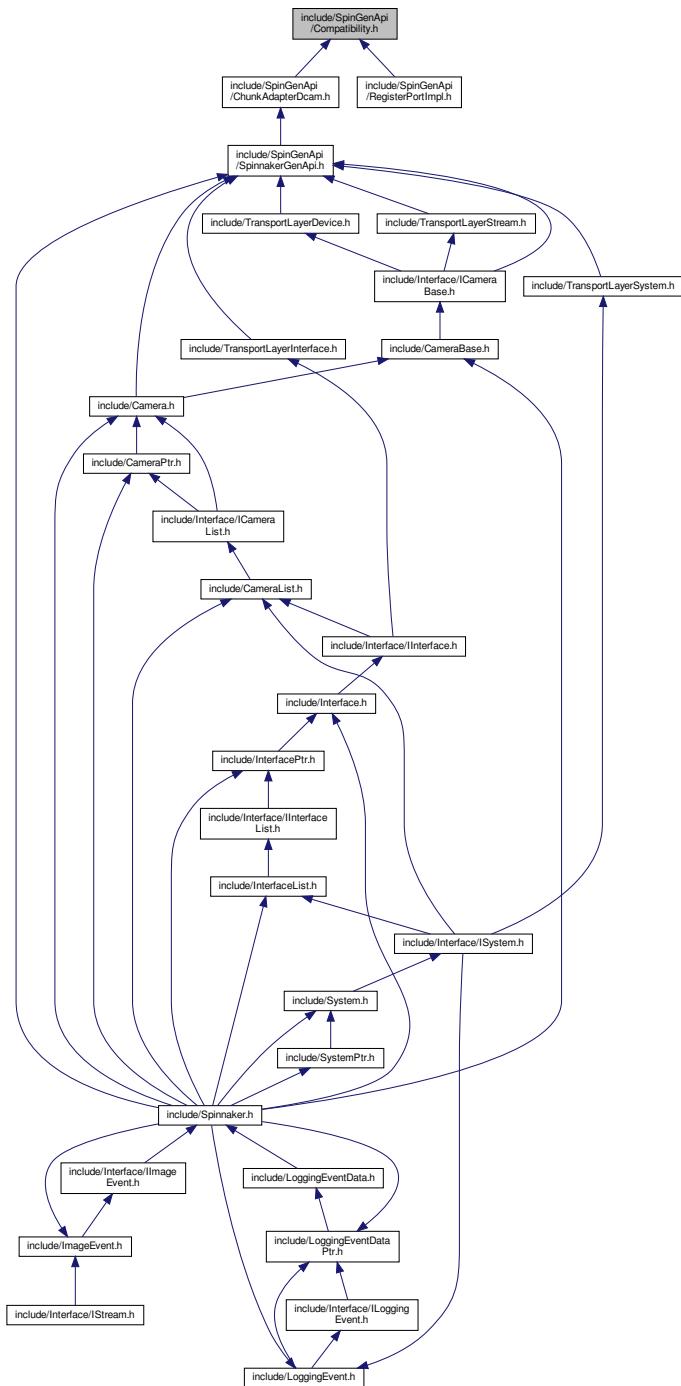
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- `typedef CommandNode CCommandRef`

## 11.63 include/SpinGenApi/Compatibility.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- #define FMT\_I64 "ll"

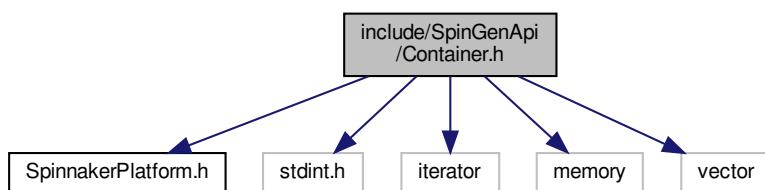
### 11.63.1 Macro Definition Documentation

#### 11.63.1.1 FMT\_I64

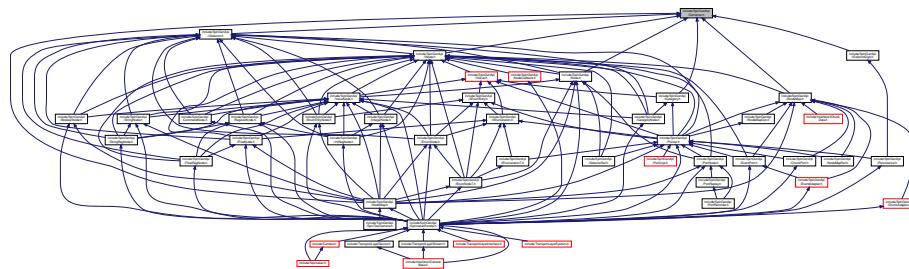
```
#define FMT_I64 "ll"
```

## 11.64 include/SpinGenApi/Container.h File Reference

Include dependency graph for Container.h:



This graph shows which files directly or indirectly include this file:



## 11.65 include/SpinGenApi/Counter.h File Reference

### Classes

- class [Counter](#)

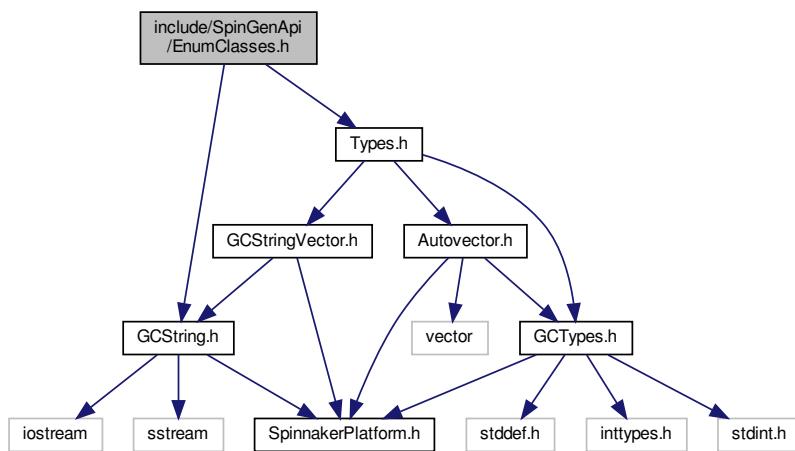
*Definition of a simple [Counter](#) class.*

### Namespaces

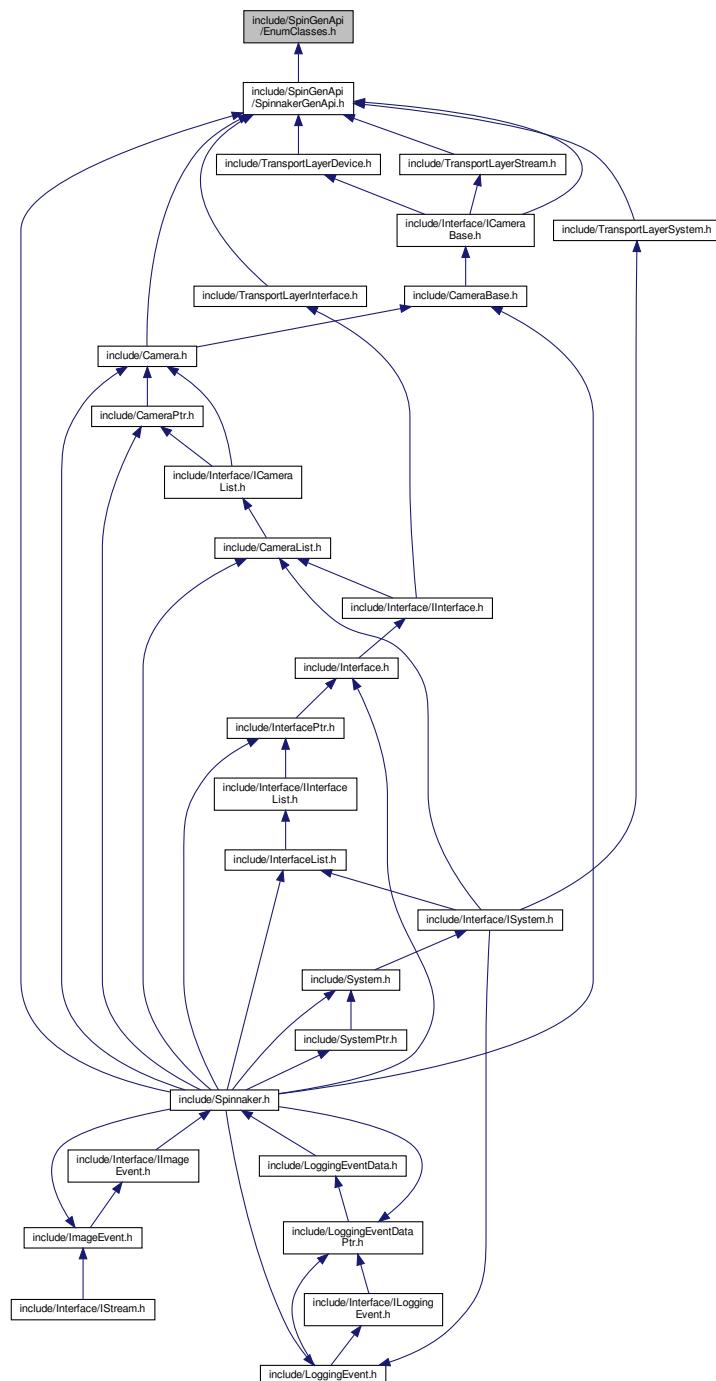
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.66 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 [EndianessClass](#)  
*Holds conversion methods for the endianess enumeration.*
- class [RepresentationClass](#)

- class [EVisibilityClass](#)

*Holds conversion methods for the representation enumeration.*
- class [EAccessModeClass](#)

*Holds conversion methods for the visibility enumeration.*
- class [ECachingModeClass](#)

*Holds conversion methods for the access mode enumeration.*
- class [ENamespaceClass](#)

*Holds conversion methods for the caching mode enumeration.*
- class [EYesNoClass](#)

*Holds conversion methods for the 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](#)

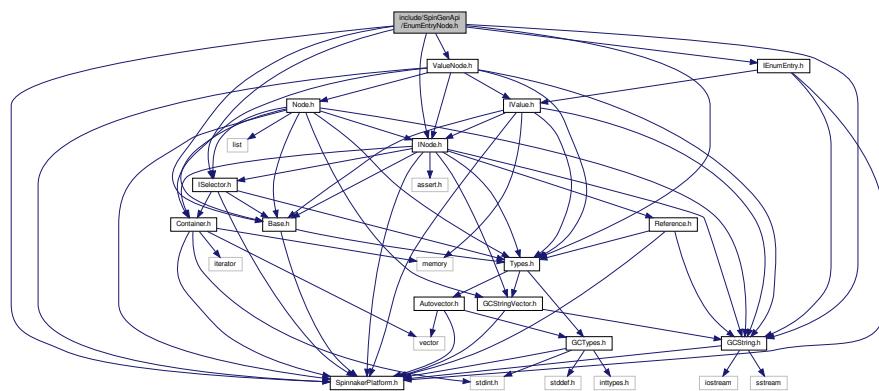
*helper class converting EGenApiSchemaVersion from and to string*

## Namespaces

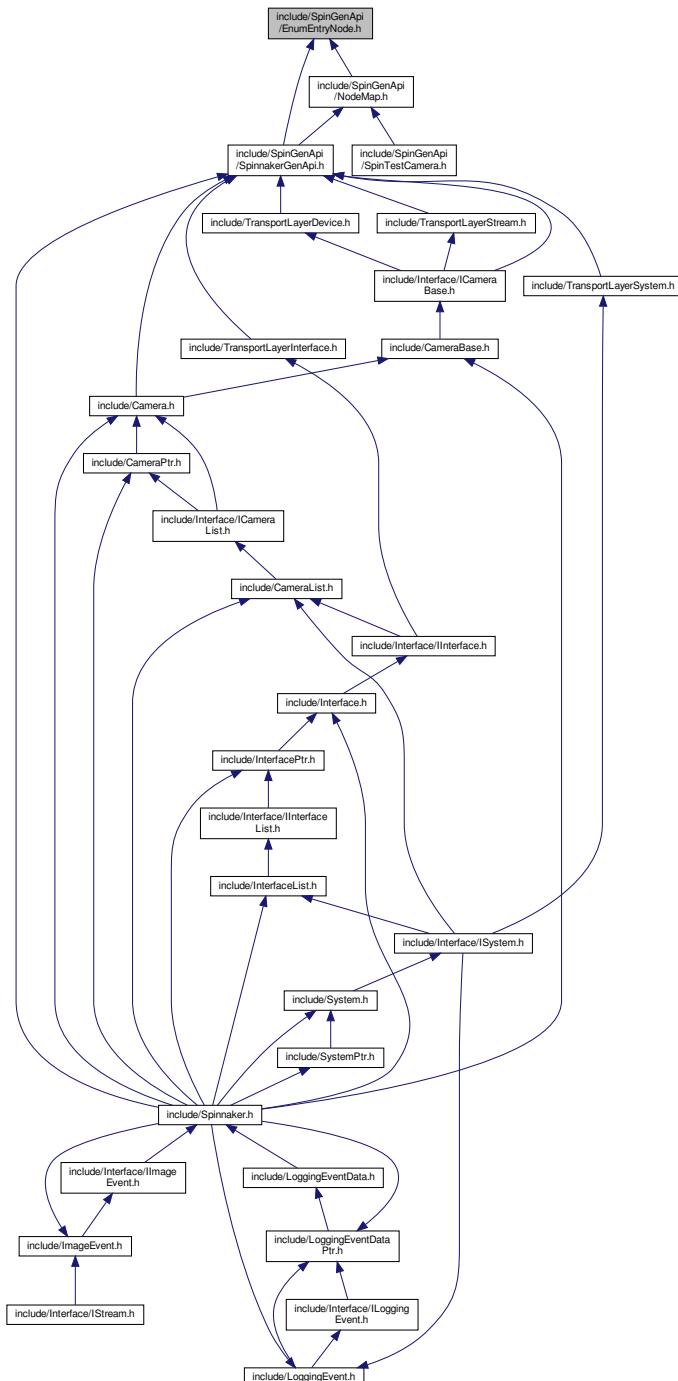
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.67 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

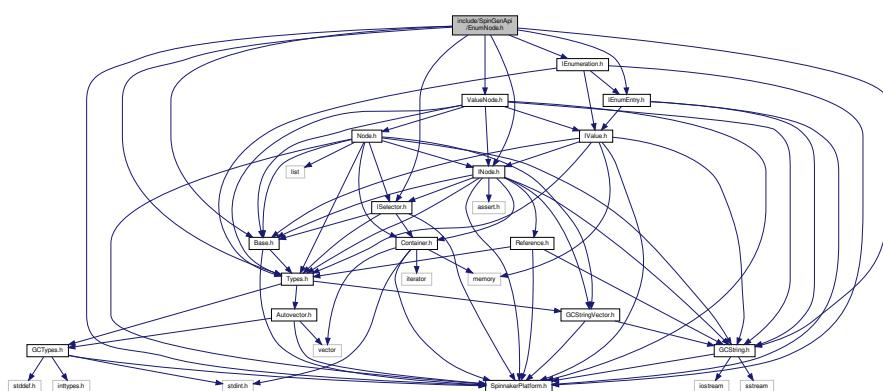
- Spinnaker
- Spinnaker::GenApi

## Typedefs

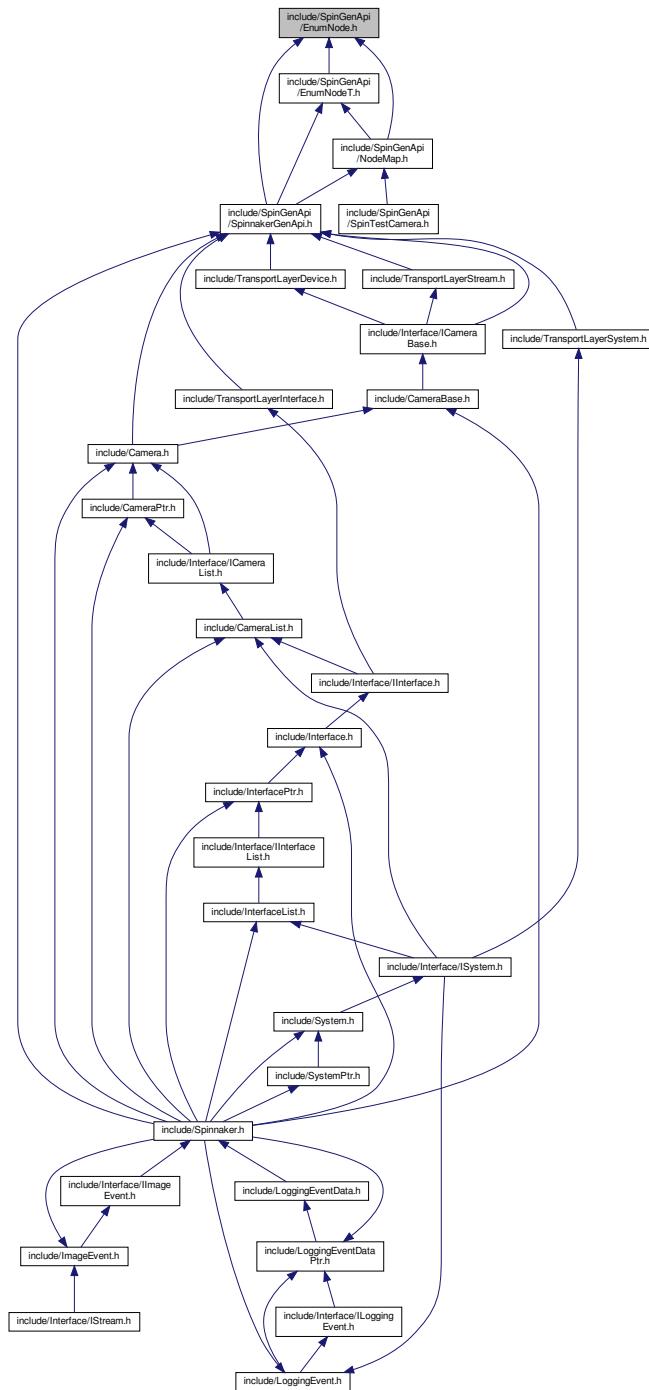
- `typedef EnumEntryNode CEnumEntryRef`

## 11.68 include/SpinGenApi/EnumNode.h File Reference

Include dependency graph for `EnumNode.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [EnumNode](#)

*Interface for string properties.*

## Namespaces

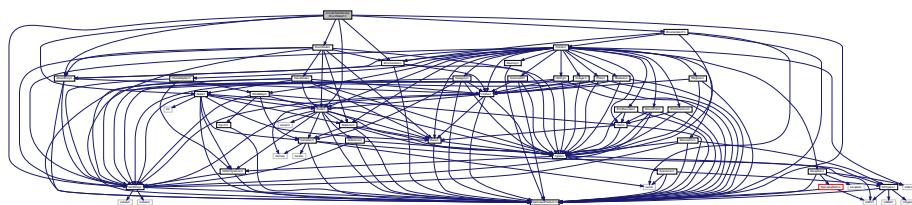
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

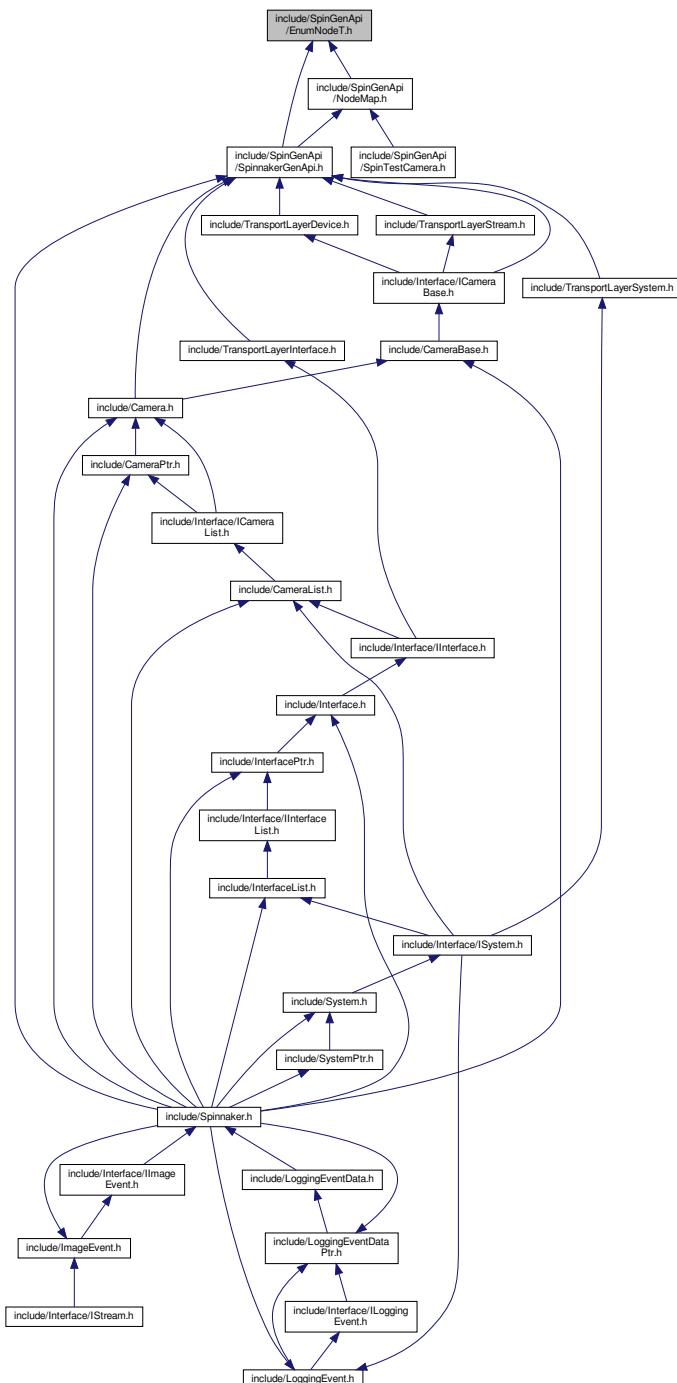
- `typedef EnumNode CEnumerationRef`

## 11.69 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 >

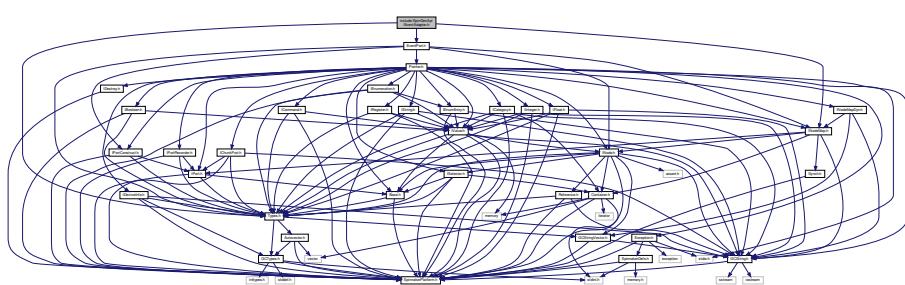
*Interface for string properties.*

## Namespaces

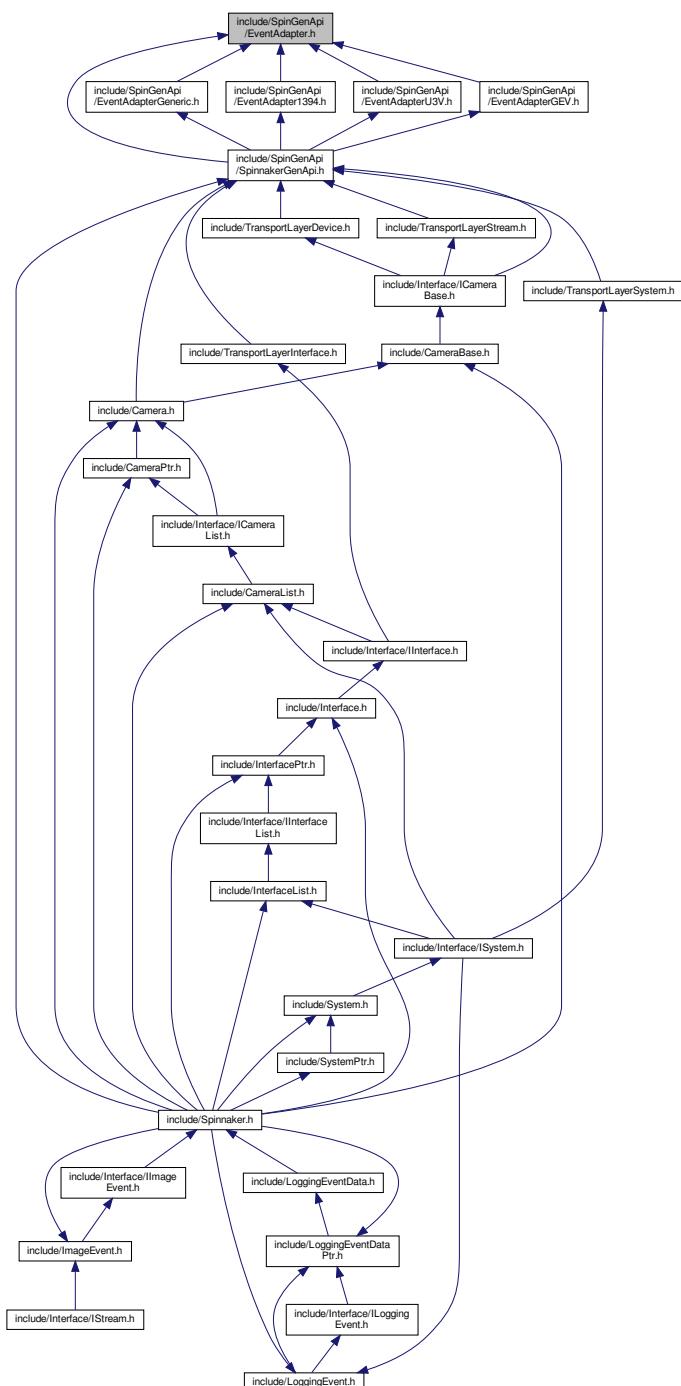
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.70 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](#)

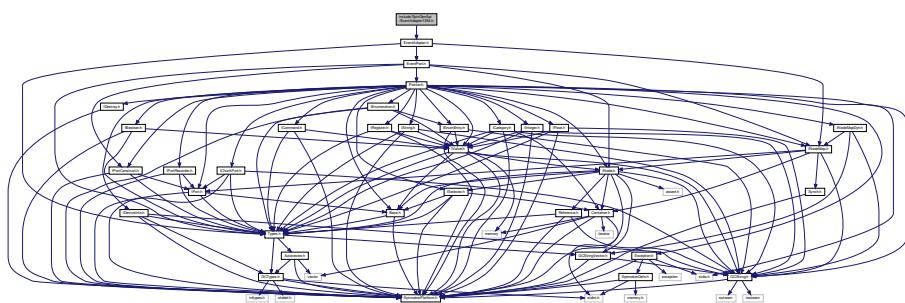
*Delivers Events to ports.*

## Namespaces

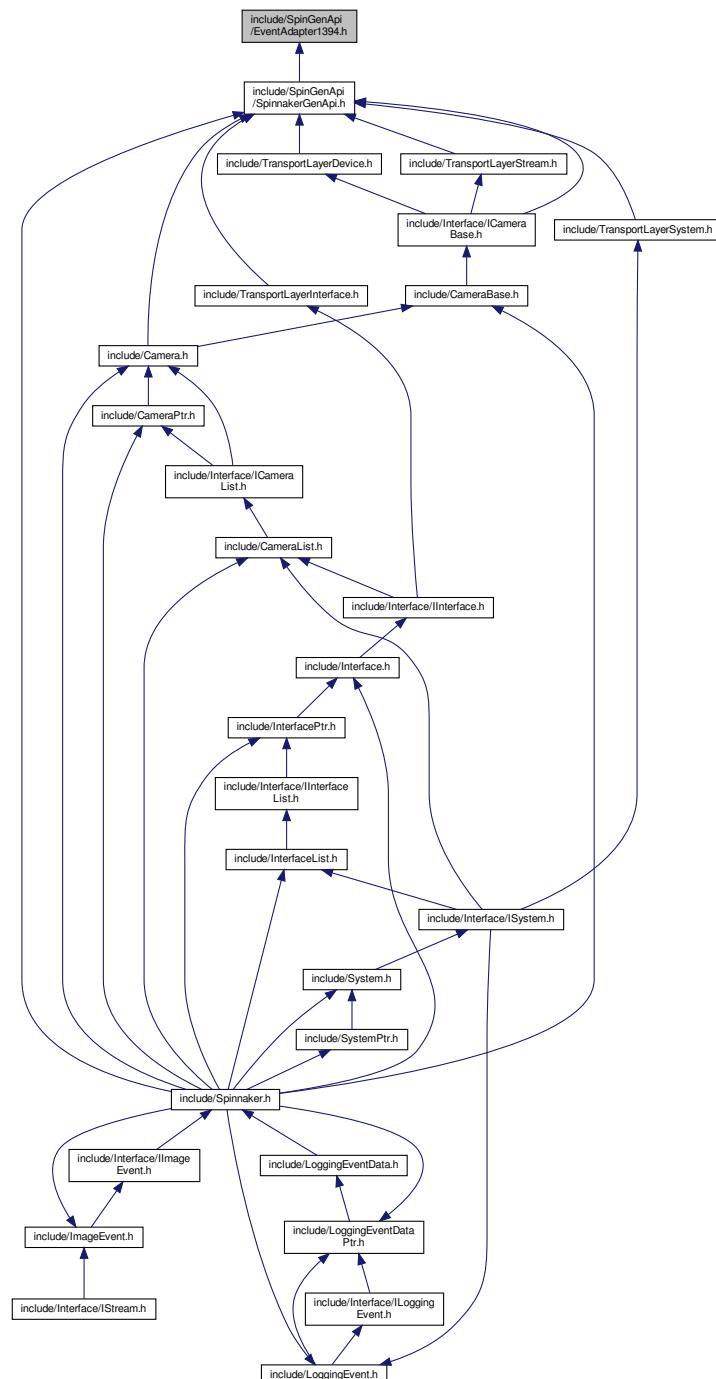
- Spinnaker
  - Spinnaker::GenApi

## 11.71 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](#)

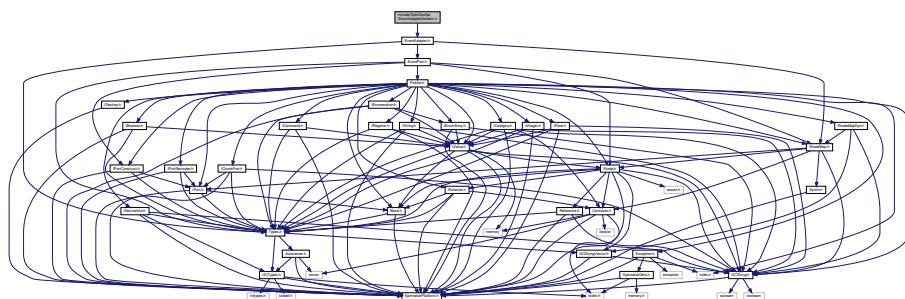
*Distribute the events to the node map.*

## Namespaces

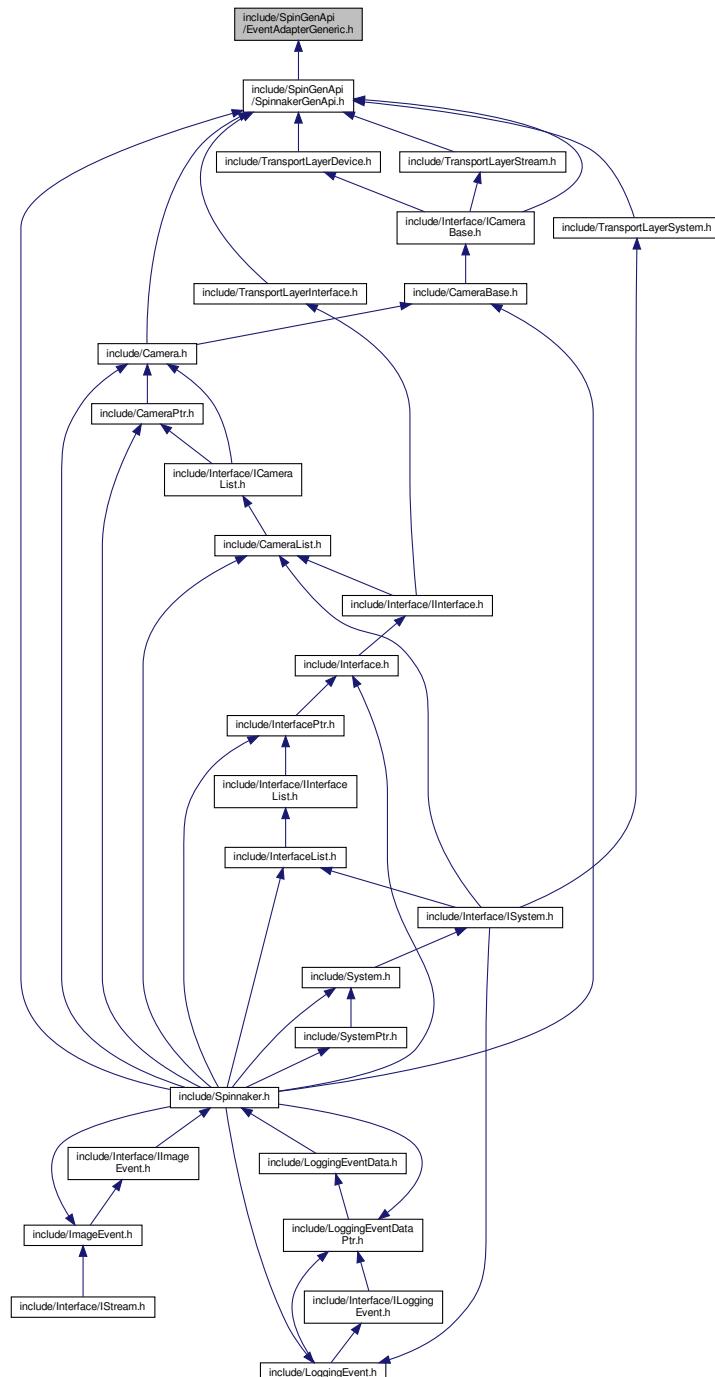
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.72 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](#)

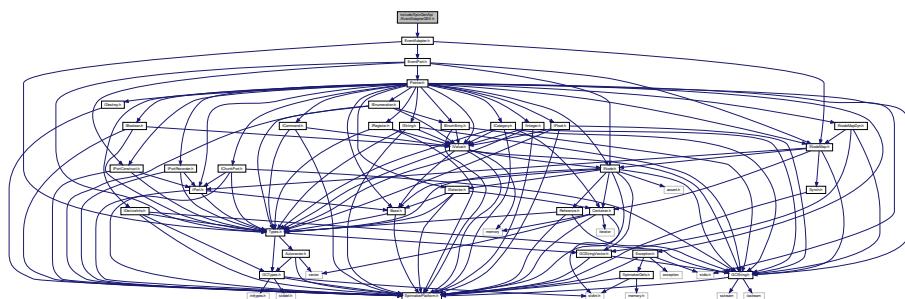
*Connects a generic event to a node map.*

## Namespaces

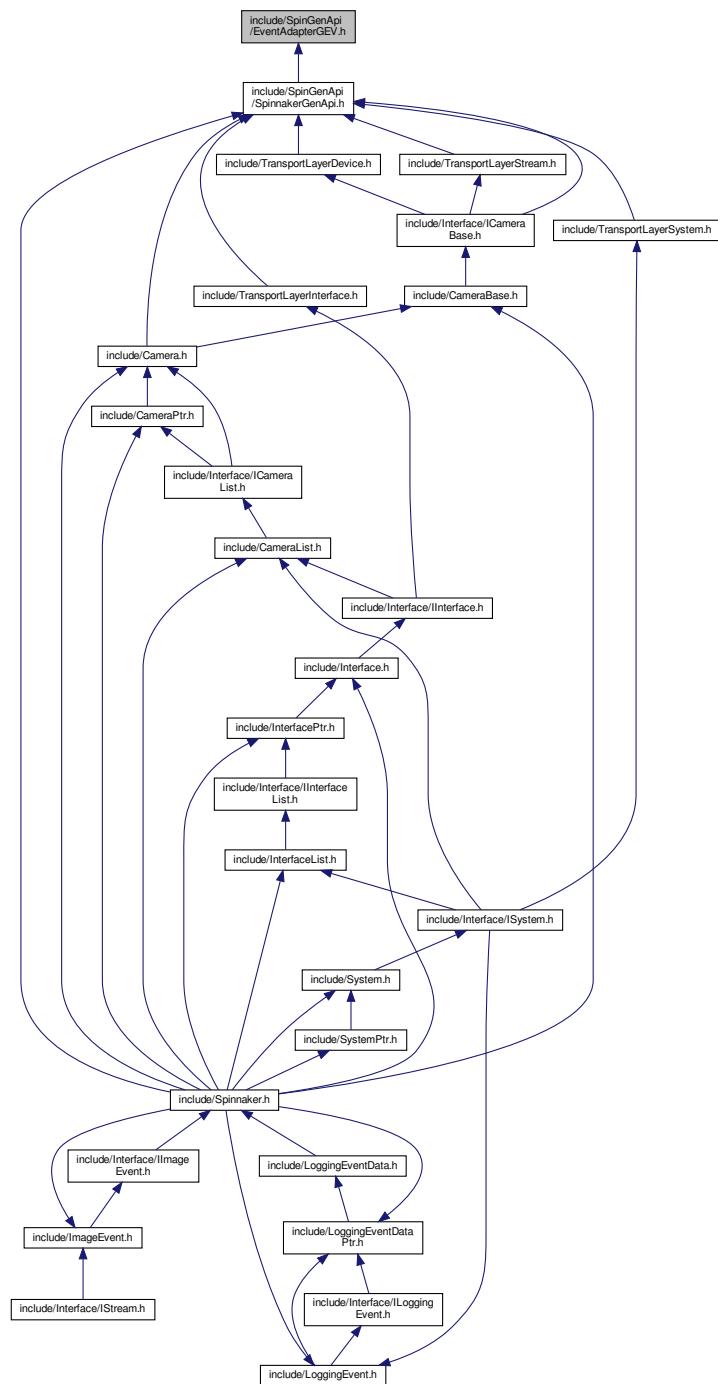
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### 11.73 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](#)

- 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

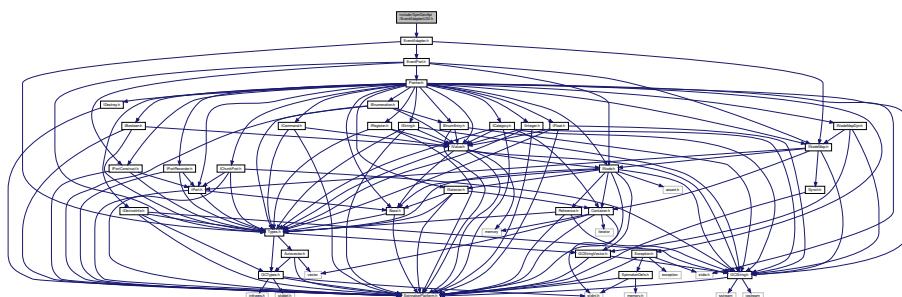
- enum [GVCP\\_MESSAGE\\_TAGS](#) {
   
**TAG\_EVENT\_CMD** = 0xc0,
   
**TAG\_EVENTDATA\_CMD** = 0xc2 }

## Variables

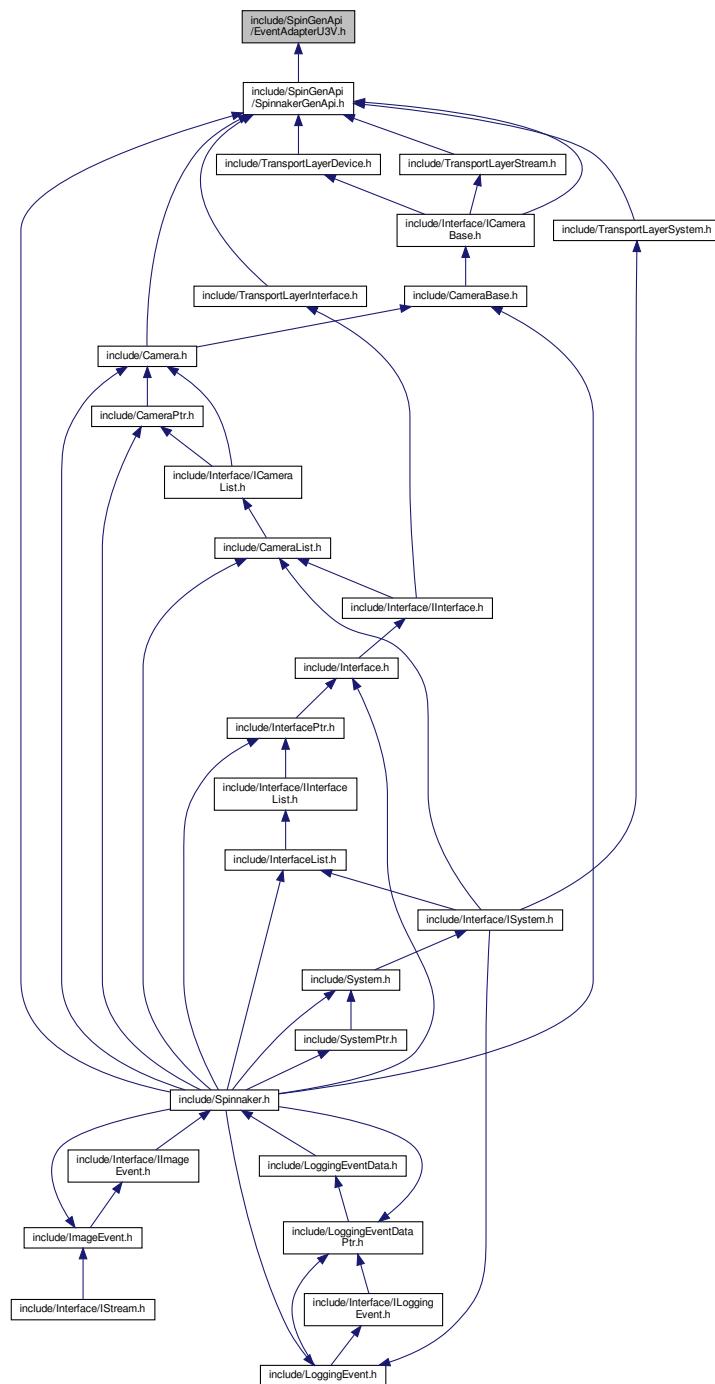
- const uint8\_t [COMMAND\\_MAGIC](#) = 0x42

## 11.74 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

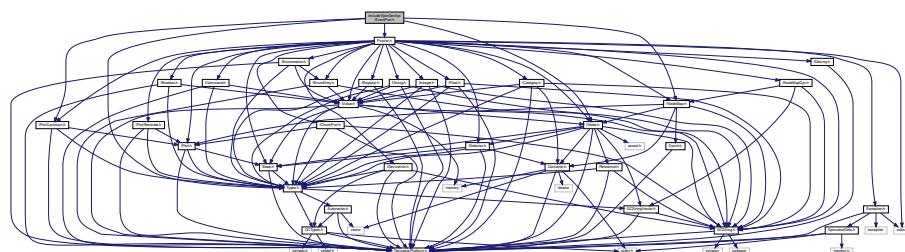
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Variables

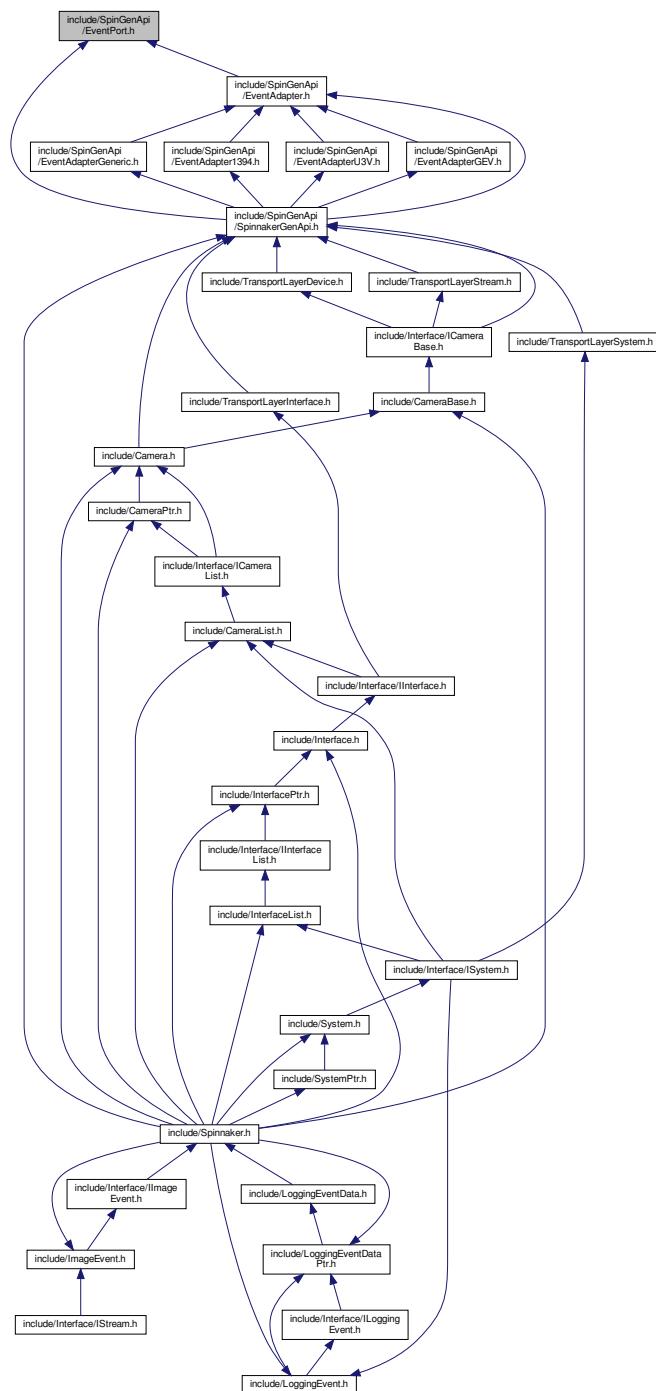
- const uint32\_t [U3V\\_EVENT\\_PREFIX](#) = 0x45563355
- const uint16\_t [GENCP\\_EVENT\\_CMD\\_ID](#) = 0x0C00
- const size\_t [GENCP\\_COMMAND\\_HEADER\\_SIZE](#) = sizeof([U3V\\_COMMAND\\_HEADER](#))
- const size\_t [GENCP\\_EVENT\\_BASIC\\_SIZE](#) = sizeof([U3V\\_EVENT\\_MESSAGE](#))

## 11.75 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](#)

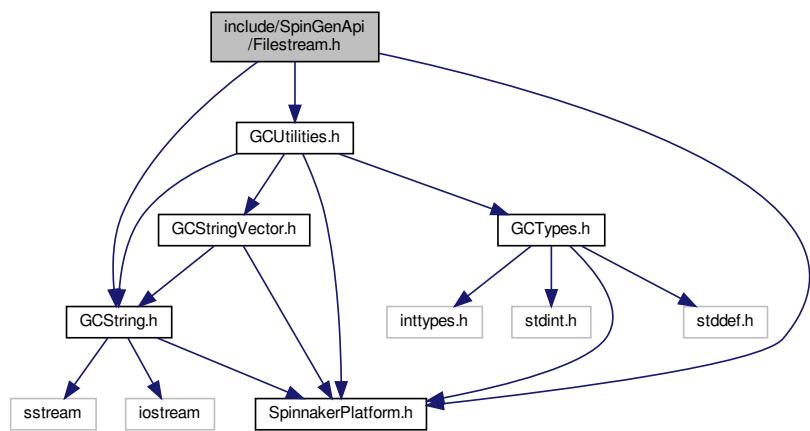
*Port attachable to an event.*

## Namespaces

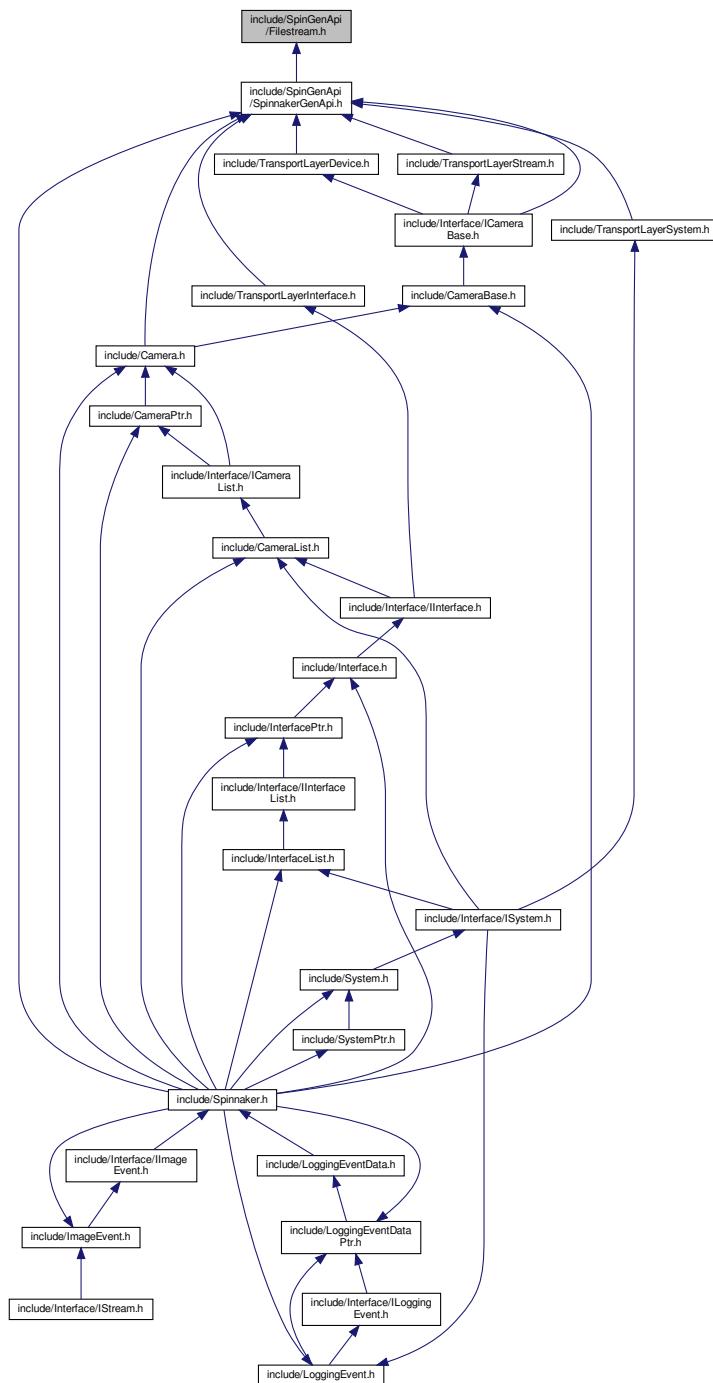
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.76 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

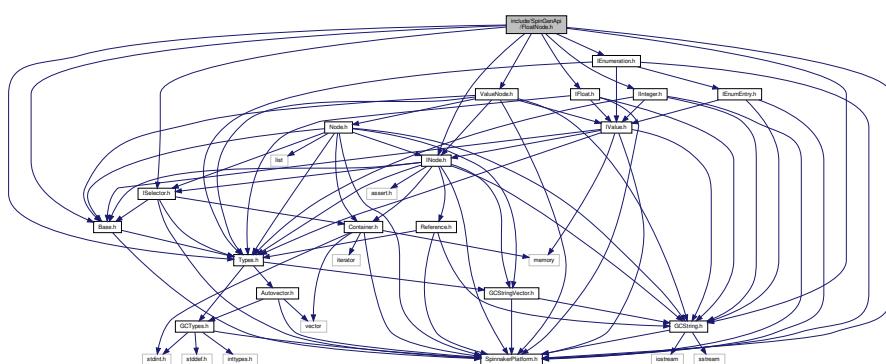
- Spinnaker
  - Spinnaker::GenApi

## TypeDefs

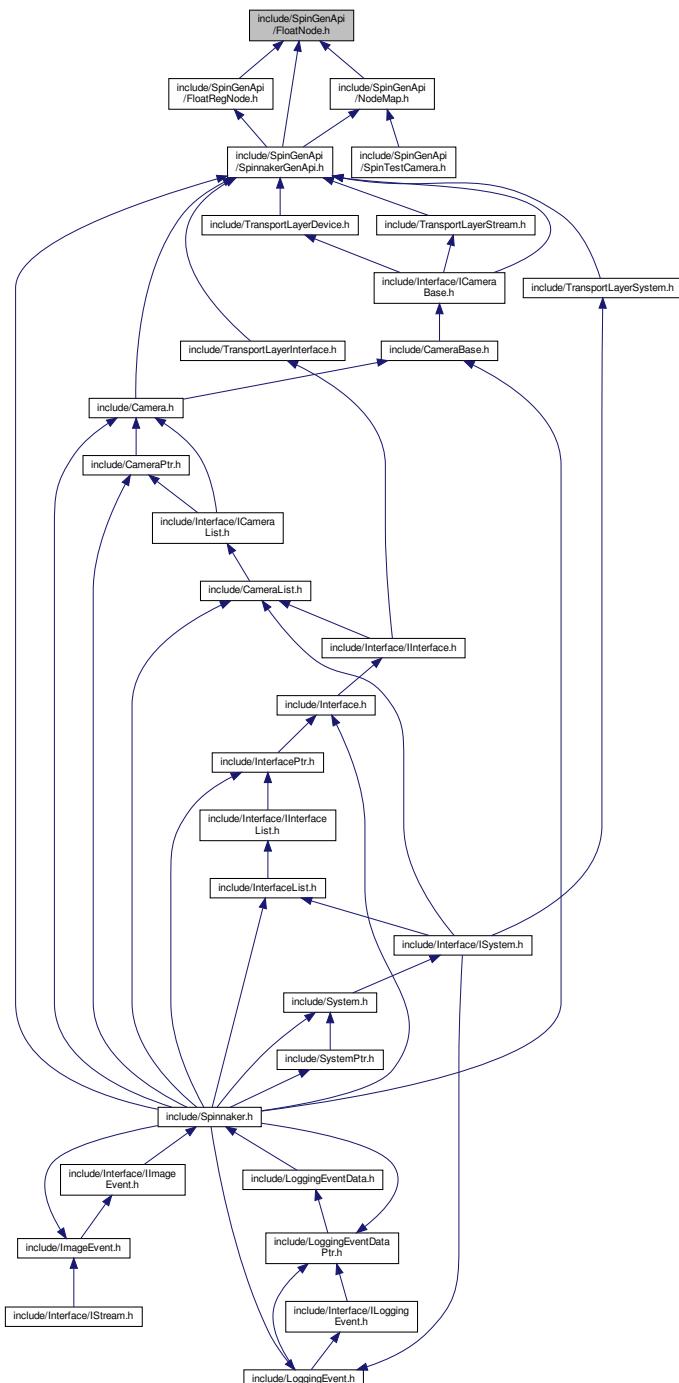
- `typedef ODevFileStreamBase< char, std::char_traits< char > > ODevFileStream`
  - `typedef IDevFileStreamBase< char, std::char_traits< char > > IDevFileStream`

## 11.77 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

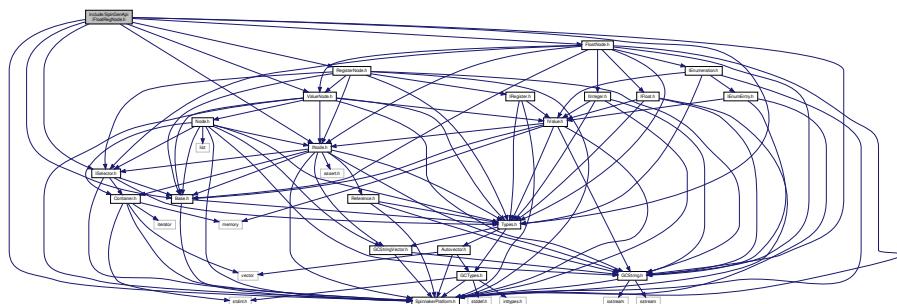
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

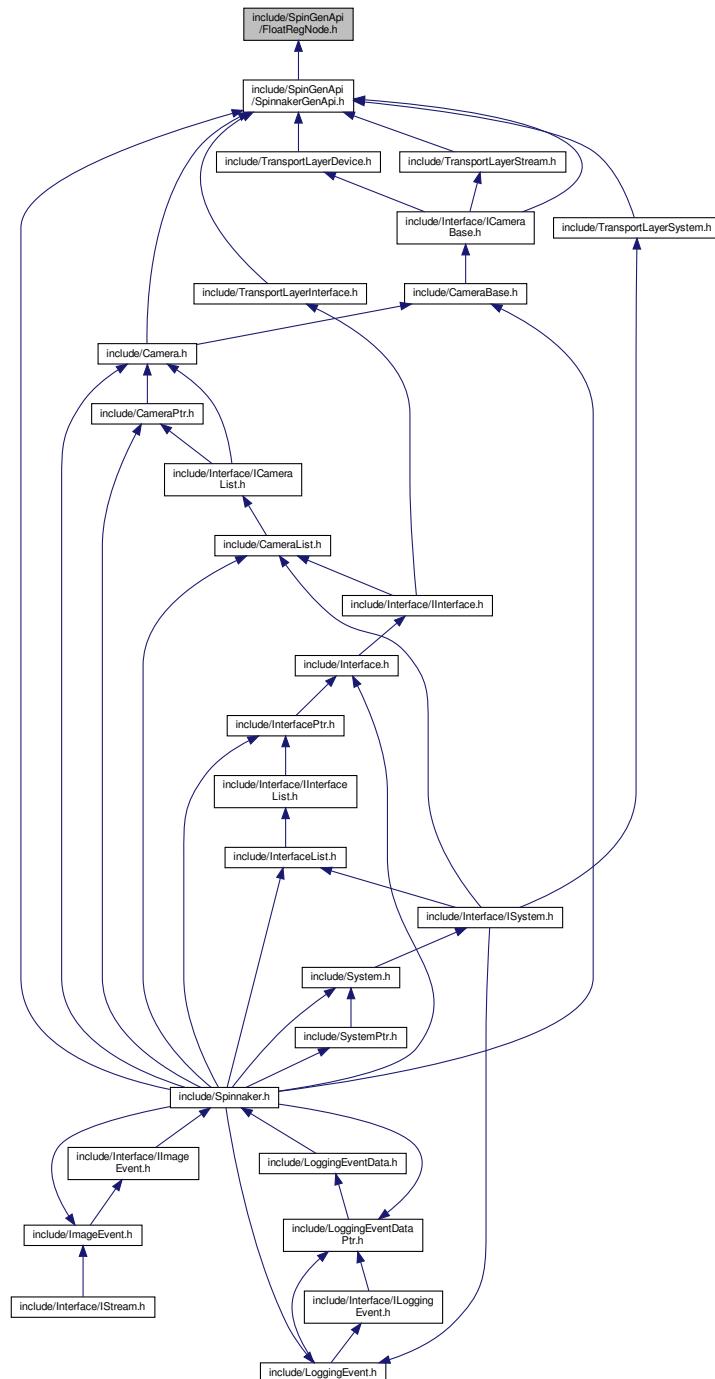
- [typedef FloatNode CFloatRef](#)

## 11.78 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](#)

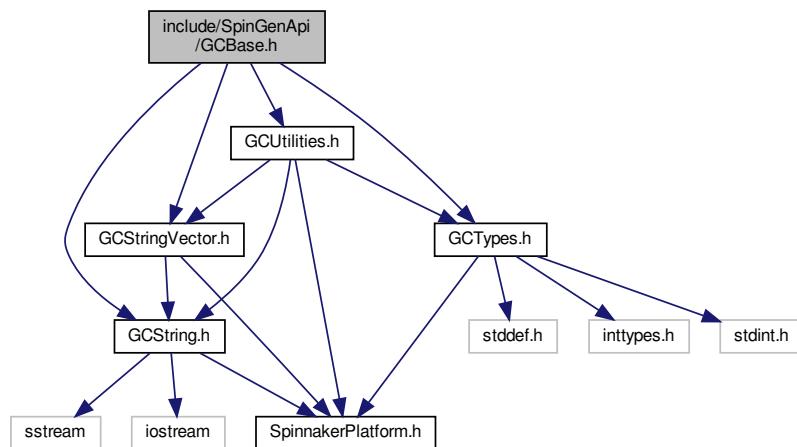
*Interface for string properties.*

## Namespaces

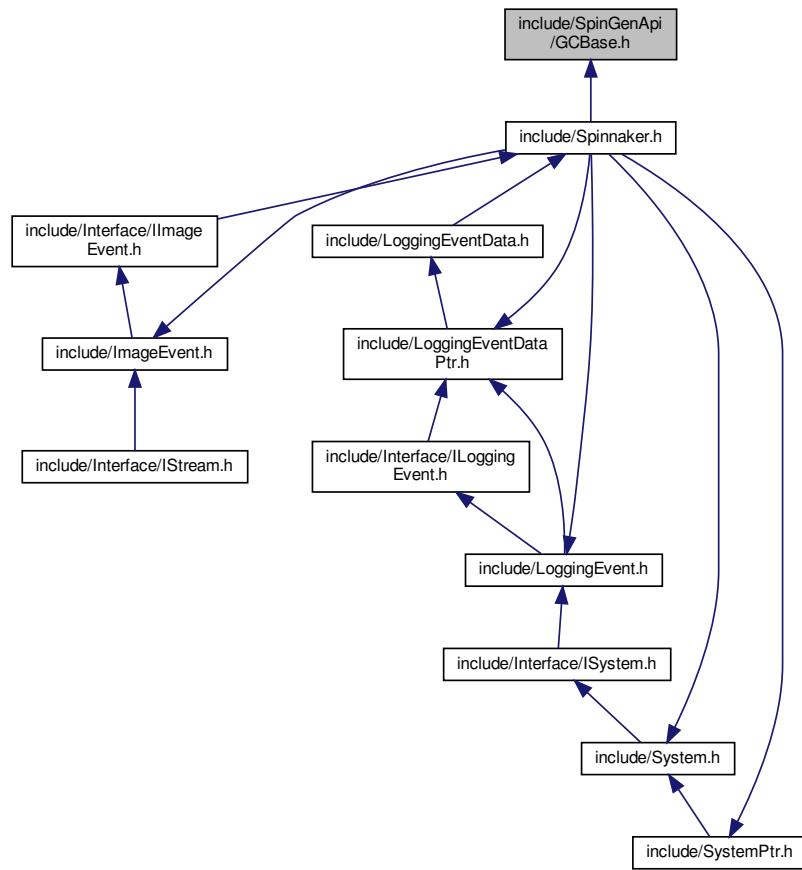
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.79 include/SpinGenApi/GCBase.h File Reference

Include dependency graph for GCBase.h:

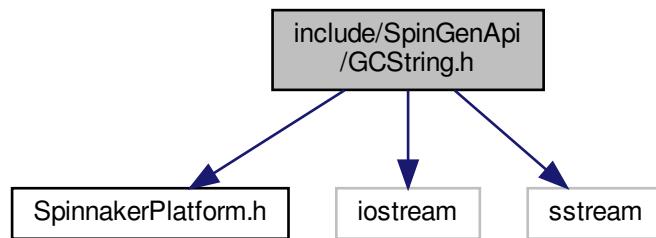


This graph shows which files directly or indirectly include this file:

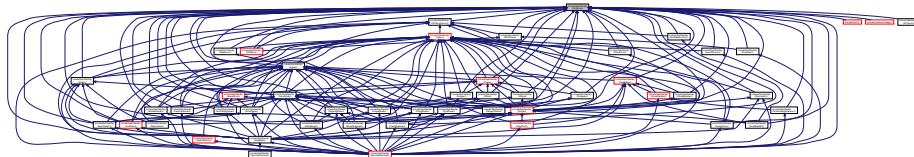


## 11.80 include/SpinGenApi/GCString.h File Reference

Include dependency graph for GCString.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gcstring](#)

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

## Macros

- `#define GCSTRING_NPOS size_t(-1)`

## 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.*
- `std::ostream & operator<< (std::ostream &ostr, const Spinnaker::GenICam::gcstring &str)`  
*STL operator out.*
- `std::istream & operator>> (std::istream &istr, Spinnaker::GenICam::gcstring &str)`  
*STL operator in.*

### 11.80.1 Macro Definition Documentation

#### 11.80.1.1 GCSTRING\_NPOS

```
#define GCSTRING_NPOS size_t(-1)
```

### 11.80.2 Function Documentation

### 11.80.2.1 operator<<()

```
std::ostream& operator<< (
 std::ostream & ostr,
 const Spinnaker::GenICam::gcstring & str) [inline]
```

STL operator out.

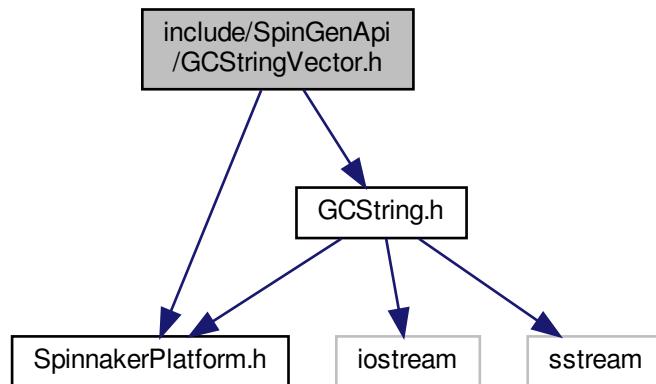
### 11.80.2.2 operator>>()

```
std::istream& operator>> (
 std::istream & istr,
 Spinnaker::GenICam::gcstring & str) [inline]
```

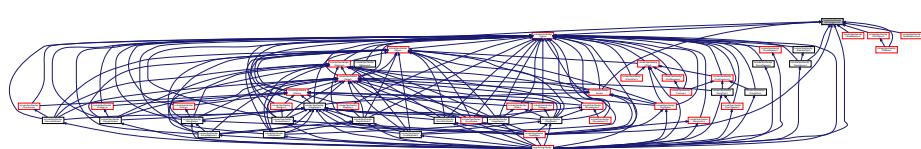
STL operator in.

## 11.81 include/SpinGenApi/GCStringVector.h File Reference

Include dependency graph for GCStringVector.h:

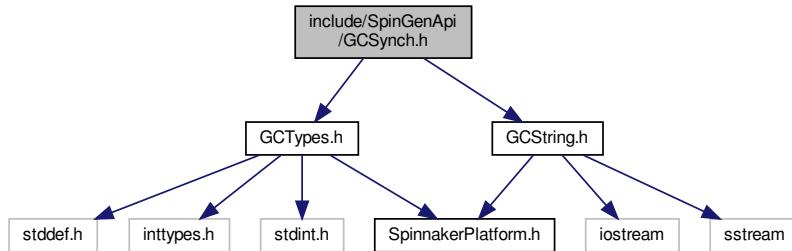


This graph shows which files directly or indirectly include this file:



## 11.82 include/SpinGenApi/GCSynch.h File Reference

Include dependency graph for GCSynch.h:



### Classes

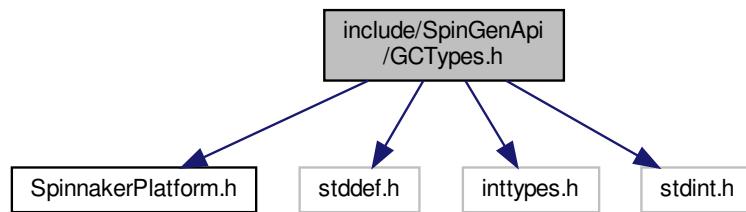
- class [CLock](#)  
*A lock class.*
- class [CLOCKEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)
- class [LockableObject< Object >](#)  
*Instance-Lock for an object.*
- class [LockableObject< Object >::Lock](#)  
*A scopelevel Lock class.*
- class [CGlobalLock](#)  
*Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)  
*Unlocks the global lock object on destruction.*

### Namespaces

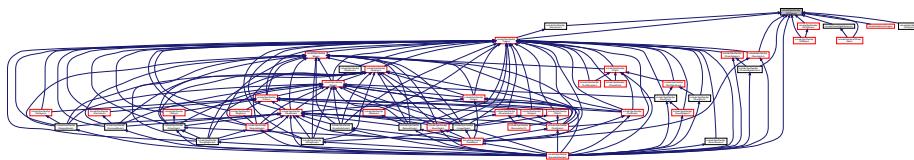
- [Spinnaker](#)
- [Spinnaker::GenICam](#)

## 11.83 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>(0xfffffffffffffffULL) /* 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>(0x00000000ffffffffffULL) /* 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>(0xffffffffffff80LL) /* minimum signed int8 value */`
- `#define GC_UINT8_MAX static_cast<uint64_t>(0x0000000000000ffULL) /* maximum unsigned int8 value */`

## Typedefs

- `typedef float float32_t`  
*32 bit floating point*
- `typedef double float64_t`  
*64 bit floating point*

### 11.83.1 Macro Definition Documentation

#### 11.83.1.1 \_\_STDC\_CONSTANT\_MACROS

```
#define __STDC_CONSTANT_MACROS
```

#### 11.83.1.2 \_\_STDC\_LIMIT\_MACROS

```
#define __STDC_LIMIT_MACROS
```

#### 11.83.1.3 GC\_INT32\_MAX

```
#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */
```

#### 11.83.1.4 GC\_INT32\_MIN

```
#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */
```

#### 11.83.1.5 GC\_INT64\_MAX

```
#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffLL) /* maximum signed int64 value */
```

### 11.83.1.6 GC\_INT64\_MIN

```
#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */
```

### 11.83.1.7 GC\_INT8\_MAX

```
#define GC_INT8_MAX static_cast<int64_t>(0x0000000000000007fLL) /* maximum signed int8 value */
```

### 11.83.1.8 GC\_INT8\_MIN

```
#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffff80LL) /* minimum signed int8 value */
```

### 11.83.1.9 GC\_UINT32\_MAX

```
#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */
```

### 11.83.1.10 GC\_UINT64\_MAX

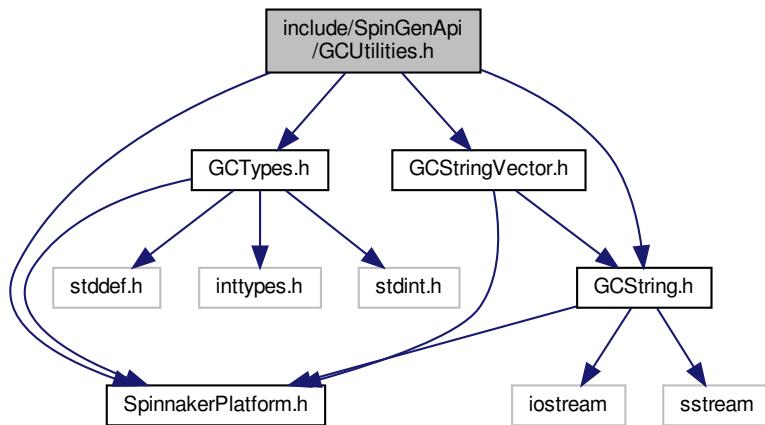
```
#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */
```

### 11.83.1.11 GC\_UINT8\_MAX

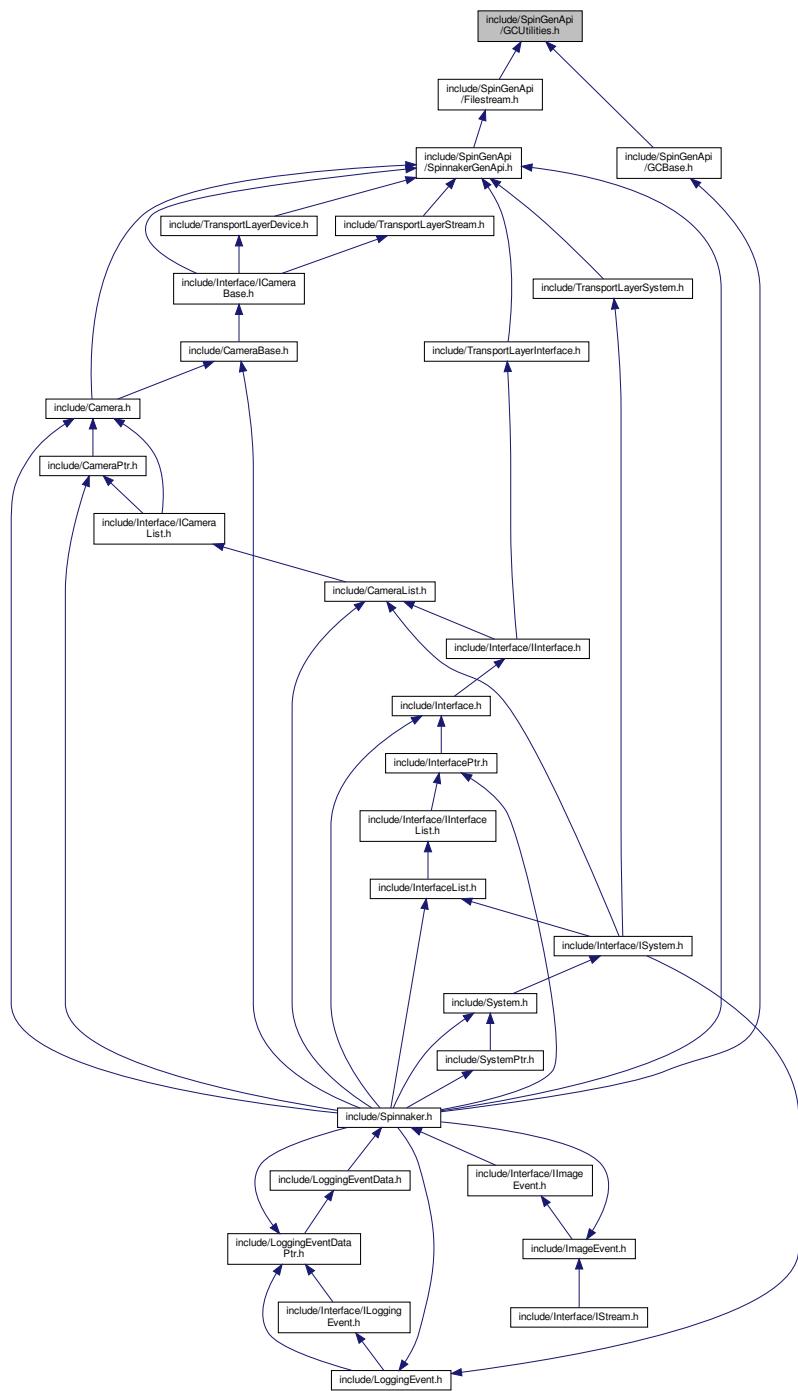
```
#define GC_UINT8_MAX static_cast<uint64_t>(0x000000000000ffULL) /* maximum unsigned int8 value */
```

## 11.84 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 **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 &delimiter=" ")  
*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.*

## 11.84.1 Macro Definition Documentation

### 11.84.1.1 [\\_\\_ERR\\_\\_](#)

```
#define __ERR__ __OUTPUT_FORMATER__("ERROR")
```

### 11.84.1.2 [\\_\\_LINE\\_STR\\_\\_](#)

```
#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)
```

### 11.84.1.3 [\\_\\_LOCATION\\_\\_](#)

```
#define __LOCATION__ __FILE__ "(" __LINE_STR__ ") "
```

### 11.84.1.4 [\\_\\_OUTPUT\\_FORMATER\\_\\_](#)

```
#define __OUTPUT_FORMATER__(
 _type) __LOCATION__ " : " _type " : "
```

### 11.84.1.5 [\\_\\_TODO\\_\\_](#)

```
#define __TODO__ __OUTPUT_FORMATER__("TBD")
```

### 11.84.1.6 [\\_\\_WARN\\_\\_](#)

```
#define __WARN__ __OUTPUT_FORMATER__("WARNING")
```

**11.84.1.7 \_TO\_STRING**

```
#define _TO_STRING(
```

```
 __stN) #__stN
```

**11.84.1.8 EXPAND\_TO\_STRINGISE**

```
#define EXPAND_TO_STRINGISE(
```

```
 __stN) _TO_STRING(__stN)
```

**11.84.1.9 GC\_COUNTOF**

```
#define GC_COUNTOF(
```

```
 arr) (sizeof(arr) / sizeof(arr)[0])
```

**11.84.1.10 GENICAM\_DEPRECATED**

```
#define GENICAM_DEPRECATED(
```

```
 FUNCTION) FUNCTION
```

**11.84.1.11 GENICAM\_UNUSED**

```
#define GENICAM_UNUSED(
```

```
 unused_var) ((void)(unused_var))
```

**11.84.1.12 USE\_TEMP\_CACHE\_FILE [1/2]**

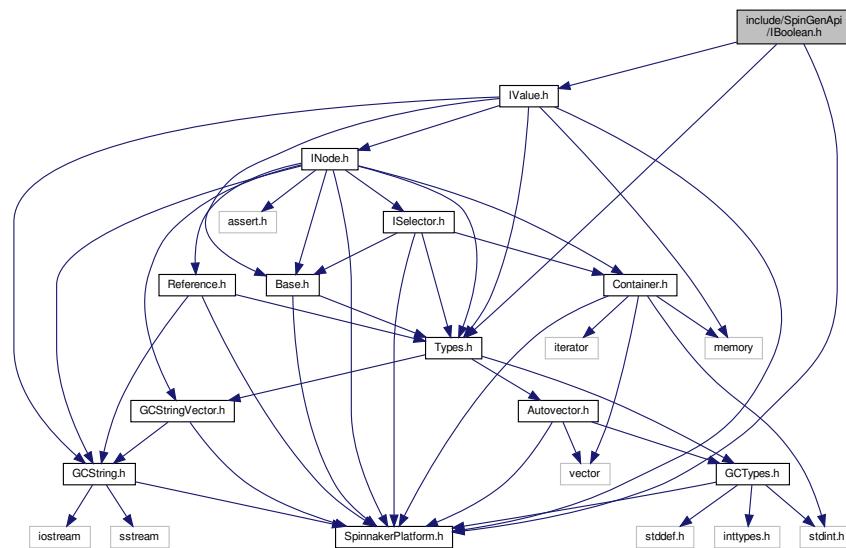
```
#define USE_TEMP_CACHE_FILE 1
```

**11.84.1.13 USE\_TEMP\_CACHE\_FILE [2/2]**

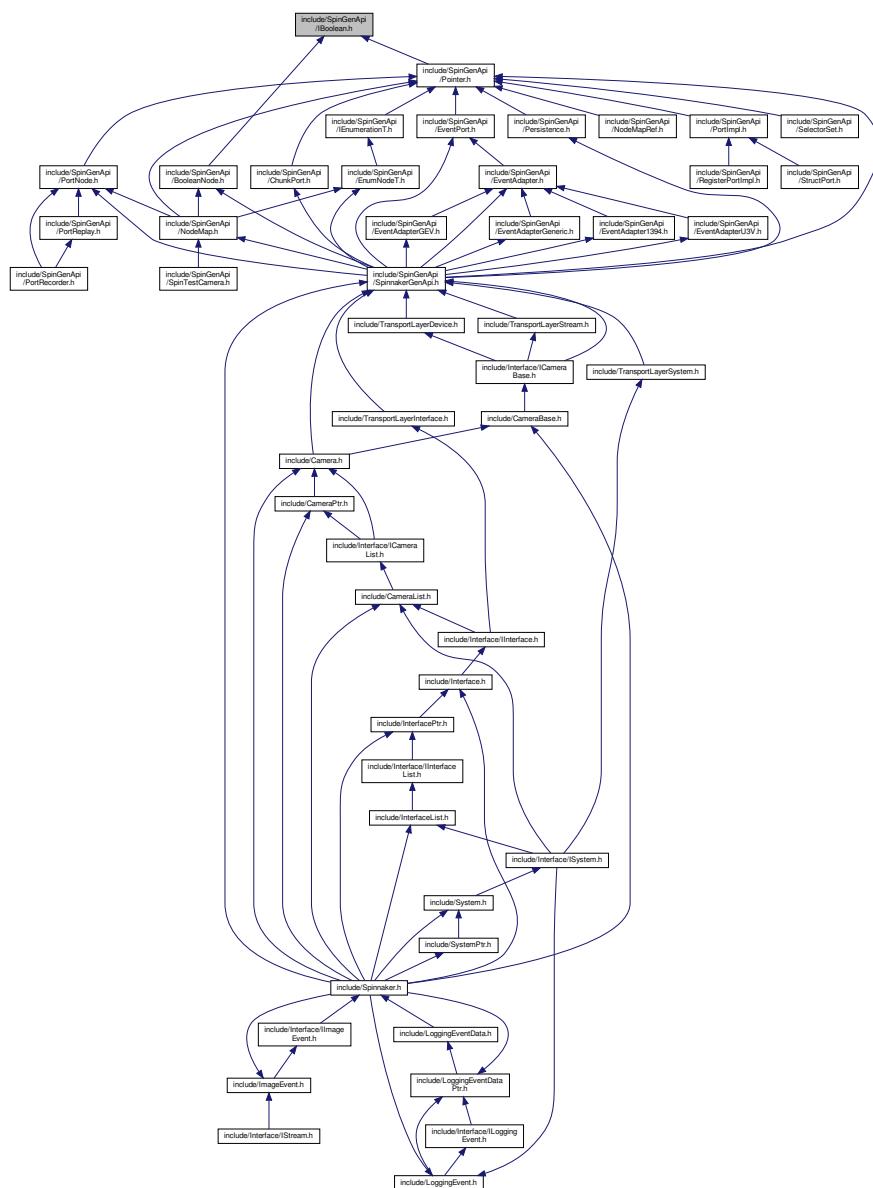
```
#define USE_TEMP_CACHE_FILE 1
```

## 11.85 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

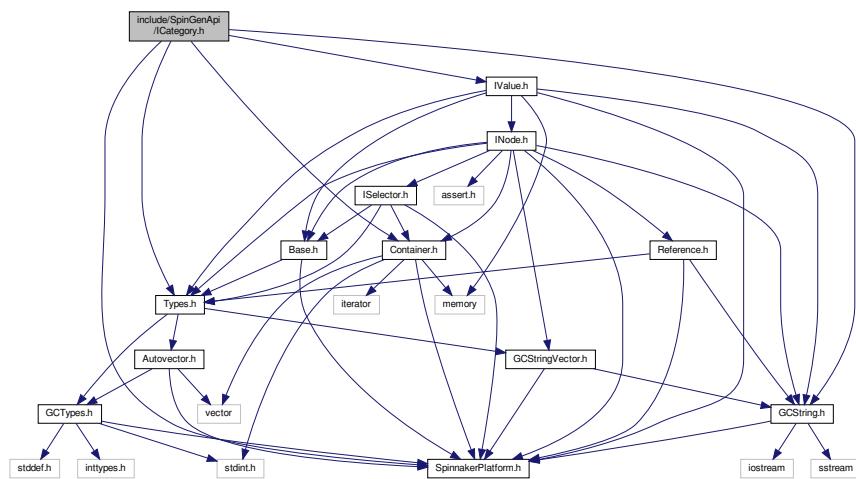
- interface SPINNAKER\_API\_ABSTRACT IBoolean

*Interface* for Boolean properties.

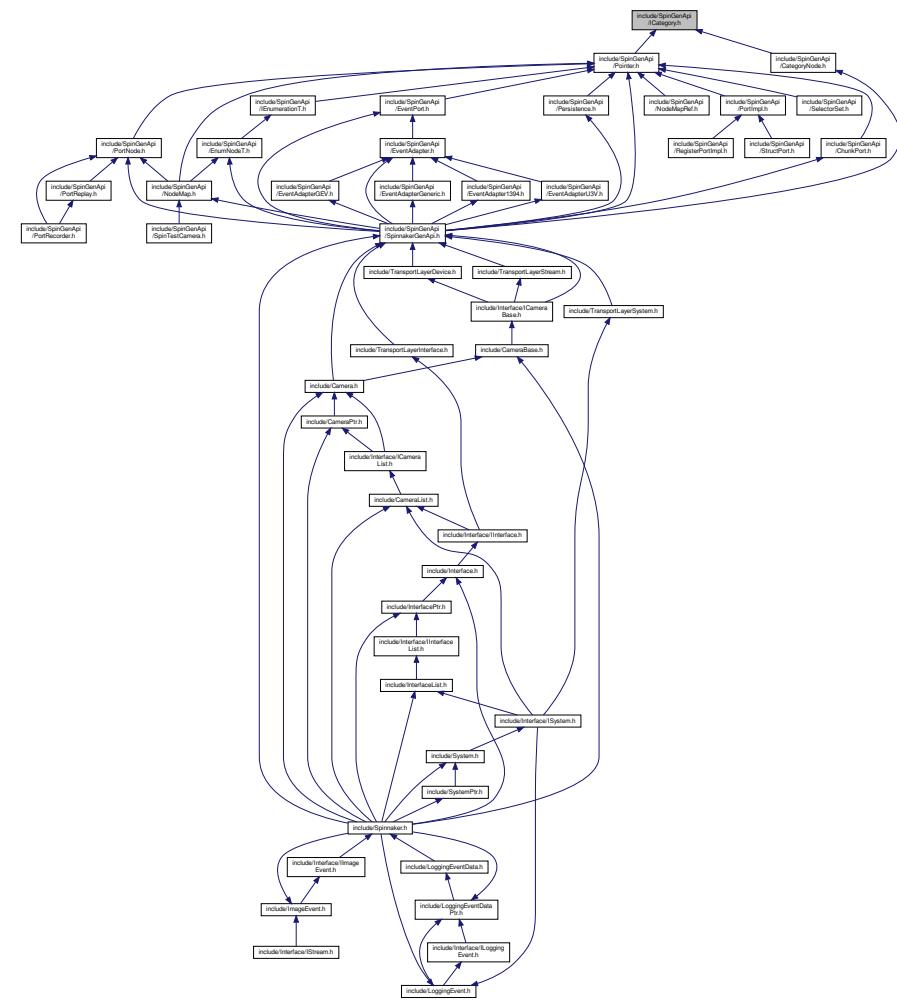
- interface SPINNAKER\_API\_ABSTRACT bool Verify = true) = 0

## 11.86 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

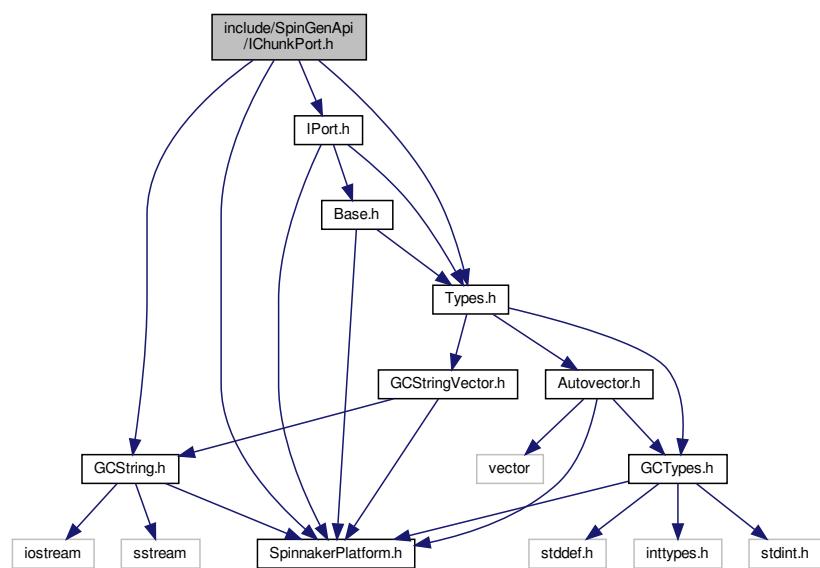
## Variables

- interface SPINNAKER\_API\_ABSTRACT ICategory

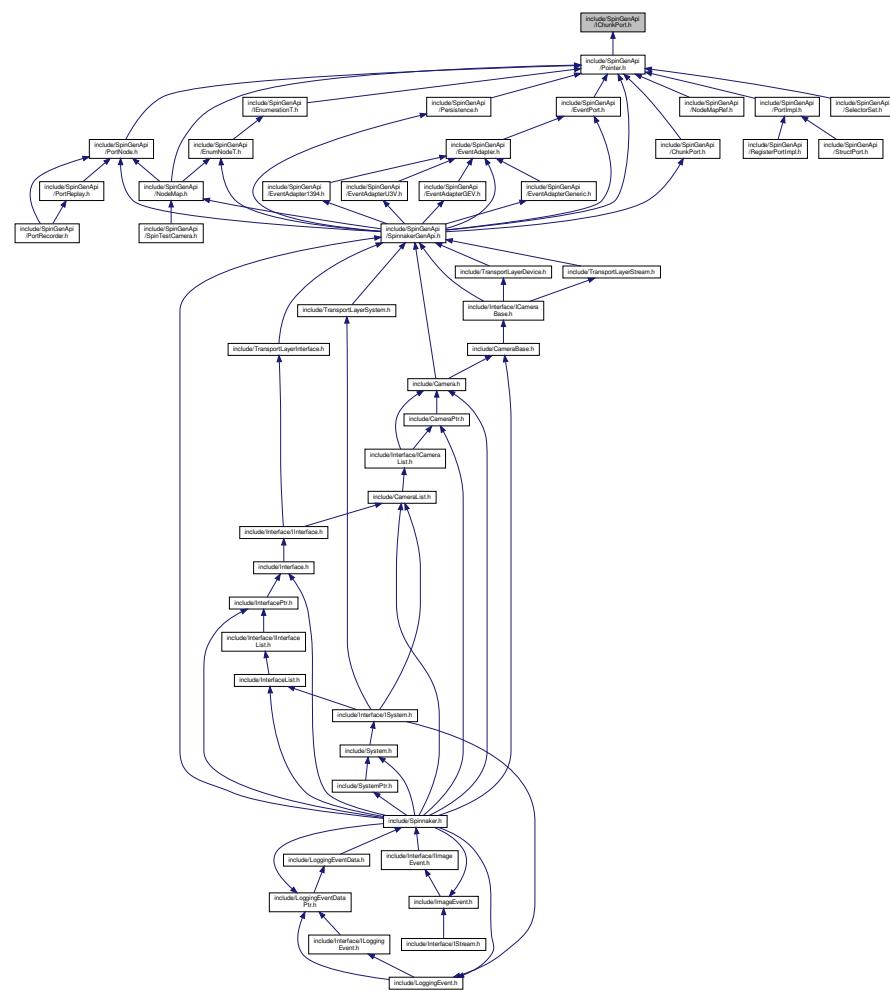
*Gives access to a category node.*

## 11.87 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**  
*Lenght 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**  
*Lenght 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.*

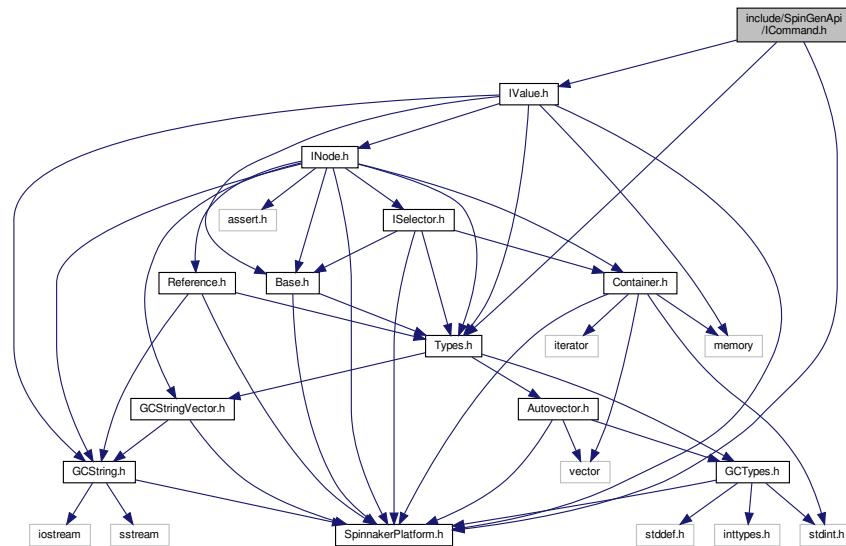
## Variables

- interface SPINNAKER\_API\_ABSTRACT IChunkPort

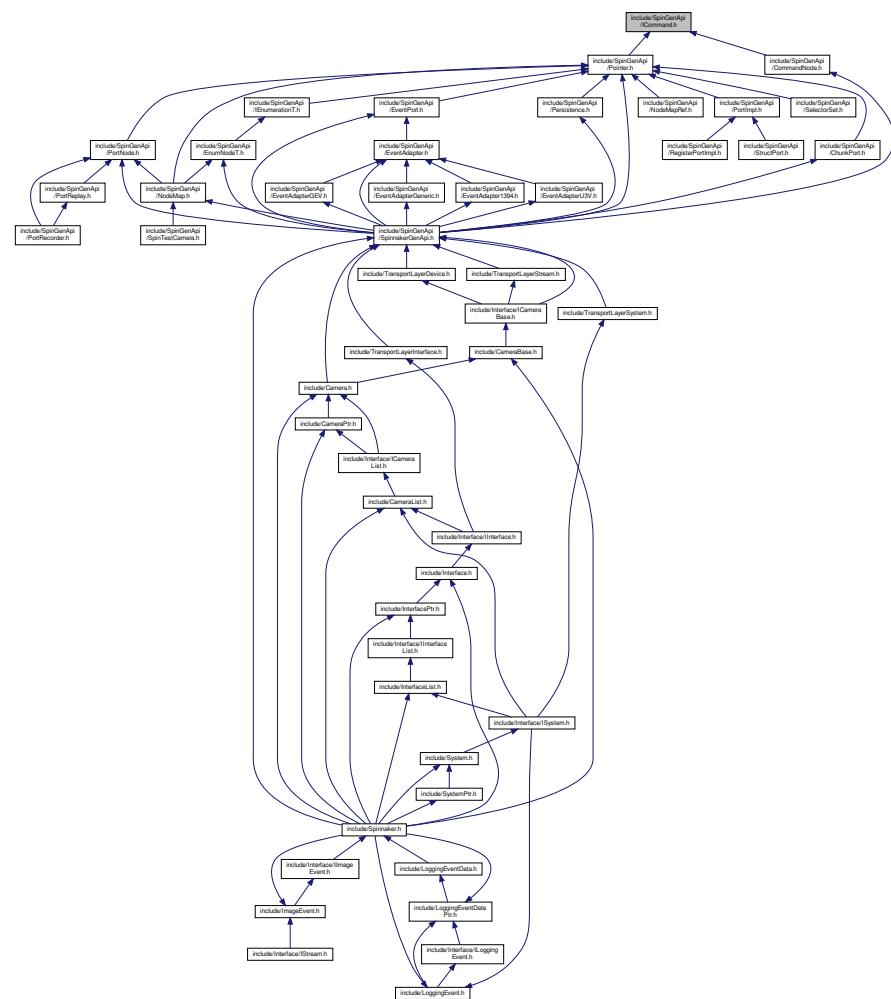
*Interface for ports attached to a chunk.*

## 11.88 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

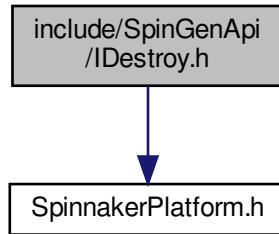
- virtual bool **operator()** () const  
*Get node value.*
  - virtual bool **IsDone** (bool Verify=true)=0  
*Query whether the command is executed.*

## Variables

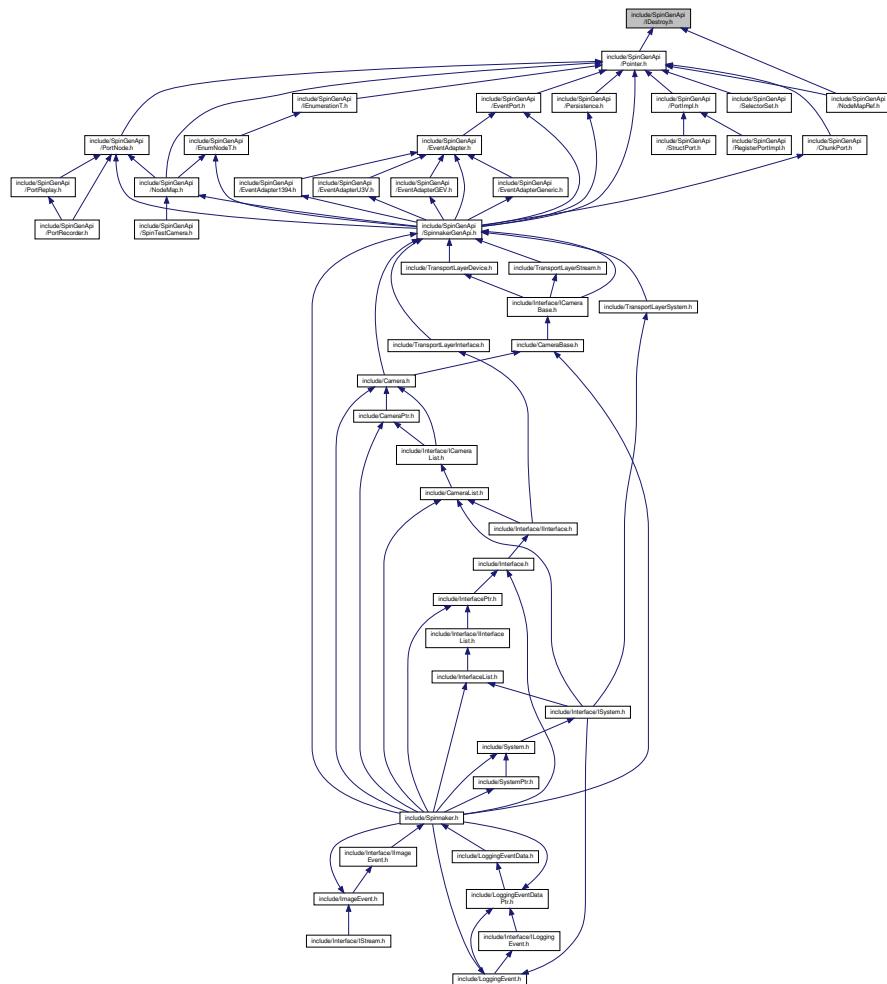
- interface **SPINNAKER\_API\_ABSTRACT ICommand**  
*Interface for command like properties.*

## 11.89 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

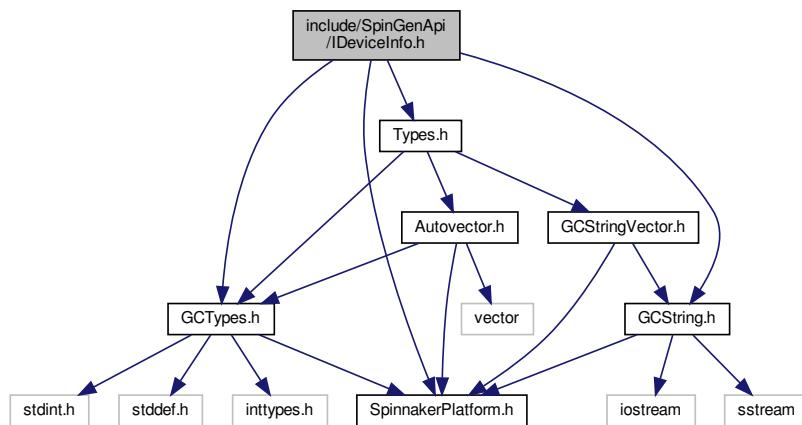
## Variables

- interface SPINNAKER\_API\_ABSTRACT IDestroy

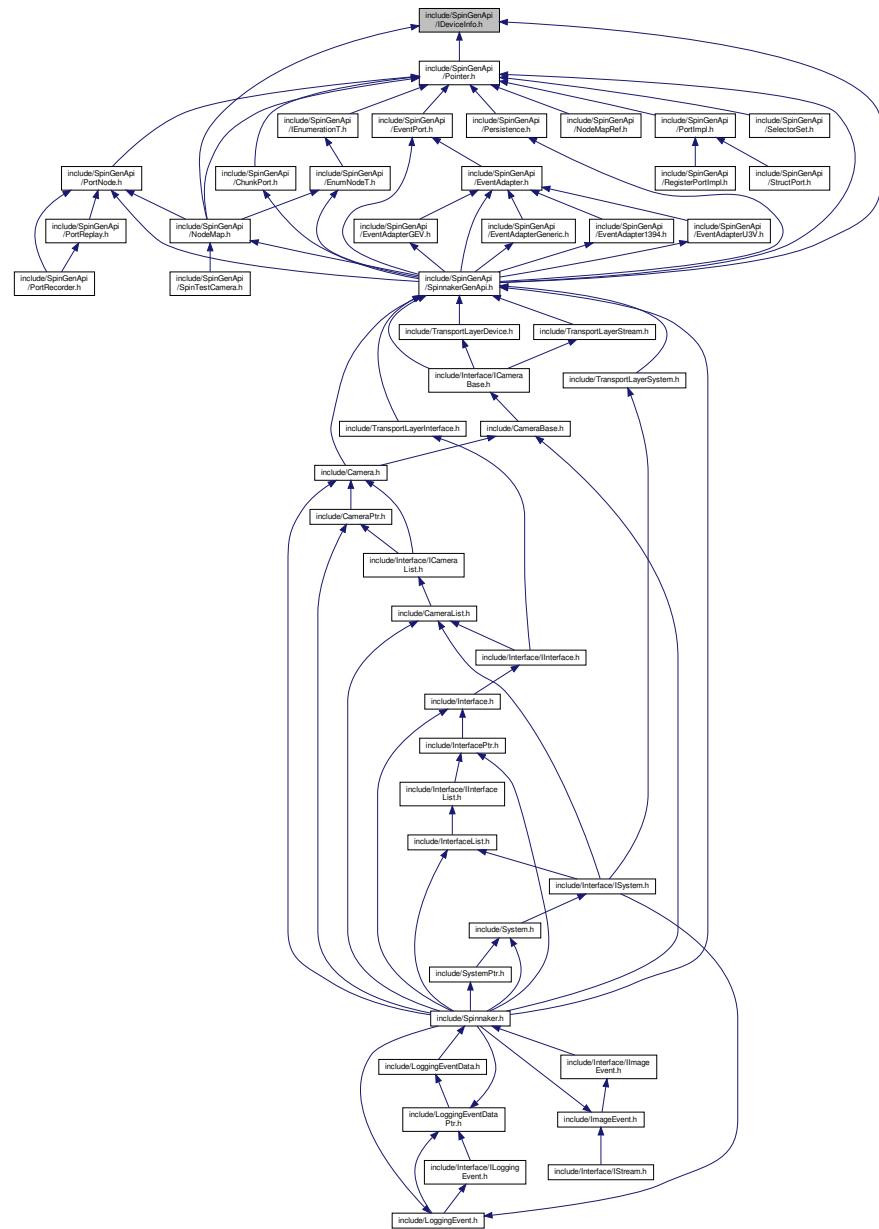
*Interface to destroy an object.*

## 11.90 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.*

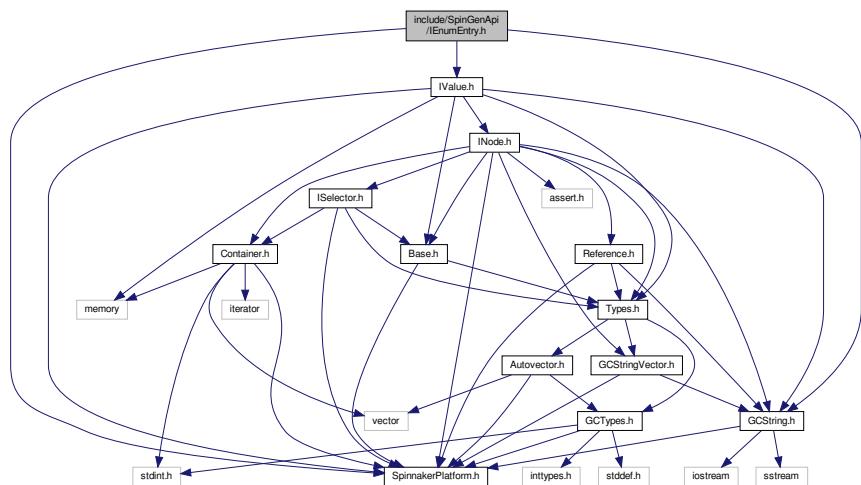
- virtual GenICam::gcstring [GetStandardNameSpace](#) ()=0  
*Get the standard name space.*
  - virtual void [GetGenApiVersion](#) (GenICam::Version\_t &Version, uint16\_t &Build)=0  
*Get the version of the DLL's [GenApi](#) implementation.*
  - virtual void [GetSchemaVersion](#) (GenICam::Version\_t &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

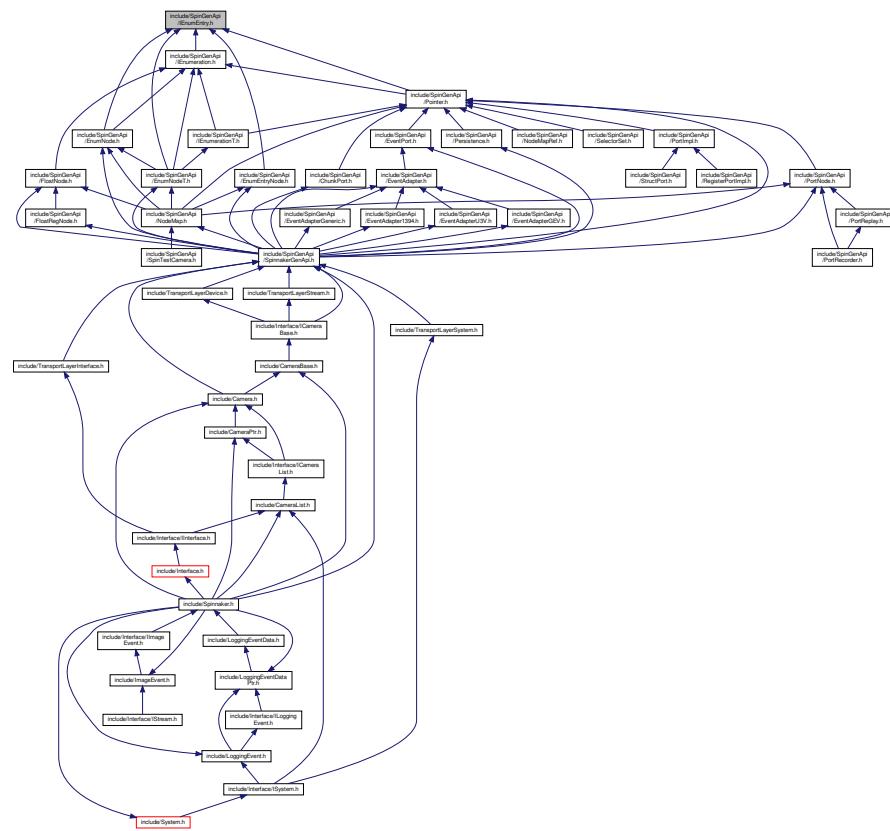
- interface SPINNAKER\_API\_ABSTRACT IDeviceInfo  
*Interface* to get information about the device (= nodemap)

11.91 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

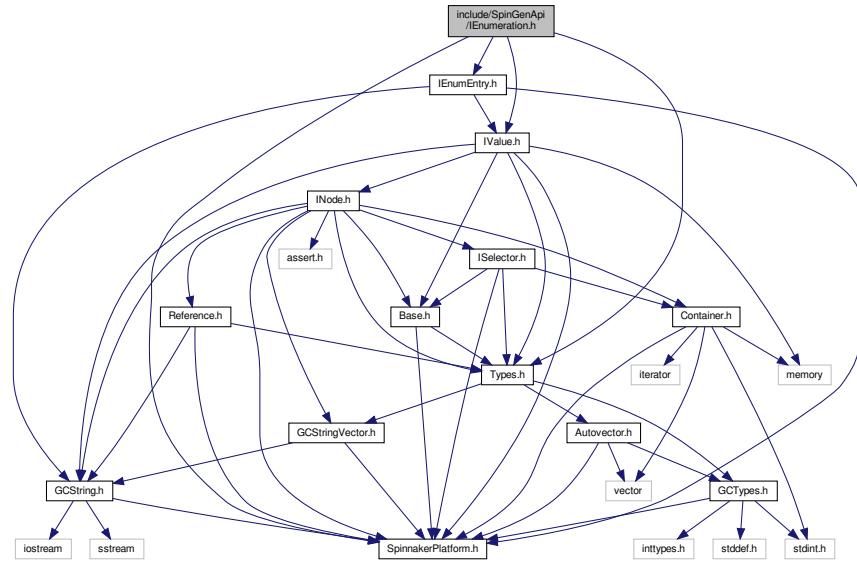
- virtual GenICam::gcstring [GetSymbolic \(\) const =0](#)  
*Get symbolic enum value.*
- virtual double [GetNumericValue \(\)=0](#)  
*Get double number associated with the entry.*
- virtual bool [IsSelfClearing \(\)=0](#)  
*Indicates if the corresponding EnumEntry is self clearing.*

## Variables

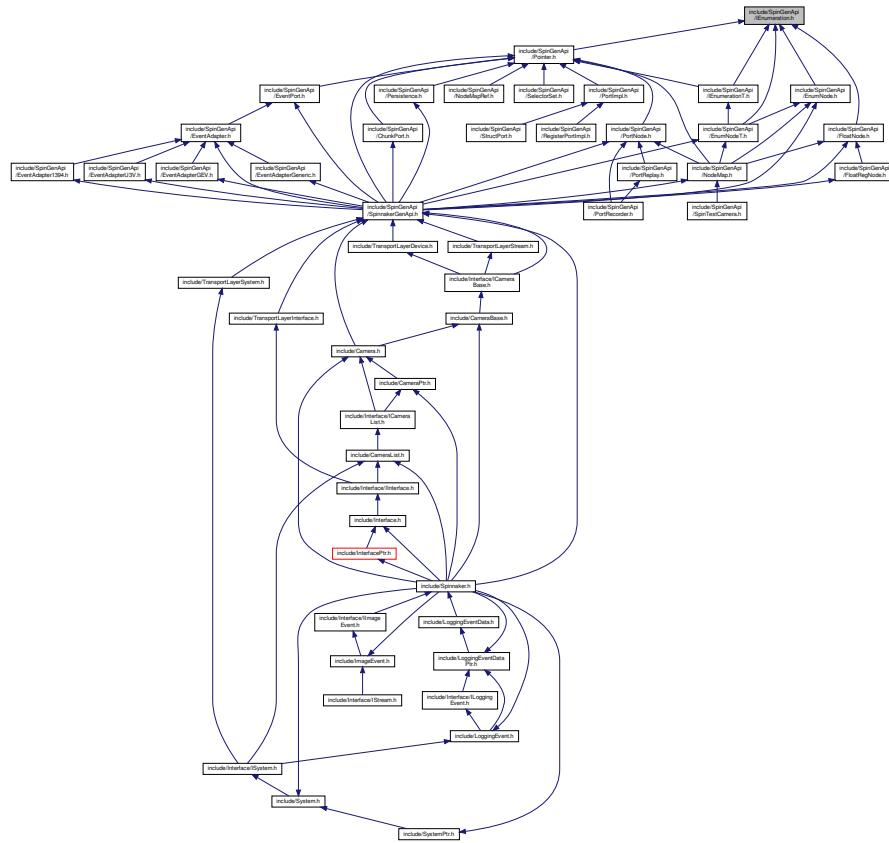
- interface SPINNAKER\_API\_ABSTRACT [IEnumEntry](#)  
*Interface of single enum value.*

## 11.92 include/SpinGenApi/IEnumeration.h File Reference

Include dependency graph for IEnumeration.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

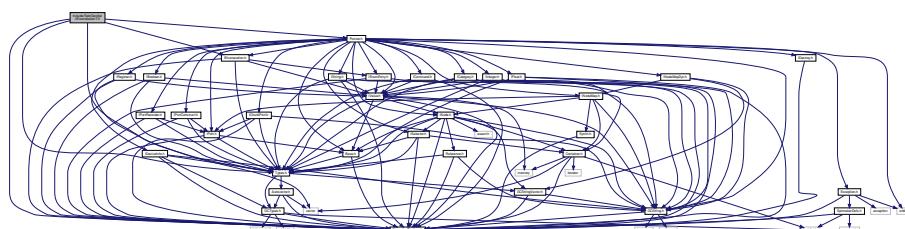
- virtual void [GetEntries](#) (NodeList\_t &Entries)=0  
*Get list of entry nodes.*
- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0  
*Set string node value.*
- virtual void [SetIntValue](#) (int64\_t Value, bool Verify=true)=0  
*Set integer node value.*
- virtual GenICam::gcstring [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

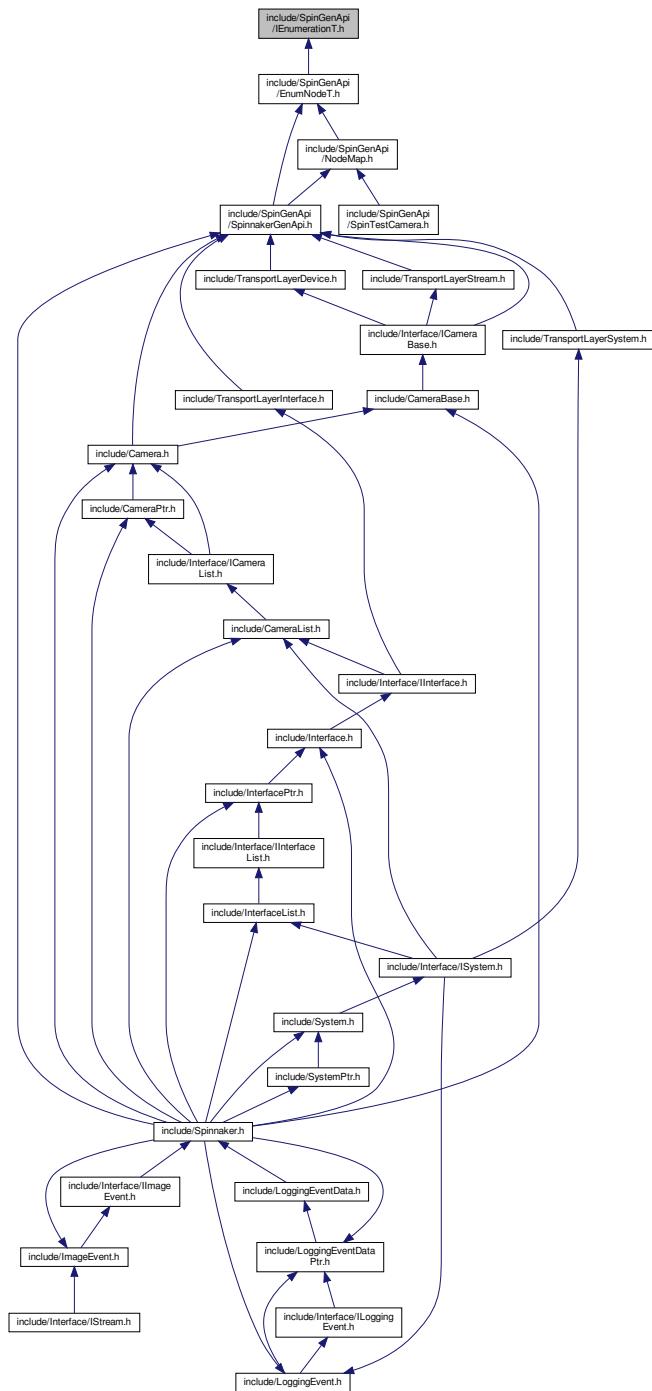
- interface SPINNAKER\_API\_ABSTRACT IEnumeration  
*Interface for enumeration properties.*

## 11.93 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

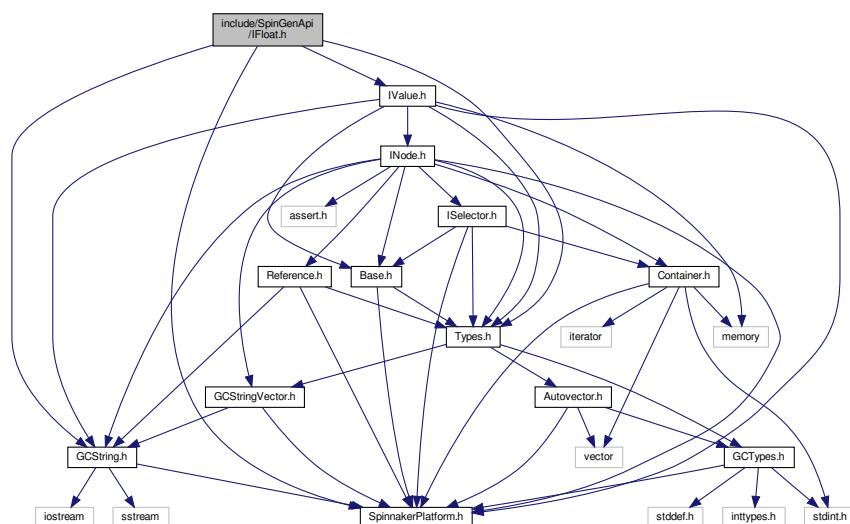
- Set node value.*
- virtual bool `GetValue` (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
  - virtual bool `operator()` () const  
*Get node value.*
  - virtual IEnumeration & `operator=` (const GenICam::gcstring &ValueStr)=0  
*Set string node value.*
  - virtual IEnumEntry \* `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

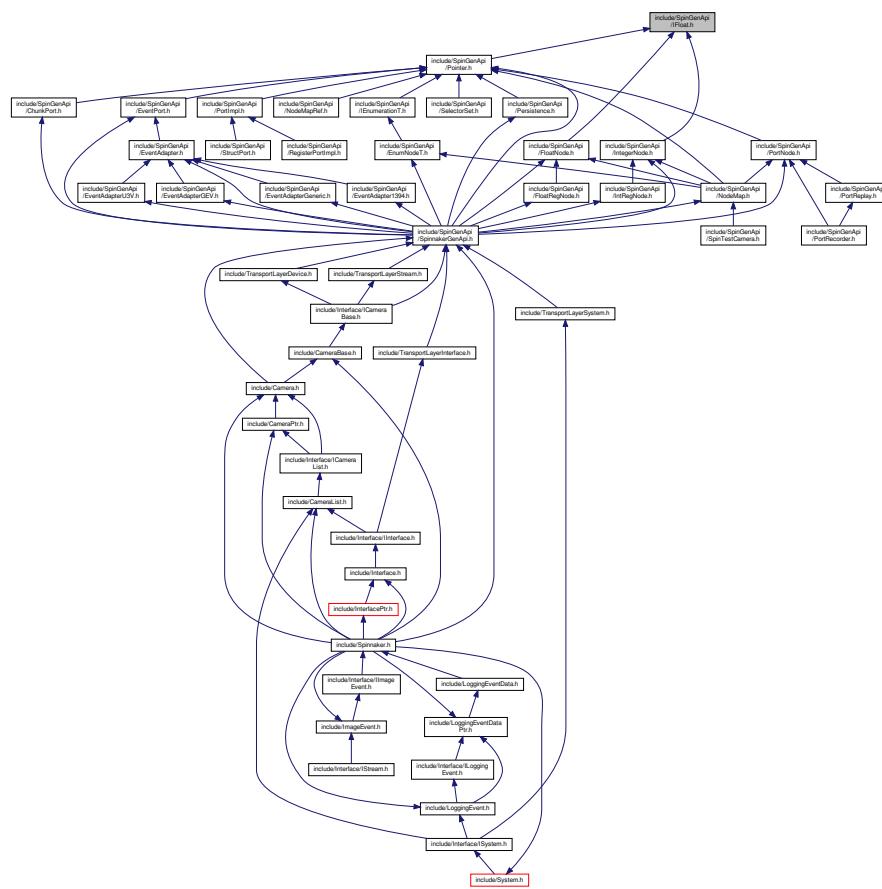
- template<typename EnumT >  
`interface SPINNAKER_API_ABSTRACT IEnumerationT`  
*Interface for enumeration properties.*
- template<typename EnumT >  
`interface SPINNAKER_API_ABSTRACT virtual public IEnumReference`  
*Interface to construct an enum reference.*

## 11.94 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

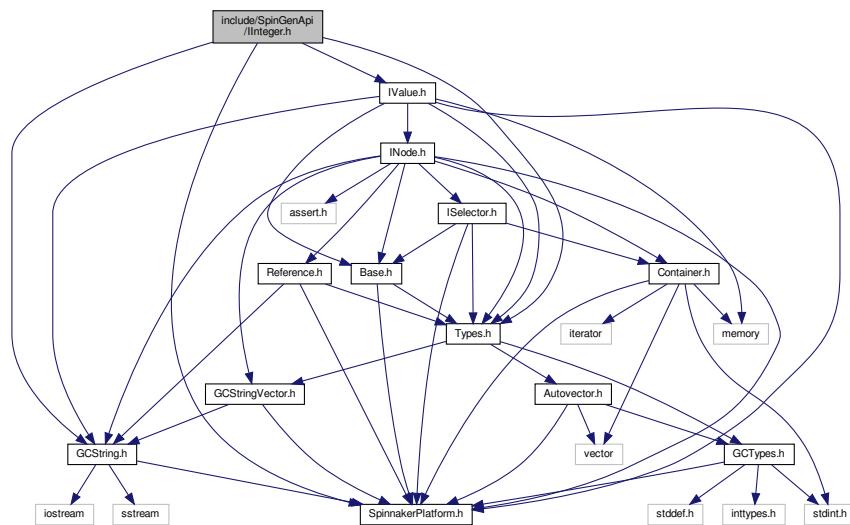
- `virtual EIncMode GetIncMode ()=0`  
*Get increment mode.*
- `virtual double GetInc ()=0`  
*Get the constant increment if there is any.*
- `virtual double _autovector_t GetListOfValidValues (bool bounded=true)=0`  
*Get list of valid value.*
- `virtual ERepresentation GetRepresentation ()=0`  
*Get 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

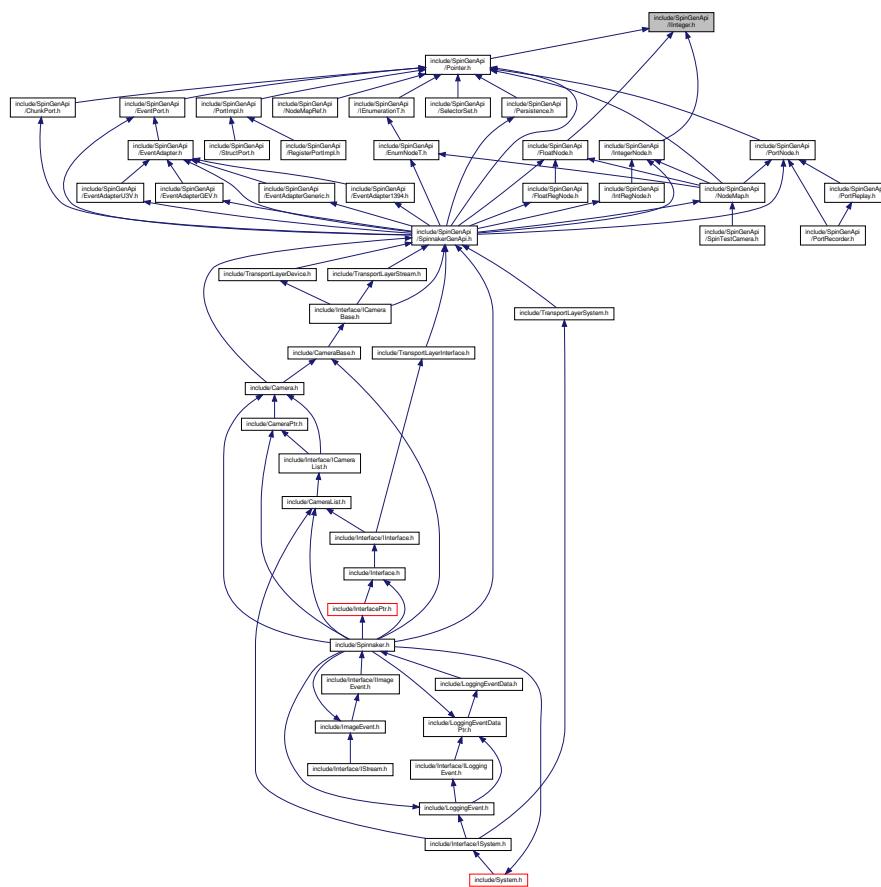
- `interface SPINNAKER_API_ABSTRACT IFloat`  
*Interface for float properties.*

## 11.95 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

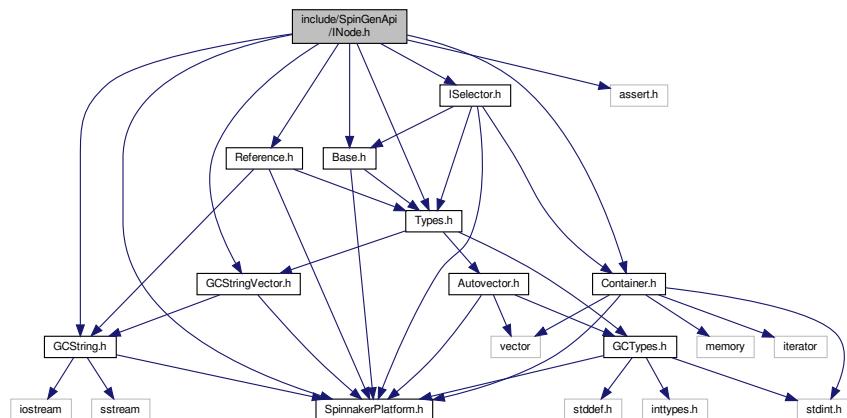
- *Get increment mode.*  
virtual double **GetInc** ()=0
- *Get the constant increment if there is any.*  
virtual double<sub>\_autovector\_t</sub> **GetListOfValidValues** (bool bounded=true)=0
- *Get list of valid value.*  
virtual ERepresentation **GetRepresentation** ()=0
- *Get recommended representation.*  
virtual GenICam::gcstring **GetUnit** () const =0
- *Get the physical unit name.*  
virtual void **ImposeMin** (int64\_t Value)=0
- *Restrict minimum value.*
- *Get maximum value.*  
virtual void **ImposeMax** (int64\_t Value)=0
- *Restrict maximum value.*

## Variables

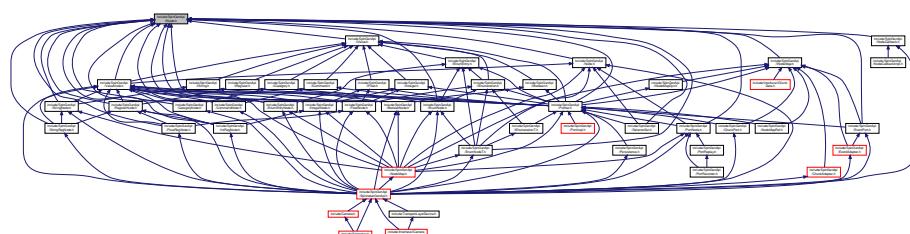
- interface SPINNAKER\_API\_ABSTRACT IInteger  
*Interface for integer properties.*

## 11.96 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 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 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 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

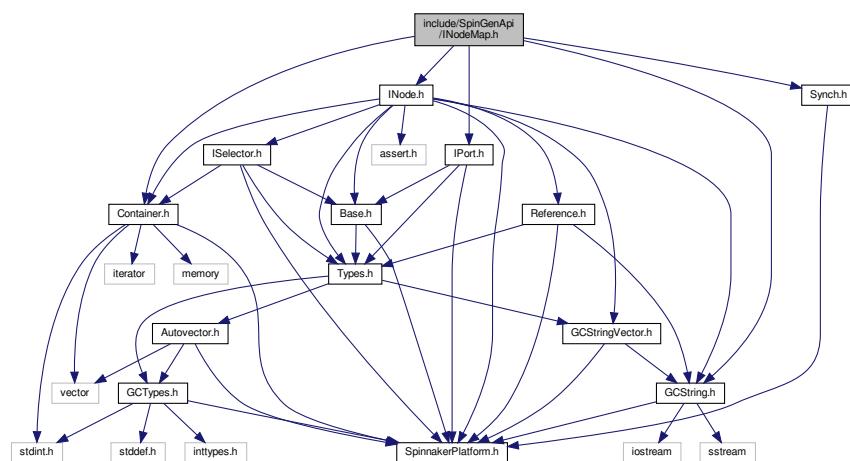
- interface SPINNAKER\_API\_ABSTRACT INode
 

*Interface common to all nodes.*
- interface SPINNAKER\_API\_ABSTRACT virtual public IReference
 

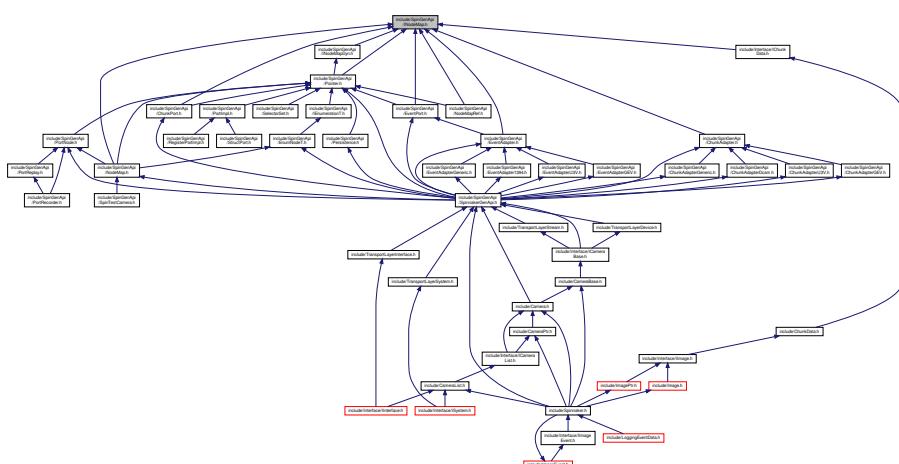
*Interface to construct a reference.*

## 11.97 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

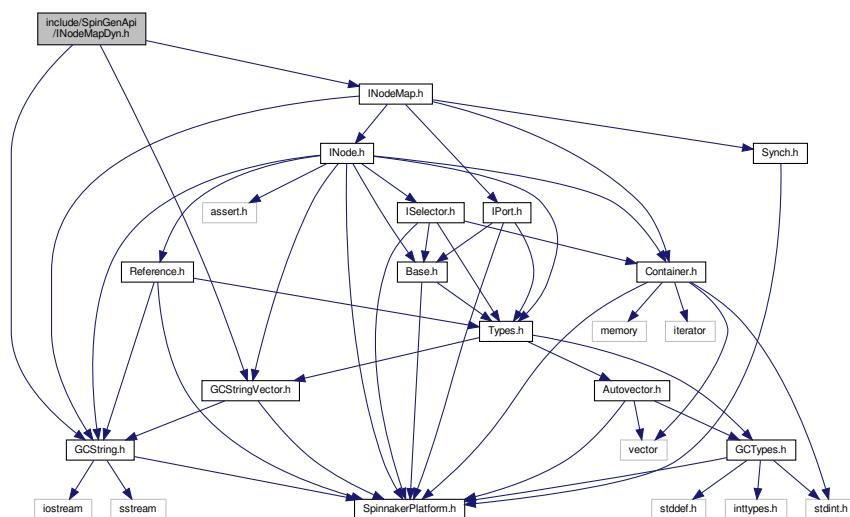
- virtual INode \* **GetNode** (const GenICam::gcstring &Name) const =0  
*Retrieves the node from the central map by Name.*
- virtual void **InvalidateNodes** () const =0  
*Invalidates all nodes.*
- virtual bool **Connect** (IPort \*pPort, const GenICam::gcstring &PortName) const =0  
*Connects a port 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

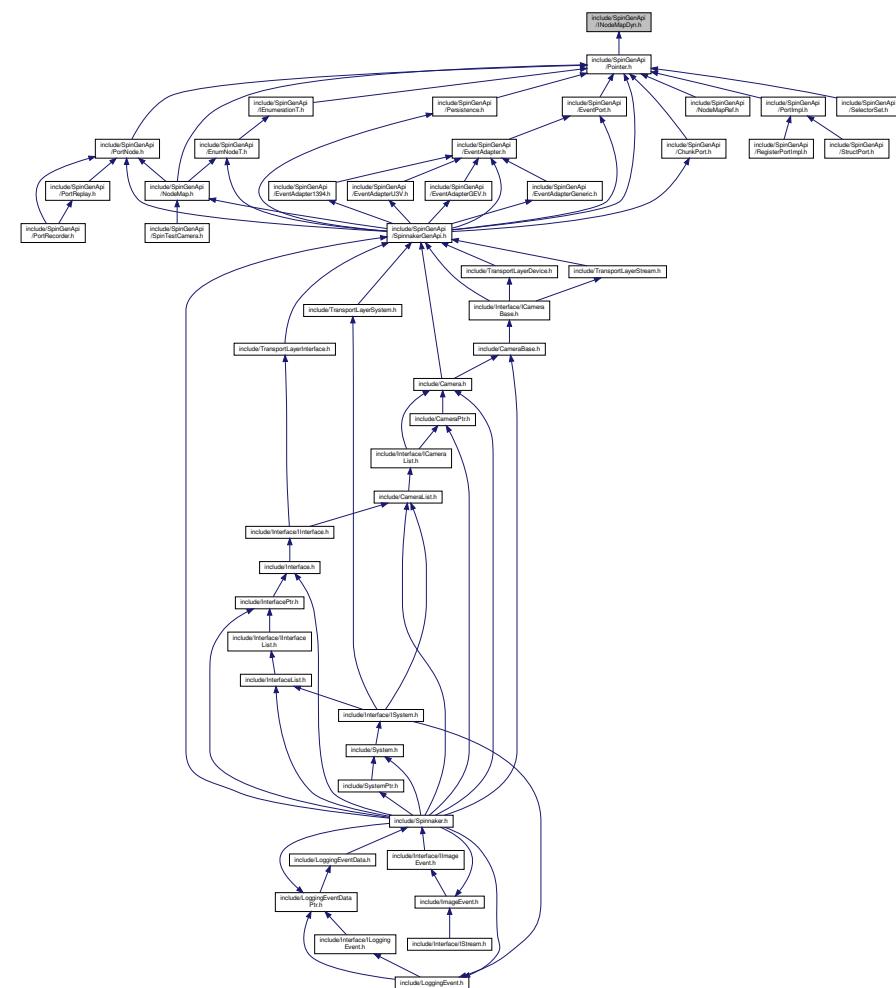
- interface SPINNAKER\_API\_ABSTRACT INodeMap  
*Interface to access the node map.*

## 11.98 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=xvDefault)=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.*

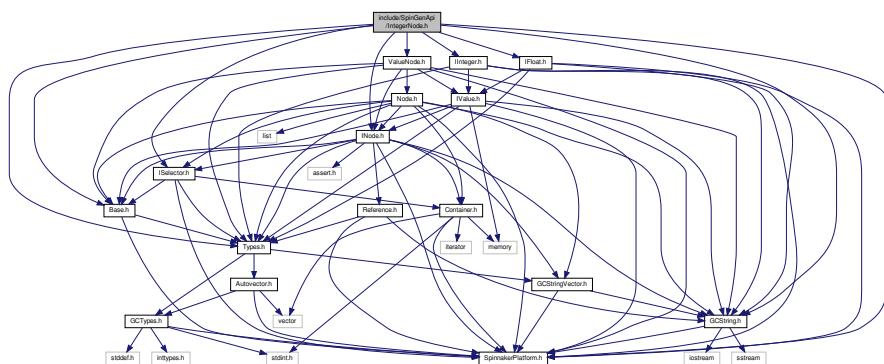
## Variables

- interface [SPINNAKER\\_API\\_ABSTRACT INodeMapDyn](#)

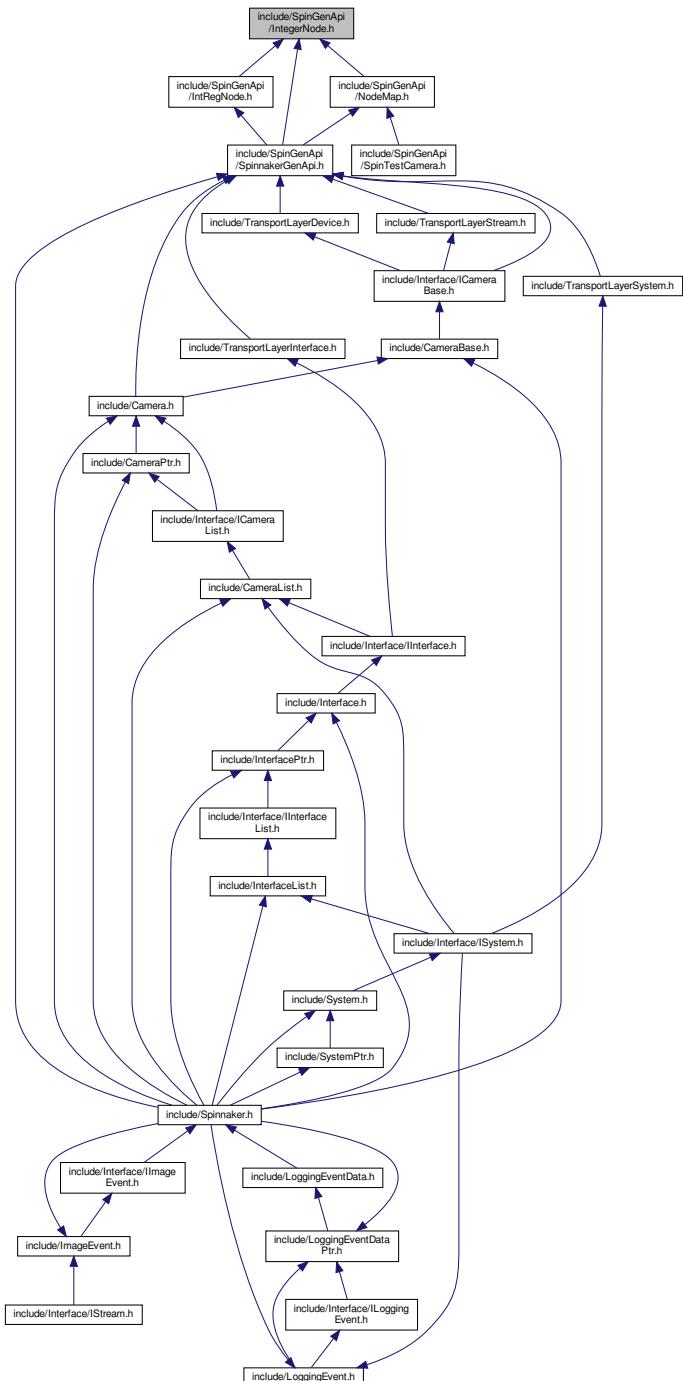
*Interface* to access the node map.

## 11.99 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

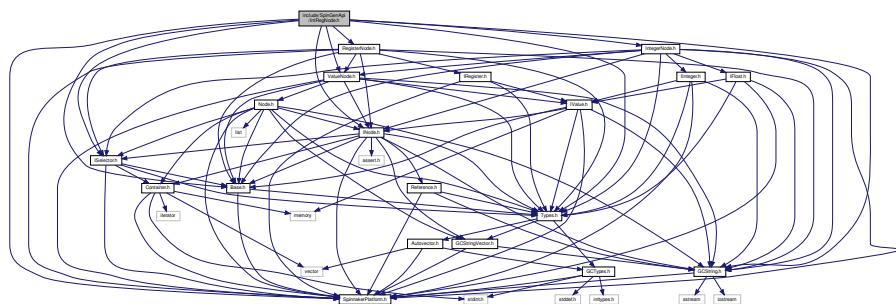
- Spinnaker
  - Spinnaker::GenApi

## TypeDefs

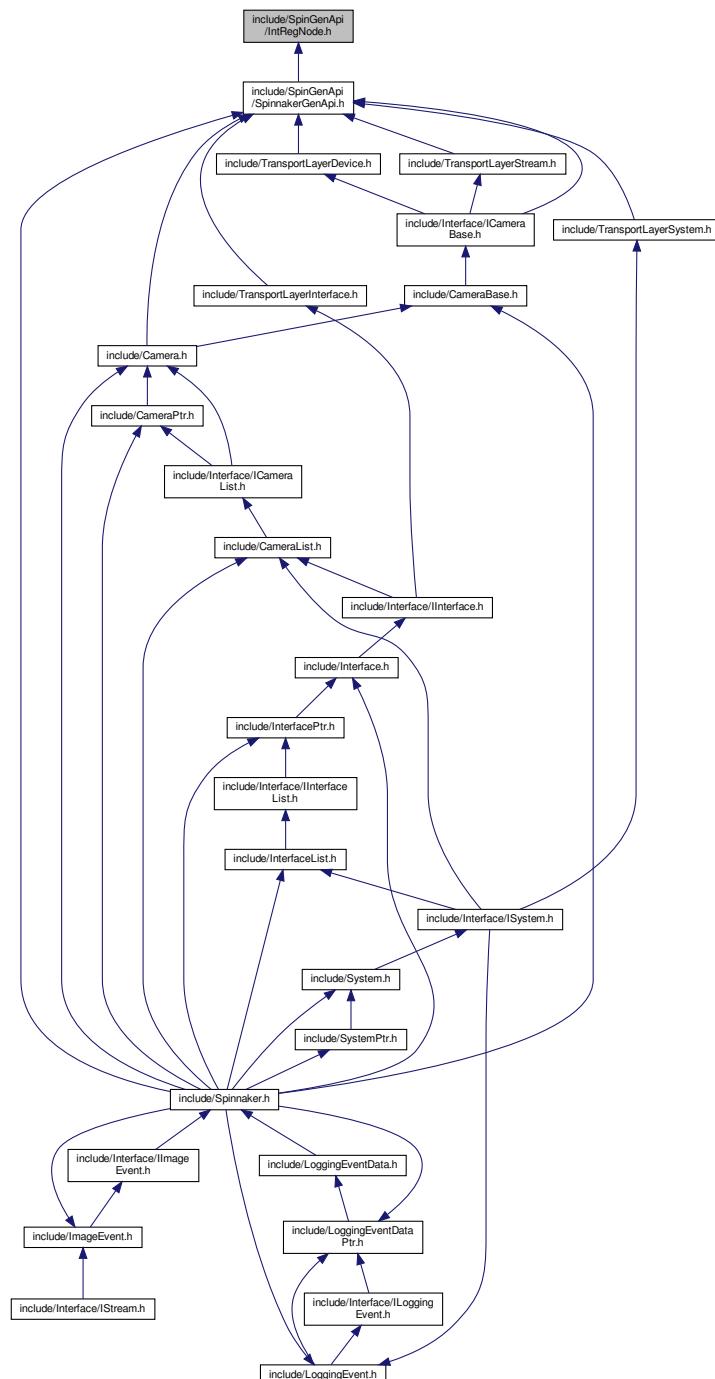
- `typedef IntegerNode CIntegerRef`

## 11.100 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](#)

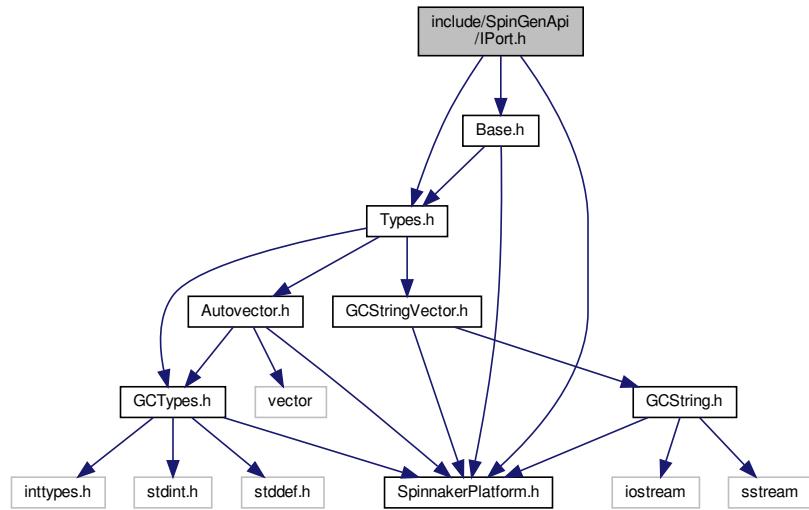
*Interface for string properties.*

## Namespaces

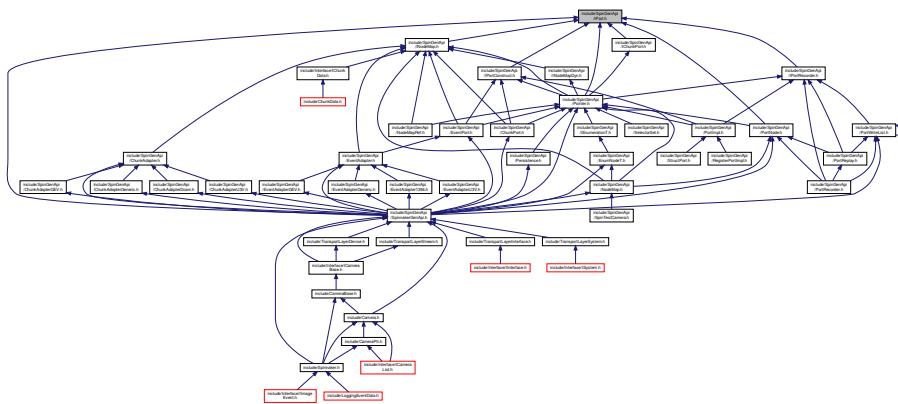
- Spinnaker
- Spinnaker::GenApi

## 11.101 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

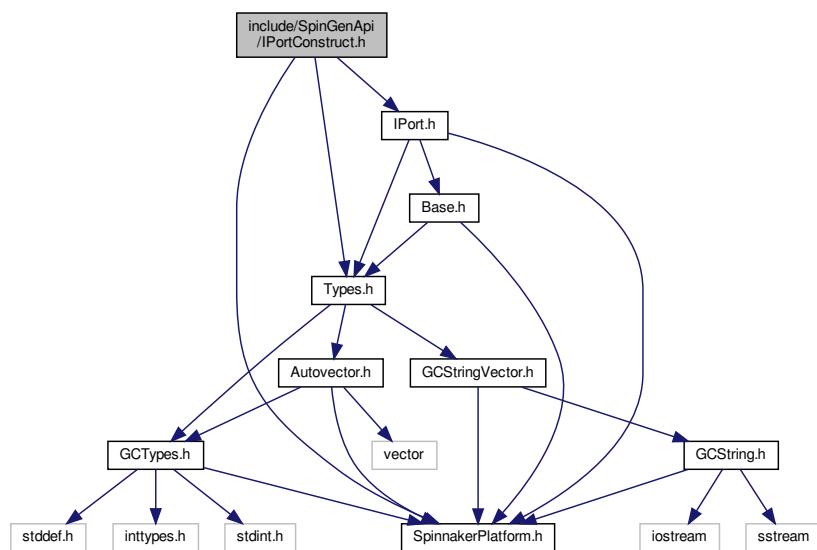
*Writes a chunk of bytes to the port.*

## Variables

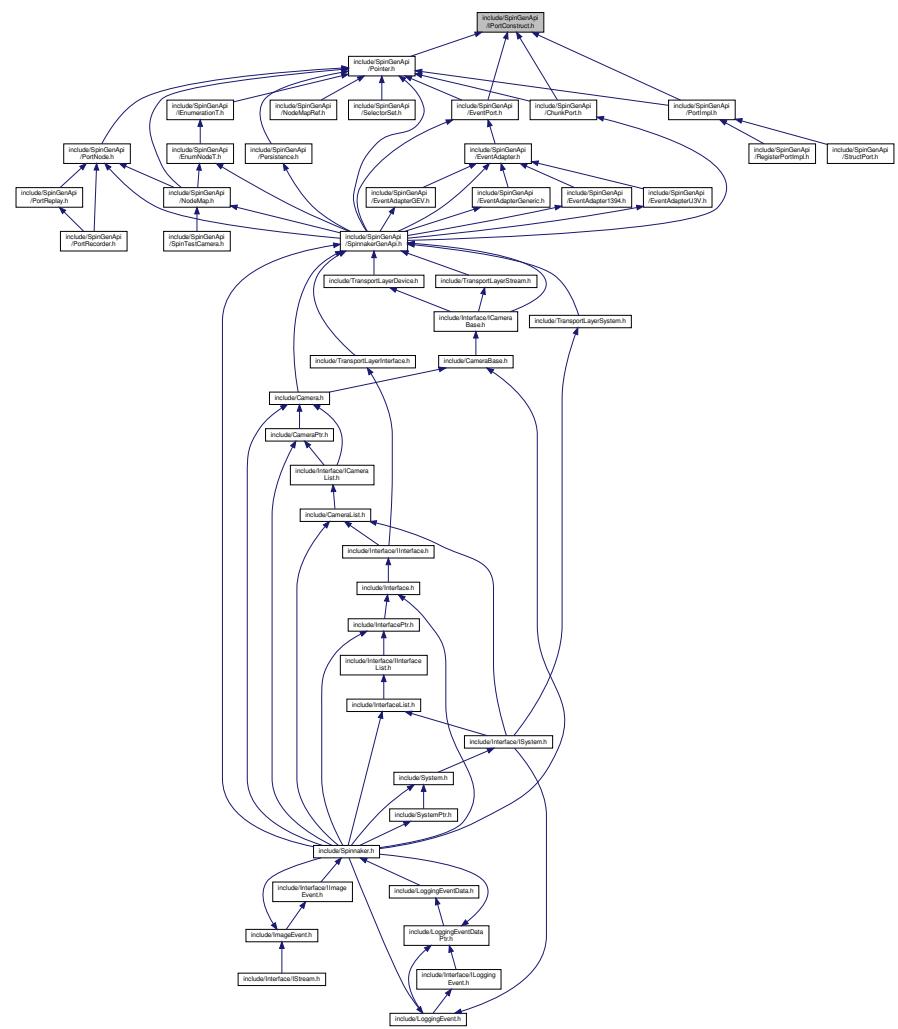
- interface SPINNAKER\_API\_ABSTRACT IPort  
*Interface for ports.*
- interface SPINNAKER\_API\_ABSTRACT int64\_t Address
- interface SPINNAKER\_API\_ABSTRACT int64\_t int64\_t Length = 0

## 11.102 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

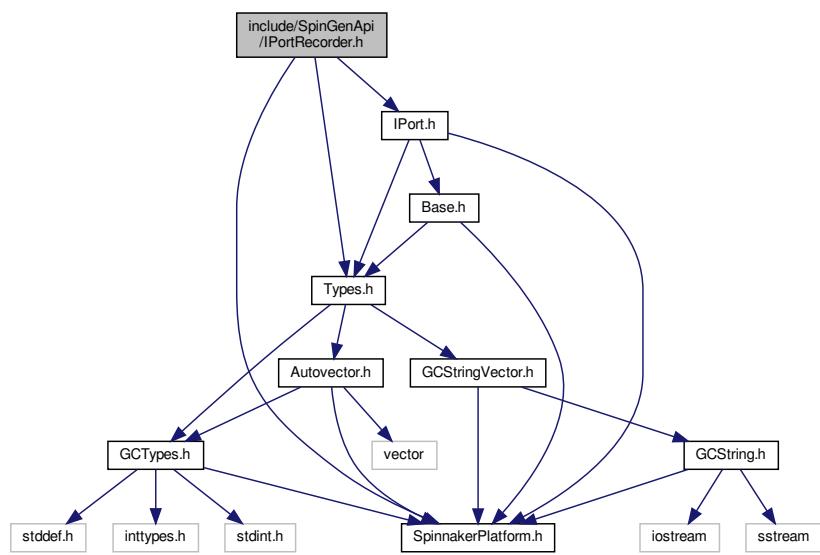
- virtual EYesNo GetSwapEndianess ()=0  
*Determines if the port adapter must perform an endianess swap.*

## Variables

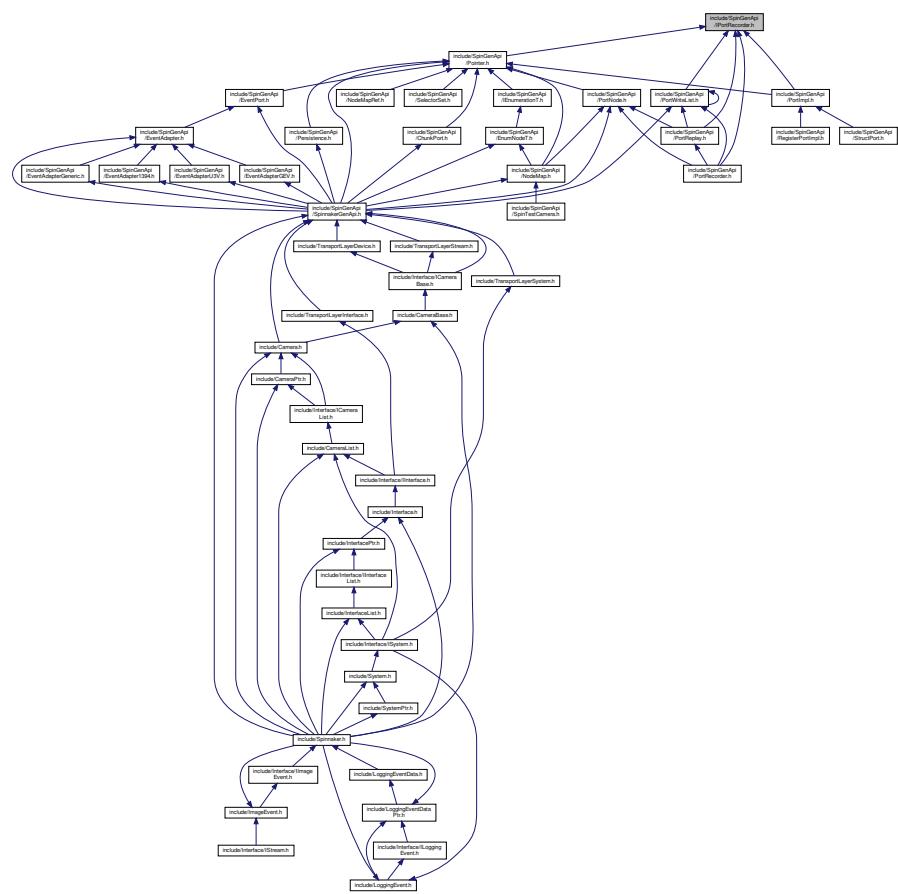
- interface SPINNAKER\_API IPortConstruct  
*Interface for ports.*

## 11.103 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

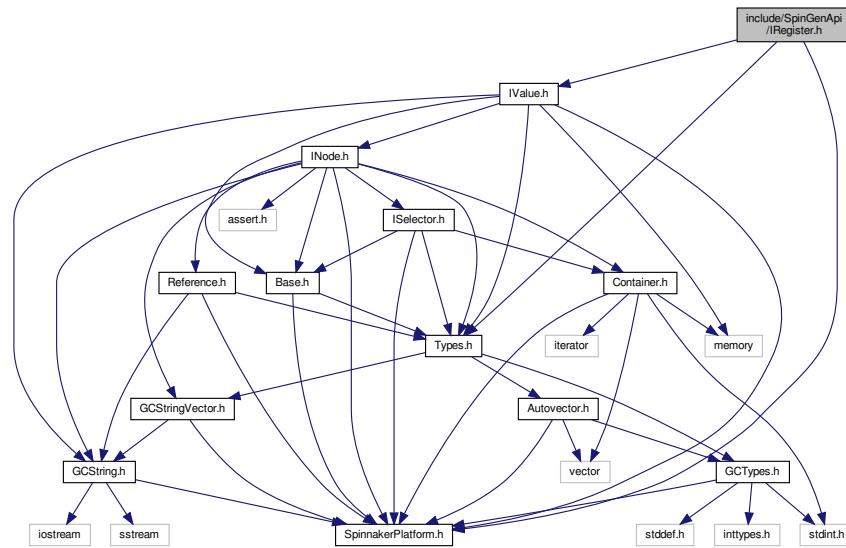
- virtual void **Replay** (IPort \*pPort)=0  
*Replays the write command to the given port interface.*
  - virtual void **SetCookie** (const int64\_t Value)=0  
*Sets a cookie in case the port implementation want to cache a command list.*
  - virtual int64\_t **GetCookie** ()=0  
*Gets the cookie a port implementation may have set for caching a command list.*
  - virtual void **StopRecording** ()=0  
*Stops recording.*

## Variables

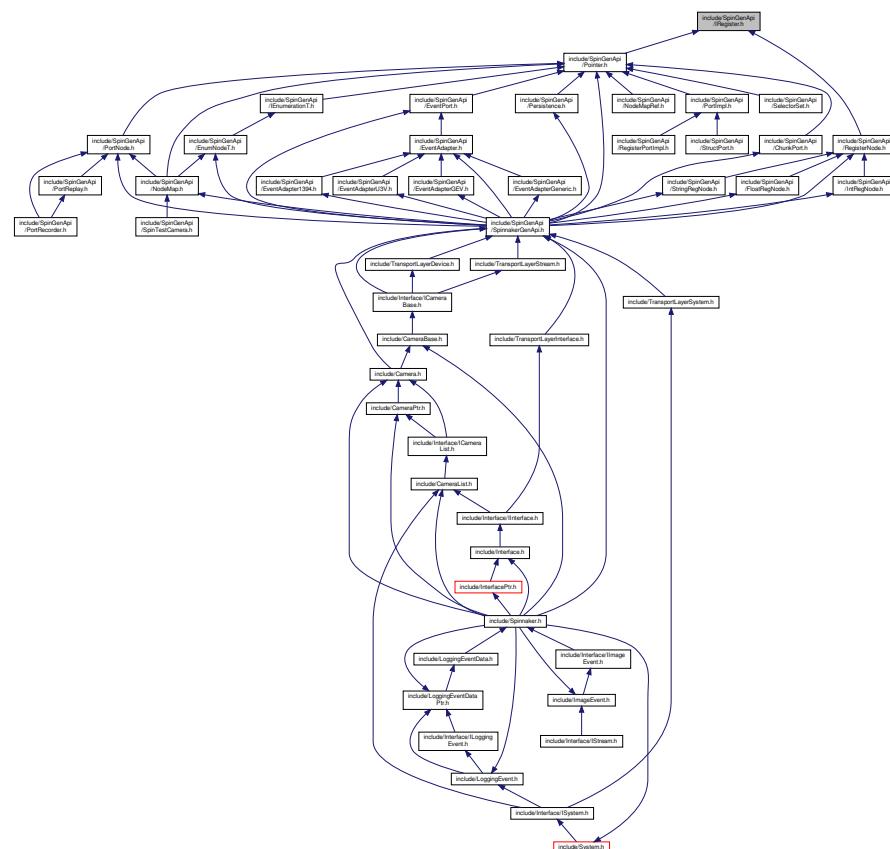
- interface SPINNAKER\_API\_ABSTRACT IPortWriteList
  - interface SPINNAKER\_API\_ABSTRACT IPortReplay
    - Interface for replaying write commands on a port.*
  - interface SPINNAKER\_API\_ABSTRACT bool Invalidate = true) = 0
  - interface SPINNAKER\_API\_ABSTRACT IPortRecorder
    - Interface for recording write commands on a port.*

## 11.104 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

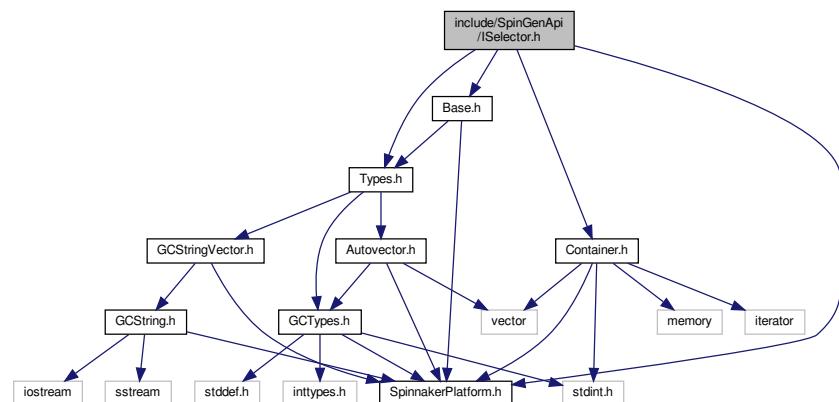
- virtual void `Get` (uint8\_t \*pBuffer, int64\_t Length, bool Verify=false, bool IgnoreCache=false)=0  
*Fills a buffer with the register's contents.*
- virtual int64\_t `GetLength` ()=0  
*Retrieves the Length of the register [Bytes].*
- virtual int64\_t `GetAddress` ()=0  
*Retrieves the Address of the register.*

## Variables

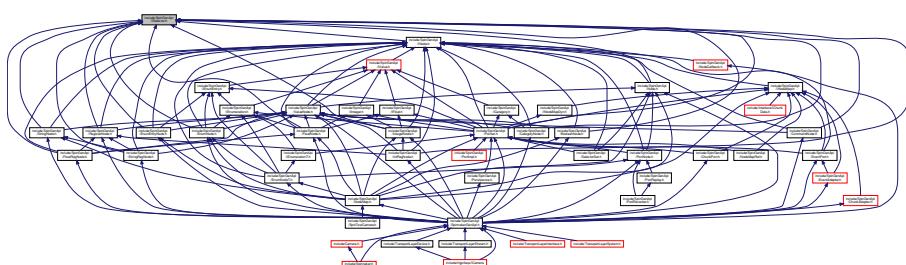
- interface SPINNAKER\_API\_ABSTRACT IRegister  
*Interface for registers.*

## 11.105 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

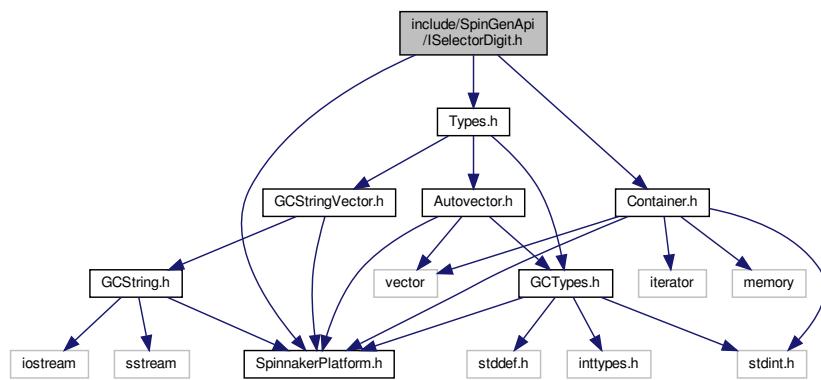
- virtual void [GetSelectedFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of selected features*
- virtual void [GetSelectingFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of features selecting this node*

## Variables

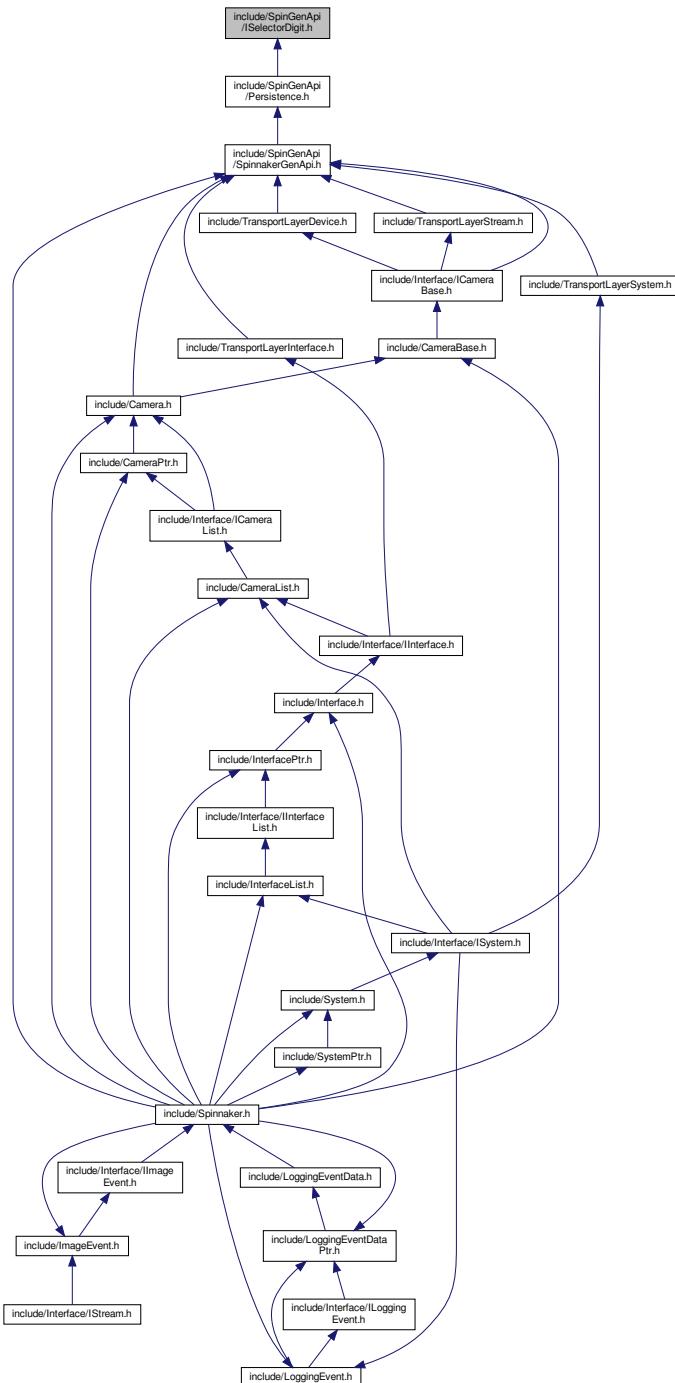
- interface [SPINNAKER\\_API\\_ABSTRACT ISelector](#)  
*Interface for groups of features selected by a single one.*

## 11.106 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.*

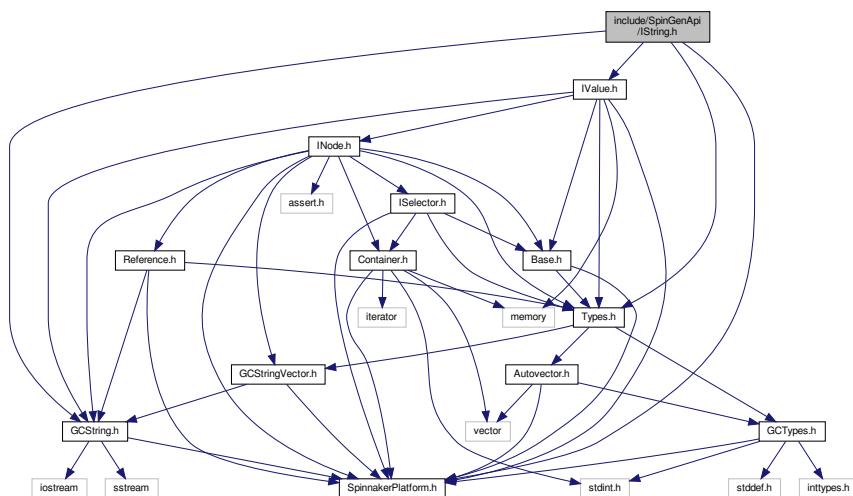
## Variables

- interface SPINNAKER\_API\_ABSTRACT [ISelectorDigit](#)

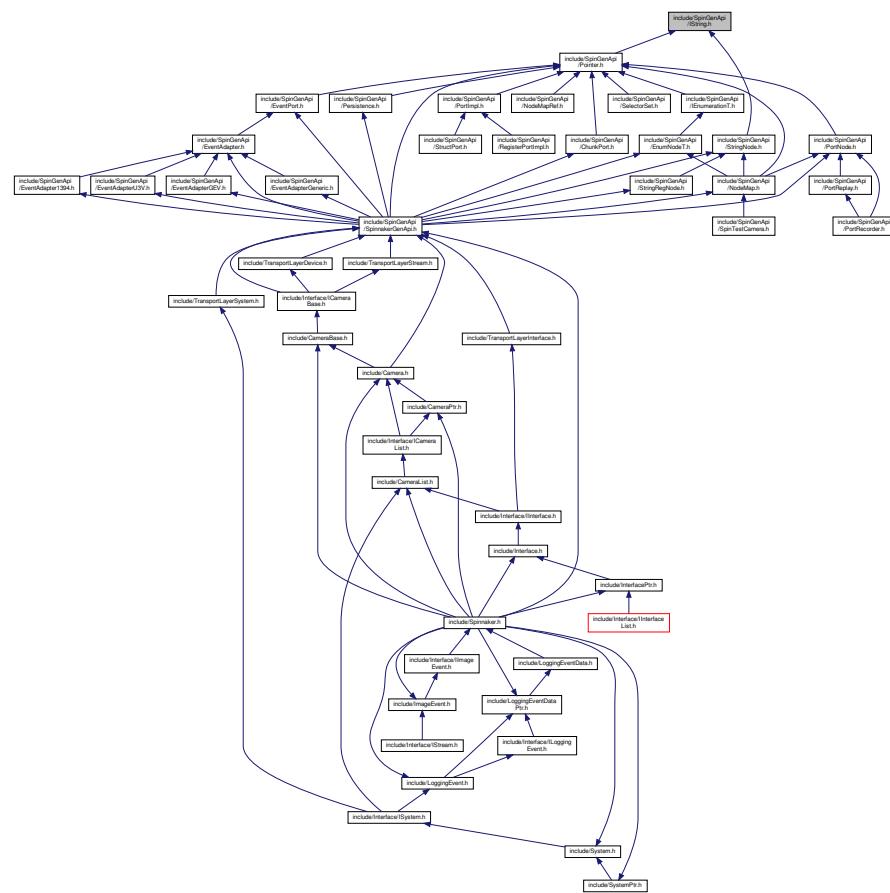
*Interface of a "digit" of the "counter" formed by the selector set.*

## 11.107 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

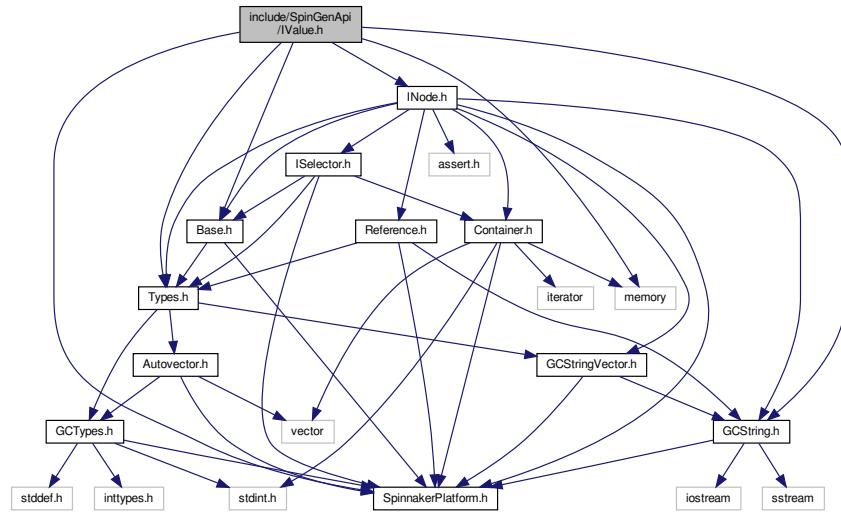
- virtual IEnumeration & `operator=` (const GenICam::gcstring &ValueStr)=0  
*Set string node value.*
- virtual bool `GetValue` (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool `operator()` () const  
*Get node value.*
- virtual GenICam::gcstring `operator*` ()=0  
*Get string node value.*
- virtual int64\_t `GetMaxLength` ()=0  
*Retrieves the maximum length of the string in bytes.*

## Variables

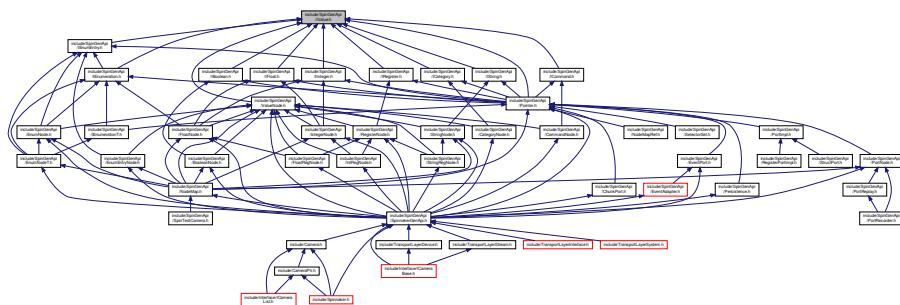
- interface SPINNAKER\_API\_ABSTRACT IString  
*Interface for string properties.*

## 11.108 include/SpinGenApi/IValue.h File Reference

Include dependency graph for IValue.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- Spinnaker
  - Spinnaker::GenApi

## 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.*

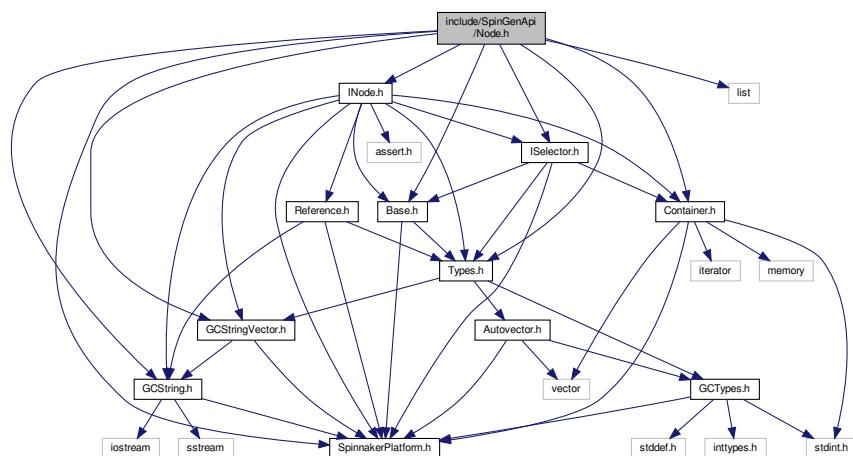
## Variables

- interface SPINNAKER\_API\_ABSTRACT IValue

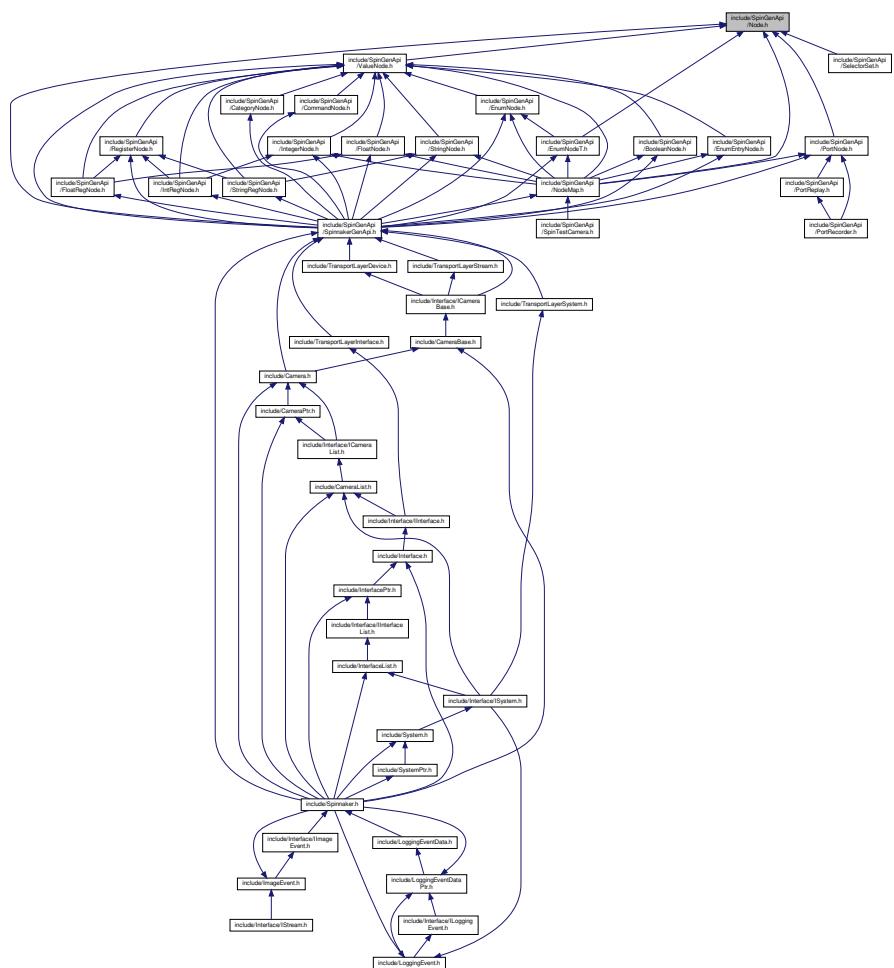
*Interface* for value properties.

## 11.109 include/SpinGenApi/Node.h File Reference

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

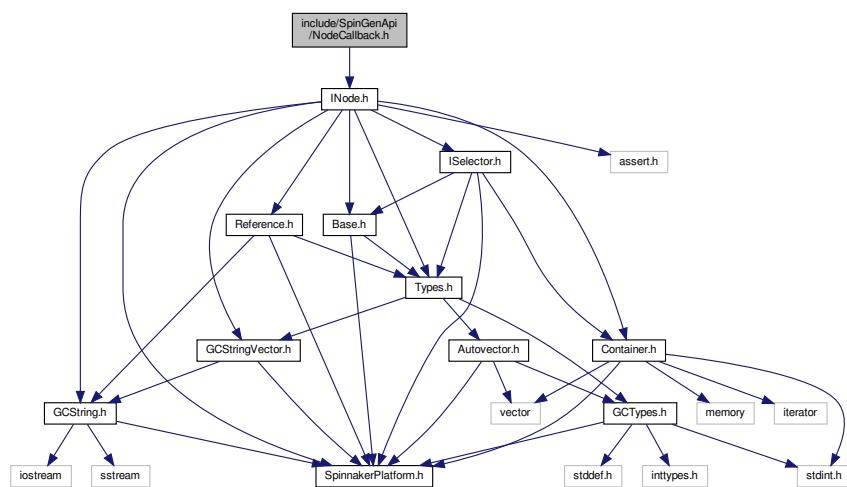
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## TypeDefs

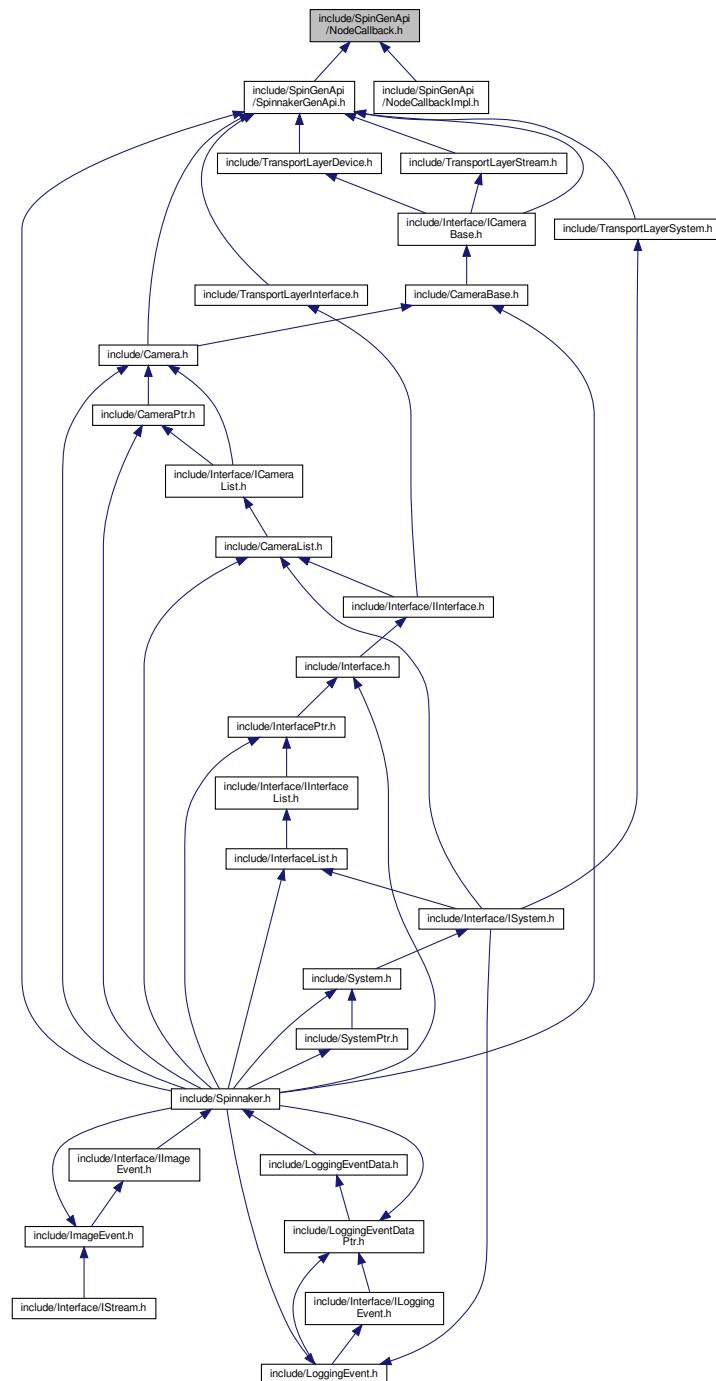
- [typedef Node CNodeRef](#)
- [typedef Node CSelectorRef](#)

## 11.110 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
 }

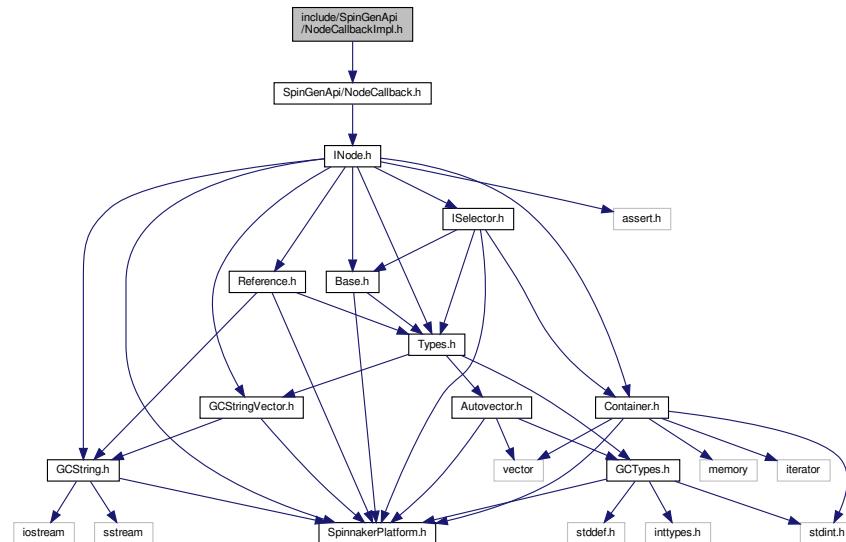
*the type of callback*

## 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.*

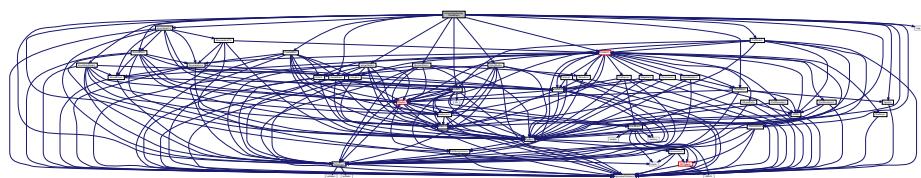
## 11.111 include/SpinGenApi/NodeCallbackImpl.h File Reference

Include dependency graph for NodeCallbackImpl.h:

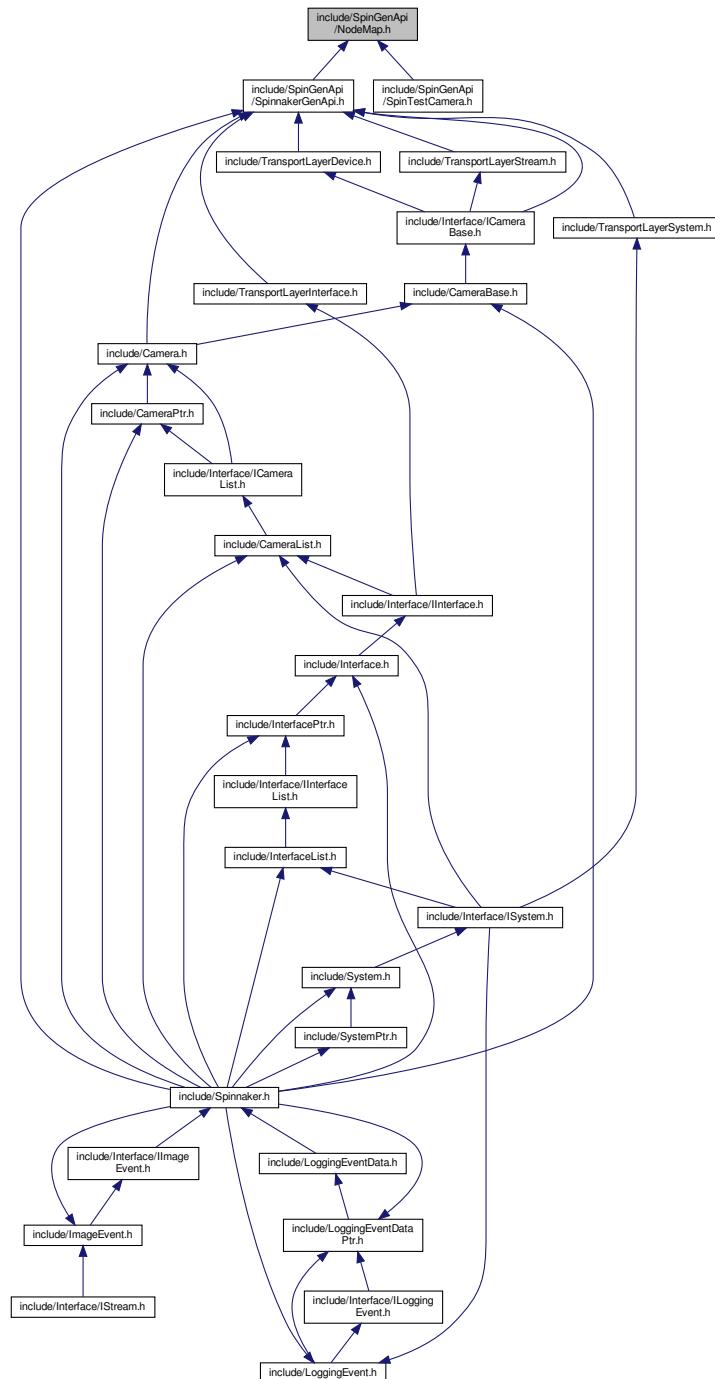


## 11.112 include/SpinGenApi/NodeMap.h File Reference

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

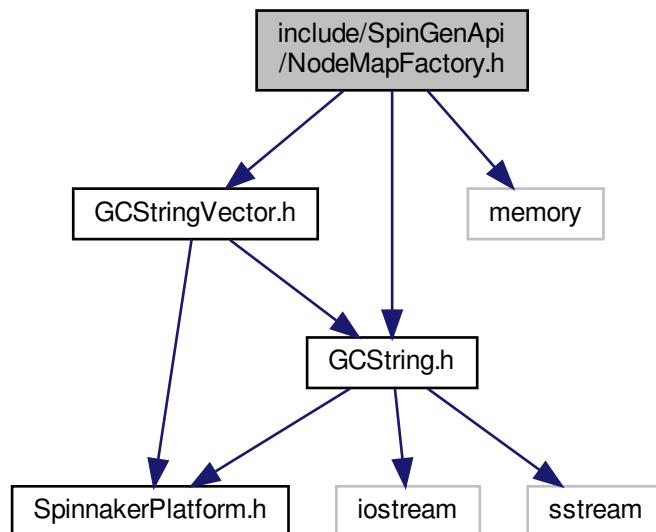
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

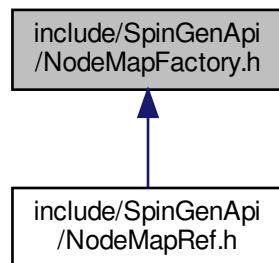
- [typedef NodeMap CNodeMapRef](#)

## 11.113 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

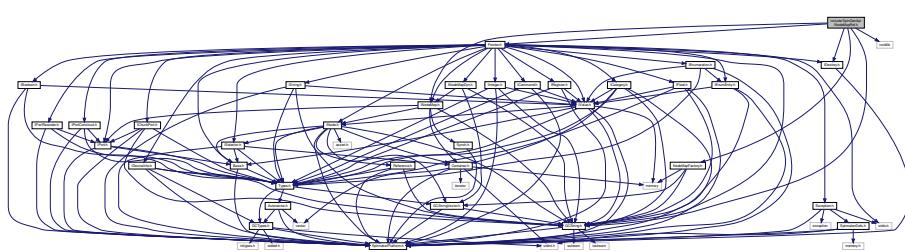
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 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.*

## 11.114 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

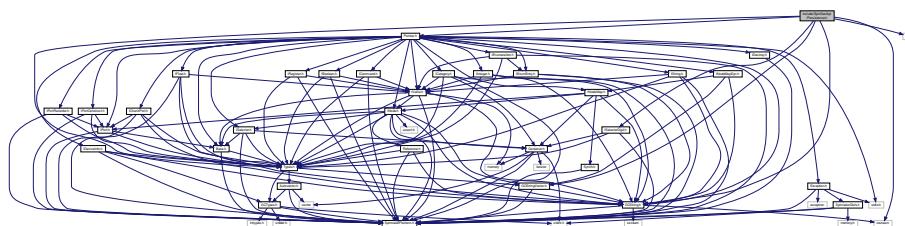
- `Spinnaker`
- `Spinnaker::GenApi`

## Functions

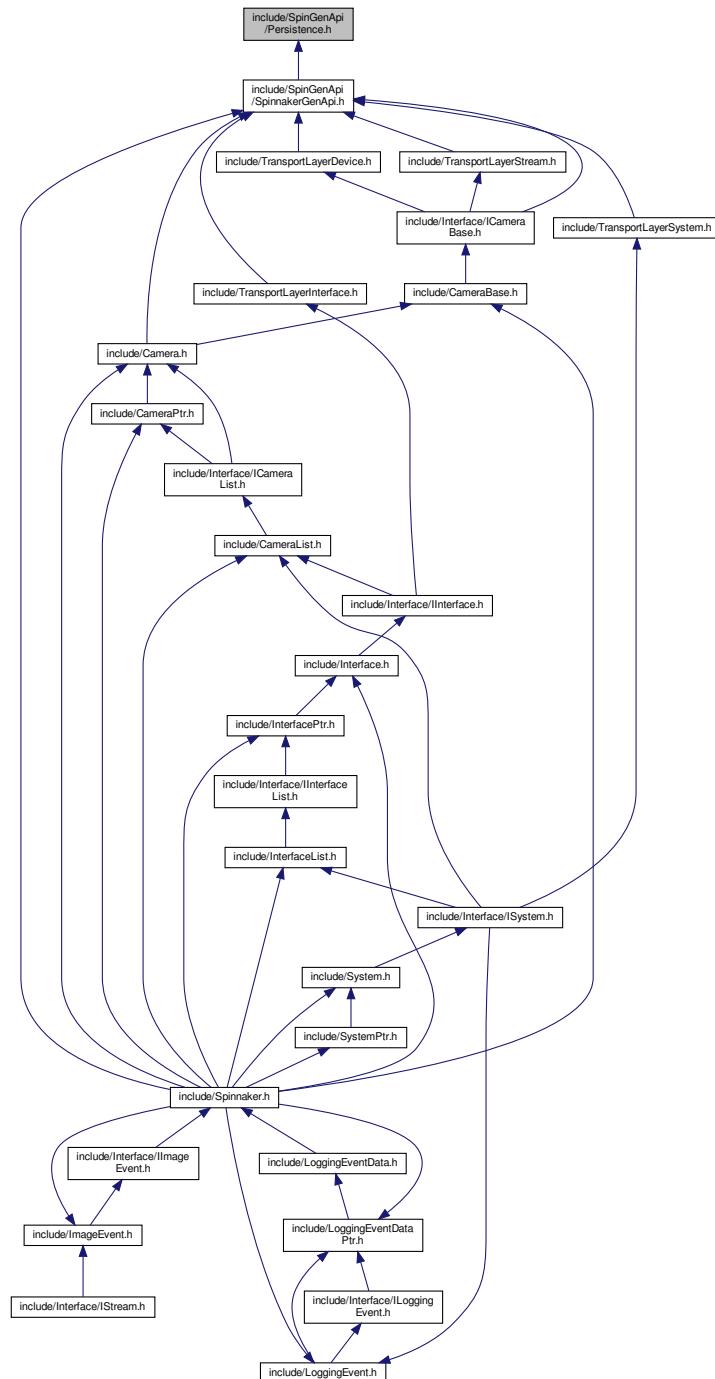
- `SPINNAKER_API IDestroy * CastToIDestroy (INodeMap *pNodeMap)`  
`makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)`
- template<class TCameraParams >  
`void _LoadXMLFromFile (const GenICam::gcstring &FileName)`
- template<class TCameraParams >  
`void _LoadXMLFromZIPFile (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 ()`

## 11.115 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

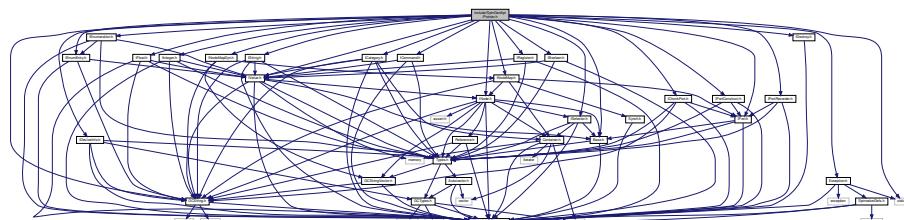
- virtual void [PersistFeature](#) (IValue &item)=0  
*Stores a feature.*
- [SPINNAKER\\_API](#) std::istream & [EatComments](#) (std::istream &is)  
*Helper function ignoring lines starting with comment character '#'.*
- [SPINNAKER\\_API](#) std::istream & [operator>>](#) (std::istream &is, CFeatureBag &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

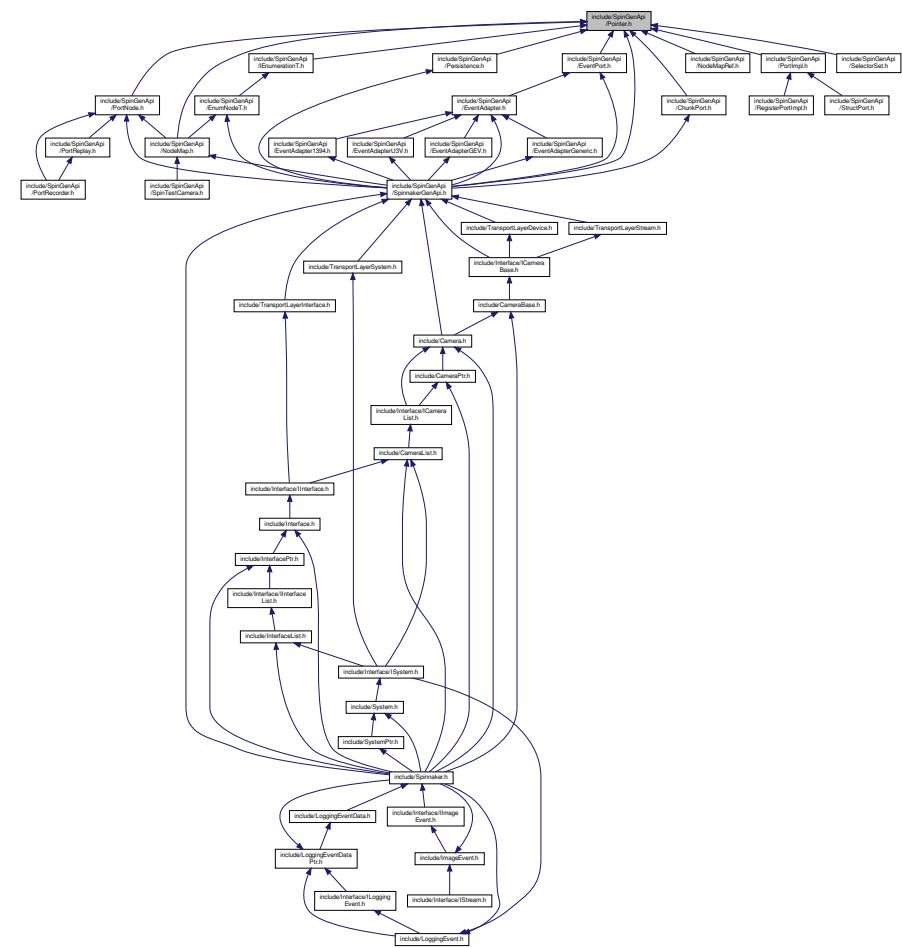
- interface [SPINNAKER\\_API\\_ABSTRACT IPersistScript](#)  
*Basic interface to persist values to.*

## 11.116 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.*

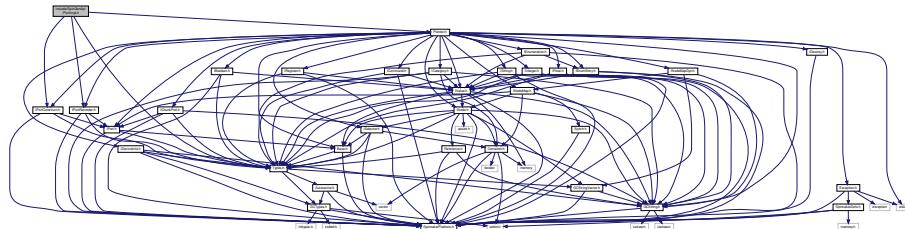
## Functions

- **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.*

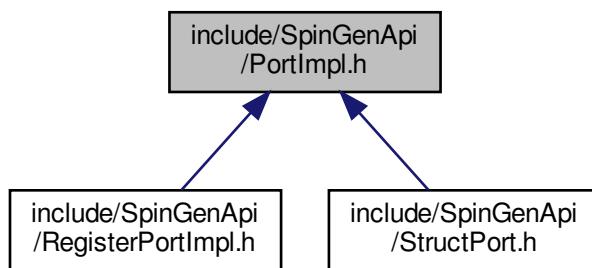
- template<class T , class B >  
bool [IsAvailable](#) (const Spinnaker::GenApi::CPointer< T, B > &ptr)  
*Checks if a node is Available.*
- GenICam::gcstring [GetInterfaceName](#) (IBase \*pBase)  
*Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.*

## 11.117 include/SpinGenApi/PortImpl.h File Reference

Include dependency graph for PortImpl.h:



This graph shows which files directly or indirectly include this file:



### Classes

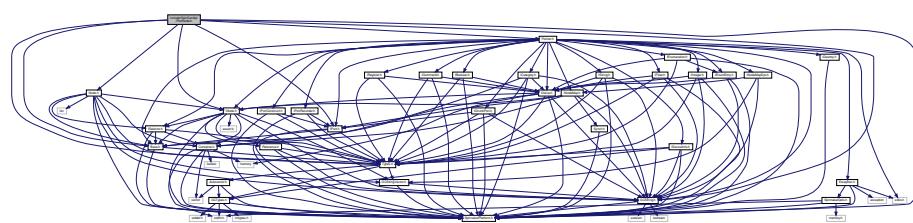
- class [CPortImpl](#)  
*Standard implementation for a port.*

### Namespaces

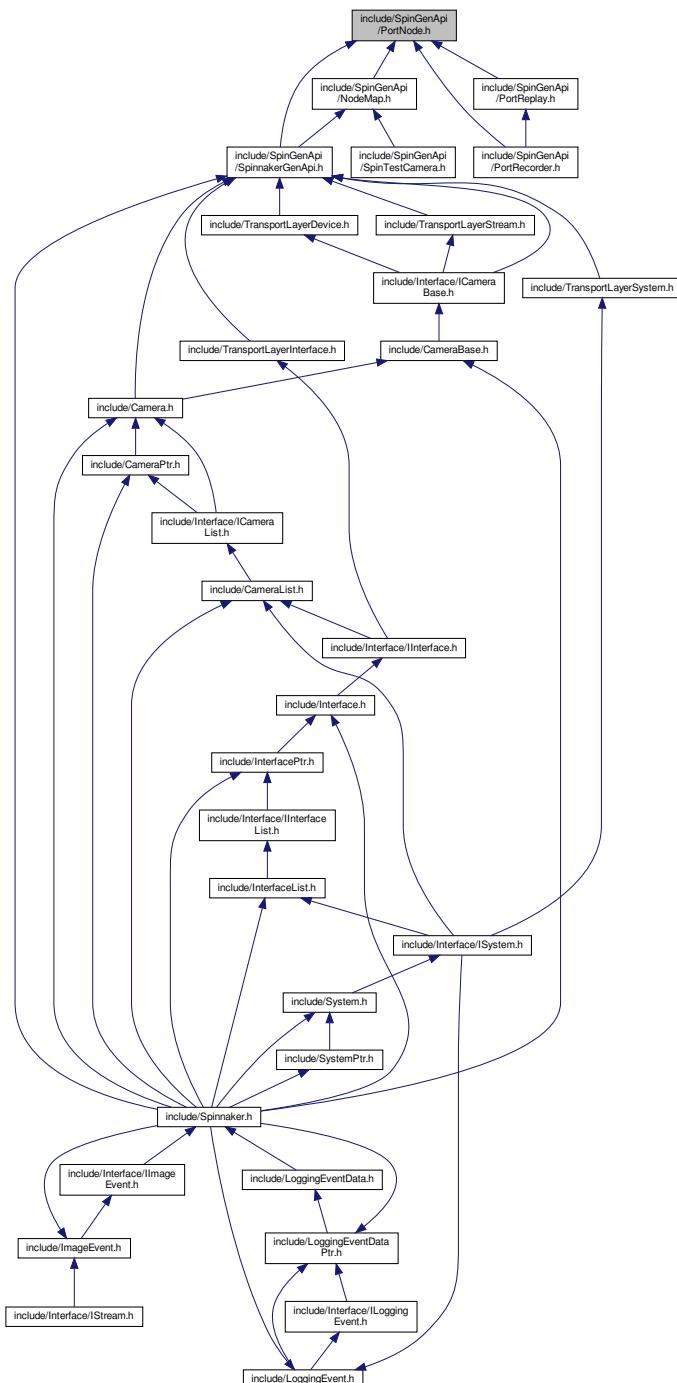
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.118 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

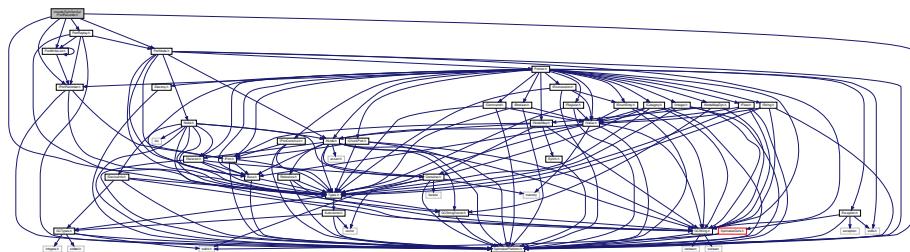
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- [typedef PortNode CPortRef](#)

## 11.119 include/SpinGenApi/PortRecorder.h File Reference

Include dependency graph for PortRecorder.h:



## Classes

- [class PortRecorder](#)  
*Interface for recording write commands on a port.*

## Namespaces

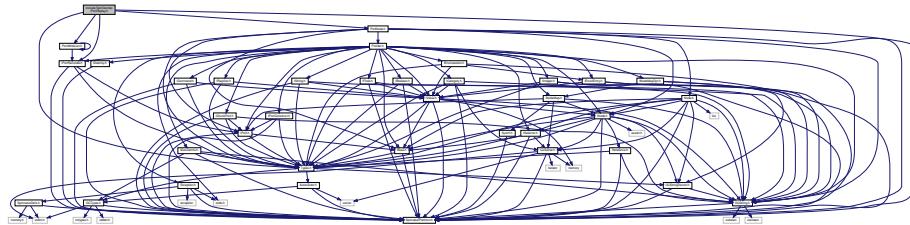
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

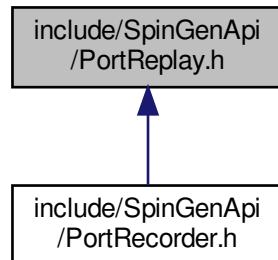
- [typedef PortRecorder CPortRecorderRef](#)  
*Reference to an IPortRecorder pointer.*

## 11.120 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](#)

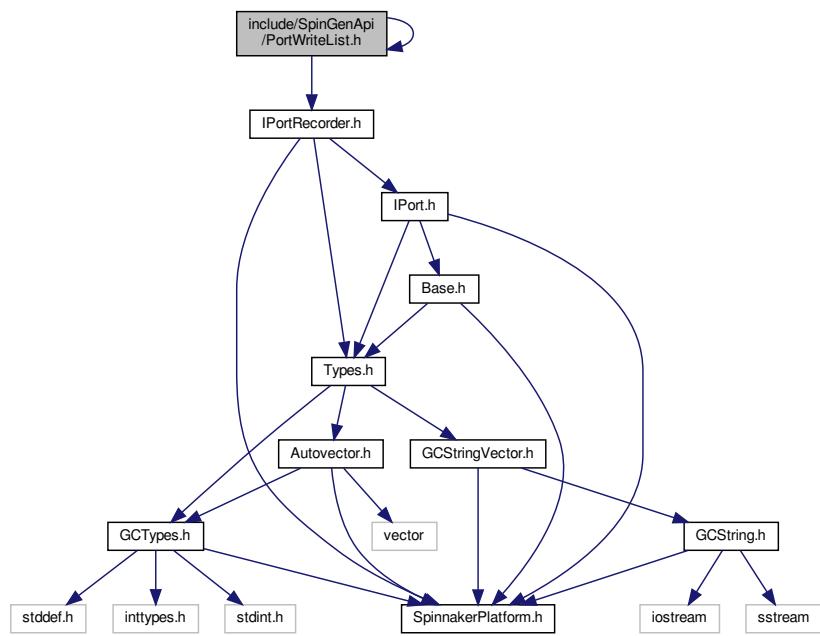
*Interface for replaying write commands on a port.*

### Namespaces

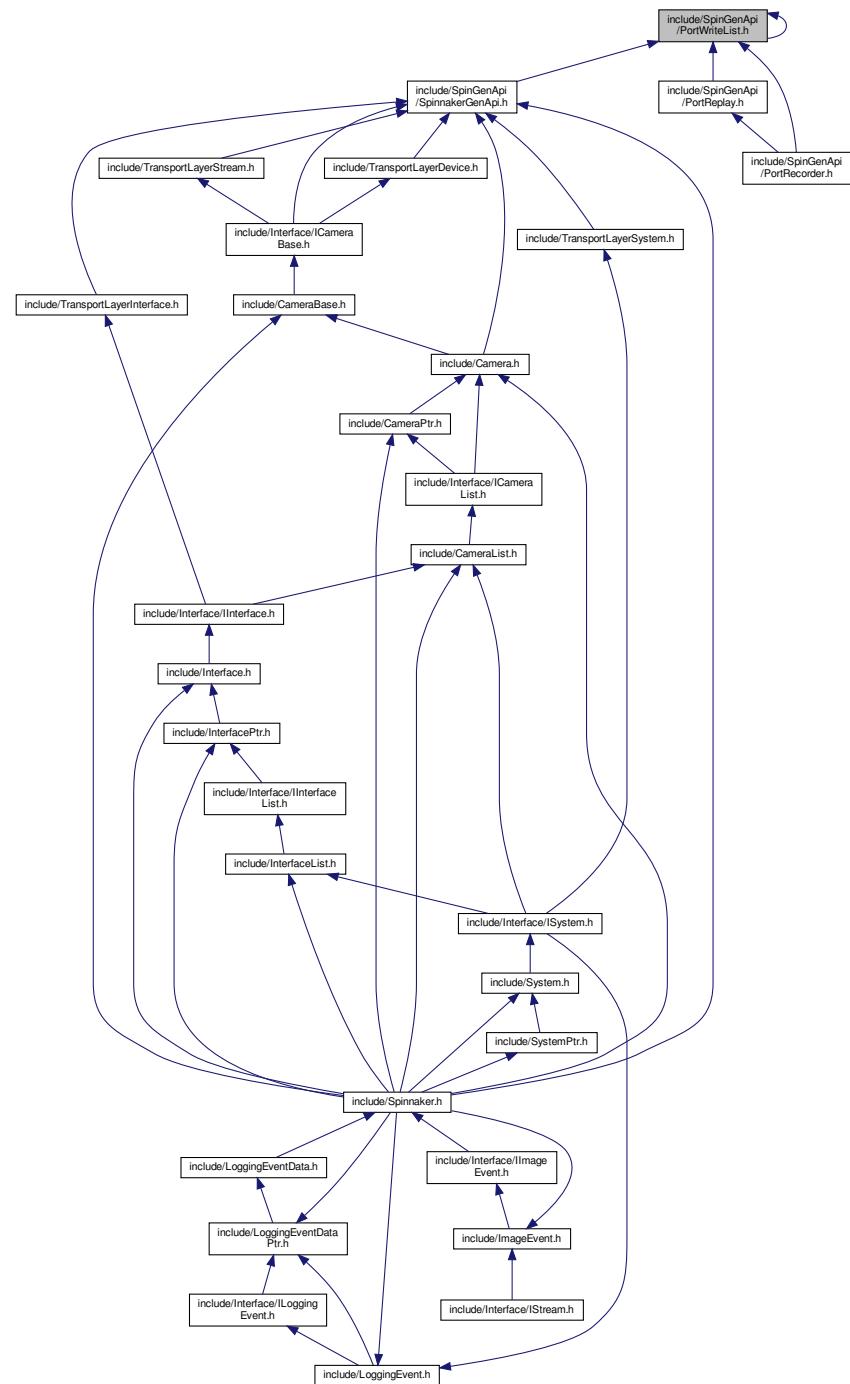
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.121 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](#)

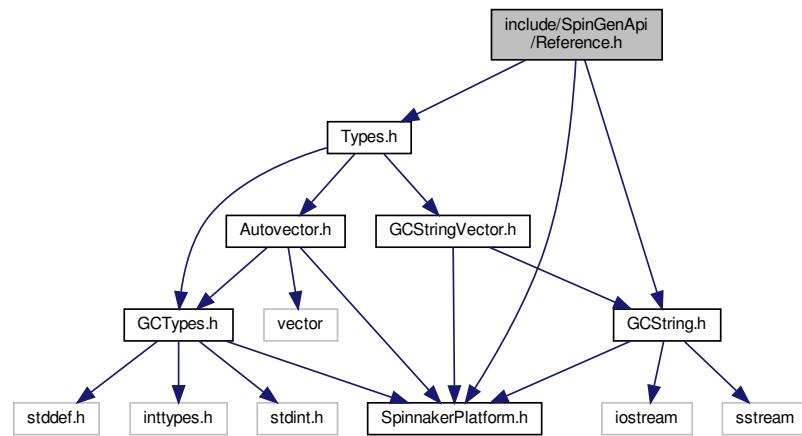
*Container holding a list of port write commands.*

# Namespaces

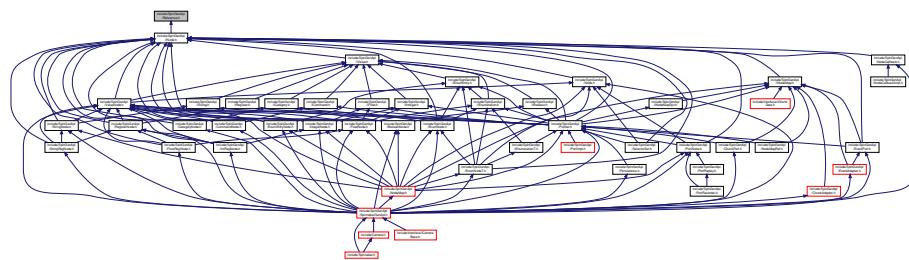
- Spinnaker
  - Spinnaker::GenApi

## 11.122 include/SpinGenApi/Reference.h File Reference

Include dependency graph for Reference.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

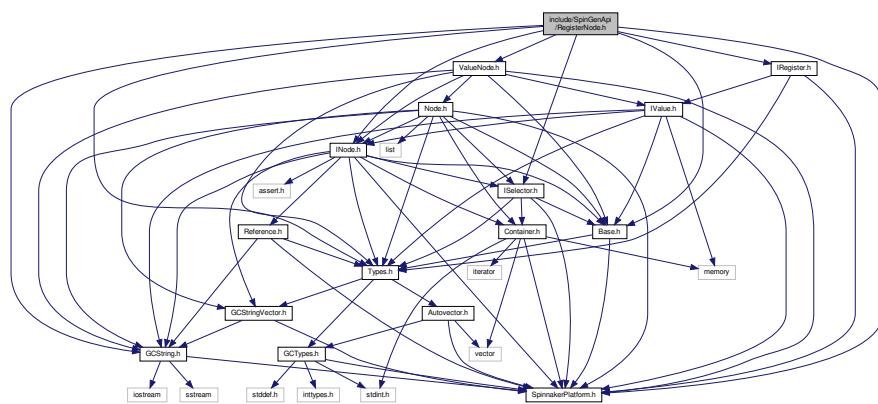
- Spinnaker
  - Spinnaker::GenApi

## Functions

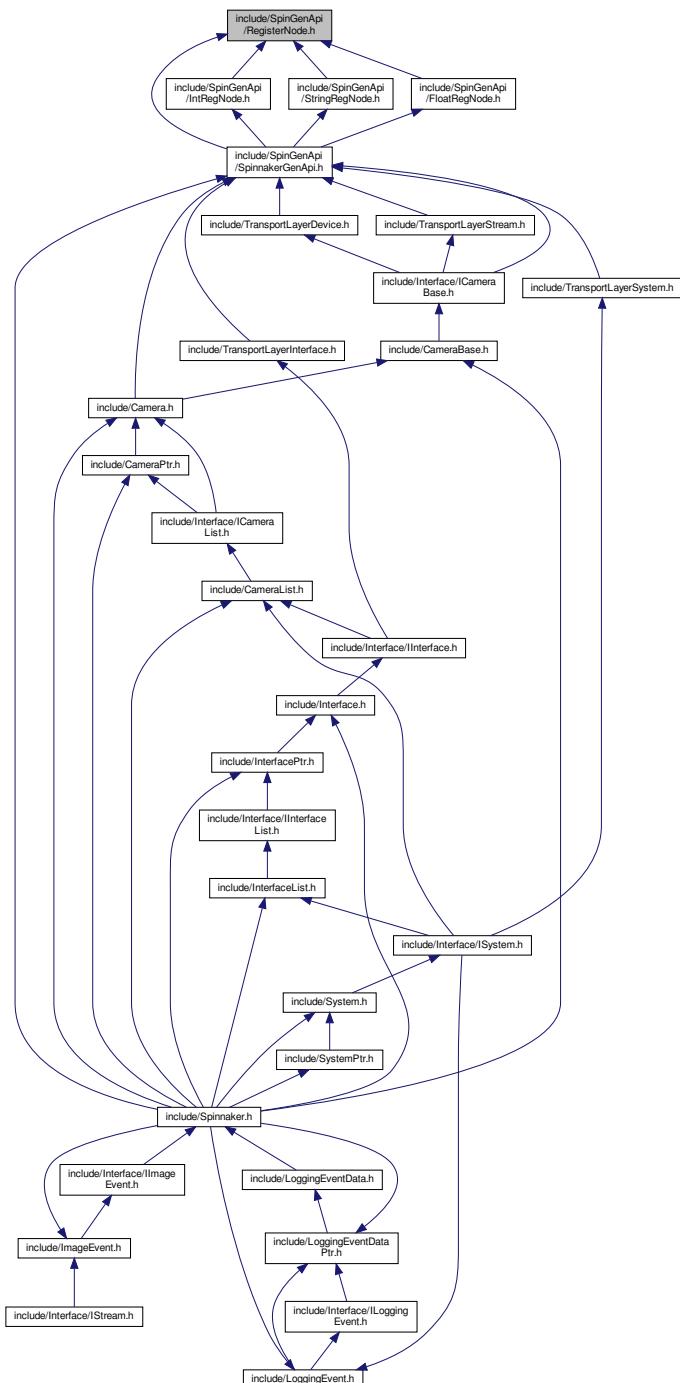
- virtual void **SetNumEnums** (int NumEnums)=0  
*sets the number of enum values*

## 11.123 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

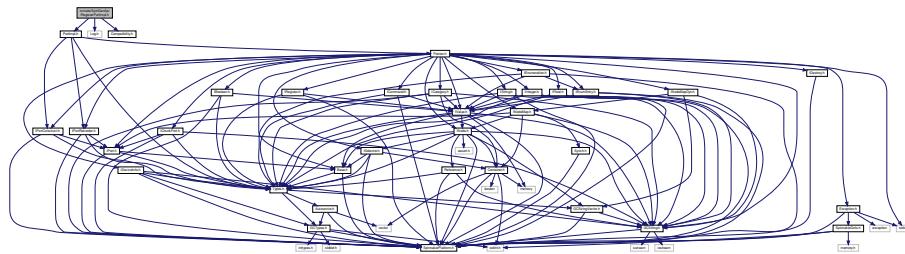
- Spinnaker
- Spinnaker::GenApi

## Typedefs

- typedef RegisterNode [CRegisterRef](#)

## 11.124 include/SpinGenApi/RegisterPortImpl.h File Reference

Include dependency graph for RegisterPortImpl.h:



## Classes

- class [CRegisterPortImpl](#)

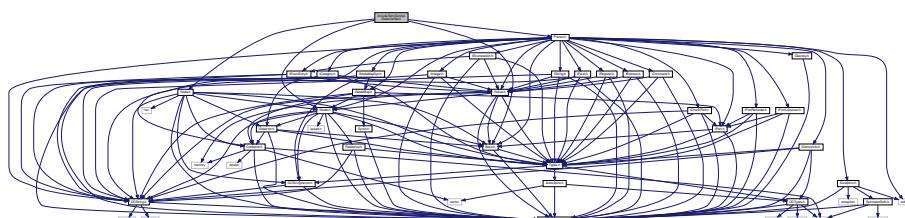
*Standard implementation for a port using a register based transport layer.*

## Namespaces

- Spinnaker
- Spinnaker::GenApi

## 11.125 include/SpinGenApi/SelectorSet.h File Reference

Include dependency graph for SelectorSet.h:



## Classes

- class [CSelectorSet](#)

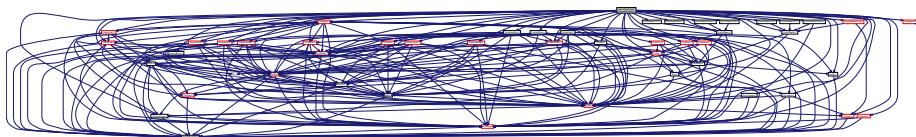
*The set of selectors selecting a given node.*

## Namespaces

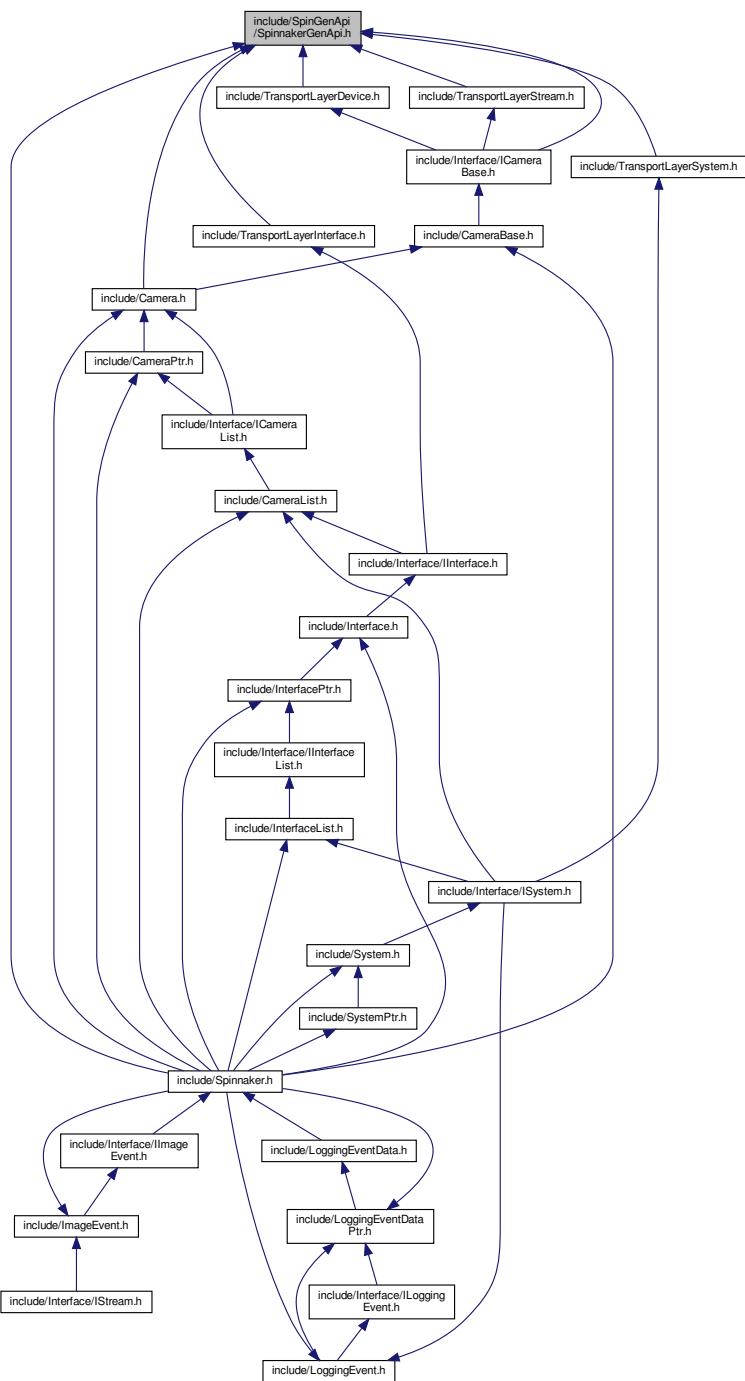
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.126 include/SpinGenApi/SpinnakerGenApi.h File Reference

Include dependency graph for SpinnakerGenApi.h:

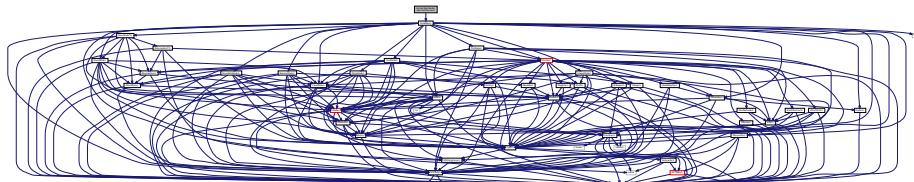


This graph shows which files directly or indirectly include this file:



## 11.127 include/SpinGenApi/SpinTestCamera.h File Reference

Include dependency graph for SpinTestCamera.h:



### Classes

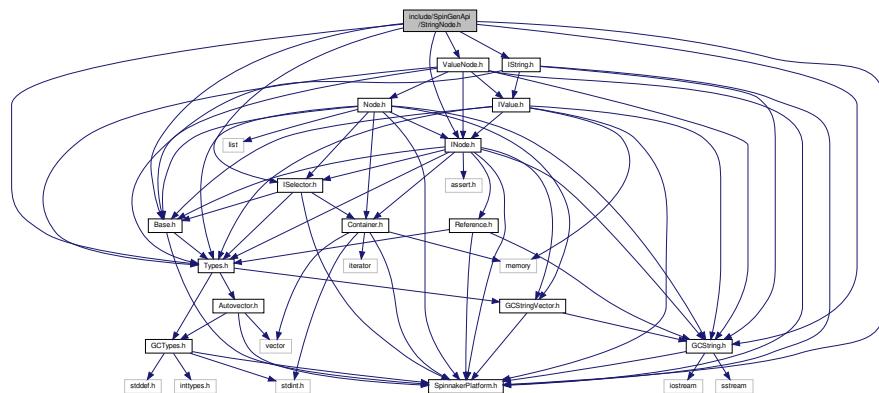
- class [SpinTestCamera](#)

### Namespaces

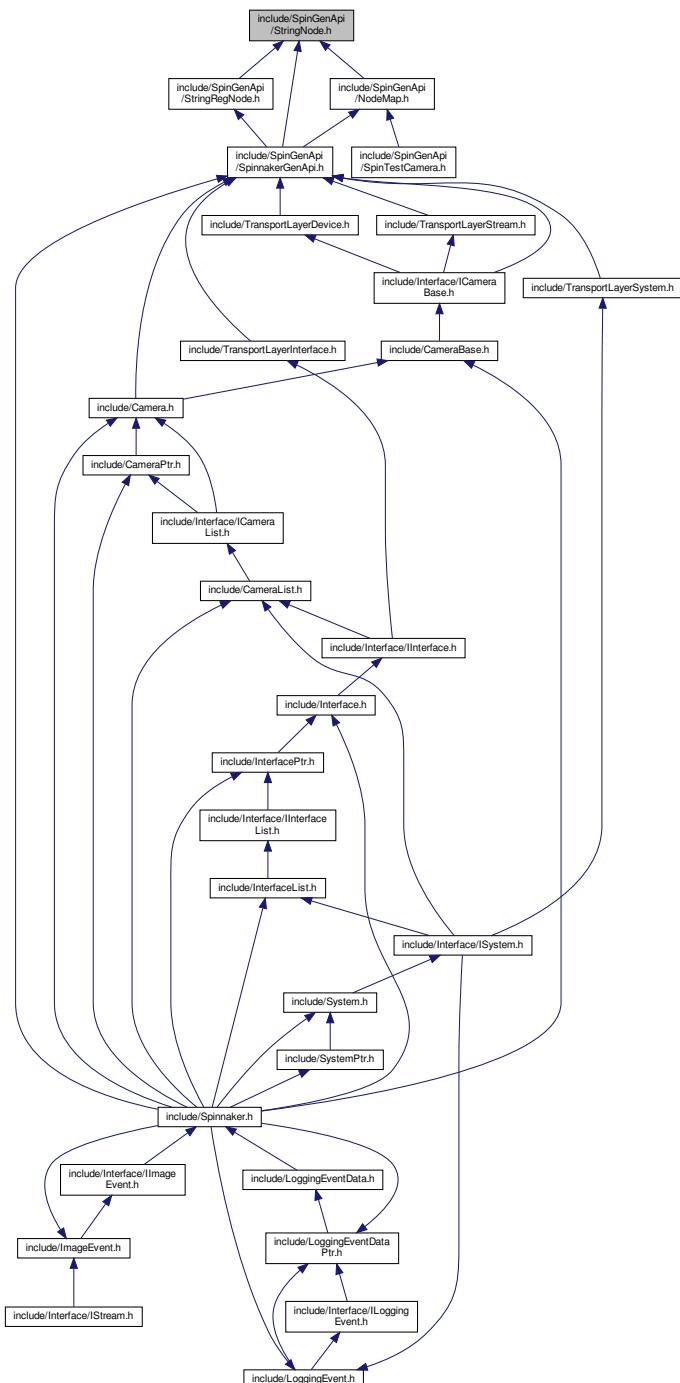
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.128 include/SpinGenApi/StringNode.h File Reference

Include dependency graph for StringNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StringNode](#)

*Interface for string properties.*

## Namespaces

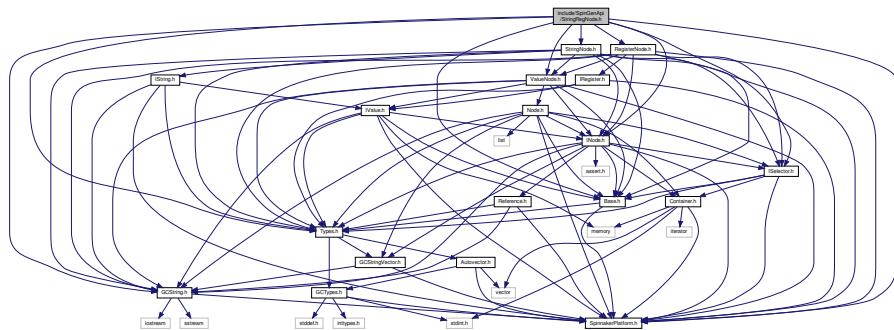
- Spinnaker
  - Spinnaker::GenApi

## TypeDefs

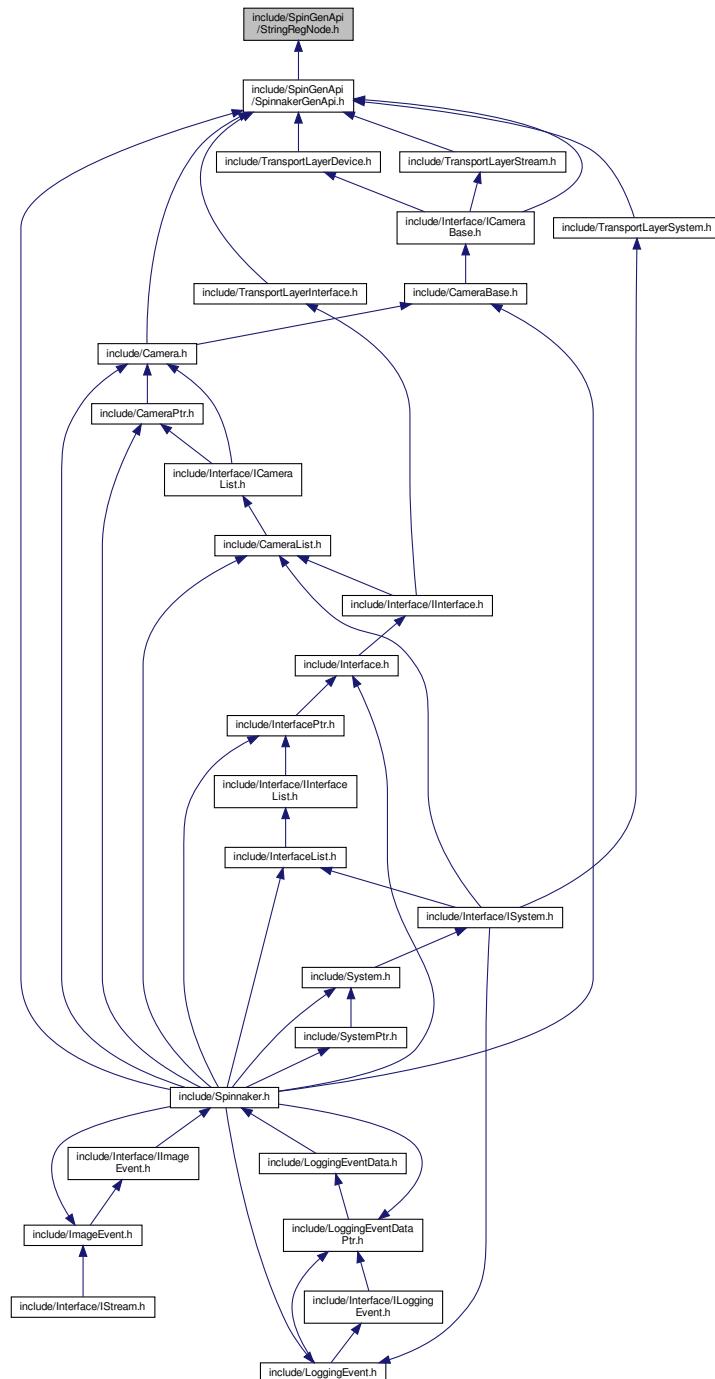
- `typedef StringNode CStringRef`

## 11.129 include/SpinGenApi/StringRegNode.h File Reference

Include dependency graph for StringRegNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StringRegNode](#)

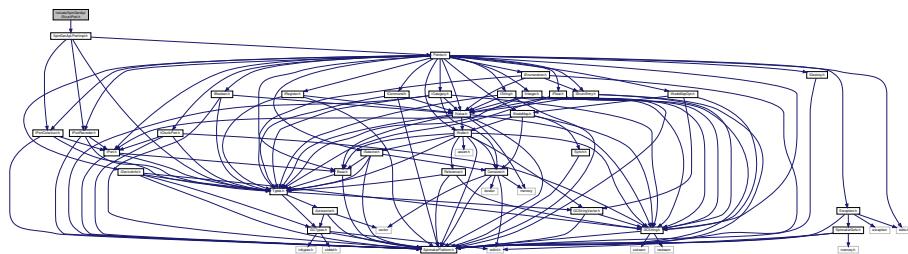
*Interface for string properties.*

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.130 include/SpinGenApi/StructPort.h File Reference

Include dependency graph for StructPort.h:



## Classes

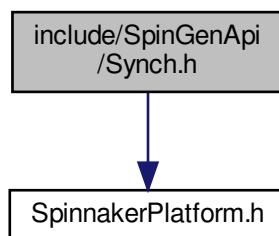
- class [CTestPortStruct< CDataStruct >](#)  
*Implements a register spaces based on a C++ struct.*

## Namespaces

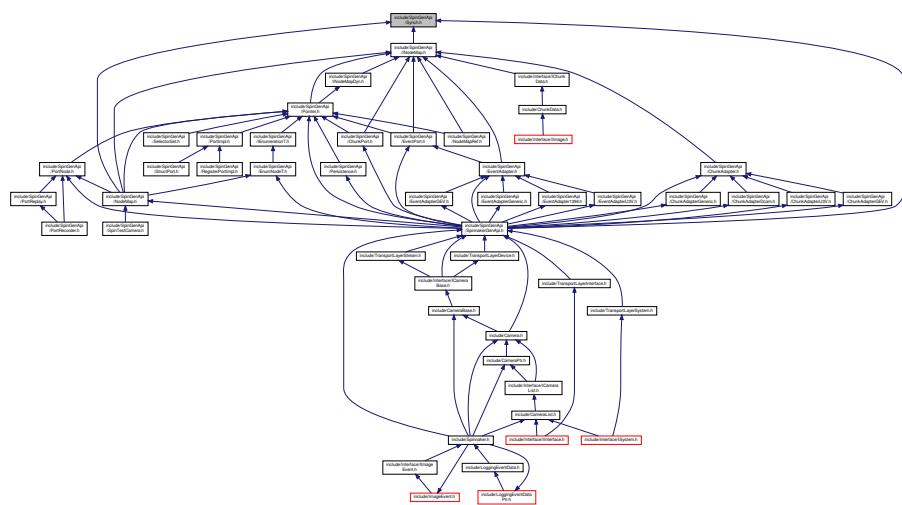
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.131 include/SpinGenApi/Synch.h File Reference

Include dependency graph for Synch.h:



This graph shows which files directly or indirectly include this file:



## Classes

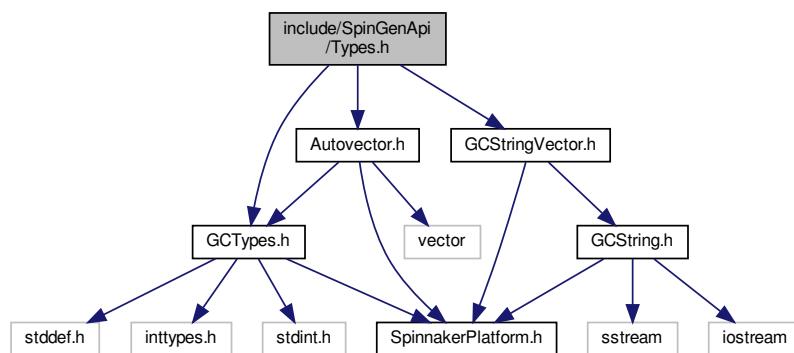
- class [Clock](#)  
*A lock class.*
- class [ClockEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)

## Namespaces

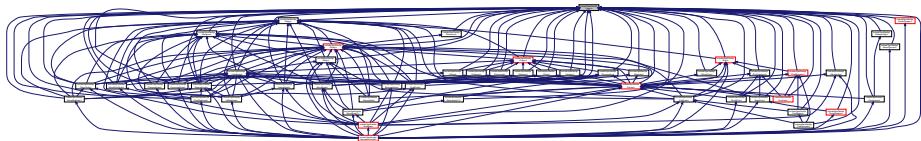
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.132 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 }`

- enum **ERepresentation** {  
  Linear,  
  Logarithmic,  
  Boolean,  
  PureNumber,  
  HexNumber,  
  IPV4Address,  
  MACAddress,  
  \_UndefinedRepresentation }
- caching mode of a register*
- enum **EEndianess** {  
  BigEndian,  
  LittleEndian,  
  \_UndefinedEndian }
- recommended representation of a node value*
- enum **ENameSpace** {  
  Custom,  
  Standard,  
  \_UndefinedNameSpace }
- Endianness of a value in a register.*
- enum **EStandardNameSpace** {  
  None,  
  GEV,  
  I IDC,  
  CL,  
  USB,  
  \_UndefinedStandardNameSpace }
- Defines if a node name is standard or custom.*
- enum **EYesNo** {  
  Yes = 1,  
  No = 0,  
  \_UndefinedYesNo = 2 }
- Defines from which standard namespace a node name comes from.*
- enum **ESlope** {  
  Increasing,  
  Decreasing,  
  Varying,  
  Automatic,  
  \_UndefinedESlope }
- Defines the choices of a Yes/No alternative.*
- enum **EXMLValidation** {  
  xvLoad = 0x00000001L,  
  xvCycles = 0x00000002L,  
  xvSFNC = 0x00000004L,  
  xvDefault = 0x00000000L,  
  xvAll = 0xffffffffL,  
  \_UndefinedEXMLValidation = 0x8000000L }
- typedef for formula type*
- 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`,  
  `intfIInteger`,  
  `intfIBoolean`,  
  `intfICommand`,  
  `intfIFloat`,  
  `intfIString`,  
  `intfIRegister`,  
  `intfICategory`,  
  `intfIEnumerator`,  
  `intfIEnumEntry`,  
  `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.*

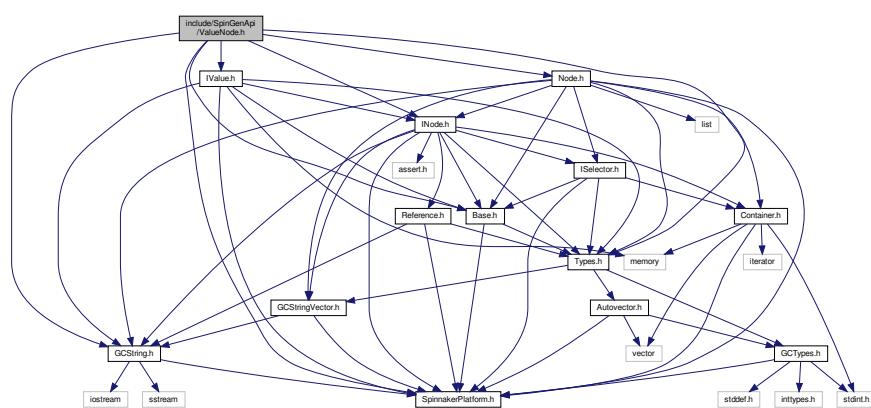
## 11.132.1 Macro Definition Documentation

### 11.132.1.1 interface

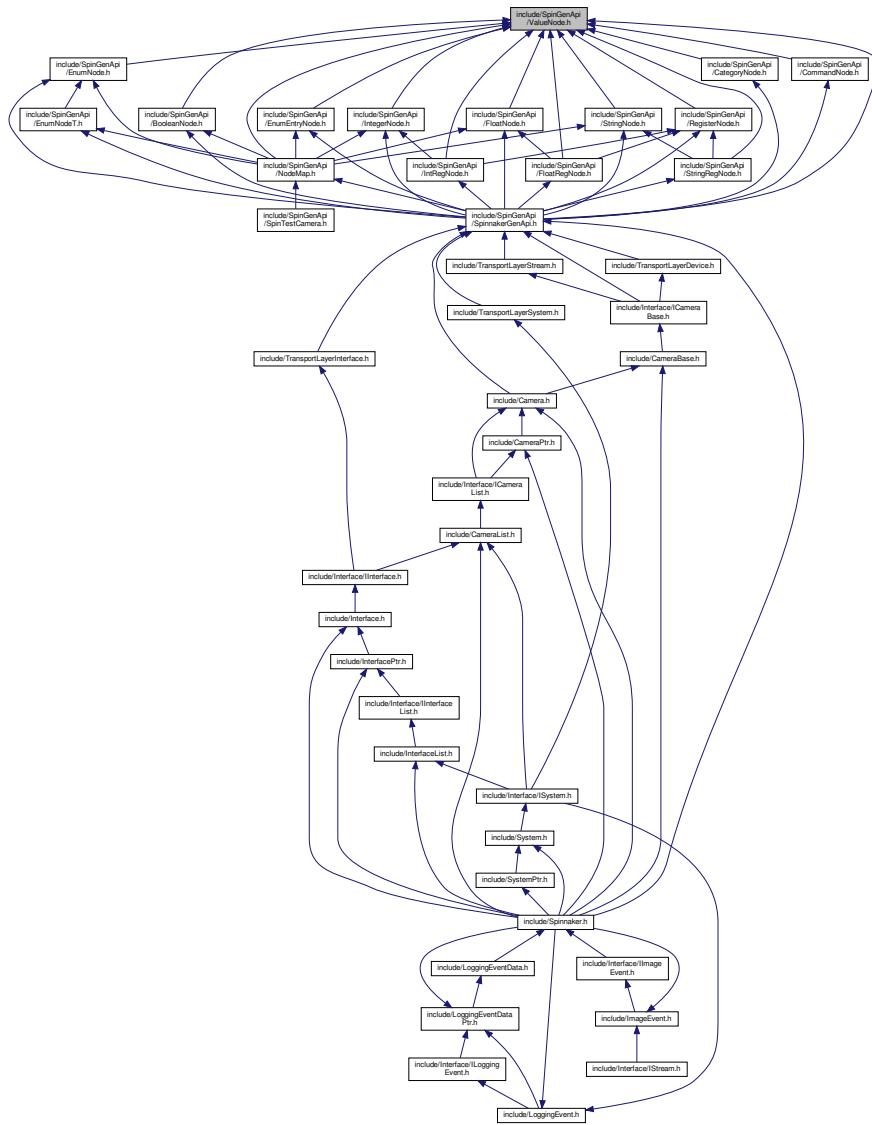
```
#define interface struct
```

## 11.133 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

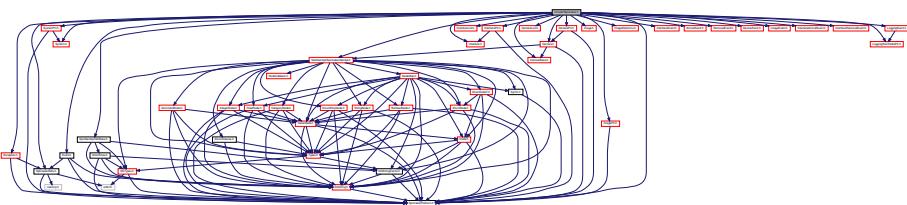
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## TypeDefs

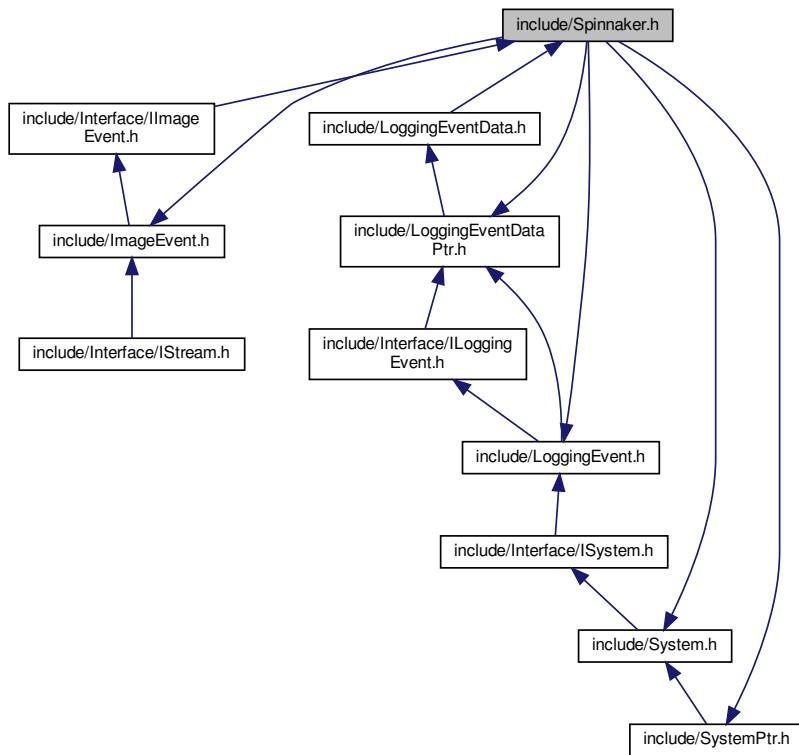
- typedef ValueNode [CValueRef](#)

## 11.134 include/Spinnaker.h File Reference

Include dependency graph for Spinnaker.h:

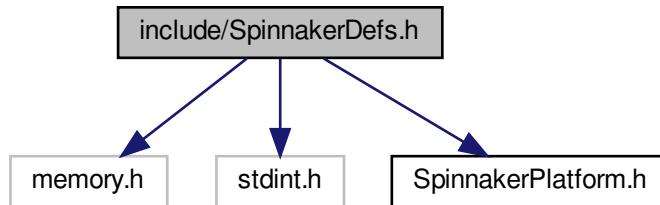


This graph shows which files directly or indirectly include this file:

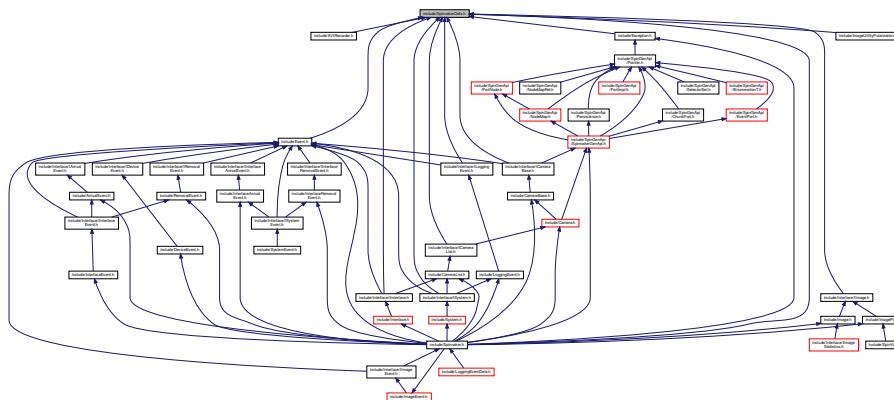


## 11.135 include/SpinnakerDefs.h File Reference

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.*

## 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 = 0xFFFFFFFF }

*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 = 3,
 IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT = 4,
 IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT = 5,
 IMAGE\_PACKETID\_INCONSISTENT = 6,
 IMAGE\_MISSING\_LEADER = 7,
 IMAGE\_MISSING\_TRAILER = 8,
 IMAGE\_DATA\_INCOMPLETE = 9,
 IMAGE\_INFO\_INCONSISTENT = 10,
 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 PayloadTypeInfolDs {  
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 }
- 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 }

## Functions

- **DEPRECATED\_ENUM** ("This enum has been deprecated. Polarization images are now created through specific " "functions in the ImageUtilityPolarization class.") `PolarizationAlgorithm`
- **DEPRECATED\_ENUM** ("This enum has been deprecated. Image scaling can now be applied through specific functions " "defined in the ImageUtility class.") `PolarizationResolution`

## Variables

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

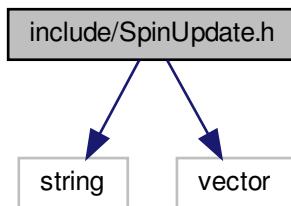
## 11.136 include/SpinnakerPlatform.h File Reference

### Macros

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

## 11.137 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 (UpdatorMessageCallback messageCallbackFunction)`
- `SPINUPDATE_API void SetProgressCallback (UpdatorProgressCallback 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)`

### 11.137.1 Macro Definition Documentation

#### 11.137.1.1 SPINUPDATE\_API

```
#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT
```

### 11.137.2 Function Documentation

#### 11.137.2.1 GetErrorMessage()

```
SPINUPDATE_API const char* GetErrorMessage ()
```

#### 11.137.2.2 SetMessageCallback()

```
SPINUPDATE_API void SetMessageCallback (UpdatorMessageCallback messageCallbackFunction)
```

#### 11.137.2.3 SetProgressCallback()

```
SPINUPDATE_API void SetProgressCallback (UpdatorProgressCallback progressCallbackFunction)
```

#### 11.137.2.4 UpdateFirmware()

```
SPINUPDATE_API int UpdateFirmware (
 std::vector< std::string > args)
```

#### 11.137.2.5 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 |

##### Returns

0 for success, otherwise non zero for failures.

Typical usage for updating is as follows: -R{serial number} [-{options} ..] {firmware zim file} -R{serial number} -UU -B {firmware zim file}

Option definitions: -B = Reboots the camera after the update has completed. If this argument is not provided, a manual power cycle will be required. -A = Updates individual portions of the firmware in flash. The code section of camera at location 0xFF08000 will be updated. ./sample\_app -AFF080000 camera.zim -U = Downgrade the firmware. Multiple U's can be used to overwrite the ROM header. -F = Force program and EEprom reload. -R = Enter a regular expression for camera serial match. For example: ./sample\_app -R.\* camera.zim Results in matching any camera serial -P = Checks the progress of the updater. -epromsave = Save the content of the EEprom to a file.

#### 11.137.2.6 UpdateFirmwareGUI()

```
SPINUPDATE_API int UpdateFirmwareGUI (
 std::string args)
```

### 11.137.3 Variable Documentation

#### 11.137.3.1 UpdatorMessageCallback

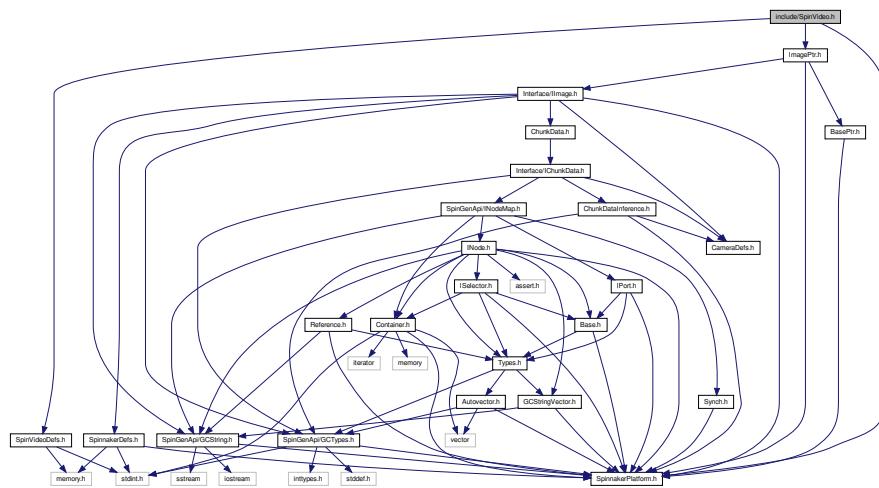
```
SPINUPDATE_API typedef int(* UpdatorMessageCallback) (const char *message)
```

### 11.137.3.2 UpdatorProgressCallback

```
SPINUPDATE_API typedef int(* UpdatorProgressCallback) (const char *action, unsigned int address,
int globalPercent, int currPercent)
```

## 11.138 include/SpinVideo.h File Reference

Include dependency graph for SpinVideo.h:



## Classes

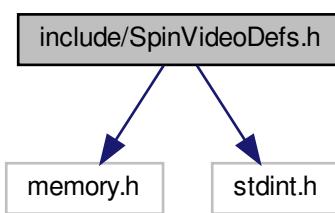
- class [SpinVideo](#)  
*Provides the functionality for the user to record images to an AVI/MP4 file.*

## Namespaces

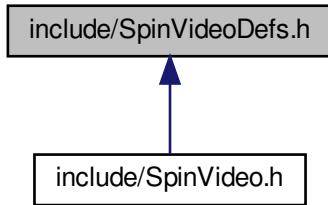
- [Spinnaker](#)
- [Spinnaker::Video](#)

## 11.139 include/SpinVideoDefs.h File Reference

Include dependency graph for SpinVideoDefs.h:



This graph shows which files directly or indirectly include this file:



## Classes

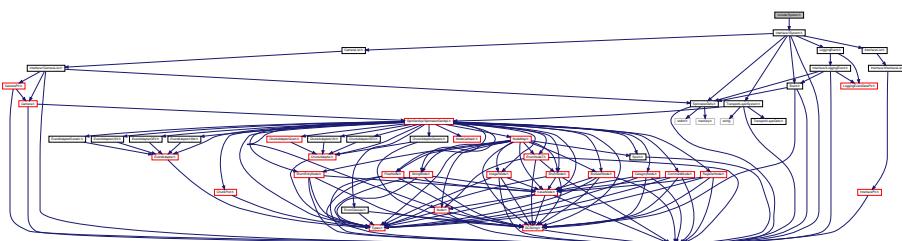
- struct [MJPEGOption](#)  
*Options for saving MJPG files.*
- struct [H264Option](#)  
*Options for saving H264 files.*
- struct [AVIOption](#)  
*Options for saving AVI files.*

## Namespaces

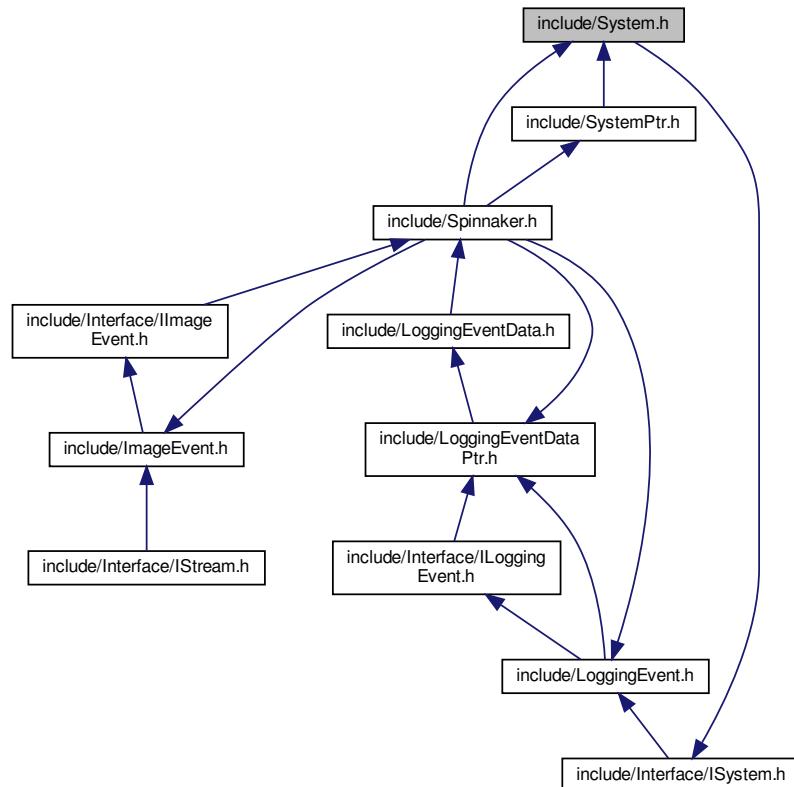
- [Spinnaker](#)
- [Spinnaker::Video](#)

## 11.140 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 1
  - #define FLIR\_SPINNAKER\_VERSION\_MINOR 27
  - #define FLIR\_SPINNAKER\_VERSION\_TYPE 0
  - #define FLIR\_SPINNAKER\_VERSION\_BUILD 48

## 11.140.1 Macro Definition Documentation

### 11.140.1.1 FLIR\_SPINNAKER\_VERSION\_BUILD

```
#define FLIR_SPINNAKER_VERSION_BUILD 48
```

### 11.140.1.2 FLIR\_SPINNAKER\_VERSION\_MAJOR

```
#define FLIR_SPINNAKER_VERSION_MAJOR 1
```

### 11.140.1.3 FLIR\_SPINNAKER\_VERSION\_MINOR

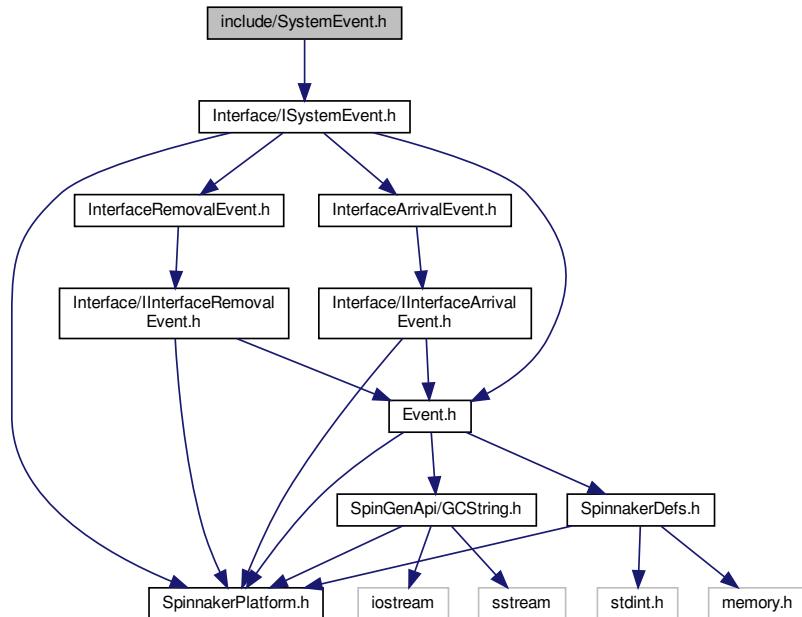
```
#define FLIR_SPINNAKER_VERSION_MINOR 27
```

### 11.140.1.4 FLIR\_SPINNAKER\_VERSION\_TYPE

```
#define FLIR_SPINNAKER_VERSION_TYPE 0
```

## 11.141 include/SystemEvent.h File Reference

Include dependency graph for SystemEvent.h:



## Classes

- class [SystemEvent](#)

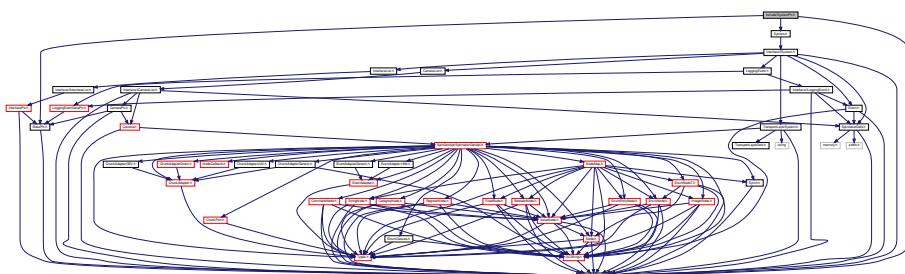
*A handler to interface arrival and removal events on the system.*

## Namespaces

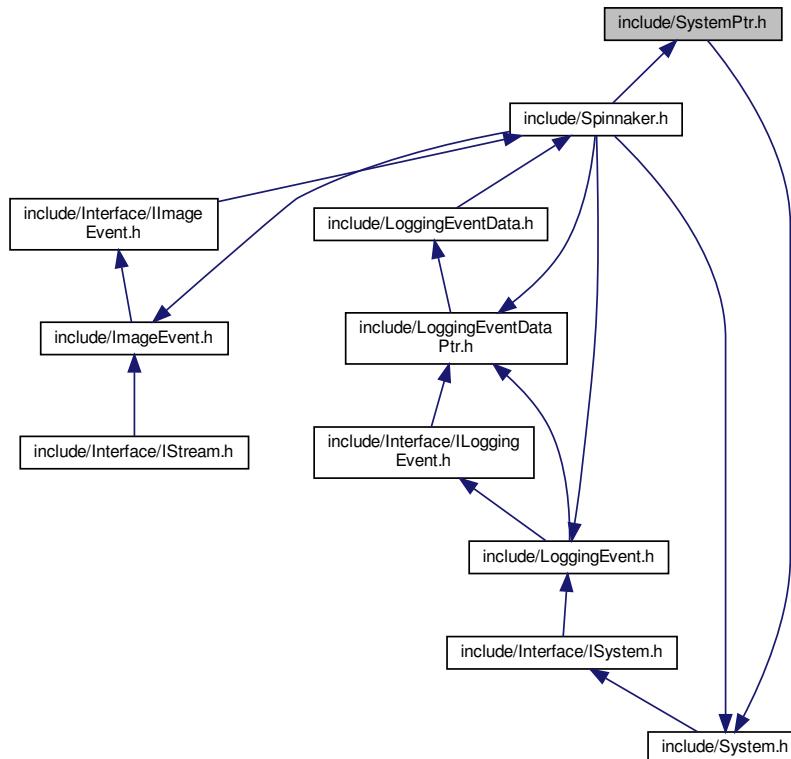
- [Spinnaker](#)

## 11.142 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](#)

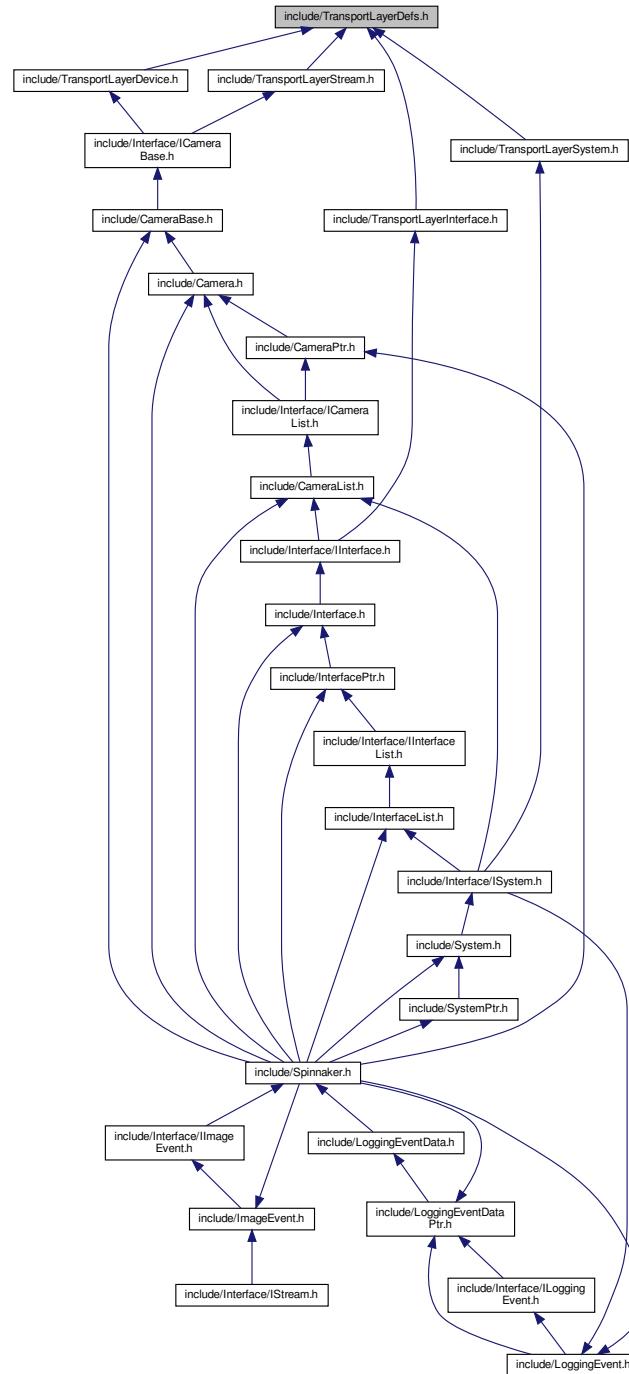
*A reference tracked pointer to a system object.*

## Namespaces

- [Spinnaker](#)

## 11.143 include/TransportLayerDefs.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- Spinnaker

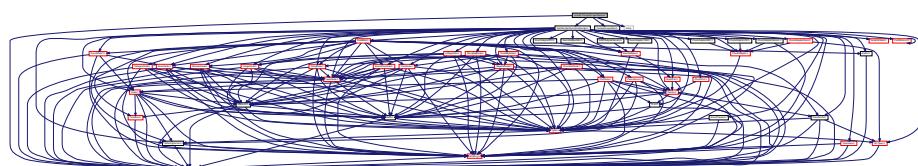
## Enumerations

- enum StreamTypeEnum {  
StreamType\_Mixed,  
StreamType\_Custom,  
StreamType\_GEV,  
StreamType\_CL,  
StreamType\_IIDC,  
StreamType\_UVC,  
StreamType\_CXP,  
StreamType\_CLHS,  
StreamType\_U3V,  
StreamType\_ETHERNET,  
StreamType\_PCI,  
NUMSTREAMTYPE }
- The enum definitions for TL Device nodes from the transport layer .xml files.
- enum StreamDefaultBufferCountModeEnum {  
StreamDefaultBufferCountMode\_Manual,  
StreamDefaultBufferCountMode\_Auto,  
NUMSTREAMDEFAULTBUFFERCOUNTMODE }
- enum StreamBufferCountModeEnum {  
StreamBufferCountMode\_Manual,  
StreamBufferCountMode\_Auto,  
NUMSTREAMBUFFERCOUNTMODE }
- enum StreamBufferHandlingModeEnum {  
StreamBufferHandlingMode\_OldestFirst,  
StreamBufferHandlingMode\_OldestFirstOverwrite,  
StreamBufferHandlingMode\_NewestFirst,  
StreamBufferHandlingMode\_NewestFirstOverwrite,  
StreamBufferHandlingMode\_NewestOnly,  
NUMSTREAMBUFFERHANDLINGMODE }
- enum DeviceTypeEnum {  
DeviceType\_Mixed,  
DeviceType\_Custom,  
DeviceType\_GEV,  
DeviceType\_CL,  
DeviceType\_IIDC,  
DeviceType\_UVC,  
DeviceType\_CXP,  
DeviceType\_CLHS,  
DeviceType\_U3V,  
DeviceType\_ETHERNET,  
DeviceType\_PCI,  
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 }

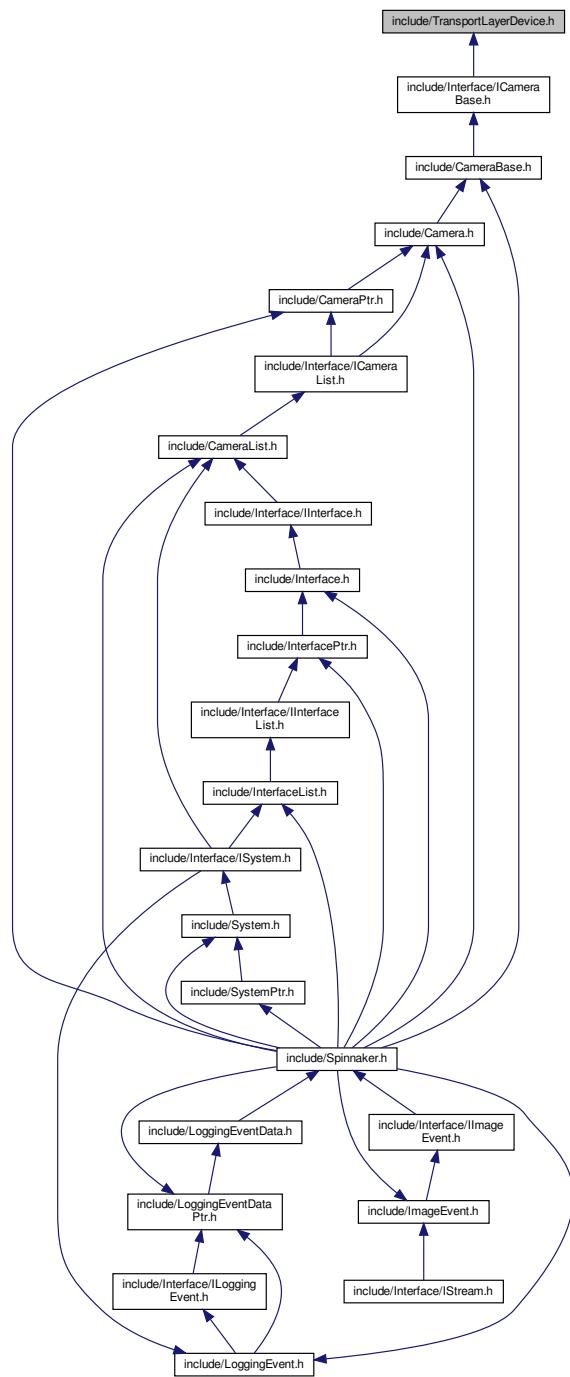
- enum `GUIXMLLocationEnum` {  
  `GUIXMLLocation_Device`,  
  `GUIXMLLocation_Host`,  
  `NUMGUIXMLLOCATION` }
- enum `GenICamXMLLocationEnum` {  
  `GenICamXMLLocation_Device`,  
  `GenICamXMLLocation_Host`,  
  `NUMGENICAMXMLLOCATION` }
- enum `DeviceEndianessMechanismEnum` {  
  `DeviceEndianessMechanism_Legacy`,  
  `DeviceEndianessMechanism_Standard`,  
  `NUMDEVICEENDIANESSMECHANISM` }
- enum `DeviceCurrentSpeedEnum` {  
  `DeviceCurrentSpeed_UnknownSpeed`,  
  `DeviceCurrentSpeed_LowSpeed`,  
  `DeviceCurrentSpeed_FullSpeed`,  
  `DeviceCurrentSpeed_HighSpeed`,  
  `DeviceCurrentSpeed_SuperSpeed`,  
  `NUMDEVICECURRENTSPEED` }
- enum `POEStatusEnum` {  
  `POEStatus_NotSupported`,  
  `POEStatus_PowerOff`,  
  `POEStatus_PowerOn`,  
  `NUMPOESTATUS` }
- enum `FilterDriverStatusEnum` {  
  `FilterDriverStatus_NotSupported`,  
  `FilterDriverStatus_Disabled`,  
  `FilterDriverStatus_Enabled`,  
  `NUMFILTERDRIVERSTATUS` }

## 11.144 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](#)

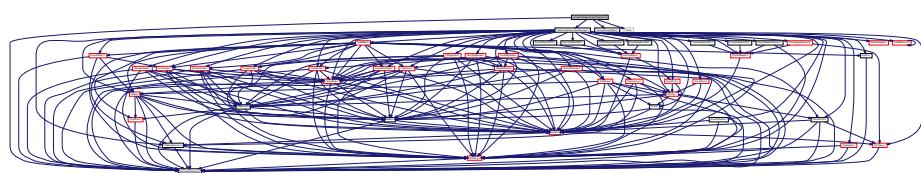
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

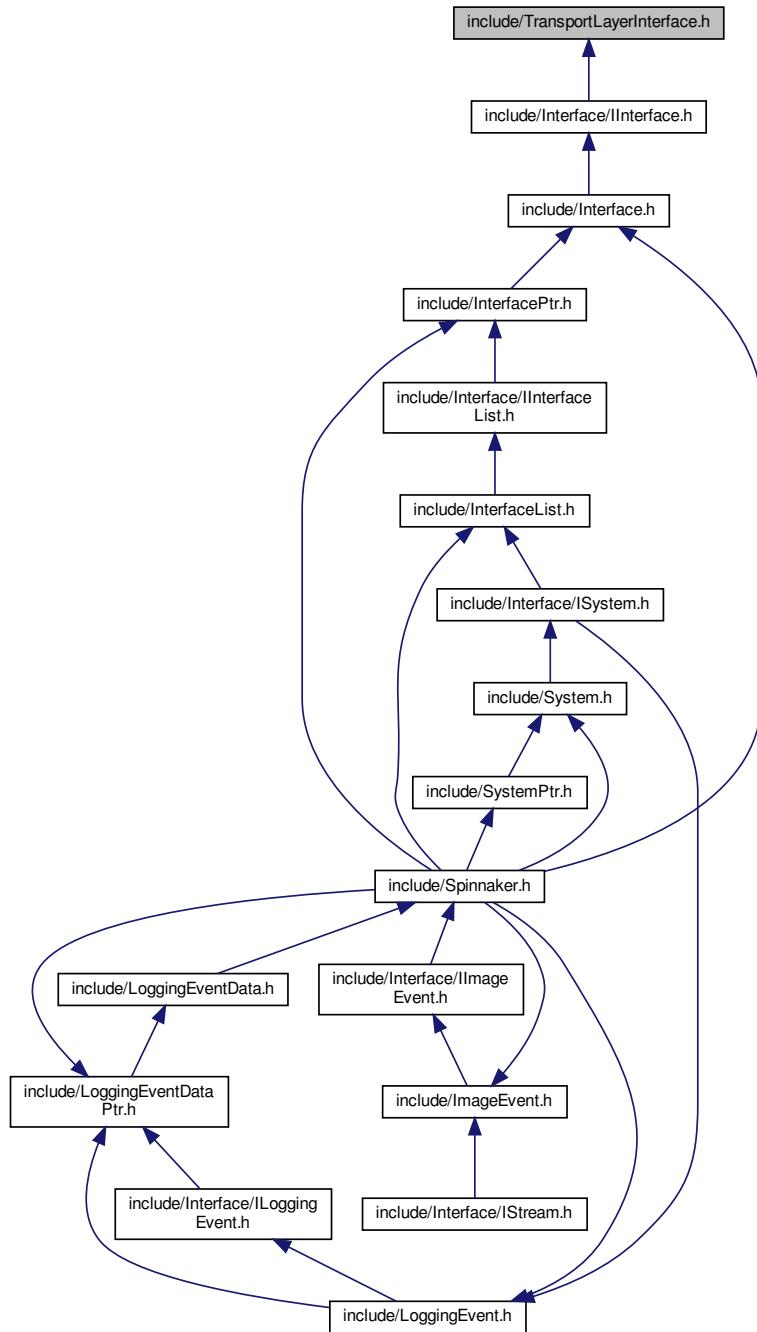
- [Spinnaker](#)

## 11.145 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](#)

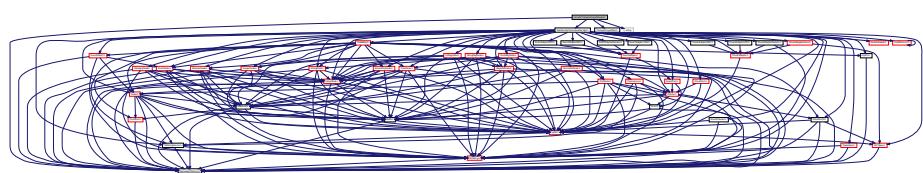
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

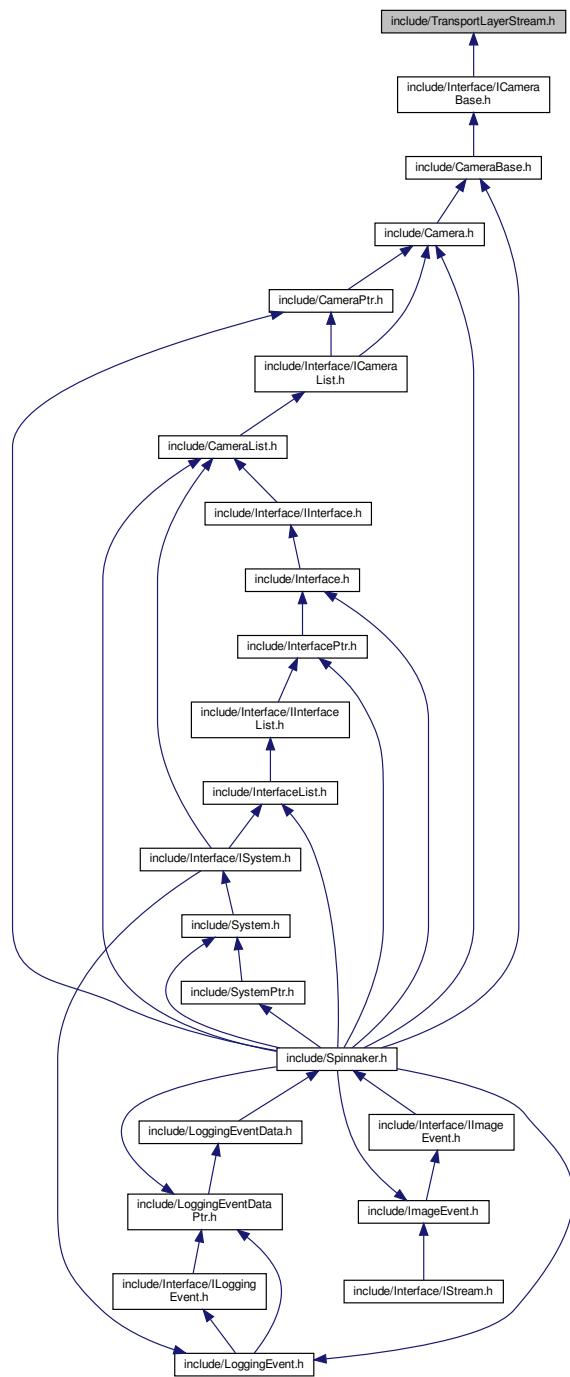
- [Spinnaker](#)

## 11.146 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](#)

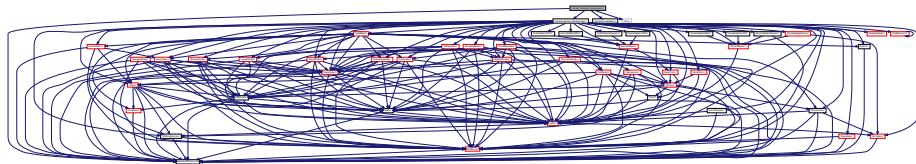
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

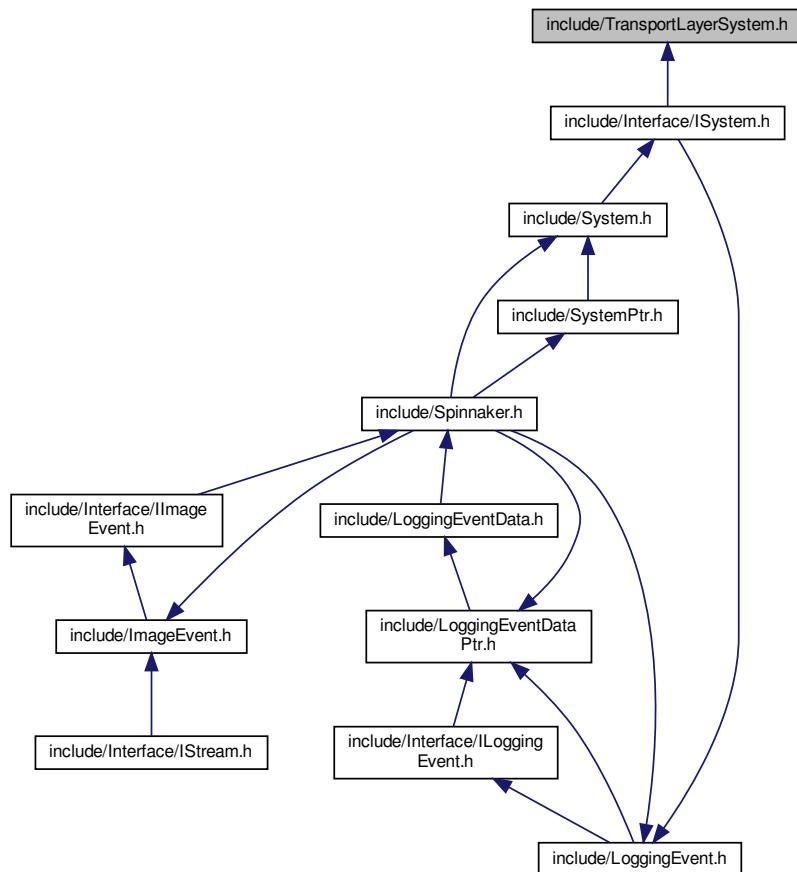
- [Spinnaker](#)

## 11.147 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](#)



# Index

\_ClearXMLCache  
    Spinnaker GenApi Classes, [233](#)  
    Spinnaker::GenApi::CNodeMapRefT, [721](#)

\_Connect  
    Spinnaker GenApi Classes, [233, 234](#)  
    Spinnaker::GenApi::CNodeMapRefT, [721](#)

\_Destroy  
    Spinnaker GenApi Classes, [234](#)

\_GetDeviceName  
    Spinnaker GenApi Classes, [234](#)  
    Spinnaker::GenApi::CNodeMapRefT, [721](#)

\_GetNode  
    Spinnaker GenApi Classes, [234](#)  
    Spinnaker::GenApi::CNodeMapRefT, [721](#)

\_GetNodes  
    Spinnaker GenApi Classes, [234](#)  
    Spinnaker::GenApi::CNodeMapRefT, [722](#)

\_GetSupportedSchemaVersions  
    Spinnaker GenApi Classes, [234](#)  
    Spinnaker::GenApi::CNodeMapRefT, [722](#)

\_Initialize  
    Spinnaker::GenApi::CGeneric\_XMLLoader←  
        Params, [681](#)

\_InvalidateNodes  
    Spinnaker GenApi Classes, [234](#)  
    Spinnaker::GenApi::CNodeMapRefT, [722](#)

\_LoadXMLFromFile  
    Spinnaker GenApi Classes, [235](#)  
    Spinnaker::GenApi::CNodeMapRefT, [722](#)

\_LoadXMLFromFileInject  
    Spinnaker GenApi Classes, [235](#)  
    Spinnaker::GenApi::CNodeMapRefT, [722](#)

\_LoadXMLFromString  
    Spinnaker GenApi Classes, [235](#)  
    Spinnaker::GenApi::CNodeMapRefT, [723](#)

\_LoadXMLFromStringInject  
    Spinnaker GenApi Classes, [235](#)  
    Spinnaker::GenApi::CNodeMapRefT, [723](#)

\_LoadXMLFromZIPData  
    Spinnaker GenApi Classes, [235](#)  
    Spinnaker::GenApi::CNodeMapRefT, [723](#)

\_LoadXMLFromZIPFile  
    Spinnaker GenApi Classes, [235](#)  
    Spinnaker::GenApi::CNodeMapRefT, [723](#)

\_Poll  
    Spinnaker GenApi Classes, [235](#)  
    Spinnaker::GenApi::CNodeMapRefT, [723](#)

\_Ptr  
    Spinnaker::GenApi::CNodeMapRefT, [724](#)

    Spinnaker::GenApi::NodeMap, [1032](#)

\_TO\_STRING  
    GCUtilities.h, [1297](#)

\_UndefinedRepresentation  
    Types Enums, [367](#)

\_ERR\_  
    GCUtilities.h, [1297](#)

\_LINE\_STR\_  
    GCUtilities.h, [1297](#)

\_LOCATION\_  
    GCUtilities.h, [1297](#)

\_OUTPUT\_FORMATER\_  
    GCUtilities.h, [1297](#)

\_STDC\_CONSTANT\_MACROS  
    GCTypes.h, [1292](#)

\_STDC\_LIMIT\_MACROS  
    GCTypes.h, [1292](#)

\_TODO\_  
    GCUtilities.h, [1297](#)

\_WARN\_  
    GCUtilities.h, [1297](#)

\_npos  
    Spinnaker::GenICam::gcstring, [807](#)

\_pCount  
    Spinnaker::GenApi::double\_autovector\_t, [757](#)  
    Spinnaker::GenApi::int64\_autovector\_t, [957](#)

\_pv  
    Spinnaker::GenApi::double\_autovector\_t, [757](#)  
    Spinnaker::GenApi::int64\_autovector\_t, [957](#)

~ArrivalEvent  
    Spinnaker::ArrivalEvent, [448](#)

~AutoLock  
    Spinnaker::GenApi::AutoLock, [451](#)  
    Spinnaker::GenICam::AutoLock, [450](#)

~BasePtr  
    Spinnaker::BasePtr, [454](#)

~BooleanNode  
    Spinnaker::GenApi::BooleanNode, [460](#)

~CChunkAdapter  
    Spinnaker::GenApi::CChunkAdapter, [638](#)

~CChunkAdapterDcam  
    Spinnaker::GenApi::CChunkAdapterDcam, [642](#)

~CChunkAdapterGEV  
    Spinnaker::GenApi::CChunkAdapterGEV, [647](#)

~CChunkAdapterGeneric  
    Spinnaker::GenApi::CChunkAdapterGeneric, [644](#)

~CChunkAdapterU3V  
    Spinnaker::GenApi::CChunkAdapterU3V, [649](#)

~CChunkPort

Spinnaker::GenApi::CChunkPort, 651  
 ~CEnumerationTRef  
     Spinnaker::GenApi::CEnumerationTRef, 657  
 ~CEventAdapter  
     Spinnaker::GenApi::CEventAdapter, 661  
 ~CEventAdapter1394  
     Spinnaker::GenApi::CEventAdapter1394, 663  
 ~CEventAdapterGEV  
     Spinnaker::GenApi::CEventAdapterGEV, 668  
 ~CEventAdapterGeneric  
     Spinnaker::GenApi::CEventAdapterGeneric, 665  
 ~CEventAdapterU3V  
     Spinnaker::GenApi::CEventAdapterU3V, 670  
 ~CEventPort  
     Spinnaker::GenApi::CEventPort, 673  
 ~CFeatureBag  
     Spinnaker::GenApi::CFeatureBag, 677  
 ~CGlobalLock  
     Spinnaker::GenICam::CGlobalLock, 683  
 ~CGlobalLockUnlocker  
     Spinnaker::GenICam::CGlobalLockUnlocker, 685  
 ~CLock  
     Spinnaker::GenApi::CLock, 701  
     Spinnaker::GenICam::CLock, 699  
 ~CNodeCallback  
     Spinnaker::GenApi::CNodeCallback, 706  
 ~CNodeMapFactory  
     Spinnaker::GenApi::CNodeMapFactory, 710  
 ~CNodeMapRefT  
     Spinnaker GenApi Classes, 237  
 ~CPointer  
     Spinnaker::GenApi::CPointer, 731  
 ~CPortImpl  
     Spinnaker::GenApi::CPortImpl, 735  
 ~CPortWriteList  
     Spinnaker::GenApi::CPortWriteList, 739  
 ~CRegisterPortImpl  
     Spinnaker::GenApi::CRegisterPortImpl, 742  
 ~CSelectorSet  
     Spinnaker::GenApi::CSelectorSet, 746  
 ~Camera  
     Spinnaker::Camera, 491  
 ~CameraBase  
     Spinnaker::CameraBase, 615  
 ~CameraList  
     Spinnaker::CameraList, 629  
 ~CategoryNode  
     Spinnaker::GenApi::CategoryNode, 637  
 ~ChunkData  
     Spinnaker::ChunkData, 689  
 ~CommandNode  
     Spinnaker::GenApi::CommandNode, 726  
 ~DeviceEvent  
     Spinnaker::DeviceEvent, 754  
 ~EnumEntryNode  
     Spinnaker::GenApi::EnumEntryNode, 766  
 ~EnumNode  
     Spinnaker::GenApi::EnumNode, 770  
 ~Event  
     Spinnaker::Event, 779  
 ~Exception  
     Spinnaker::Exception, 785  
 ~FileProtocolAdapter  
     Spinnaker::GenApi::FileProtocolAdapter, 789  
 ~FloatNode  
     Spinnaker::GenApi::FloatNode, 796  
 ~FloatRegNode  
     Spinnaker::GenApi::FloatRegNode, 802  
 ~IArrivalEvent  
     Spinnaker::IArrivalEvent, 827  
 ~ICameraBase  
     Spinnaker::ICameraBase, 830  
 ~ICameraList  
     Spinnaker::ICameraList, 838  
 ~IChunkData  
     Spinnaker::IChunkData, 842  
 ~IDataStream  
     Spinnaker::IDataStream, 849  
 ~IDevFileStreamBuf  
     Spinnaker::GenApi::IDevFileStreamBuf, 858  
 ~IDeviceEvent  
     Spinnaker::IDeviceEvent, 860  
 ~IImage  
     Spinnaker::IImage, 863  
 ~IImageEvent  
     Spinnaker::IImageEvent, 875  
 ~IImageStatistics  
     Spinnaker::IImageStatistics, 877  
 ~IInterface  
     Spinnaker::IInterface, 881  
 ~IInterfaceArrivalEvent  
     Spinnaker::IInterfaceArrivalEvent, 885  
 ~IInterfaceEvent  
     Spinnaker::IInterfaceEvent, 887  
 ~IInterfaceList  
     Spinnaker::IInterfaceList, 889  
 ~IInterfaceRemovalEvent  
     Spinnaker::IInterfaceRemovalEvent, 892  
 ~ILoggingEvent  
     Spinnaker::ILoggingEvent, 894  
 ~IRemovalEvent  
     Spinnaker::IRemovalEvent, 987  
 ~ISystem  
     Spinnaker::ISystem, 989  
 ~ISystemEvent  
     Spinnaker::ISystemEvent, 995  
 ~Image  
     Spinnaker::Image, 899  
 ~ImageEvent  
     Spinnaker::ImageEvent, 926  
 ~ImagePtr  
     Spinnaker::ImagePtr, 928  
 ~ImageStatistics  
     Spinnaker::ImageStatistics, 931  
 ~InferenceBoundingBoxResult  
     Chunk Data Inference Class, 162

~IntRegNode  
    Spinnaker::GenApi::IntRegNode, 984

~IntegerNode  
    Spinnaker::GenApi::IntegerNode, 960

~Interface  
    Spinnaker::Interface, 965

~InterfaceArrivalEvent  
    Spinnaker::InterfaceArrivalEvent, 970

~InterfaceEvent  
    Spinnaker::InterfaceEvent, 973

~InterfaceList  
    Spinnaker::InterfaceList, 975

~InterfaceRemovalEvent  
    Spinnaker::InterfaceRemovalEvent, 981

~Lock  
    Spinnaker::GenICam::LockableObject::Lock, 1000

~LoggingEvent  
    Spinnaker::LoggingEvent, 1004

~LoggingEventData  
    Spinnaker::LoggingEventData, 1006

~Node  
    Spinnaker::GenApi::Node, 1017

~NodeMap  
    Spinnaker::GenApi::NodeMap, 1027

~ODevFileStreamBuf  
    Spinnaker::GenApi::ODevFileStreamBuf, 1037

~PortNode  
    Spinnaker::GenApi::PortNode, 1043

~PortRecorder  
    Spinnaker::GenApi::PortRecorder, 1048

~PortReplay  
    Spinnaker::GenApi::PortReplay, 1050

~RegisterNode  
    Spinnaker::GenApi::RegisterNode, 1055

~RemovalEvent  
    Spinnaker::RemovalEvent, 1058

~SpinVideo  
    Spinnaker::Video::SpinVideo, 1062

~StringNode  
    Spinnaker::GenApi::StringNode, 1068

~StringRegNode  
    Spinnaker::GenApi::StringRegNode, 1072

~System  
    Spinnaker::System, 1075

~SystemEvent  
    Spinnaker::SystemEvent, 1084

~SystemPtr  
    Spinnaker::SystemPtr, 1087

~TransportLayerDevice  
    Spinnaker::TransportLayerDevice, 1091

~TransportLayerInterface  
    Spinnaker::TransportLayerInterface, 1102

~TransportLayerStream  
    Spinnaker::TransportLayerStream, 1111

~TransportLayerSystem  
    Spinnaker::TransportLayerSystem, 1117

~ValueNode  
    Spinnaker::GenApi::ValueNode, 1123

~double\_automvector\_t  
    Spinnaker::GenApi::double\_automvector\_t, 756

~gcstring  
    Spinnaker::GenICam::gcstring, 807

~int64\_automvector\_t  
    Spinnaker::GenApi::int64\_automvector\_t, 956

ADAPTERCONFIG\_API  
    AdapterConfig.h, 1128

aPAUSEMACCtrlFramesReceived  
    Spinnaker::Camera, 497

aPAUSEMACCtrlFramesTransmitted  
    Spinnaker::Camera, 497

AVI Recorder Class, 35  
    DEPRECATED\_CLASS, 35

AVIOption, 451  
    Spinnaker::Video::AVIOption, 452

AasRoiEnable  
    Spinnaker::Camera, 492

AasRoiHeight  
    Spinnaker::Camera, 492

AasRoiOffsetX  
    Spinnaker::Camera, 492

AasRoiOffsetY  
    Spinnaker::Camera, 493

AasRoiWidth  
    Spinnaker::Camera, 493

AcquisitionAbort  
    Spinnaker::Camera, 493

AcquisitionArm  
    Spinnaker::Camera, 493

AcquisitionBurstFrameCount  
    Spinnaker::Camera, 493

AcquisitionFrameCount  
    Spinnaker::Camera, 494

AcquisitionFrameRate  
    Spinnaker::Camera, 494

AcquisitionFrameRateEnable  
    Spinnaker::Camera, 494

AcquisitionLineRate  
    Spinnaker::Camera, 494

AcquisitionMode  
    Spinnaker::Camera, 494

AcquisitionModeEnums  
    CameraDefs Class, 73

AcquisitionResultingFrameRate  
    Spinnaker::Camera, 495

AcquisitionStart  
    Spinnaker::Camera, 495

AcquisitionStatus  
    Spinnaker::Camera, 495

AcquisitionStatusSelector  
    Spinnaker::Camera, 495

AcquisitionStatusSelectorEnums  
    CameraDefs Class, 73

AcquisitionStop  
    Spinnaker::Camera, 495

ActionCommand  
    Spinnaker::TransportLayerInterface, 1102

ActionCommandResult, 441  
 ActionCommandStatus  
     Spinnaker Definitions, 193  
 ActionDeviceKey  
     Spinnaker::Camera, 495  
 ActionGroupKey  
     Spinnaker::Camera, 496  
 ActionGroupMask  
     Spinnaker::Camera, 496  
 ActionQueueSize  
     Spinnaker::Camera, 496  
 ActionSelector  
     Spinnaker::Camera, 496  
 ActionUnconditionalMode  
     Spinnaker::Camera, 496  
 ActionUnconditionalModeEnums  
     CameraDefs Class, 74  
 AdapterConfig, 377  
     AdapterConfigErr, 378  
     AutoPopulateAdapterInfo, 378  
     AutoPopulateAdvancedProperties, 378  
     ConfigureAdapter, 379  
     GetAuto10GDesc, 379  
     GetAutoGigabitDesc, 379  
     GetAutoStartGateway, 379  
     GetAutoStartIp, 379  
     GetAutoSubnetMask, 379  
     GetAutoSubnetMaskLength, 379  
     GetConfigLogFileName, 380  
     GetEnumerationLogFileName, 380  
     GetGatewayAddress, 380  
     GetMaxIpAddress, 380  
     GetMinIpAddress, 380  
     GetSubnetMaskLength, 380  
     IsOnSameSubnet, 380  
     IsValidIpAddress, 381  
     IsValidSubnetMask, 381  
     PopulateAdapterIplInfo, 381  
     RetrieveAllAdapters, 381  
     ValidateIpAddress, 381  
 AdapterConfig.h  
     ADAPTERCONFIG\_API, 1128  
 AdapterConfig::AdapterInfo  
     adapterDescription, 444  
     adapterGUID, 444  
     AdapterInfo, 444  
     adapterName, 444  
     dhcpEnabled, 445  
     iplInfo, 445  
     jumboPacketValidValues, 445  
     jumboPackets, 445  
     jumboPacketsRegKey, 445  
     receiveBuffers, 445  
     receiveBuffersMax, 445  
     receiveBuffersMin, 445  
     receiveBuffersRegKey, 446  
     receiveBuffersStep, 446  
     transmitBuffers, 446  
     transmitBuffersMax, 446  
     transmitBuffersMin, 446  
     transmitBuffersRegKey, 446  
     transmitBuffersStep, 446  
 AdapterConfig::IplInfo  
     gateway, 985  
     ipAddress, 985  
     IplInfo, 985  
     subnetLength, 985  
     subnetMask, 985  
 AdapterConfigErr  
     AdapterConfig, 378  
 AdapterConfigException, 442  
     AdapterConfigException, 442  
     ErrCode, 443  
     GetParamStr, 443  
 adapterDescription  
     AdapterConfig::AdapterInfo, 444  
 adapterGUID  
     AdapterConfig::AdapterInfo, 444  
 AdapterInfo, 443  
     AdapterConfig::AdapterInfo, 444  
 adapterName  
     AdapterConfig::AdapterInfo, 444  
 AdaptiveCompressionEnable  
     Spinnaker::Camera, 496  
 AdcBitDepth  
     Spinnaker::Camera, 497  
 AdcBitDepthEnums  
     CameraDefs Class, 74  
 AddInjectionData  
     Spinnaker::GenApi::CNodeMapFactory, 712  
 Address  
     IPort Interface, 321  
 AnnounceImage  
     Spinnaker::IDataStream, 850  
 Append  
     Spinnaker::CameraList, 630  
     Spinnaker::ICameraList, 838  
     Spinnaker::Video::SpinVideo, 1062  
 append  
     Spinnaker::GenICam::gcstring, 807  
 ApplyStyleSheet  
     Spinnaker::GenApi::CNodeMapFactory, 712  
 ArrivalEvent, 447  
     Spinnaker::ArrivalEvent, 448  
 ArrivalEvent Class, 30  
 assign  
     Spinnaker::GenICam::gcstring, 807, 808  
 attach  
     Spinnaker::GenApi::FileProtocolAdapter, 790  
 AttachBuffer  
     Spinnaker::GenApi::CChunkAdapter, 639  
     Spinnaker::GenApi::CChunkAdapterDcam, 642  
     Spinnaker::GenApi::CChunkAdapterGEV, 647  
     Spinnaker::GenApi::CChunkAdapterGeneric, 645  
     Spinnaker::GenApi::CChunkAdapterU3V, 649  
     Spinnaker::IDataStream, 850

AttachChunk  
    Spinnaker::GenApi::CChunkPort, 652

AttachEvent  
    Spinnaker::GenApi::CEventPort, 673

AttachNode  
    Spinnaker::GenApi::CEventPort, 673

AttachNodeMap  
    Spinnaker::GenApi::CChunkAdapter, 639  
    Spinnaker::GenApi::CEventAdapter, 661

AttachPort  
    Spinnaker::GenApi::CChunkPort, 652

AttachStatistics\_t, 449  
    NumAttachedChunks, 449  
    NumChunkPorts, 450  
    NumChunks, 450

AutoAlgorithmSelector  
    Spinnaker::Camera, 497

AutoAlgorithmSelectorEnums  
    CameraDefs Class, 74

AutoExposureControlLoopDamping  
    Spinnaker::Camera, 497

AutoExposureControlPriority  
    Spinnaker::Camera, 498

AutoExposureControlPriorityEnums  
    CameraDefs Class, 75

AutoExposureEVCompensation  
    Spinnaker::Camera, 498

AutoExposureExposureTimeLowerLimit  
    Spinnaker::Camera, 498

AutoExposureExposureTimeUpperLimit  
    Spinnaker::Camera, 499

AutoExposureGainLowerLimit  
    Spinnaker::Camera, 499

AutoExposureGainUpperLimit  
    Spinnaker::Camera, 499

AutoExposureGreyValueLowerLimit  
    Spinnaker::Camera, 499

AutoExposureGreyValueUpperLimit  
    Spinnaker::Camera, 499

AutoExposureLightingMode  
    Spinnaker::Camera, 500

AutoExposureLightingModeEnums  
    CameraDefs Class, 75

AutoExposureMeteringMode  
    Spinnaker::Camera, 500

AutoExposureMeteringModeEnums  
    CameraDefs Class, 75

AutoExposureTargetGreyValue  
    Spinnaker::Camera, 500

AutoExposureTargetGreyValueAuto  
    Spinnaker::Camera, 501

AutoExposureTargetGreyValueAutoEnums  
    CameraDefs Class, 76

AutoForceIP  
    Spinnaker::TransportLayerInterface, 1103  
    Spinnaker::TransportLayerSystem, 1117

AutoLock, 450, 451  
    Spinnaker::GenApi::AutoLock, 451

    Spinnaker::GenCam::AutoLock, 450

AutoPopulateAdapterInfo  
    AdapterConfig, 378

AutoPopulateAdvancedProperties  
    AdapterConfig, 378

AutoVector Class, 238

Automatic  
    Types Enums, 373

BMPOption, 457  
    Spinnaker::BMPOption, 457

BalanceRatio  
    Spinnaker::Camera, 501

BalanceRatioSelector  
    Spinnaker::Camera, 501

BalanceRatioSelectorEnums  
    CameraDefs Class, 76

BalanceWhiteAuto  
    Spinnaker::Camera, 502

BalanceWhiteAutoDamping  
    Spinnaker::Camera, 502

BalanceWhiteAutoEnums  
    CameraDefs Class, 77

BalanceWhiteAutoLowerLimit  
    Spinnaker::Camera, 502

BalanceWhiteAutoProfile  
    Spinnaker::Camera, 502

BalanceWhiteAutoProfileEnums  
    CameraDefs Class, 77

BalanceWhiteAutoUpperLimit  
    Spinnaker::Camera, 503

BasePtr  
    Spinnaker::BasePtr, 453, 454

BasePtr Class, 38

BasePtr< T, B >, 452

BeginAcquisition  
    Spinnaker::CameraBase, 616  
    Spinnaker::ICameraBase, 830

Beginner  
    Types Enums, 374

binaryFile  
    Spinnaker::PGMOption, 1039  
    Spinnaker::PPMOption, 1052

BinningHorizontal  
    Spinnaker::Camera, 503

BinningHorizontalMode  
    Spinnaker::Camera, 503

BinningHorizontalModeEnums  
    CameraDefs Class, 77

BinningSelector  
    Spinnaker::Camera, 503

BinningSelectorEnums  
    CameraDefs Class, 78

BinningVertical  
    Spinnaker::Camera, 503

BinningVerticalMode  
    Spinnaker::Camera, 504

BinningVerticalModeEnums  
    CameraDefs Class, 78

bitrate  
     Spinnaker::Video::H264Option, 824

BlackLevel  
     Spinnaker::Camera, 504

BlackLevelAuto  
     Spinnaker::Camera, 504

BlackLevelAutoBalance  
     Spinnaker::Camera, 504

BlackLevelAutoBalanceEnums  
     CameraDefs Class, 78

BlackLevelAutoEnums  
     CameraDefs Class, 79

BlackLevelClampingEnable  
     Spinnaker::Camera, 504

BlackLevelRaw  
     Spinnaker::Camera, 505

BlackLevelSelector  
     Spinnaker::Camera, 505

BlackLevelSelectorEnums  
     CameraDefs Class, 79

BlockId  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 817  
     GVCP\_EVENT\_ITEM, 815

BlockId64High  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 817

BlockId64Low  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 817

Boolean  
     Types Enums, 372

BooleanNode, 458  
     Spinnaker::GenApi::BooleanNode, 459, 460

BooleanNode Class, 242  
     CBooleanRef, 242

bottomRightXCoord  
     Chunk Data Inference Class, 163

bottomRightYCoord  
     Chunk Data Inference Class, 163

boxType  
     Chunk Data Inference Class, 163

BufferOwnership  
     Spinnaker Definitions, 193

build  
     Spinnaker::LibraryVersion, 999

c\_str  
     Spinnaker::GenICam::gcstring, 808

CBasePtr  
     Pointer Class, 345

CBooleanPtr  
     Pointer Class, 345

CBooleanRef  
     BooleanNode Class, 242

CCategoryPtr  
     Pointer Class, 346

CCategoryRef  
     CategoryNode Class, 243

CChunkAdapter, 637  
     Spinnaker::GenApi::CChunkAdapter, 638

CChunkAdapterDcam, 641  
     Spinnaker::GenApi::CChunkAdapterDcam, 642

CChunkAdapterGEV, 646  
     Spinnaker::GenApi::CChunkAdapterGEV, 647

CChunkAdapterGeneric, 643  
     Spinnaker::GenApi::CChunkAdapterGeneric, 644

CChunkAdapterU3V, 648  
     Spinnaker::GenApi::CChunkAdapterU3V, 649

CChunkPort, 650  
     Spinnaker::GenApi::CChunkPort, 651

CChunkPortPtr  
     Pointer Class, 346

CCommandPtr  
     Pointer Class, 346

CCommandRef  
     CommandNode Class, 249

CDeviceInfoPtr  
     Pointer Class, 346

CEnumEntryPtr  
     Pointer Class, 346

CEnumEntryRef  
     EnumEntryNode Class, 254

CEnumerationPtr  
     Pointer Class, 346

CEnumerationRef  
     EnumNode Class, 255

CEnumerationTRef  
     Spinnaker::GenApi::CEnumerationTRef, 657

CEnumerationTRef< EnumT >, 655

CEventAdapter, 660  
     Spinnaker::GenApi::CEventAdapter, 661

CEventAdapter1394, 662  
     Spinnaker::GenApi::CEventAdapter1394, 663

CEventAdapterGEV, 667  
     Spinnaker::GenApi::CEventAdapterGEV, 668

CEventAdapterGeneric, 664  
     Spinnaker::GenApi::CEventAdapterGeneric, 665

CEventAdapterU3V, 669  
     Spinnaker::GenApi::CEventAdapterU3V, 670

CEventPort, 671  
     Spinnaker::GenApi::CEventPort, 672

CFeatureBag, 676  
     Spinnaker::GenApi::CFeatureBag, 677

CFloatPtr, 679  
     Spinnaker::GenApi::CFloatPtr, 680

CFloatRef  
     FloatNode Class, 264

CGeneric\_XMLLoaderParams, 681

CGlobalLock, 682  
     Spinnaker::GenICam::CGlobalLock, 682, 683

CGlobalLockUnlocker, 684  
     Spinnaker::GenICam::CGlobalLockUnlocker, 685

CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN  
     IChunkPort Interface, 279

CHUNK\_BASE\_ADDRESS\_REGISTER  
     IChunkPort Interface, 279

CHUNK\_LENGTH\_REGISTER\_LEN  
     IChunkPort Interface, 280

CHUNK\_LENGTH\_REGISTER

IChunkPort Interface, 280  
CIntegerPtr  
    Pointer Class, 347  
CIntegerRef  
    IntegerNode Class, 319  
Clock, 698, 700  
    Spinnaker::GenApi::Clock, 701  
    Spinnaker::GenICam::Clock, 699  
ClockEx, 703, 704  
CNodeCallback, 705  
    Spinnaker::GenApi::CNodeCallback, 706  
CNodeMapDynPtr  
    Pointer Class, 347  
CNodeMapFactory, 708  
    Spinnaker::GenApi::CNodeMapFactory, 710, 711  
CNodeMapFactory::NodeStatistics\_t, 1033  
CNodeMapPtr  
    Pointer Class, 347  
CNodeMapRef, 716  
    Spinnaker GenApi Classes, 233  
    Spinnaker::GenApi::CNodeMapRef, 717, 718  
CNodeMapRefT< TCameraParams >, 719  
CNodeMapRefT  
    Spinnaker GenApi Classes, 236  
CNodePtr  
    Pointer Class, 347  
CNodeRef  
    Spinnaker GenApi Classes, 233  
COMMAND\_MAGIC  
    Spinnaker::GenApi, 437  
CPointer  
    Spinnaker::GenApi::CPointer, 730  
CPointer< T, B >, 729  
CPortConstructPtr  
    Pointer Class, 347  
CPortImpl, 734  
    Spinnaker::GenApi::CPortImpl, 735  
CPortPtr  
    Pointer Class, 347  
CPortRecorderPtr  
    Pointer Class, 348  
CPortRecorderRef  
    PortRecorder Class, 352  
CPortRef  
    PortNode Class, 351  
CPortReplayPtr  
    Pointer Class, 348  
CPortWriteList, 738  
    Spinnaker::GenApi::CPortWriteList, 739  
CPortWriteListPtr  
    Pointer Class, 348  
CRCChecksum  
    DCAM\_CHECKSUM, 751  
CRegisterPortImpl, 741  
    Spinnaker::GenApi::CRegisterPortImpl, 742  
CRegisterPtr  
    Pointer Class, 348  
CRegisterRef  
    RegisterNode Class, 356  
CSelectorPtr  
    Pointer Class, 348  
CSelectorRef  
    Spinnaker GenApi Classes, 233  
CSelectorSet, 744  
    Spinnaker::GenApi::CSelectorSet, 745  
CStringPtr  
    Pointer Class, 348  
CStringRef  
    StringNode Class, 360  
CTestPortStruct  
    Spinnaker::GenApi::CTestPortStruct, 749  
CTestPortStruct< CDataStruct >, 747  
CValuePtr  
    Pointer Class, 349  
CValueRef  
    ValueNode Class, 375  
CacheChunkData  
    IChunkPort Interface, 280  
    Spinnaker::GenApi::PortNode, 1043  
CalculateStatistics  
    Spinnaker::IlImage, 864  
    Spinnaker::Image, 900  
CallbackHandleType  
    Spinnaker GenApi Interfaces, 240  
Camera, 461  
    Spinnaker::Camera, 491  
Camera Base Class, 40  
Camera Base Interface Class, 220  
Camera Class, 39  
Camera List Class, 156  
CameraBase, 613  
    Spinnaker::CameraBase, 615, 616  
    Spinnaker::TransportLayerDevice, 1092  
    Spinnaker::TransportLayerStream, 1111  
CameraDefs Class, 41  
    AcquisitionModeEnums, 73  
    AcquisitionStatusSelectorEnums, 73  
    ActionUnconditionalModeEnums, 74  
    AdcBitDepthEnums, 74  
    AutoAlgorithmSelectorEnums, 74  
    AutoExposureControlPriorityEnums, 75  
    AutoExposureLightingModeEnums, 75  
    AutoExposureMeteringModeEnums, 75  
    AutoExposureTargetGreyValueAutoEnums, 76  
    BalanceRatioSelectorEnums, 76  
    BalanceWhiteAutoEnums, 77  
    BalanceWhiteAutoProfileEnums, 77  
    BinningHorizontalModeEnums, 77  
    BinningSelectorEnums, 78  
    BinningVerticalModeEnums, 78  
    BlackLevelAutoBalanceEnums, 78  
    BlackLevelAutoEnums, 79  
    BlackLevelSelectorEnums, 79  
    ChunkBlackLevelSelectorEnums, 79  
    ChunkCounterSelectorEnums, 79  
    ChunkEncoderSelectorEnums, 80

ChunkEncoderStatusEnums, 80  
 ChunkExposureTimeSelectorEnums, 80  
 ChunkGainSelectorEnums, 81  
 ChunkImageComponentEnums, 81  
 ChunkPixelFormatEnums, 82  
 ChunkRegionIDEnums, 82  
 ChunkScan3dCoordinateReferenceSelector Enums, 82  
 ChunkScan3dCoordinateSelectorEnums, 83  
 ChunkScan3dCoordinateSystemEnums, 83  
 ChunkScan3dCoordinateSystemReferenceEnums, 83  
 ChunkScan3dCoordinateTransformSelector Enums, 84  
 ChunkScan3dDistanceUnitEnums, 84  
 ChunkScan3dOutputModeEnums, 85  
 ChunkSelectorEnums, 85  
 ChunkSourceIDEnums, 86  
 ChunkTimerSelectorEnums, 86  
 ChunkTransferStreamIDEnums, 87  
 CIConfigurationEnums, 87  
 CITimeSlotsCountEnums, 87  
 ColorTransformationSelectorEnums, 88  
 ColorTransformationValueSelectorEnums, 88  
 CounterEventActivationEnums, 89  
 CounterEventSourceEnums, 89  
 CounterResetActivationEnums, 90  
 CounterResetSourceEnums, 90  
 CounterSelectorEnums, 90  
 CounterStatusEnums, 91  
 CounterTriggerActivationEnums, 91  
 CounterTriggerSourceEnums, 91  
 CxpConnectionTestModeEnums, 92  
 CxpLinkConfigurationEnums, 92  
 CxpLinkConfigurationPreferredEnums, 93  
 CxpLinkConfigurationStatusEnums, 94  
 CxpPoCxpStatusEnums, 95  
 DecimationHorizontalModeEnums, 96  
 DecimationSelectorEnums, 96  
 DecimationVerticalModeEnums, 96  
 DefectCorrectionModeEnums, 96  
 DeinterlacingEnums, 97  
 DeviceCharacterSetEnums, 97  
 DeviceClockSelectorEnums, 97  
 DeviceConnectionStatusEnums, 98  
 DeviceIndicatorModeEnums, 98  
 DeviceLinkHeartbeatModeEnums, 98  
 DeviceLinkThroughputLimitModeEnums, 100  
 DevicePowerSupplySelectorEnums, 100  
 DeviceRegistersEndiannessEnums, 100  
 DeviceScanTypeEnums, 101  
 DeviceSerialPortBaudRateEnums, 101  
 DeviceSerialPortSelectorEnums, 101  
 DeviceStreamChannelEndiannessEnums, 101  
 DeviceStreamChannelTypeEnums, 102  
 DeviceTLTypeEnums, 104  
 DeviceTapGeometryEnums, 102  
 DeviceTemperatureSelectorEnums, 103  
 DeviceTypeEnums, 104  
 EncoderModeEnums, 104  
 EncoderOutputModeEnums, 105  
 EncoderResetActivationEnums, 105  
 EncoderResetSourceEnums, 106  
 EncoderSelectorEnums, 107  
 EncoderSourceAEnums, 107  
 EncoderSourceBEnums, 107  
 EncoderStatusEnums, 108  
 EventNotificationEnums, 108  
 EventSelectorEnums, 108  
 ExposureActiveModeEnums, 109  
 ExposureAutoEnums, 109  
 ExposureModeEnums, 109  
 ExposureTimeModeEnums, 110  
 ExposureTimeSelectorEnums, 110  
 FileOpenModeEnums, 111  
 FileOperationSelectorEnums, 111  
 FileOperationStatusEnums, 111  
 FileSelectorEnums, 112  
 GainAutoBalanceEnums, 112  
 GainAutoEnums, 112  
 GainSelectorEnums, 113  
 GevCCPEnums, 113  
 GevCurrentPhysicalLinkConfigurationEnums, 113  
 GevGVCPExtendedStatusCodesSelectorEnums, 113  
 GevGVSPExtendedIDModeEnums, 114  
 GevIEEE1588ClockAccuracyEnums, 114  
 GevIEEE1588ModeEnums, 114  
 GevIEEE1588StatusEnums, 115  
 GevIPConfigurationStatusEnums, 115  
 GevPhysicalLinkConfigurationEnums, 115  
 GevSupportedOptionSelectorEnums, 116  
 ImageComponentSelectorEnums, 117  
 ImageCompressionJPEGFormatOptionEnums, 117  
 ImageCompressionModeEnums, 118  
 ImageCompressionRateOptionEnums, 118  
 LUTSelectorEnums, 122  
 LineFormatEnums, 118  
 LineInputFilterSelectorEnums, 119  
 LineModeEnums, 119  
 LineSelectorEnums, 119  
 LineSourceEnums, 120  
 LogicBlockLUTInputActivationEnums, 120  
 LogicBlockLUTInputSelectorEnums, 121  
 LogicBlockLUTInputSourceEnums, 121  
 LogicBlockLUTSelectorEnums, 122  
 LogicBlockSelectorEnums, 122  
 PixelColorFilterEnums, 123  
 PixelFormatEnums, 123  
 PixelFormatInfoSelectorEnums, 129  
 PixelSizeEnums, 134  
 RegionDestinationEnums, 135  
 RegionModeEnums, 135  
 RegionSelectorEnums, 136  
 RgbTransformLightSourceEnums, 136

Scan3dCoordinateReferenceSelectorEnums, 136  
Scan3dCoordinateSelectorEnums, 137  
Scan3dCoordinateSystemEnums, 137  
Scan3dCoordinateSystemReferenceEnums, 137  
Scan3dCoordinateTransformSelectorEnums, 138  
Scan3dDistanceUnitEnums, 138  
Scan3dOutputModeEnums, 139  
SensorDigitizationTapsEnums, 139  
SensorShutterModeEnums, 140  
SensorTapsEnums, 140  
SequencerConfigurationModeEnums, 141  
SequencerConfigurationValidEnums, 141  
SequencerModeEnums, 141  
SequencerSetValidEnums, 141  
SequencerTriggerActivationEnums, 142  
SequencerTriggerSourceEnums, 142  
SerialPortBaudRateEnums, 142  
SerialPortParityEnums, 143  
SerialPortSelectorEnums, 143  
SerialPortSourceEnums, 144  
SerialPortStopBitsEnums, 144  
SoftwareSignalSelectorEnums, 144  
SourceSelectorEnums, 145  
TestPatternEnums, 145  
TestPatternGeneratorSelectorEnums, 145  
TimerSelectorEnums, 146  
TimerStatusEnums, 146  
TimerTriggerActivationEnums, 146  
TimerTriggerSourceEnums, 147  
TransferComponentSelectorEnums, 148  
TransferControlModeEnums, 148  
TransferOperationModeEnums, 149  
TransferQueueModeEnums, 149  
TransferSelectorEnums, 149  
TransferStatusSelectorEnums, 150  
TransferTriggerActivationEnums, 150  
TransferTriggerModeEnums, 150  
TransferTriggerSelectorEnums, 151  
TransferTriggerSourceEnums, 151  
TriggerActivationEnums, 152  
TriggerModeEnums, 153  
TriggerOverlapEnums, 153  
TriggerSelectorEnums, 153  
TriggerSourceEnums, 153  
UserOutputSelectorEnums, 154  
UserSetDefaultEnums, 154  
UserSetSelectorEnums, 155  
WhiteClipSelectorEnums, 155

CameraInternal  
    Spinnaker::ICameraBase, 836  
    Spinnaker::TransportLayerDevice, 1092  
    Spinnaker::TransportLayerStream, 1111

CameraList, 628  
    Spinnaker::CameraList, 629

CameraListImpl  
    Spinnaker::ICameraList, 840

CameraPtr, 634  
    CameraPtr Class, 157, 158

    CameraPtr Class, 157  
        CameraPtr, 157, 158

    CastToIDestroy  
        Spinnaker GenApi Classes, 236

    CategoryNode, 635  
        Spinnaker::GenApi::CategoryNode, 636

    CategoryNode Class, 243  
        CCategoryRef, 243

    centerXCoord  
        Chunk Data Inference Class, 163

    centerYCoord  
        Chunk Data Inference Class, 163

    CheckBufferLayout  
        Spinnaker::GenApi::CChunkAdapter, 639  
        Spinnaker::GenApi::CChunkAdapterDcam, 642  
        Spinnaker::GenApi::CChunkAdapterGEV, 647  
        Spinnaker::GenApi::CChunkAdapterGeneric, 645  
        Spinnaker::GenApi::CChunkAdapterU3V, 649

    CheckCRC  
        Spinnaker::GenApi::CChunkAdapterDcam, 643  
        Spinnaker::Image, 864  
        Spinnaker::Image, 901

    CheckChunkID  
        Spinnaker::GenApi::CChunkPort, 652

    CheckEventID  
        Spinnaker::GenApi::CEventPort, 673

    Chunk Data Inference Class, 160  
        ~InferenceBoundingBoxResult, 162  
        bottomRightXCoord, 163  
        bottomRightYCoord, 163  
        boxType, 163  
        centerXCoord, 163  
        centerYCoord, 163  
        circle, 164  
        classId, 164  
        confidence, 164  
        GetBoxAt, 161  
        GetBoxCount, 161  
        GetBoxSize, 161  
        GetVersion, 161  
        InferenceBoundingBoxResult, 162  
        operator=, 162  
        radius, 164  
        rect, 164  
        rotatedRect, 164  
        rotationAngle, 164  
        topLeftXCoord, 164, 165  
        topLeftYCoord, 165

    ChunkAdapter Class, 244  
        ChunkAdapterDcam Class, 245  
        ChunkAdapterGEV Class, 247  
        ChunkAdapterGeneric Class, 246  
        ChunkAdapterU3V Class, 376

    ChunkBlackLevel  
        Spinnaker::Camera, 505

    ChunkBlackLevelSelector  
        Spinnaker::Camera, 505

    ChunkBlackLevelSelectorEnums

CameraDefs Class, 79  
ChunkCRC  
    Spinnaker::Camera, 506  
ChunkCounterSelector  
    Spinnaker::Camera, 505  
ChunkCounterSelectorEnums  
    CameraDefs Class, 79  
ChunkCounterValue  
    Spinnaker::Camera, 506  
ChunkData, 686  
    Spinnaker::ChunkData, 688  
ChunkData Class, 159  
ChunkEnable  
    Spinnaker::Camera, 506  
ChunkEncoderSelector  
    Spinnaker::Camera, 506  
ChunkEncoderSelectorEnums  
    CameraDefs Class, 80  
ChunkEncoderStatus  
    Spinnaker::Camera, 506  
ChunkEncoderStatusEnums  
    CameraDefs Class, 80  
ChunkEncoderValue  
    Spinnaker::Camera, 506  
ChunkExposureEndLineStatusAll  
    Spinnaker::Camera, 507  
ChunkExposureTime  
    Spinnaker::Camera, 507  
ChunkExposureTimeSelector  
    Spinnaker::Camera, 507  
ChunkExposureTimeSelectorEnums  
    CameraDefs Class, 80  
ChunkFrameID  
    Spinnaker::Camera, 507  
ChunkGain  
    Spinnaker::Camera, 507  
ChunkGainSelector  
    Spinnaker::Camera, 507  
ChunkGainSelectorEnums  
    CameraDefs Class, 81  
ChunkHeight  
    Spinnaker::Camera, 508  
ChunkID  
    DCAM\_CHUNK\_TRAILER, 752  
    GVCP\_CHUNK\_TRAILER, 814  
    SingleChunkData\_t, 1059  
    SingleChunkDataStr\_t, 1060  
    U3V\_CHUNK\_TRAILER, 1118  
ChunkImage  
    Spinnaker::Camera, 508  
ChunkImageComponent  
    Spinnaker::Camera, 508  
ChunkImageComponentEnums  
    CameraDefs Class, 81  
ChunkInferenceBoundingBoxResult  
    Spinnaker::Camera, 508  
ChunkInferenceConfidence  
    Spinnaker::Camera, 508  
    ChunkInferenceResult  
        Spinnaker::Camera, 508  
    ChunkLength  
        DCAM\_CHUNK\_TRAILER, 752  
        GVCP\_CHUNK\_TRAILER, 814  
        SingleChunkData\_t, 1059  
        SingleChunkDataStr\_t, 1060  
        U3V\_CHUNK\_TRAILER, 1118  
    ChunkLinePitch  
        Spinnaker::Camera, 509  
    ChunkLineStatusAll  
        Spinnaker::Camera, 509  
    ChunkModeActive  
        Spinnaker::Camera, 509  
    ChunkOffset  
        SingleChunkData\_t, 1059  
        SingleChunkDataStr\_t, 1060  
    ChunkOffsetX  
        Spinnaker::Camera, 509  
    ChunkOffsetY  
        Spinnaker::Camera, 509  
    ChunkPartSelector  
        Spinnaker::Camera, 509  
    ChunkPixelDynamicRangeMax  
        Spinnaker::Camera, 510  
    ChunkPixelDynamicRangeMin  
        Spinnaker::Camera, 510  
    ChunkPixelFormat  
        Spinnaker::Camera, 510  
    ChunkPixelFormatEnums  
        CameraDefs Class, 82  
    ChunkPort Class, 248  
    ChunkRegionIDEnums  
        CameraDefs Class, 82  
    ChunkRegionID  
        Spinnaker::Camera, 510  
    ChunkScan3dAxisMax  
        Spinnaker::Camera, 510  
    ChunkScan3dAxisMin  
        Spinnaker::Camera, 510  
    ChunkScan3dCoordinateOffset  
        Spinnaker::Camera, 511  
    ChunkScan3dCoordinateReferenceSelector  
        Spinnaker::Camera, 511  
    ChunkScan3dCoordinateReferenceSelectorEnums  
        CameraDefs Class, 82  
    ChunkScan3dCoordinateReferenceValue  
        Spinnaker::Camera, 511  
    ChunkScan3dCoordinateScale  
        Spinnaker::Camera, 511  
    ChunkScan3dCoordinateSelector  
        Spinnaker::Camera, 511  
    ChunkScan3dCoordinateSelectorEnums  
        CameraDefs Class, 83  
    ChunkScan3dCoordinateSystem  
        Spinnaker::Camera, 511  
    ChunkScan3dCoordinateSystemEnums  
        CameraDefs Class, 83

ChunkScan3dCoordinateSystemReference  
    Spinnaker::Camera, 512

ChunkScan3dCoordinateSystemReferenceEnums  
    CameraDefs Class, 83

ChunkScan3dCoordinateTransformSelector  
    Spinnaker::Camera, 512

ChunkScan3dCoordinateTransformSelectorEnums  
    CameraDefs Class, 84

ChunkScan3dDistanceUnit  
    Spinnaker::Camera, 512

ChunkScan3dDistanceUnitEnums  
    CameraDefs Class, 84

ChunkScan3dInvalidDataFlag  
    Spinnaker::Camera, 512

ChunkScan3dInvalidHeaderValue  
    Spinnaker::Camera, 512

ChunkScan3dOutputMode  
    Spinnaker::Camera, 512

ChunkScan3dOutputModeEnums  
    CameraDefs Class, 85

ChunkScan3dTransformValue  
    Spinnaker::Camera, 513

ChunkScanLineSelector  
    Spinnaker::Camera, 513

ChunkSelector  
    Spinnaker::Camera, 513

ChunkSelectorEnums  
    CameraDefs Class, 85

ChunkSequencerSetActive  
    Spinnaker::Camera, 513

ChunkSerialData  
    Spinnaker::Camera, 513

ChunkSerialDataLength  
    Spinnaker::Camera, 513

ChunkSerialReceiveOverflow  
    Spinnaker::Camera, 514

ChunkSourceIDEnums  
    CameraDefs Class, 86

ChunkSourceID  
    Spinnaker::Camera, 514

ChunkStreamChannelID  
    Spinnaker::Camera, 514

ChunkTimerSelector  
    Spinnaker::Camera, 514

ChunkTimerSelectorEnums  
    CameraDefs Class, 86

ChunkTimerValue  
    Spinnaker::Camera, 514

ChunkTimestamp  
    Spinnaker::Camera, 514

ChunkTimestampLatchValue  
    Spinnaker::Camera, 515

ChunkTransferBlockID  
    Spinnaker::Camera, 515

ChunkTransferQueueCurrentBlockCount  
    Spinnaker::Camera, 515

ChunkTransferStreamIDEnums  
    CameraDefs Class, 87

ChunkTransferStreamID  
    Spinnaker::Camera, 515

ChunkWidth  
    Spinnaker::Camera, 515

circle  
    Chunk Data Inference Class, 164

CL  
    Types Enums, 373

CIConfiguration  
    Spinnaker::Camera, 515

CIConfigurationEnums  
    CameraDefs Class, 87

CITimeSlotsCount  
    Spinnaker::Camera, 516

CITimeSlotsCountEnums  
    CameraDefs Class, 87

classId  
    Chunk Data Inference Class, 164

CleanupChunkAdapter  
    Spinnaker::IDataStream, 850

Clear  
    Spinnaker::CameraList, 631  
    Spinnaker::ICameraList, 838  
    Spinnaker::IInterfaceList, 890  
    Spinnaker::InterfaceList, 975

ClearCache  
    Spinnaker::GenApi::CChunkPort, 652  
    Spinnaker::GenApi::CNodeMapFactory, 712

ClearCaches  
    Spinnaker::GenApi::CChunkAdapter, 639

ClearXMLCache  
    Spinnaker::GenApi::NodeMap, 1027

Close  
    Spinnaker::Video::SpinVideo, 1063

close  
    Spinnaker::GenApi::IDevFileStreamBase, 856  
    Spinnaker::GenApi::IDevFileStreamBuf, 858  
    Spinnaker::GenApi::ODevFileStreamBase, 1035  
    Spinnaker::GenApi::ODevFileStreamBuf, 1037

closeFile  
    Spinnaker::GenApi::FileProtocolAdapter, 791

ColorProcessingAlgorithm  
    Spinnaker Definitions, 193

ColorTransformationEnable  
    Spinnaker::Camera, 516

ColorTransformationSelector  
    Spinnaker::Camera, 516

ColorTransformationSelectorEnums  
    CameraDefs Class, 88

ColorTransformationValue  
    Spinnaker::Camera, 516

ColorTransformationValueSelector  
    Spinnaker::Camera, 516

ColorTransformationValueSelectorEnums  
    CameraDefs Class, 88

Combine  
    INode Interface, 302

Command

GVCP\_REQUEST\_HEADER, 823  
 CommandHeader  
   U3V\_EVENT\_MESSAGE, 1121  
 CommandId  
   U3V\_COMMAND\_HEADER, 1119  
 CommandNode, 724  
   Spinnaker::GenApi::CommandNode, 725, 726  
 CommandNode Class, 249  
   CCommandRef, 249  
 compare  
   Spinnaker::GenICam::gcstring, 808  
 Compatibility.h  
   FMT\_I64, 1259  
 compression  
   Spinnaker::TIFFOption, 1089  
 compressionLevel  
   Spinnaker::PNGOption, 1040  
 CompressionMethod  
   Spinnaker::TIFFOption, 1088  
 CompressionRatio  
   Spinnaker::Camera, 517  
 confidence  
   Chunk Data Inference Class, 164  
 ConfigureAdapter  
   AdapterConfig, 379  
 Connect  
   INodeMap Interface, 311, 312  
   Spinnaker::GenApi::NodeMap, 1027  
 Container Class, 250  
 Convert  
   Spinnaker::IImage, 864  
   Spinnaker::Image, 901, 902  
 Counter, 727  
   Spinnaker::GenApi::Counter, 728  
 Counter Class, 251  
 CounterDelay  
   Spinnaker::Camera, 517  
 CounterDuration  
   Spinnaker::Camera, 517  
 CounterEventActivation  
   Spinnaker::Camera, 517  
 CounterEventActivationEnums  
   CameraDefs Class, 89  
 CounterEventSource  
   Spinnaker::Camera, 517  
 CounterEventSourceEnums  
   CameraDefs Class, 89  
 CounterReset  
   Spinnaker::Camera, 517  
 CounterResetActivation  
   Spinnaker::Camera, 518  
 CounterResetActivationEnums  
   CameraDefs Class, 90  
 CounterResetSource  
   Spinnaker::Camera, 518  
 CounterResetSourceEnums  
   CameraDefs Class, 90  
 CounterSelector  
   Spinnaker::Camera, 518  
   CounterSelectorEnums  
     CameraDefs Class, 90  
 CounterStatus  
   Spinnaker::Camera, 518  
 CounterStatusEnums  
   CameraDefs Class, 91  
 CounterTriggerActivation  
   Spinnaker::Camera, 518  
 CounterTriggerActivationEnums  
   CameraDefs Class, 91  
 CounterTriggerSource  
   Spinnaker::Camera, 518  
 CounterTriggerSourceEnums  
   CameraDefs Class, 91  
 CounterValue  
   Spinnaker::Camera, 519  
 CounterValueAtReset  
   Spinnaker::Camera, 519  
 Create  
   Spinnaker::Image, 902, 903  
 CreateAlop  
   Spinnaker::ImageUtilityPolarization, 947  
 CreateDlop  
   Spinnaker::ImageUtilityPolarization, 947, 948  
 CreateEmptyNodeMap  
   Spinnaker::GenApi::CNodeMapFactory, 713  
 CreateGlareReduced  
   Spinnaker::ImageUtilityPolarization, 948, 949  
 CreateHeatmap  
   Spinnaker::ImageUtilityHeatmap, 942, 943  
 CreateNodeDataFromNodeMap  
   Spinnaker::GenApi::CNodeMapFactory, 713  
 CreateNodeMap  
   Spinnaker::GenApi::CNodeMapFactory, 713  
 CreateNormalized  
   Spinnaker::ImageUtility, 938–940  
 CreateScaled  
   Spinnaker::ImageUtility, 940  
 CreateShared  
   Spinnaker::Image, 903  
 CreateStokesS0  
   Spinnaker::ImageUtilityPolarization, 949  
 CreateStokesS1  
   Spinnaker::ImageUtilityPolarization, 950  
 CreateStokesS2  
   Spinnaker::ImageUtilityPolarization, 951  
 Custom  
   Types Enums, 372  
 CxpConnectionSelector  
   Spinnaker::Camera, 519  
 CxpConnectionTestErrorCount  
   Spinnaker::Camera, 519  
 CxpConnectionTestMode  
   Spinnaker::Camera, 519  
 CxpConnectionTestModeEnums  
   CameraDefs Class, 92  
 CxpConnectionTestPacketCount

Spinnaker::Camera, 519  
CxpLinkConfiguration  
    Spinnaker::Camera, 520  
CxpLinkConfigurationEnums  
    CameraDefs Class, 92  
CxpLinkConfigurationPreferred  
    Spinnaker::Camera, 520  
CxpLinkConfigurationPreferredEnums  
    CameraDefs Class, 93  
CxpLinkConfigurationStatus  
    Spinnaker::Camera, 520  
CxpLinkConfigurationStatusEnums  
    CameraDefs Class, 94  
CxpPoCxpAuto  
    Spinnaker::Camera, 520  
CxpPoCxpStatus  
    Spinnaker::Camera, 520  
CxpPoCxpStatusEnums  
    CameraDefs Class, 95  
CxpPoCxpTripReset  
    Spinnaker::Camera, 520  
CxpPoCxpTurnOff  
    Spinnaker::Camera, 521  
  
DCAM\_CHECKSUM, 751  
    CRCChecksum, 751  
DCAM\_CHUNK\_TRAILER, 751  
    ChunkID, 752  
    ChunkLength, 752  
    InverseChunkLength, 752  
DEPRECATED\_CLASS  
    AVI Recorder Class, 35  
DEPRECATED\_ENUM  
    Spinnaker Definitions, 200  
DEPRECATED\_FUNC  
    Spinnaker::lImage, 865  
    Spinnaker::Image, 904–906  
Data  
    GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID, 822  
    GVCP\_EVENTDATA\_REQUEST, 821  
DelInit  
    Spinnaker::CameraBase, 616  
    Spinnaker::ICameraBase, 830  
DecimationHorizontal  
    Spinnaker::Camera, 521  
DecimationHorizontalMode  
    Spinnaker::Camera, 521  
DecimationHorizontalModeEnums  
    CameraDefs Class, 96  
DecimationSelector  
    Spinnaker::Camera, 521  
DecimationSelectorEnums  
    CameraDefs Class, 96  
DecimationVertical  
    Spinnaker::Camera, 522  
DecimationVerticalMode  
    Spinnaker::Camera, 522  
DecimationVerticalModeEnums  
    CameraDefs Class, 96  
Decreasing  
    Types Enums, 373  
DeepCopy  
    Spinnaker::lImage, 864  
    Spinnaker::Image, 903  
DefectCorrectStaticEnable  
    Spinnaker::Camera, 522  
DefectCorrectionMode  
    Spinnaker::Camera, 522  
DefectCorrectionModeEnums  
    CameraDefs Class, 96  
DefectTableApply  
    Spinnaker::Camera, 523  
DefectTableCoordinateX  
    Spinnaker::Camera, 523  
DefectTableCoordinateY  
    Spinnaker::Camera, 523  
DefectTableFactoryRestore  
    Spinnaker::Camera, 523  
DefectTableIndex  
    Spinnaker::Camera, 524  
DefectTablePixelCount  
    Spinnaker::Camera, 524  
DefectTableSave  
    Spinnaker::Camera, 524  
Deinterlacing  
    Spinnaker::Camera, 524  
DeinterlacingEnums  
    CameraDefs Class, 97  
deleteFile  
    Spinnaker::GenApi::FileProtocolAdapter, 791  
DeliverEventMessage  
    Spinnaker::GenApi::CEventAdapter1394, 664  
    Spinnaker::GenApi::CEventAdapterGEV, 668  
    Spinnaker::GenApi::CEventAdapterU3V, 670  
DeliverMessage  
    Spinnaker::GenApi::CEventAdapter, 661  
    Spinnaker::GenApi::CEventAdapter1394, 664  
    Spinnaker::GenApi::CEventAdapterGEV, 668  
    Spinnaker::GenApi::CEventAdapterGeneric, 666  
    Spinnaker::GenApi::CEventAdapterU3V, 670  
Deregister  
    NodeCallback Class, 337  
DeregisterCallback  
    INode Interface, 302  
    Spinnaker::GenApi::Node, 1017  
Destroy  
    Spinnaker::GenApi::CNodeCallback, 706  
    Spinnaker::GenApi::Function\_NodeCallback, 804  
    Spinnaker::GenApi::Member\_NodeCallback, 1011  
    Spinnaker::GenApi::NodeMap, 1027  
DetachBuffer  
    Spinnaker::GenApi::CChunkAdapter, 640  
DetachChunk  
    Spinnaker::GenApi::CChunkPort, 653  
DetachEvent  
    Spinnaker::GenApi::CEventPort, 674

DetachNode  
     Spinnaker::GenApi::CEventPort, 674

DetachNodeMap  
     Spinnaker::GenApi::CChunkAdapter, 640  
     Spinnaker::GenApi::CEventAdapter, 662

DetachPort  
     Spinnaker::GenApi::CChunkPort, 653

DeviceAccessStatus  
     Spinnaker::TransportLayerDevice, 1092  
     Spinnaker::TransportLayerInterface, 1103

DeviceAccessStatusEnum  
     TransportLayerDfs Class, 210

DeviceAddress  
     Spinnaker::ActionCommandResult, 441

DeviceCharacterSet  
     Spinnaker::Camera, 524

DeviceCharacterSetEnums  
     CameraDfs Class, 97

DeviceClockFrequency  
     Spinnaker::Camera, 525

DeviceClockSelector  
     Spinnaker::Camera, 525

DeviceClockSelectorEnums  
     CameraDfs Class, 97

DeviceConnectionSelector  
     Spinnaker::Camera, 525

DeviceConnectionSpeed  
     Spinnaker::Camera, 525

DeviceConnectionStatus  
     Spinnaker::Camera, 525

DeviceConnectionStatusEnums  
     CameraDfs Class, 98

DeviceCount  
     Spinnaker::TransportLayerInterface, 1103

DeviceCurrentSpeed  
     Spinnaker::TransportLayerDevice, 1092

DeviceCurrentSpeedEnum  
     TransportLayerDfs Class, 211

DeviceDisplayName  
     Spinnaker::TransportLayerDevice, 1093

DeviceDriverVersion  
     Spinnaker::TransportLayerDevice, 1093

DeviceEndianessMechanism  
     Spinnaker::TransportLayerDevice, 1093

DeviceEndianessMechanismEnum  
     TransportLayerDfs Class, 211

DeviceEvent, 752  
     Spinnaker::DeviceEvent, 753

DeviceEvent Class, 166

DeviceEventChannelCount  
     Spinnaker::Camera, 525

DeviceFamilyName  
     Spinnaker::Camera, 526

DeviceFeaturePersistenceEnd  
     Spinnaker::Camera, 526

DeviceFeaturePersistenceStart  
     Spinnaker::Camera, 526

DeviceFirmwareVersion

    Spinnaker::Camera, 526

    DeviceGenCPVersionMajor  
         Spinnaker::Camera, 526

    DeviceGenCPVersionMinor  
         Spinnaker::Camera, 526

    DeviceID  
         Spinnaker::Camera, 527

    DeviceIndicatorMode  
         Spinnaker::Camera, 527

    DeviceIndicatorModeEnums  
         CameraDfs Class, 98

    DeviceInstanceld  
         Spinnaker::TransportLayerDevice, 1093

    DeviceIsUpdater  
         Spinnaker::TransportLayerDevice, 1093

    DeviceLinkBandwidthReserve  
         Spinnaker::Camera, 527

    DeviceLinkCommandTimeout  
         Spinnaker::Camera, 527

    DeviceLinkConnectionCount  
         Spinnaker::Camera, 527

    DeviceLinkCurrentThroughput  
         Spinnaker::Camera, 527

    DeviceLinkHeartbeatMode  
         Spinnaker::Camera, 528

    DeviceLinkHeartbeatModeEnums  
         CameraDfs Class, 98

    DeviceLinkHeartbeatTimeout  
         Spinnaker::Camera, 528

    DeviceLinkSelector  
         Spinnaker::Camera, 528

    DeviceLinkSpeed  
         Spinnaker::Camera, 528

    DeviceLinkThroughputLimit  
         Spinnaker::Camera, 528

    DeviceLinkThroughputLimitMode  
         Spinnaker::Camera, 529

    DeviceLinkThroughputLimitModeEnums  
         CameraDfs Class, 100

    DeviceLocation  
         Spinnaker::TransportLayerDevice, 1094

    DeviceManifestEntrySelector  
         Spinnaker::Camera, 529

    DeviceManifestPrimaryURL  
         Spinnaker::Camera, 529

    DeviceManifestSchemaMajorVersion  
         Spinnaker::Camera, 529

    DeviceManifestSchemaMinorVersion  
         Spinnaker::Camera, 529

    DeviceManifestSecondaryURL  
         Spinnaker::Camera, 530

    DeviceManifestXMLMajorVersion  
         Spinnaker::Camera, 530

    DeviceManifestXMLMinorVersion  
         Spinnaker::Camera, 530

DeviceManifestXMLSubMinorVersion  
    Spinnaker::Camera, 530

DeviceManufacturerInfo  
    Spinnaker::Camera, 530

DeviceMaxThroughput  
    Spinnaker::Camera, 530

DeviceModelName  
    Spinnaker::Camera, 531  
    Spinnaker::TransportLayerDevice, 1094  
    Spinnaker::TransportLayerInterface, 1103

DeviceMulticastMonitorMode  
    Spinnaker::TransportLayerDevice, 1094

DevicePowerSupplySelector  
    Spinnaker::Camera, 531

DevicePowerSupplySelectorEnums  
    CameraDefs Class, 100

DeviceRegistersCheck  
    Spinnaker::Camera, 531

DeviceRegistersEndianness  
    Spinnaker::Camera, 531

DeviceRegistersEndiannessEnums  
    CameraDefs Class, 100

DeviceRegistersStreamingEnd  
    Spinnaker::Camera, 531

DeviceRegistersStreamingStart  
    Spinnaker::Camera, 532

DeviceRegistersValid  
    Spinnaker::Camera, 532

DeviceReset  
    Spinnaker::Camera, 532

DeviceSFNCVersionMajor  
    Spinnaker::Camera, 533

DeviceSFNCVersionMinor  
    Spinnaker::Camera, 533

DeviceSFNCVersionSubMinor  
    Spinnaker::Camera, 533

DeviceScanType  
    Spinnaker::Camera, 532

DeviceScanTypeEnums  
    CameraDefs Class, 101

DeviceSelector  
    Spinnaker::TransportLayerInterface, 1103

DeviceSerialNumber  
    Spinnaker::Camera, 532  
    Spinnaker::TransportLayerDevice, 1094

DeviceSerialPortBaudRate  
    Spinnaker::Camera, 532

DeviceSerialPortBaudRateEnums  
    CameraDefs Class, 101

DeviceSerialPortSelector  
    Spinnaker::Camera, 533

DeviceSerialPortSelectorEnums  
    CameraDefs Class, 101

DeviceStreamChannelCount  
    Spinnaker::Camera, 533

DeviceStreamChannelEndianness  
    Spinnaker::Camera, 533

DeviceStreamChannelEndiannessEnums  
    CameraDefs Class, 101

DeviceStreamChannelLink  
    Spinnaker::Camera, 534

DeviceStreamChannelPacketSize  
    Spinnaker::Camera, 534

DeviceStreamChannelSelector  
    Spinnaker::Camera, 534

DeviceStreamChannelType  
    Spinnaker::Camera, 534

DeviceStreamChannelTypeEnums  
    CameraDefs Class, 102

DeviceTLType  
    Spinnaker::Camera, 535

DeviceTLTypeEnums  
    CameraDefs Class, 104

DeviceTLVersionMajor  
    Spinnaker::Camera, 535

DeviceTLVersionMinor  
    Spinnaker::Camera, 535

DeviceTLVersionSubMinor  
    Spinnaker::Camera, 535

DeviceTapGeometry  
    Spinnaker::Camera, 534

DeviceTapGeometryEnums  
    CameraDefs Class, 102

DeviceTemperature  
    Spinnaker::Camera, 534

DeviceTemperatureSelector  
    Spinnaker::Camera, 535

DeviceTemperatureSelectorEnums  
    CameraDefs Class, 103

DeviceType  
    Spinnaker::Camera, 536  
    Spinnaker::TransportLayerDevice, 1094

DeviceTypeEnum  
    TransportLayerDefs Class, 212

DeviceTypeEnums  
    CameraDefs Class, 104

DeviceU3VProtocol  
    Spinnaker::TransportLayerDevice, 1095

DeviceUnlock  
    Spinnaker::TransportLayerInterface, 1104

DeviceUpdateList  
    Spinnaker::TransportLayerInterface, 1104

DeviceUptime  
    Spinnaker::Camera, 536

DeviceUserID  
    Spinnaker::Camera, 536  
    Spinnaker::TransportLayerDevice, 1095

DeviceVendorName  
    Spinnaker::Camera, 536  
    Spinnaker::TransportLayerDevice, 1095  
    Spinnaker::TransportLayerInterface, 1104

DeviceVersion  
    Spinnaker::Camera, 536  
    Spinnaker::TransportLayerDevice, 1095

dhcpEnabled  
    AdapterConfig::AdapterInfo, 445

DisableAll  
     Spinnaker::IImageStatistics, 877  
     Spinnaker::ImageStatistics, 931

DiscoverMaxPacketSize  
     Spinnaker::CameraBase, 616  
     Spinnaker::ICameraBase, 830

doc/Doxxygen/spindocs/Licensing.dox, 1127

doc/Doxxygen/spindocs/MainPage.dox, 1127

DoesEnvironmentVariableExist  
     GCUtilities Utility, 271

double\_autovector\_t, 755  
     Spinnaker::GenApi::double\_autovector\_t, 756

EAccessMode  
     Types Enums, 368

EAccessModeClass, 758

ECacheUsage\_t  
     NodeMapFactory Class, 340

ECachingMode  
     Types Enums, 368

ECachingModeClass, 759

ECallbackType  
     NodeCallback Class, 337

EContentType\_t  
     NodeMapFactory Class, 341

EDisplayNotation  
     Types Enums, 368

EDisplayNotationClass, 760

EEndianess  
     Types Enums, 370

EEndianessClass, 761

EGenApiSchemaVersion  
     Types Enums, 370

EGenApiSchemaVersionClass, 762

EIncMode  
     Types Enums, 370

EInputDirection  
     Types Enums, 371

EInputDirectionClass, 763

EInterfaceType  
     Types Enums, 371

ELinkType  
     Types Enums, 371

ENamespace  
     Types Enums, 372

ENamespaceClass, 764

ERepresentation  
     Types Enums, 372

ERepresentationClass, 773

ESign  
     Types Enums, 372

ESignClass, 774

ESlope  
     Types Enums, 373

ESlopeClass, 775

EStandardNameSpace  
     Types Enums, 373

EStandardNameSpaceClass, 776

EVENT\_TIMEOUT\_INFINITE  
     Spinnaker Headers, 187

EVENT\_TIMEOUT\_NONE  
     Spinnaker Headers, 187

EVisibility  
     Types Enums, 373

EVisibilityClass, 781

EXMLValidation  
     Types Enums, 374

EXPAND\_TO\_STRINGISE  
     GCUtilities.h, 1298

EYesNo  
     Types Enums, 374

EYesNoClass, 788

EatComments  
     Spinnaker GenApi Classes, 236

empty  
     Spinnaker::GenICam::gcstring, 808

EnableAll  
     Spinnaker::IImageStatistics, 877  
     Spinnaker::ImageStatistics, 931

EnableGreyOnly  
     Spinnaker::IImageStatistics, 877  
     Spinnaker::ImageStatistics, 931

EnableHSOnly  
     Spinnaker::IImageStatistics, 877  
     Spinnaker::ImageStatistics, 932

EnableRGBOnly  
     Spinnaker::IImageStatistics, 878  
     Spinnaker::ImageStatistics, 932

EncoderDivider  
     Spinnaker::Camera, 536

EncoderMode  
     Spinnaker::Camera, 537

EncoderModeEnums  
     CameraDefs Class, 104

EncoderOutputMode  
     Spinnaker::Camera, 537

EncoderReset  
     Spinnaker::Camera, 537

EncoderResetActivation  
     Spinnaker::Camera, 537

EncoderResetActivationEnums  
     CameraDefs Class, 105

EncoderResetSource  
     Spinnaker::Camera, 537

EncoderResetSourceEnums  
     CameraDefs Class, 106

EncoderSelector  
     Spinnaker::Camera, 537

EncoderSelectorEnums  
     CameraDefs Class, 107

EncoderSourceAEnums  
     CameraDefs Class, 107

EncoderSourceBEnums  
     CameraDefs Class, 107

EncoderSourceA

Spinnaker::Camera, 538  
EncoderSourceB  
    Spinnaker::Camera, 538  
EncoderStatus  
    Spinnaker::Camera, 538  
EncoderStatusEnums  
    CameraDefs Class, 108  
EncoderTimeout  
    Spinnaker::Camera, 538  
EncoderValue  
    Spinnaker::Camera, 538  
EncoderValueAtReset  
    Spinnaker::Camera, 538  
EndAcquisition  
    Spinnaker::CameraBase, 617  
    Spinnaker::ICameraBase, 831  
EnumClasses Class, 252  
EnumEntryNode, 765  
    Spinnaker::GenApi::EnumEntryNode, 766  
EnumEntryNode Class, 254  
    CEnumEntryRef, 254  
EnumNode, 768  
    Spinnaker::GenApi::EnumNode, 770  
EnumNode Class, 255  
    CEnumerationRef, 255  
EnumNodeT Class, 256  
EnumerateGEVInterfaces  
    Spinnaker::TransportLayerSystem, 1118  
EnumerationCount  
    Spinnaker::Camera, 539  
ErrCode  
    AdapterConfigException, 443  
Error  
    Spinnaker Definitions, 194  
Event, 778  
    GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID, 822  
    GVCP\_EVENTDATA\_REQUEST, 821  
    Spinnaker::Event, 779  
Event Class, 167  
EventAcquisitionEnd  
    Spinnaker::Camera, 539  
EventAcquisitionEndFrameID  
    Spinnaker::Camera, 539  
EventAcquisitionEndTimestamp  
    Spinnaker::Camera, 539  
EventAcquisitionError  
    Spinnaker::Camera, 539  
EventAcquisitionErrorFrameID  
    Spinnaker::Camera, 539  
EventAcquisitionErrorTimestamp  
    Spinnaker::Camera, 540  
EventAcquisitionStart  
    Spinnaker::Camera, 540  
EventAcquisitionStartFrameID  
    Spinnaker::Camera, 540  
EventAcquisitionStartTimestamp  
    Spinnaker::Camera, 540  
EventAcquisitionTransferEnd  
    Spinnaker::Camera, 540  
EventAcquisitionTransferEndFrameID  
    Spinnaker::Camera, 540  
EventAcquisitionTransferEndTimestamp  
    Spinnaker::Camera, 541  
EventAcquisitionTransferStart  
    Spinnaker::Camera, 541  
EventAcquisitionTransferStartFrameID  
    Spinnaker::Camera, 541  
EventAcquisitionTransferStartTimestamp  
    Spinnaker::Camera, 541  
EventAcquisitionTrigger  
    Spinnaker::Camera, 541  
EventAcquisitionTriggerFrameID  
    Spinnaker::Camera, 541  
EventAcquisitionTriggerTimestamp  
    Spinnaker::Camera, 542  
EventActionLate  
    Spinnaker::Camera, 542  
EventActionLateFrameID  
    Spinnaker::Camera, 542  
EventActionLateTimestamp  
    Spinnaker::Camera, 542  
EventAdapter Class, 257  
EventAdapter1394 Class, 258  
EventAdapterGEV Class, 260  
EventAdapterGeneric Class, 259  
EventAdapterU3V Class, 261  
EventCounter0End  
    Spinnaker::Camera, 542  
EventCounter0EndFrameID  
    Spinnaker::Camera, 542  
EventCounter0EndTimestamp  
    Spinnaker::Camera, 543  
EventCounter0Start  
    Spinnaker::Camera, 543  
EventCounter0StartFrameID  
    Spinnaker::Camera, 543  
EventCounter0StartTimestamp  
    Spinnaker::Camera, 543  
EventCounter1End  
    Spinnaker::Camera, 543  
EventCounter1EndFrameID  
    Spinnaker::Camera, 543  
EventCounter1EndTimestamp  
    Spinnaker::Camera, 544  
EventCounter1Start  
    Spinnaker::Camera, 544  
EventCounter1StartFrameID  
    Spinnaker::Camera, 544  
EventCounter1StartTimestamp  
    Spinnaker::Camera, 544  
EventData  
    U3V\_EVENT\_MESSAGE, 1121  
EventEncoder0Restarted  
    Spinnaker::Camera, 544  
EventEncoder0RestartedFrameID

Spinnaker::Camera, 544  
 EventEncoder0RestartedTimestamp  
     Spinnaker::Camera, 545  
 EventEncoder0Stopped  
     Spinnaker::Camera, 545  
 EventEncoder0StoppedFrameID  
     Spinnaker::Camera, 545  
 EventEncoder0StoppedTimestamp  
     Spinnaker::Camera, 545  
 EventEncoder1Restarted  
     Spinnaker::Camera, 545  
 EventEncoder1RestartedFrameID  
     Spinnaker::Camera, 545  
 EventEncoder1RestartedTimestamp  
     Spinnaker::Camera, 546  
 EventEncoder1Stopped  
     Spinnaker::Camera, 546  
 EventEncoder1StoppedFrameID  
     Spinnaker::Camera, 546  
 EventEncoder1StoppedTimestamp  
     Spinnaker::Camera, 546  
 EventError  
     Spinnaker::Camera, 546  
 EventErrorCode  
     Spinnaker::Camera, 546  
 EventErrorFrameID  
     Spinnaker::Camera, 547  
 EventErrorTimestamp  
     Spinnaker::Camera, 547  
 EventExposureEnd  
     Spinnaker::Camera, 547  
 EventExposureEndFrameID  
     Spinnaker::Camera, 547  
 EventExposureEndTimestamp  
     Spinnaker::Camera, 547  
 EventExposureStart  
     Spinnaker::Camera, 547  
 EventExposureStartFrameID  
     Spinnaker::Camera, 548  
 EventExposureStartTimestamp  
     Spinnaker::Camera, 548  
 EventFrameBurstEnd  
     Spinnaker::Camera, 548  
 EventFrameBurstEndFrameID  
     Spinnaker::Camera, 548  
 EventFrameBurstEndTimestamp  
     Spinnaker::Camera, 548  
 EventFrameBurstStart  
     Spinnaker::Camera, 548  
 EventFrameBurstStartFrameID  
     Spinnaker::Camera, 549  
 EventFrameBurstStartTimestamp  
     Spinnaker::Camera, 549  
 EventFrameEnd  
     Spinnaker::Camera, 549  
 EventFrameEndFrameID  
     Spinnaker::Camera, 549  
 EventFrameEndTimestamp

Spinnaker::Camera, 549  
 EventFrameStart  
     Spinnaker::Camera, 549  
 EventFrameStartFrameID  
     Spinnaker::Camera, 550  
 EventFrameStartTimestamp  
     Spinnaker::Camera, 550  
 EventFrameTransferEnd  
     Spinnaker::Camera, 550  
 EventFrameTransferEndFrameID  
     Spinnaker::Camera, 550  
 EventFrameTransferEndTimestamp  
     Spinnaker::Camera, 550  
 EventFrameTransferStart  
     Spinnaker::Camera, 550  
 EventFrameTransferStartFrameID  
     Spinnaker::Camera, 551  
 EventFrameTransferStartTimestamp  
     Spinnaker::Camera, 551  
 EventFrameTrigger  
     Spinnaker::Camera, 551  
 EventFrameTriggerFrameID  
     Spinnaker::Camera, 551  
 EventFrameTriggerTimestamp  
     Spinnaker::Camera, 551  
 EventId  
     GVCP\_EVENT\_ITEM\_BASIC, 816  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 817  
     GVCP\_EVENT\_ITEM, 815  
     U3V\_EVENT\_DATA, 1120  
 EventLine0AnyEdge  
     Spinnaker::Camera, 551  
 EventLine0AnyEdgeFrameID  
     Spinnaker::Camera, 552  
 EventLine0AnyEdgeTimestamp  
     Spinnaker::Camera, 552  
 EventLine0FallingEdge  
     Spinnaker::Camera, 552  
 EventLine0FallingEdgeFrameID  
     Spinnaker::Camera, 552  
 EventLine0FallingEdgeTimestamp  
     Spinnaker::Camera, 552  
 EventLine0RisingEdge  
     Spinnaker::Camera, 552  
 EventLine0RisingEdgeFrameID  
     Spinnaker::Camera, 553  
 EventLine0RisingEdgeTimestamp  
     Spinnaker::Camera, 553  
 EventLine1AnyEdge  
     Spinnaker::Camera, 553  
 EventLine1AnyEdgeFrameID  
     Spinnaker::Camera, 553  
 EventLine1AnyEdgeTimestamp  
     Spinnaker::Camera, 553  
 EventLine1FallingEdge  
     Spinnaker::Camera, 553  
 EventLine1FallingEdgeFrameID  
     Spinnaker::Camera, 554

EventLine1FallingEdgeTimestamp  
    Spinnaker::Camera, 554

EventLine1RisingEdge  
    Spinnaker::Camera, 554

EventLine1RisingEdgeFrameID  
    Spinnaker::Camera, 554

EventLine1RisingEdgeTimestamp  
    Spinnaker::Camera, 554

EventLinkSpeedChange  
    Spinnaker::Camera, 554

EventLinkSpeedChangeFrameID  
    Spinnaker::Camera, 555

EventLinkSpeedChangeTimestamp  
    Spinnaker::Camera, 555

EventLinkTrigger0  
    Spinnaker::Camera, 555

EventLinkTrigger0FrameID  
    Spinnaker::Camera, 555

EventLinkTrigger0Timestamp  
    Spinnaker::Camera, 555

EventLinkTrigger1  
    Spinnaker::Camera, 555

EventLinkTrigger1FrameID  
    Spinnaker::Camera, 556

EventLinkTrigger1Timestamp  
    Spinnaker::Camera, 556

EventNotification  
    Spinnaker::Camera, 556

EventNotificationEnums  
    CameraDefs Class, 108

EventPort Class, 262

EventProcessor  
    Spinnaker::Event, 781

EventSelector  
    Spinnaker::Camera, 556

EventSelectorEnums  
    CameraDefs Class, 108

EventSequencerSetChange  
    Spinnaker::Camera, 556

EventSequencerSetChangeFrameID  
    Spinnaker::Camera, 556

EventSequencerSetChangeTimestamp  
    Spinnaker::Camera, 557

EventSerialData  
    Spinnaker::Camera, 557

EventSerialDataLength  
    Spinnaker::Camera, 557

EventSerialPortReceive  
    Spinnaker::Camera, 557

EventSerialPortReceiveTimestamp  
    Spinnaker::Camera, 557

EventSerialReceiveOverflow  
    Spinnaker::Camera, 557

EventStream0TransferBlockEnd  
    Spinnaker::Camera, 558

EventStream0TransferBlockEndFrameID  
    Spinnaker::Camera, 558

EventStream0TransferBlockEndTimestamp  
    Spinnaker::Camera, 558

EventStream0TransferBlockStart  
    Spinnaker::Camera, 558

EventStream0TransferBlockStartFrameID  
    Spinnaker::Camera, 558

EventStream0TransferBlockStartTimestamp  
    Spinnaker::Camera, 558

EventStream0TransferBlockTrigger  
    Spinnaker::Camera, 559

EventStream0TransferBlockTriggerFrameID  
    Spinnaker::Camera, 559

EventStream0TransferBlockTriggerTimestamp  
    Spinnaker::Camera, 559

EventStream0TransferBurstEnd  
    Spinnaker::Camera, 559

EventStream0TransferBurstEndFrameID  
    Spinnaker::Camera, 559

EventStream0TransferBurstEndTimestamp  
    Spinnaker::Camera, 559

EventStream0TransferBurstStart  
    Spinnaker::Camera, 560

EventStream0TransferBurstStartFrameID  
    Spinnaker::Camera, 560

EventStream0TransferBurstStartTimestamp  
    Spinnaker::Camera, 560

EventStream0TransferEnd  
    Spinnaker::Camera, 560

EventStream0TransferEndFrameID  
    Spinnaker::Camera, 560

EventStream0TransferEndTimestamp  
    Spinnaker::Camera, 560

EventStream0TransferOverflow  
    Spinnaker::Camera, 561

EventStream0TransferOverflowFrameID  
    Spinnaker::Camera, 561

EventStream0TransferOverflowTimestamp  
    Spinnaker::Camera, 561

EventStream0TransferPause  
    Spinnaker::Camera, 561

EventStream0TransferPauseFrameID  
    Spinnaker::Camera, 561

EventStream0TransferPauseTimestamp  
    Spinnaker::Camera, 561

EventStream0TransferResume  
    Spinnaker::Camera, 562

EventStream0TransferResumeFrameID  
    Spinnaker::Camera, 562

EventStream0TransferResumeTimestamp  
    Spinnaker::Camera, 562

EventStream0TransferStart  
    Spinnaker::Camera, 562

EventStream0TransferStartFrameID  
    Spinnaker::Camera, 562

EventStream0TransferStartTimestamp  
    Spinnaker::Camera, 562

EventTest  
    Spinnaker::Camera, 563

EventTestTimestamp

Spinnaker::Camera, 563  
 EventTimer0End  
     Spinnaker::Camera, 563  
 EventTimer0EndFrameID  
     Spinnaker::Camera, 563  
 EventTimer0EndTimestamp  
     Spinnaker::Camera, 563  
 EventTimer0Start  
     Spinnaker::Camera, 563  
 EventTimer0StartFrameID  
     Spinnaker::Camera, 564  
 EventTimer0StartTimestamp  
     Spinnaker::Camera, 564  
 EventTimer1End  
     Spinnaker::Camera, 564  
 EventTimer1EndFrameID  
     Spinnaker::Camera, 564  
 EventTimer1EndTimestamp  
     Spinnaker::Camera, 564  
 EventTimer1Start  
     Spinnaker::Camera, 564  
 EventTimer1StartFrameID  
     Spinnaker::Camera, 565  
 EventTimer1StartTimestamp  
     Spinnaker::Camera, 565  
 EventType  
     Spinnaker Definitions, 195  
 Exception, 783  
     Spinnaker::Exception, 784, 785  
 Exception Class, 168  
 Execute  
     Spinnaker::GenApi::CommandNode, 726  
 Expert  
     Types Enums, 374  
 ExposureActiveMode  
     Spinnaker::Camera, 565  
 ExposureActiveModeEnums  
     CameraDefs Class, 109  
 ExposureAuto  
     Spinnaker::Camera, 565  
 ExposureAutoEnums  
     CameraDefs Class, 109  
 ExposureMode  
     Spinnaker::Camera, 565  
 ExposureModeEnums  
     CameraDefs Class, 109  
 ExposureTime  
     Spinnaker::Camera, 565  
 ExposureTimeMode  
     Spinnaker::Camera, 566  
 ExposureTimeModeEnums  
     CameraDefs Class, 110  
 ExposureTimeSelector  
     Spinnaker::Camera, 566  
 ExposureTimeSelectorEnums  
     CameraDefs Class, 110  
 ExtractIndependentSubtree  
     INodeMapDyn Interface, 315  
 ExtractPolarQuadrant  
     Spinnaker::ImageUtilityPolarization, 952  
 ExtractPolarization  
     Spinnaker::Image, 906  
 ExtractSubtree  
     Spinnaker::GenApi::CNodeMapFactory, 713  
 FLIR\_SPINNAKER\_VERSION\_BUILD  
     System.h, 1389  
 FLIR\_SPINNAKER\_VERSION\_MAJOR  
     System.h, 1390  
 FLIR\_SPINNAKER\_VERSION\_MINOR  
     System.h, 1390  
 FLIR\_SPINNAKER\_VERSION\_TYPE  
     System.h, 1390  
 FMT\_I64  
     Compatibility.h, 1259  
 FactoryReset  
     Spinnaker::Camera, 566  
 FileAccessBuffer  
     Spinnaker::Camera, 566  
 FileAccessLength  
     Spinnaker::Camera, 566  
 FileAccessOffset  
     Spinnaker::Camera, 566  
 FileMode  
     Spinnaker::Camera, 567  
 FileModeEnums  
     CameraDefs Class, 111  
 FileOperationExecute  
     Spinnaker::Camera, 567  
 FileOperationResult  
     Spinnaker::Camera, 567  
 FileOperationSelector  
     Spinnaker::Camera, 567  
 FileOperationSelectorEnums  
     CameraDefs Class, 111  
 FileOperationStatus  
     Spinnaker::Camera, 567  
 FileOperationStatusEnums  
     CameraDefs Class, 111  
 FileProtocolAdapter, 789  
     Spinnaker::GenApi::FileProtocolAdapter, 789  
 FileSelector  
     Spinnaker::Camera, 568  
 FileSelectorEnums  
     CameraDefs Class, 112  
 FileSize  
     Spinnaker::Camera, 568  
 filebuf\_type  
     Spinnaker::GenApi::IDevFileStreamBase, 856  
     Spinnaker::GenApi::ODevFileStreamBase, 1035  
 Filestream Class, 263  
 FilterDriverStatus  
     Spinnaker::TransportLayerInterface, 1104  
 FilterDriverStatusEnum  
     TransportLayerDefs Class, 212  
 find  
     Spinnaker::GenICam::gcstring, 808, 809

find\_first\_not\_of  
    Spinnaker::GenICam::gcstring, 809

find\_first\_of  
    Spinnaker::GenICam::gcstring, 809

Flags  
    GVCP\_REQUEST\_HEADER, 823  
    U3V\_COMMAND\_HEADER, 1119

float32\_t  
    GCTypes Class, 268

float64\_t  
    GCTypes Class, 268

FloatNode, 793  
    Spinnaker::GenApi::FloatNode, 796

FloatNode Class, 264  
    CFloatRef, 264

FloatRegNode, 800  
    Spinnaker::GenApi::FloatRegNode, 801, 802

FloatRegNode Class, 265

FlushQueueAllDiscard  
    Spinnaker::IDataStream, 851

ForceIP  
    Spinnaker::CameraBase, 617  
    Spinnaker::ICameraBase, 831

frameRate  
    Spinnaker::Video::AVIOption, 452  
    Spinnaker::Video::H264Option, 824  
    Spinnaker::Video::MJPGOption, 1013

FromString  
    IValue Class, 333  
    Spinnaker::GenApi::EAccessModeClass, 758  
    Spinnaker::GenApi::ECachingModeClass, 759  
    Spinnaker::GenApi::EDisplayNotationClass, 760  
    Spinnaker::GenApi::EEndianessClass, 761  
    Spinnaker::GenApi::EGenApiSchemaVersion←  
        Class, 762  
    Spinnaker::GenApi::EInputDirectionClass, 763  
    Spinnaker::GenApi::ENamespaceClass, 764  
    Spinnaker::GenApi::ERepresentationClass, 773  
    Spinnaker::GenApi::ESignClass, 775  
    Spinnaker::GenApi::ESlopeClass, 776  
    Spinnaker::GenApi::ESTandardNameSpaceClass,  
        777  
    Spinnaker::GenApi::EVisibilityClass, 782  
    Spinnaker::GenApi::EYesNoClass, 788  
    Spinnaker::GenApi::ValueNode, 1124

Function\_NodeCallback  
    Spinnaker::GenApi::Function\_NodeCallback, 804

Function\_NodeCallback< Function >, 803

GC\_COUNTOF  
    GCUtillities.h, 1298

GC\_INT32\_MAX  
    GCTypes.h, 1292

GC\_INT32\_MIN  
    GCTypes.h, 1292

GC\_INT64\_MAX  
    GCTypes.h, 1292

GC\_INT64\_MIN  
    GCTypes.h, 1292

GC\_INT8\_MAX  
    GCTypes.h, 1293

GC\_INT8\_MIN  
    GCTypes.h, 1293

GC\_UINT32\_MAX  
    GCTypes.h, 1293

GC\_UINT64\_MAX  
    GCTypes.h, 1293

GC\_UINT8\_MAX  
    GCTypes.h, 1293

GCSTRING\_NPOS  
    GCString.h, 1288

GCString Class, 266  
    GCString.h  
        GCSTRING\_NPOS, 1288  
        operator<<, 1288  
        operator>>, 1289

GCSynch Class, 267

GCTypes Class, 268  
    float32\_t, 268  
    float64\_t, 268

GCTypes.h  
    \_\_STDC\_CONSTANT\_MACROS, 1292  
    \_\_STDC\_LIMIT\_MACROS, 1292

GC\_INT32\_MAX, 1292

GC\_INT32\_MIN, 1292

GC\_INT64\_MAX, 1292

GC\_INT64\_MIN, 1292

GC\_INT8\_MAX, 1293

GC\_INT8\_MIN, 1293

GC\_UINT32\_MAX, 1293

GC\_UINT64\_MAX, 1293

GC\_UINT8\_MAX, 1293

GCUtillities Utility, 270  
    DoesEnvironmentVariableExist, 271  
    GetFiles, 271  
    GetGenICamCLProtocolFolder, 272  
    GetGenICamCacheFolder, 271  
    GetGenICamLogConfig, 272  
    GetModulePathFromFunction, 272  
    GetValueOfEnvironmentVariable, 272, 273

INTEGRAL\_CAST2, 273

INTEGRAL\_CAST, 273

ReplaceEnvironmentVariables, 273

SetGenICamCLProtocolFolder, 274

SetGenICamCacheFolder, 273

SetGenICamLogConfig, 274

Tokenize, 274

UrlDecode, 274

UrlEncode, 275

GCUtillities.h  
    \_\_TO\_STRING, 1297  
    \_\_ERR\_\_, 1297  
    \_\_LINE\_STR\_\_, 1297  
    \_\_LOCATION\_\_, 1297  
    \_\_OUTPUT\_FORMATER\_\_, 1297  
    \_\_TODO\_\_, 1297  
    \_\_WARN\_\_, 1297

EXPAND\_TO\_STRINGISE, 1298  
 GC\_COUNTOF, 1298  
 GENICAM\_DEPRECATED, 1298  
 GENICAM\_UNUSED, 1298  
 USE\_TEMP\_CACHE\_FILE, 1298  
 GENCP\_COMMAND\_HEADER\_SIZE  
     Spinnaker::GenApi, 437  
 GENCP\_EVENT\_BASIC\_SIZE  
     Spinnaker::GenApi, 437  
 GENCP\_EVENT\_CMD\_ID  
     Spinnaker::GenApi, 437  
 GENICAM\_DEPRECATED  
     GCUtilities.h, 1298  
 GENICAM\_UNUSED  
     GCUtilities.h, 1298  
 GUIXMLLocation  
     Spinnaker::TransportLayerDevice, 1099  
 GUIXMLLocationEnum  
     TransportLayerDfs Class, 213  
 GUIXMLPath  
     Spinnaker::TransportLayerDevice, 1099  
 GVCP\_CHUNK\_TRAILER, 814  
     ChunkID, 814  
     ChunkLength, 814  
 GVCP\_EVENT\_ITEM\_BASIC, 816  
     EventId, 816  
     ReservedOrEventSize, 816  
 GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 817  
     BlockId, 817  
     BlockId64High, 817  
     BlockId64Low, 817  
     EventId, 817  
     ReservedOrEventSize, 817  
     StreamChannelId, 818  
     TimestampHigh, 818  
     TimestampLow, 818  
 GVCP\_EVENT\_ITEM, 815  
     BlockId, 815  
     EventId, 815  
     ReservedOrEventSize, 815  
     StreamChannelId, 815  
     TimestampHigh, 815  
     TimestampLow, 816  
 GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, 819  
     Header, 820  
     Items, 820  
 GVCP\_EVENT\_REQUEST, 818  
     Header, 819  
     Items, 819  
 GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID, 821  
     Data, 822  
     Event, 822  
     Header, 822  
 GVCP\_EVENTDATA\_REQUEST, 820  
     Data, 821  
     Event, 821  
     Header, 821  
 GVCP\_MESSAGE\_TAGS

Spinnaker::GenApi, 436  
 GVCP\_REQUEST\_HEADER, 822  
     Command, 823  
     Flags, 823  
     Length, 823  
     Magic, 823  
     ReqId, 823  
 Gain  
     Spinnaker::Camera, 568  
 GainAuto  
     Spinnaker::Camera, 568  
 GainAutoBalance  
     Spinnaker::Camera, 568  
 GainAutoBalanceEnums  
     CameraDefs Class, 112  
 GainAutoEnums  
     CameraDefs Class, 112  
 GainSelector  
     Spinnaker::Camera, 569  
 GainSelectorEnums  
     CameraDefs Class, 113  
 Gamma  
     Spinnaker::Camera, 569  
 GammaEnable  
     Spinnaker::Camera, 569  
 gateway  
     AdapterConfig::IpInfo, 985  
 gcstring, 805  
     Spinnaker::GenICam::gcstring, 806  
 GenICamXMLLocation  
     Spinnaker::TransportLayerDevice, 1095  
 GenICamXMLLocationEnum  
     TransportLayerDfs Class, 212  
 GenICamXMLPath  
     Spinnaker::TransportLayerDevice, 1095  
 Get  
     IRegister Interfaces, 326  
     Spinnaker::GenApi::RegisterNode, 1055  
 get  
     Spinnaker::BasePtr, 454  
 GetAccessMode  
     Spinnaker::CameraBase, 617  
     Spinnaker::GenApi::CChunkPort, 653  
     Spinnaker::GenApi::CEventPort, 674  
     Spinnaker::GenApi::CPortImpl, 736  
     Spinnaker::GenApi::CRegisterPortImpl, 742  
     Spinnaker::GenApi::CTestPortStruct, 749  
     Spinnaker::GenApi::Node, 1017  
     Spinnaker::GenApi::PortRecorder, 1048  
     Spinnaker::ICameraBase, 831  
 GetAddress  
     IRegister Interfaces, 327  
     Spinnaker::GenApi::RegisterNode, 1055  
 GetAlias  
     INode Interface, 303  
     Spinnaker::GenApi::Node, 1017  
 GetAuto10GDesc  
     AdapterConfig, 379

GetAutoGigabitDesc  
    AdapterConfig, 379

GetAutoStartGateway  
    AdapterConfig, 379

GetAutoStartIp  
    AdapterConfig, 379

GetAutoSubnetMask  
    AdapterConfig, 379

GetAutoSubnetMaskLength  
    AdapterConfig, 379

GetBitsPerPixel  
    Spinnaker::IImage, 865  
    Spinnaker::Image, 907

GetBlackLevel  
    Spinnaker::ChunkData, 689  
    Spinnaker::IChunkData, 842

GetBoxAt  
    Chunk Data Inference Class, 161

GetBoxCount  
    Chunk Data Inference Class, 161

GetBoxSize  
    Chunk Data Inference Class, 161

getBufSize  
    Spinnaker::GenApi::FileProtocolAdapter, 791

GetBufferChunkData  
    Spinnaker::IDataStream, 851

GetBufferInfoBool8Type  
    Spinnaker::IDataStream, 851

GetBufferInfoPtrType  
    Spinnaker::IDataStream, 851

GetBufferInfoSizeType  
    Spinnaker::IDataStream, 851

GetBufferInfoUInt64Type  
    Spinnaker::IDataStream, 851

GetBufferOwnership  
    Spinnaker::CameraBase, 618  
    Spinnaker::ICameraBase, 831

GetBufferSize  
    Spinnaker::IImage, 865  
    Spinnaker::Image, 907

GetBuildDate  
    Spinnaker::Exception, 786

GetBuildTime  
    Spinnaker::Exception, 786

GetByIndex  
    Spinnaker::CameraList, 631  
    Spinnaker::ICameraList, 838  
    Spinnaker::IInterfaceList, 890  
    Spinnaker::InterfaceList, 976

GetBySerial  
    Spinnaker::CameraList, 631  
    Spinnaker::ICameraList, 839

GetCRC  
    Spinnaker::ChunkData, 689  
    Spinnaker::IChunkData, 842

GetCachingMode  
    INode Interface, 303  
    Spinnaker::GenApi::Node, 1017

GetCallbackType  
    Spinnaker::GenApi::CNodeCallback, 706

GetCameras  
    Spinnaker::IInterface, 882  
    Spinnaker::ISystem, 990  
    Spinnaker::Interface, 965  
    Spinnaker::System, 1075

GetCastAlias  
    INode Interface, 303  
    Spinnaker::GenApi::Node, 1018

GetCategoryName  
    Spinnaker::LoggingEventData, 1006

GetChannelStatus  
    Spinnaker::IImageStatistics, 878  
    Spinnaker::ImageStatistics, 932

GetChildren  
    INode Interface, 303  
    Spinnaker::GenApi::Node, 1018

GetChunkData  
    Spinnaker::IImage, 866  
    Spinnaker::Image, 907

GetChunkIDLength  
    Spinnaker::GenApi::CChunkPort, 653

GetChunkID  
    Spinnaker::GenApi::PortNode, 1043

GetChunkLayoutId  
    Spinnaker::IImage, 866  
    Spinnaker::Image, 907

GetColorProcessing  
    Spinnaker::IImage, 866  
    Spinnaker::Image, 908

GetConfigLogFileName  
    AdapterConfig, 380

GetCookie  
    IPortRecorder Interface, 324  
    Spinnaker::GenApi::CPortWriteList, 739

GetCounterValue  
    Spinnaker::ChunkData, 689  
    Spinnaker::IChunkData, 842

GetCurrentEntry  
    IEnumerator Interface, 289  
    Spinnaker::GenApi::CEnumerationTRef, 657  
    Spinnaker::GenApi::EnumNode, 770

GetData  
    Spinnaker::IImage, 866  
    Spinnaker::Image, 908

GetDataAbsoluteMax  
    Spinnaker::IImage, 866  
    Spinnaker::Image, 908

GetDataAbsoluteMin  
    Spinnaker::IImage, 866  
    Spinnaker::Image, 909

GetDefaultColorProcessing  
    Spinnaker::Image, 909

GetDescription  
    INode Interface, 304  
    Spinnaker::GenApi::Node, 1018

GetDeviceEventId

Spinnaker::DeviceEvent, 754  
 Spinnaker::IDeviceEvent, 861  
 GetDeviceEventName  
     Spinnaker::DeviceEvent, 754  
     Spinnaker::IDeviceEvent, 861  
 GetDeviceName  
     INodeMap Interface, 312  
     Spinnaker::GenApi::Node, 1018  
     Spinnaker::GenApi::NodeMap, 1028  
 GetDeviceNodeMap  
     Spinnaker::IDataStream, 852  
 GetDeviceVersion  
     IDeviceInfo Interface, 284  
     Spinnaker::GenApi::NodeMap, 1028  
 GetDisplayName  
     INode Interface, 304  
     Spinnaker::GenApi::Node, 1018  
 GetDisplayNotation  
     IFloat Interface, 295  
     Spinnaker::GenApi::FloatNode, 796  
 GetDisplayPrecision  
     IFloat Interface, 295  
     Spinnaker::GenApi::FloatNode, 796  
 GetDocuURL  
     INode Interface, 304  
     Spinnaker::GenApi::Node, 1019  
 GetEncoderValue  
     Spinnaker::ChunkData, 689  
     Spinnaker::IChunkData, 842  
 GetEntries  
     IEnumerator Interface, 289  
     Spinnaker::GenApi::EnumNode, 770  
 GetEntry  
     IEnumerator Interface, 290  
     IEnumeratorT Interface, 292  
     Spinnaker::GenApi::CEnumerationTRef, 658  
     Spinnaker::GenApi::EnumNode, 771  
 GetEntryByName  
     IEnumerator Interface, 290  
     Spinnaker::GenApi::EnumNode, 771  
 GetEnumAlias  
     Spinnaker::GenApi::CFloatPtr, 680  
     Spinnaker::GenApi::FloatNode, 796  
 GetEnumerationLogFileName  
     AdapterConfig, 380  
 GetError  
     Spinnaker::Exception, 786  
 GetErrorMessage  
     SpinUpdate.h, 1385  
     Spinnaker::Exception, 786  
 GetEventIDLength  
     Spinnaker::GenApi::CEventPort, 674  
 GetEventID  
     INode Interface, 304  
     Spinnaker::GenApi::Node, 1019  
 GetEventPayloadData  
     Spinnaker::Event, 779  
 GetEventPayloadDataSize  
     Spinnaker::Event, 779  
 GetEventType  
     Spinnaker::Event, 780  
 GetExposureEndLineStatusAll  
     Spinnaker::ChunkData, 690  
     Spinnaker::IChunkData, 843  
 GetExposureTime  
     Spinnaker::ChunkData, 690  
     Spinnaker::IChunkData, 843  
 GetFeatureBagHandle  
     Spinnaker::GenApi::CFeatureBag, 677  
 GetFeatures  
     Spinnaker::GenApi::CategoryNode, 637  
 GetFileName  
     Spinnaker::Exception, 786  
 GetFiles  
     GCUtilities Utility, 271  
 GetFloatAlias  
     Spinnaker::GenApi::IntegerNode, 960  
 GetFrameID  
     Spinnaker::ChunkData, 690  
     Spinnaker::IChunkData, 843  
     Spinnaker::IImage, 867  
     Spinnaker::Image, 909  
 GetFullErrorMessage  
     Spinnaker::Exception, 786  
 GetFunctionName  
     Spinnaker::Exception, 786  
 GetGain  
     Spinnaker::ChunkData, 690  
     Spinnaker::IChunkData, 843  
 GetGatewayAddress  
     AdapterConfig, 380  
 GetGenApiVersion  
     IDeviceInfo Interface, 284  
     Spinnaker::GenApi::NodeMap, 1028  
 GetGenICamCLProtocolFolder  
     GCUtilities Utility, 272  
 GetGenICamCacheFolder  
     GCUtilities Utility, 271  
 GetGenICamLogConfig  
     GCUtilities Utility, 272  
 GetGuiXml  
     Spinnaker::CameraBase, 618  
     Spinnaker::ICameraBase, 831  
 GetHeatmapColorGradient  
     Spinnaker::ImageUtilityHeatmap, 943  
 GetHeatmapRange  
     Spinnaker::ImageUtilityHeatmap, 944  
 GetHeight  
     Spinnaker::ChunkData, 691  
     Spinnaker::IChunkData, 843  
     Spinnaker::IImage, 867  
     Spinnaker::Image, 910  
 GetHistogram  
     Spinnaker::IImageStatistics, 878  
     Spinnaker::ImageStatistics, 932  
 GetID

Spinnaker::IImage, 867  
Spinnaker::Image, 910  
GetImage  
    Spinnaker::ChunkData, 691  
    Spinnaker::IChunkData, 843  
GetImageData  
    Spinnaker::IImage, 867  
    Spinnaker::Image, 910  
GetImageSize  
    Spinnaker::IImage, 867  
    Spinnaker::Image, 910  
GetImageStatus  
    Spinnaker::IImage, 867  
    Spinnaker::Image, 911  
GetImageStatusDescription  
    Spinnaker::Image, 911  
GetInc  
    IFloat Interface, 295  
    Spinnaker::GenApi::FloatNode, 796  
    Spinnaker::GenApi::IntegerNode, 961  
GetIncMode  
    IFloat Interface, 295  
    Spinnaker::GenApi::FloatNode, 797  
    Spinnaker::GenApi::IntegerNode, 961  
GetInferenceBoundingBoxResult  
    Spinnaker::ChunkData, 691  
    Spinnaker::IChunkData, 844  
GetInferenceConfidence  
    Spinnaker::ChunkData, 691  
    Spinnaker::IChunkData, 844  
GetInferenceResult  
    Spinnaker::ChunkData, 692  
    Spinnaker::IChunkData, 844  
GetInstance  
    Spinnaker::System, 1076  
GetIntAlias  
    Spinnaker::GenApi::CFloatPtr, 680  
    Spinnaker::GenApi::FloatNode, 797  
GetIntValue  
    IEnumeration Interface, 290  
    Spinnaker::GenApi::EnumNode, 771  
GetInterfaceName  
    Pointer Class, 349  
GetInterfaces  
    Spinnaker::ISystem, 990  
    Spinnaker::System, 1076  
GetLength  
    IRegister Interfaces, 327  
    Spinnaker::GenApi::RegisterNode, 1056  
GetLibraryVersion  
    Spinnaker::ISystem, 990  
    Spinnaker::System, 1076  
GetLineNumber  
    Spinnaker::Exception, 787  
GetLinePitch  
    Spinnaker::ChunkData, 692  
    Spinnaker::IChunkData, 844  
GetLineStatusAll  
    Spinnaker::ChunkData, 692  
    Spinnaker::IChunkData, 844  
    Spinnaker::ChunkData, 692  
    Spinnaker::IChunkData, 844  
GetListOfValidValues  
    IFloat Interface, 295  
    Spinnaker::GenApi::FloatNode, 797  
    Spinnaker::GenApi::IntegerNode, 961  
GetLock  
    INodeMap Interface, 312  
    Spinnaker::GenApi::NodeMap, 1028  
    Spinnaker::GenCam::LockableObject, 1002  
GetLogMessage  
    Spinnaker::LoggingEventData, 1006  
GetLoggingEventPriorityLevel  
    Spinnaker::ISystem, 990  
    Spinnaker::System, 1077  
GetMax  
    IFloat Interface, 295  
    Spinnaker::GenApi::FloatNode, 797  
    Spinnaker::GenApi::IntegerNode, 961  
GetMaxIpAddress  
    AdapterConfig, 380  
GetMaxLength  
    IString Class, 332  
    Spinnaker::GenApi::StringNode, 1068  
GetMean  
    Spinnaker::IImageStatistics, 878  
    Spinnaker::ImageStatistics, 933  
GetMin  
    IFloat Interface, 296  
    Spinnaker::GenApi::FloatNode, 797  
    Spinnaker::GenApi::IntegerNode, 961  
GetMinIpAddress  
    AdapterConfig, 380  
GetmodelName  
    Spinnaker::GenApi::NodeMap, 1028  
GetModulePathFromFunction  
    GCUtilities Utility, 272  
GetNDC  
    Spinnaker::LoggingEventData, 1006  
GetName  
    Spinnaker::GenApi::Node, 1019  
GetNameSpace  
    INode Interface, 304  
    Spinnaker::GenApi::Node, 1019  
GetNextImage  
    Spinnaker::CameraBase, 618  
    Spinnaker::ICameraBase, 831  
    Spinnaker::IDataStream, 852  
GetNextImageInternal  
    Spinnaker::IDataStream, 852  
GetNode  
    INodeMap Interface, 312  
    Spinnaker::GenApi::CNodeCallback, 707  
    Spinnaker::GenApi::NodeMap, 1028  
    Spinnaker::GenApi::ValueNode, 1124  
GetNodeHandle  
    Spinnaker::GenApi::Node, 1019  
GetNodeMap

INode Interface, 304  
 Spinnaker::CameraBase, 619  
 Spinnaker::GenApi::Node, 1019  
 Spinnaker::ICameraBase, 832  
 Spinnaker::IDataStream, 852  
 GetNodeMapHandle  
     Spinnaker::GenApi::NodeMap, 1029  
 GetNodeStatistics  
     Spinnaker::GenApi::CNodeMapFactory, 714  
 GetNodes  
     Spinnaker::GenApi::NodeMap, 1029  
 GetNumChannels  
     Spinnaker::IImage, 868  
     Spinnaker::Image, 911  
 GetNumDataStreams  
     Spinnaker::CameraBase, 619  
     Spinnaker::ICameraBase, 832  
 GetNumImagesInUse  
     Spinnaker::CameraBase, 620  
     Spinnaker::ICameraBase, 832  
     Spinnaker::IDataStream, 852  
 GetNumNodes  
     INode Map Interface, 312  
     Spinnaker::GenApi::NodeMap, 1029  
 GetNumPixelValues  
     Spinnaker::IImageStatistics, 878  
     Spinnaker::ImageStatistics, 933  
 GetNumReads  
     Spinnaker::GenApi::CTestPortStruct, 749  
 GetNumWrites  
     Spinnaker::GenApi::CTestPortStruct, 749  
 GetNumericValue  
     IEnumEntry Interface, 287  
     Spinnaker::GenApi::EnumEntryNode, 766  
 GetOffsetX  
     Spinnaker::ChunkData, 692  
     Spinnaker::IChunkData, 844  
 GetOffsetY  
     Spinnaker::ChunkData, 693  
     Spinnaker::IChunkData, 845  
 GetParamStr  
     AdapterConfigException, 443  
 GetParents  
     INode Interface, 304  
     Spinnaker::GenApi::Node, 1019  
 GetPartSelector  
     Spinnaker::ChunkData, 693  
     Spinnaker::IChunkData, 845  
 GetPayloadType  
     Spinnaker::IImage, 868  
     Spinnaker::Image, 911  
 GetPixelDynamicRangeMax  
     Spinnaker::ChunkData, 693  
     Spinnaker::IChunkData, 845  
 GetPixelDynamicRangeMin  
     Spinnaker::ChunkData, 693  
     Spinnaker::IChunkData, 845  
 GetPixelFormat  
     Spinnaker::IImage, 868  
     Spinnaker::Image, 912  
 GetPixelFormatIntType  
     Spinnaker::IImage, 868  
     Spinnaker::Image, 912  
 GetPixelFormatName  
     Spinnaker::IImage, 868  
     Spinnaker::Image, 912  
 GetPixelValueRange  
     Spinnaker::IImageStatistics, 879  
     Spinnaker::ImageStatistics, 933  
 GetPolarizationAlgorithm  
     Spinnaker::Image, 913  
 GetPolarizationValues  
     Spinnaker::Image, 913  
 GetPollingTime  
     INode Interface, 305  
     Spinnaker::GenApi::Node, 1020  
 GetPort  
     Spinnaker::IDataStream, 852  
 GetPortHandle  
     Spinnaker::GenApi::PortNode, 1044  
 GetPortReplayHandle  
     Spinnaker::GenApi::PortReplay, 1051  
 GetPortWriteListHandle  
     Spinnaker::GenApi::CPortWriteList, 739  
 GetPrincipalInterfaceType  
     INode Interface, 305  
     Spinnaker::GenApi::CChunkPort, 653  
     Spinnaker::GenApi::CEventPort, 674  
     Spinnaker::GenApi::CTestPortStruct, 749  
     Spinnaker::GenApi::Node, 1020  
 GetPriority  
     Spinnaker::LoggingEventData, 1007  
 GetPriorityName  
     Spinnaker::LoggingEventData, 1007  
 GetPrivateData  
     Spinnaker::IImage, 868  
     Spinnaker::Image, 913  
 GetProductGuid  
     IDeviceInfo Interface, 285  
     Spinnaker::GenApi::NodeMap, 1029  
 GetProperty  
     INode Interface, 305  
     Spinnaker::GenApi::Node, 1020  
 GetPropertyNames  
     INode Interface, 305  
     Spinnaker::GenApi::Node, 1020  
 GetRange  
     Spinnaker::IImageStatistics, 879  
     Spinnaker::ImageStatistics, 934  
 GetRepresentation  
     IFloat Interface, 296  
     Spinnaker::GenApi::FloatNode, 797  
     Spinnaker::GenApi::IntegerNode, 961  
 GetScan3dAxisMax  
     Spinnaker::ChunkData, 694  
     Spinnaker::IChunkData, 845

GetScan3dAxisMin  
    Spinnaker::ChunkData, 694  
    Spinnaker::IChunkData, 845

GetScan3dCoordinateOffset  
    Spinnaker::ChunkData, 694  
    Spinnaker::IChunkData, 846

GetScan3dCoordinateReferenceValue  
    Spinnaker::ChunkData, 694  
    Spinnaker::IChunkData, 846

GetScan3dCoordinateScale  
    Spinnaker::ChunkData, 695  
    Spinnaker::IChunkData, 846

GetScan3dInvalidHeaderValue  
    Spinnaker::ChunkData, 695  
    Spinnaker::IChunkData, 846

GetScan3dTransformValue  
    Spinnaker::ChunkData, 695  
    Spinnaker::IChunkData, 846

GetScanLineSelector  
    Spinnaker::ChunkData, 695  
    Spinnaker::IChunkData, 846

GetSchemaVersion  
    IDeviceInfo Interface, 285  
    Spinnaker::GenApi::NodeMap, 1029

GetSelectedFeatures  
    ISelector Interface, 328  
    Spinnaker::GenApi::Node, 1020

GetSelectingFeatures  
    ISelector Interface, 328  
    Spinnaker::GenApi::Node, 1020

GetSelectorList  
    ISelectorDigit Interface, 329  
    Spinnaker::GenApi::CSelectorSet, 746

GetSequencerSetActive  
    Spinnaker::ChunkData, 696  
    Spinnaker::IChunkData, 847

GetSerialDataLength  
    Spinnaker::ChunkData, 696  
    Spinnaker::IChunkData, 847

GetSize  
    Spinnaker::CameraList, 632  
    Spinnaker::ICameraList, 839  
    Spinnaker::IInterfaceList, 890  
    Spinnaker::InterfaceList, 976

GetStandardNameSpace  
    IDeviceInfo Interface, 285  
    Spinnaker::GenApi::NodeMap, 1029

GetStatistics  
    Spinnaker::IImageStatistics, 879  
    Spinnaker::ImageStatistics, 934

GetStreamChannelID  
    Spinnaker::ChunkData, 696  
    Spinnaker::IChunkData, 847

GetStreamType  
    Spinnaker::IDataStream, 852

GetStride  
    Spinnaker::IImage, 869  
    Spinnaker::Image, 914

GetSubnetMaskLength  
    AdapterConfig, 380

GetSupportedSchemaVersions  
    INodeMapDyn Interface, 315  
    Spinnaker::GenApi::CNodeMapFactory, 714  
    Spinnaker::GenApi::NodeMap, 1030

GetSwapEndianess  
    IPortConstruct Interface, 323  
    Spinnaker::GenApi::CChunkPort, 653  
    Spinnaker::GenApi::CEventPort, 674  
    Spinnaker::GenApi::CPortImpl, 736  
    Spinnaker::GenApi::PortNode, 1044

GetSymbolic  
    IEnumEntry Interface, 287  
    Spinnaker::GenApi::EnumEntryNode, 767

GetSymbolics  
    Spinnaker::GenApi::EnumNode, 771

GetTLDeviceNodeMap  
    Spinnaker::CameraBase, 620  
    Spinnaker::ICameraBase, 832

GetTLNodemap  
    Spinnaker::IInterface, 882  
    Spinnaker::ISystem, 990  
    Spinnaker::Interface, 966  
    Spinnaker::System, 1077

GetTLPayloadType  
    Spinnaker::IImage, 869  
    Spinnaker::Image, 914

GetTLPixelFormat  
    Spinnaker::IImage, 869  
    Spinnaker::Image, 915

GetTLPixelFormatNamespace  
    Spinnaker::IImage, 869  
    Spinnaker::Image, 915

GetTLStreamNodeMap  
    Spinnaker::CameraBase, 620  
    Spinnaker::ICameraBase, 832

GetThreadName  
    Spinnaker::LoggingEventData, 1007

GetTimeStamp  
    Spinnaker::IImage, 869  
    Spinnaker::Image, 914

GetTimerValue  
    Spinnaker::ChunkData, 696  
    Spinnaker::IChunkData, 847

GetTimestamp  
    Spinnaker::ChunkData, 697  
    Spinnaker::IChunkData, 847  
    Spinnaker::LoggingEventData, 1007

GetTimestampLatchValue  
    Spinnaker::ChunkData, 697  
    Spinnaker::IChunkData, 847

GetToolTip  
    IDeviceInfo Interface, 285  
    Spinnaker::GenApi::Node, 1021  
    Spinnaker::GenApi::NodeMap, 1030

GetTransferBlockID  
    Spinnaker::ChunkData, 697

Spinnaker::IChunkData, 848  
 GetTransferQueueCurrentBlockCount  
     Spinnaker::ChunkData, 697  
     Spinnaker::IChunkData, 848  
 GetUniqueId  
     Spinnaker::CameraBase, 621  
     Spinnaker::ICameraBase, 832  
 GetUnit  
     IFloat Interface, 296  
     Spinnaker::GenApi::FloatNode, 798  
     Spinnaker::GenApi::IntegerNode, 962  
 GetUserBufferCount  
     Spinnaker::CameraBase, 621  
     Spinnaker::ICameraBase, 833  
 GetUserBufferSize  
     Spinnaker::CameraBase, 621  
     Spinnaker::ICameraBase, 833  
 GetUserBufferTotalSize  
     Spinnaker::CameraBase, 622  
     Spinnaker::ICameraBase, 833  
 GetValidPayloadSize  
     Spinnaker::IImage, 869  
     Spinnaker::Image, 915  
 GetValue  
     IBoolean Interface, 276  
     Spinnaker::GenApi::BooleanNode, 460  
     Spinnaker::GenApi::CEnumerationTRef, 658  
     Spinnaker::GenApi::Counter, 728  
     Spinnaker::GenApi::EnumEntryNode, 767  
     Spinnaker::GenApi::FloatNode, 798  
     Spinnaker::GenApi::IntegerNode, 962  
     Spinnaker::GenApi::StringNode, 1068  
 GetValueOfEnvironmentVariable  
     GCUtilities Utility, 272, 273  
 GetVendorName  
     IDeviceInfo Interface, 285  
     Spinnaker::GenApi::NodeMap, 1030  
 GetVersion  
     Chunk Data Inference Class, 161  
 GetVersionGuid  
     IDeviceInfo Interface, 285  
     Spinnaker::GenApi::NodeMap, 1030  
 GetVisibility  
     INode Interface, 305  
     Spinnaker::GenApi::Node, 1021  
 GetWidth  
     Spinnaker::ChunkData, 698  
     Spinnaker::IChunkData, 848  
     Spinnaker::IImage, 870  
     Spinnaker::Image, 916  
 GetXOffset  
     Spinnaker::IImage, 870  
     Spinnaker::Image, 916  
 GetXPadding  
     Spinnaker::IImage, 870  
     Spinnaker::Image, 916  
 GetYOffset  
     Spinnaker::IImage, 870  
     Spinnaker::Image, 916  
 Spinnaker::Image, 917  
 GetYPadding  
     Spinnaker::IImage, 870  
     Spinnaker::Image, 917  
 getline  
     Spinnaker::GenICam, 439  
 GevActionDeviceKey  
     Spinnaker::TransportLayerInterface, 1104  
 GevActionGroupKey  
     Spinnaker::TransportLayerInterface, 1104  
 GevActionGroupMask  
     Spinnaker::TransportLayerInterface, 1105  
 GevActionTime  
     Spinnaker::TransportLayerInterface, 1105  
 GevActiveLinkCount  
     Spinnaker::Camera, 569  
 GevCCPEnum  
     TransportLayerDefs Class, 213  
 GevCCPEnums  
     CameraDefs Class, 113  
 GevCCP  
     Spinnaker::Camera, 569  
     Spinnaker::TransportLayerDevice, 1096  
 GevCurrentDefaultGateway  
     Spinnaker::Camera, 569  
 GevCurrentIPAddress  
     Spinnaker::Camera, 570  
 GevCurrentIPConfigurationDHCP  
     Spinnaker::Camera, 570  
 GevCurrentIPConfigurationLLA  
     Spinnaker::Camera, 570  
 GevCurrentIPConfigurationPersistentIP  
     Spinnaker::Camera, 570  
 GevCurrentPhysicalLinkConfiguration  
     Spinnaker::Camera, 570  
 GevCurrentPhysicalLinkConfigurationEnums  
     CameraDefs Class, 113  
 GevCurrentSubnetMask  
     Spinnaker::Camera, 570  
 GevDeviceDiscoverMaximumPacketSize  
     Spinnaker::TransportLayerDevice, 1096  
 GevDeviceForceGateway  
     Spinnaker::TransportLayerDevice, 1096  
 GevDeviceForceIPAddress  
     Spinnaker::TransportLayerDevice, 1096  
 GevDeviceForceIPEx  
     Spinnaker::TransportLayerDevice, 1096  
 GevDeviceForceIP  
     Spinnaker::TransportLayerDevice, 1096  
 GevDeviceForceSubnetMask  
     Spinnaker::TransportLayerDevice, 1097  
 GevDeviceGateway  
     Spinnaker::TransportLayerDevice, 1097  
 GevDeviceIPAddress  
     Spinnaker::TransportLayerDevice, 1097  
     Spinnaker::TransportLayerInterface, 1105  
 GevDeviceIsWrongSubnet  
     Spinnaker::TransportLayerDevice, 1097

GevDeviceMACAddress  
    Spinnaker::TransportLayerDevice, 1097  
    Spinnaker::TransportLayerInterface, 1105  
GevDeviceMaximumPacketSize  
    Spinnaker::TransportLayerDevice, 1097  
GevDeviceMaximumRetryCount  
    Spinnaker::TransportLayerDevice, 1098  
GevDeviceModelsBigEndian  
    Spinnaker::TransportLayerDevice, 1098  
GevDevicePort  
    Spinnaker::TransportLayerDevice, 1098  
GevDeviceReadAndWriteTimeout  
    Spinnaker::TransportLayerDevice, 1098  
GevDeviceSubnetMask  
    Spinnaker::TransportLayerDevice, 1098  
    Spinnaker::TransportLayerInterface, 1105  
GevDiscoveryAckDelay  
    Spinnaker::Camera, 571  
GevFailedPacketCount  
    Spinnaker::TransportLayerStream, 1112  
GevFirstURL  
    Spinnaker::Camera, 571  
GevGVCPExtendedStatusCodes  
    Spinnaker::Camera, 571  
GevGVCPExtendedStatusCodesSelector  
    Spinnaker::Camera, 571  
GevGVCPExtendedStatusCodesSelectorEnums  
    CameraDefs Class, 113  
GevGVCPHeartbeatDisable  
    Spinnaker::Camera, 571  
GevGVCPPendingAck  
    Spinnaker::Camera, 571  
GevGVCPPendingTimeout  
    Spinnaker::Camera, 572  
GevGVSPExtendedIDMode  
    Spinnaker::Camera, 572  
GevGVSPExtendedIDModeEnums  
    CameraDefs Class, 114  
GevHeartbeatTimeout  
    Spinnaker::Camera, 572  
GevIEEE1588  
    Spinnaker::Camera, 572  
GevIEEE1588ClockAccuracy  
    Spinnaker::Camera, 572  
GevIEEE1588ClockAccuracyEnums  
    CameraDefs Class, 114  
GevIEEE1588Mode  
    Spinnaker::Camera, 572  
GevIEEE1588ModeEnums  
    CameraDefs Class, 114  
GevIEEE1588Status  
    Spinnaker::Camera, 573  
GevIEEE1588StatusEnums  
    CameraDefs Class, 115  
GevIPConfigurationStatus  
    Spinnaker::Camera, 573  
GevIPConfigurationStatusEnums  
    CameraDefs Class, 115

GevInterfaceGateway  
    Spinnaker::TransportLayerInterface, 1105  
GevInterfaceIPAddress  
    Spinnaker::TransportLayerInterface, 1106  
GevInterfaceMACAddress  
    Spinnaker::TransportLayerInterface, 1106  
GevInterfaceMTU  
    Spinnaker::TransportLayerInterface, 1106  
GevInterfaceReceiveLinkSpeed  
    Spinnaker::TransportLayerInterface, 1106  
GevInterfaceSelector  
    Spinnaker::Camera, 573  
GevInterfaceSubnetMask  
    Spinnaker::TransportLayerInterface, 1106  
GevInterfaceTransmitLinkSpeed  
    Spinnaker::TransportLayerInterface, 1106  
GevMACAddress  
    Spinnaker::Camera, 573  
GevMCDA  
    Spinnaker::Camera, 573  
GevMCPHostPort  
    Spinnaker::Camera, 573  
GevMCRC  
    Spinnaker::Camera, 574  
GevMCSP  
    Spinnaker::Camera, 574  
GevMCTT  
    Spinnaker::Camera, 574  
GevMaximumNumberResendBuffers  
    Spinnaker::TransportLayerStream, 1112  
GevMaximumNumberResendRequests  
    Spinnaker::TransportLayerStream, 1112  
GevNumberOfInterfaces  
    Spinnaker::Camera, 574  
GevPAUSEFrameReception  
    Spinnaker::Camera, 574  
GevPAUSEFrameTransmission  
    Spinnaker::Camera, 574  
GevPacketResendMode  
    Spinnaker::TransportLayerStream, 1112  
GevPacketResendTimeout  
    Spinnaker::TransportLayerStream, 1112  
GevPersistentDefaultGateway  
    Spinnaker::Camera, 575  
GevPersistentIPAddress  
    Spinnaker::Camera, 575  
GevPersistentSubnetMask  
    Spinnaker::Camera, 575  
GevPhysicalLinkConfiguration  
    Spinnaker::Camera, 575  
GevPhysicalLinkConfigurationEnums  
    CameraDefs Class, 115  
GevPrimaryApplicationIPAddress  
    Spinnaker::Camera, 575  
GevPrimaryApplicationSocket  
    Spinnaker::Camera, 575  
GevPrimaryApplicationSwitchoverKey  
    Spinnaker::Camera, 576

GevResendPacketCount  
     Spinnaker::TransportLayerStream, 1113

GevResendRequestCount  
     Spinnaker::TransportLayerStream, 1113

GevSCCFGAllInTransmission  
     Spinnaker::Camera, 576

GevSCCFGExtendedChunkData  
     Spinnaker::Camera, 576

GevSCCFGPacketResendDestination  
     Spinnaker::Camera, 576

GevSCFGUnconditionalStreaming  
     Spinnaker::Camera, 576

GevSCDA  
     Spinnaker::Camera, 576

GevSCPDirection  
     Spinnaker::Camera, 577

GevSCPHostPort  
     Spinnaker::Camera, 577

GevSCPIfaceIndex  
     Spinnaker::Camera, 577

GevSCPSBigEndian  
     Spinnaker::Camera, 577

GevSCPSDoNotFragment  
     Spinnaker::Camera, 577

GevSCPSFireTestPacket  
     Spinnaker::Camera, 578

GevSCPSPacketSize  
     Spinnaker::Camera, 578

GevSCPD  
     Spinnaker::Camera, 577

GevSCSP  
     Spinnaker::Camera, 578

GevSCZoneConfigurationLock  
     Spinnaker::Camera, 578

GevSCZoneCount  
     Spinnaker::Camera, 578

GevSCZoneDirectionAll  
     Spinnaker::Camera, 578

GevSecondURL  
     Spinnaker::Camera, 579

GevStreamChannelSelector  
     Spinnaker::Camera, 579

GevSupportedOption  
     Spinnaker::Camera, 579

GevSupportedOptionSelector  
     Spinnaker::Camera, 579

GevSupportedOptionSelectorEnums  
     CameraDefs Class, 116

GevTimestampTickFrequency  
     Spinnaker::Camera, 579

GevTotalPacketCount  
     Spinnaker::TransportLayerStream, 1113

GevVersionMajor  
     Spinnaker::TransportLayerDevice, 1098

GevVersionMinor  
     Spinnaker::TransportLayerDevice, 1099

GuiXmlManifestAddress  
     Spinnaker::Camera, 579

Guru  
     Types Enums, 374

H264Option, 823  
     Spinnaker::Video::H264Option, 824

HasCRC  
     Spinnaker::GenApi::CChunkAdapterDcam, 643

HasInc  
     IFloat Interface, 296

Header  
     GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, 820

HeatMapColor  
     Spinnaker Definitions, 201

HeatmapColor  
     Spinnaker::ImageUtilityHeatmap, 942

Height  
     Spinnaker::Camera, 580

height  
     Spinnaker::Video::H264Option, 825

HeightMax  
     Spinnaker::Camera, 580

HostAdapterDriverVersion  
     Spinnaker::TransportLayerInterface, 1107

HostAdapterName  
     Spinnaker::TransportLayerInterface, 1107

HostAdapterVendor  
     Spinnaker::TransportLayerInterface, 1107

IArrivalEvent, 826  
     Spinnaker::IArrivalEvent, 827

IBase  
     IBase Interface, 241

IBase Interface, 241  
     IBase, 241

IBoolean  
     IBoolean Interface, 277

IBoolean Interface, 276  
     GetValue, 276

IBoolean, 277  
     operator(), 277

operator=, 277  
     Verify, 277

ICameraBase, 828  
     Spinnaker::ICameraBase, 830

ICameraList, 837  
     Spinnaker::ICameraList, 838

ICategory  
     ICategory Interfaces, 278

ICategory Interfaces, 278  
     ICategory, 278

IChunkData, 840  
    Spinnaker::IChunkData, 842  
IChunkData Class, 221  
IChunkPort  
    IChunkPort Interface, 280  
IChunkPort Interface, 279  
    CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN, 279  
    CHUNK\_BASE\_ADDRESS\_REGISTER, 279  
    CHUNK\_LENGTH\_REGISTER\_LEN, 280  
    CHUNK\_LENGTH\_REGISTER, 280  
    CacheChunkData, 280  
    IChunkPort, 280  
 ICommand  
    ICommand Interface, 281  
ICommand Interface, 281  
    ICommand, 281  
    IsDone, 281  
IDataStream, 849  
    Spinnaker::Event, 781  
    Spinnaker::IDataStream, 850  
    Spinnaker::Image, 923  
IDestroy  
    IDestroy Interface, 283  
IDestroy Interface, 283  
    IDestroy, 283  
IDevFileStream  
    Spinnaker::GenApi, 435  
IDevFileStreamBase< CharType, Traits >, 855  
IDevFileStreamBuf  
    Spinnaker::GenApi::IDevFileStreamBuf, 858  
IDevFileStreamBuf< CharType, Traits >, 857  
IDeviceEvent, 859  
    Spinnaker::IDeviceEvent, 860  
IDeviceInfo  
    IDeviceInfo Interface, 286  
IDeviceInfo Interface, 284  
    GetDeviceVersion, 284  
    GetGenApiVersion, 284  
    GetProductGuid, 285  
    GetSchemaVersion, 285  
    GetStandardNameSpace, 285  
    GetToolTip, 285  
    GetVendorName, 285  
    GetVersionGuid, 285  
    IDeviceInfo, 286  
IEnumEntry  
    IEnumEntry Interface, 288  
IEnumEntry Interface, 287  
    GetNumericValue, 287  
    GetSymbolic, 287  
    IEnumEntry, 288  
    IsSelfClearing, 287  
IEnumReference  
    IEnumerationT Interface, 293  
IEnumeration  
    IEnumeration Interface, 291  
IEnumeration Interface, 289  
    GetCurrentEntry, 289  
    GetEntries, 289  
    GetEntry, 290  
    GetEntryByName, 290  
    GetIntValue, 290  
    IEnumeration, 291  
    operator\*, 290  
    SetIntValue, 291  
IEnumerationT Interface, 292  
    GetEntry, 292  
    IEnumReference, 293  
    IEnumerationT, 293  
    operator=, 292, 293  
IEnumerationT  
    IEnumerationT Interface, 293  
IFloat  
    IFloat Interface, 297  
IFloat Interface, 294  
    GetDisplayNotation, 295  
    GetDisplayPrecision, 295  
    GetInc, 295  
    GetIncMode, 295  
    GetListOfValidValues, 295  
    GetMax, 295  
    GetMin, 296  
    GetRepresentation, 296  
    GetUnit, 296  
    HasInc, 296  
    IFloat, 297  
    ImposeMax, 296  
    ImposeMin, 296  
    operator=, 297  
IImage, 862  
    Spinnaker::IImage, 864  
IImage Class, 222  
IImageEvent, 874  
    Spinnaker::IImageEvent, 875  
IImageStatistics, 876  
    Spinnaker::IImageStatistics, 877  
IImageStatistics Class, 223  
IInteger  
    IInteger Interface, 299  
IInteger Interface, 298  
    IInteger, 299  
    ImposeMax, 298  
    ImposeMin, 298  
    operator=, 298  
IInterface, 880  
    Spinnaker::IInterface, 881, 882  
    Spinnaker::TransportLayerInterface, 1102  
IInterface Class, 224  
IInterfaceArrivalEvent, 884  
    Spinnaker::IInterfaceArrivalEvent, 885  
IInterfaceEvent, 886  
    Spinnaker::IInterfaceEvent, 887  
IInterfaceList, 889  
    Spinnaker::IInterfaceList, 890  
IInterfaceList Class, 225  
IInterfaceRemovalEvent, 891

Spinnaker::IInterfaceRemovalEvent, 892  
 ILoggingEvent, 893  
   Spinnaker::ILoggingEvent, 894  
 INTEGRAL\_CAST2  
   GCUtilities Utility, 273  
 INTEGRAL\_CAST  
   GCUtilities Utility, 273  
 INode  
   INode Interface, 310  
 INode Interface, 300  
   Combine, 302  
   DeregisterCallback, 302  
   GetAlias, 303  
   GetCachingMode, 303  
   GetCastAlias, 303  
   GetChildren, 303  
   GetDescription, 304  
   GetDisplayName, 304  
   GetDocuURL, 304  
   GetEventID, 304  
   GetNameSpace, 304  
   GetNodeMap, 304  
   GetParents, 304  
   GetPollingTime, 305  
   GetPrincipalInterfaceType, 305  
   GetProperty, 305  
   GetPropertyNames, 305  
   GetVisibility, 305  
 INode, 310  
 IReference, 310  
   ImposeAccessMode, 305  
   ImposeVisibility, 306  
   InvalidateNode, 306  
   IsAccessModeCacheable, 306  
   IsAvailable, 306  
   IsCachable, 307  
   IsCacheable, 307  
   IsDeprecated, 307  
   IsFeature, 307  
   IsImplemented, 307, 308  
   IsReadable, 308  
   IsStreamable, 308  
   IsVisible, 308  
   IsWritable, 309  
   operator!=, 309  
   operator==, 309  
   RegisterCallback, 309  
 INodeMap  
   INodeMap Interface, 313  
 INodeMap Interface, 311  
   Connect, 311, 312  
   GetDeviceName, 312  
   GetLock, 312  
   GetNode, 312  
   GetNumNodes, 312  
 INodeMap, 313  
   InvalidateNodes, 313  
   Poll, 313  
 INodeMapDyn  
   INodeMapDyn Interface, 318  
 INodeMapDyn Interface, 314  
   ExtractIndependentSubtree, 315  
   GetSupportedSchemaVersions, 315  
   INodeMapDyn, 318  
   LoadXMLFromFile, 315  
   LoadXMLFromFileInject, 315  
   LoadXMLFromString, 316  
   LoadXMLFromStringInject, 316  
   LoadXMLFromZIPData, 316  
   LoadXMLFromZIPFile, 316  
   MergeXMLFiles, 316  
   PreprocessXMLFromFile, 317  
   PreprocessXMLFromZIPFile, 317  
 IPersistScript  
   Spinnaker::GenApi, 437  
 IPort  
   IPort Interface, 322  
 IPort Interface, 321  
   Address, 321  
   IPort, 322  
   Length, 322  
   Write, 321  
 IPortConstruct  
   IPortConstruct Interface, 323  
 IPortConstruct Interface, 323  
   GetSwapEndianess, 323  
   IPortConstruct, 323  
 IPortRecorder  
   IPortRecorder Interface, 325  
 IPortRecorder Interface, 324  
   GetCookie, 324  
   IPortRecorder, 325  
   IPortReplay, 325  
   IPortWriteList, 325  
   Invalidate, 325  
   Replay, 324  
   SetCookie, 325  
   StopRecording, 325  
 IPortReplay  
   IPortRecorder Interface, 325  
 IPortWriteList  
   IPortRecorder Interface, 325  
 IReference  
   INode Interface, 310  
 IRegister  
   IRegister Interfaces, 327  
 IRegister Interfaces, 326  
   Get, 326  
   GetAddress, 327  
   GetLength, 327  
   IRegister, 327  
 IRemovalEvent, 986  
   Spinnaker::IRemovalEvent, 987  
 ISelector  
   ISelector Interface, 328  
 ISelector Interface, 328

GetSelectedFeatures, 328  
GetSelectingFeatures, 328  
ISelector, 328  
ISelectorDigit  
    ISelectorDigit Interface, 330  
ISelectorDigit Interface, 329  
    GetSelectorList, 329  
    ISelectorDigit, 330  
    Restore, 330  
    SetNext, 330  
    ToString, 330  
IString  
    IString Class, 332  
IString Class, 332  
    GetMaxLength, 332  
    IString, 332  
ISystem, 988  
    Spinnaker::ISystem, 989  
    Spinnaker::TransportLayerSystem, 1117  
ISystem Class, 226  
ISystemEvent, 994  
    Spinnaker::ISystemEvent, 995  
IValue  
    IValue Class, 334  
IValue Class, 333  
    FromString, 333  
    IValue, 334  
    IsValueCacheValid, 334  
    ToString, 334  
Image, 895  
    Spinnaker::Image, 900  
Image Class, 169  
Image Utility Class, 173  
Image Utility Heatmap Class, 174  
Image Utility Polarization Class, 175  
ImageComponentEnable  
    Spinnaker::Camera, 580  
ImageComponentSelector  
    Spinnaker::Camera, 580  
ImageComponentSelectorEnums  
    CameraDefs Class, 117  
ImageCompressionBitrate  
    Spinnaker::Camera, 580  
ImageCompressionJPEGFormatOption  
    Spinnaker::Camera, 580  
ImageCompressionJPEGFormatOptionEnums  
    CameraDefs Class, 117  
ImageCompressionMode  
    Spinnaker::Camera, 581  
ImageCompressionModeEnums  
    CameraDefs Class, 118  
ImageCompressionQuality  
    Spinnaker::Camera, 581  
ImageCompressionRateOption  
    Spinnaker::Camera, 581  
ImageCompressionRateOptionEnums  
    CameraDefs Class, 118  
ImageConverter  
    Spinnaker::Image, 923  
ImageEvent, 924  
    Spinnaker::ImageEvent, 925  
ImageEvent Class, 170  
ImageFileFormat  
    Spinnaker Definitions, 195  
ImageFiler  
    Spinnaker::Image, 923  
ImagePtr, 927  
    Spinnaker::ImagePtr, 928  
ImagePtr Class, 171  
ImageScalingAlgorithm  
    Spinnaker::ImageUtility, 937  
ImageStatistics, 929  
    Spinnaker::ImageStatistics, 931  
ImageStatistics Class, 172  
ImageStatsCalculator  
    Spinnaker::Image, 923  
    Spinnaker::ImageStatistics, 936  
ImageStatus  
    Spinnaker Definitions, 197  
ImageUtility, 936  
ImageUtilityHeatmap, 941  
ImageUtilityImpl  
    Spinnaker::Image, 923  
ImageUtilityPolarization, 945  
ImageUtilityPolarizationImpl  
    Spinnaker::Image, 923  
ImposeAccessMode  
    INode Interface, 305  
    Spinnaker::GenApi::Node, 1021  
ImposeMax  
    IFloat Interface, 296  
    IInteger Interface, 298  
    Spinnaker::GenApi::FloatNode, 798  
    Spinnaker::GenApi::IntegerNode, 962  
ImposeMin  
    IFloat Interface, 296  
    IInteger Interface, 298  
    Spinnaker::GenApi::FloatNode, 798  
    Spinnaker::GenApi::IntegerNode, 962  
ImposeVisibility  
    INode Interface, 306  
    Spinnaker::GenApi::Node, 1021  
include/AVIRecorder.h, 1131  
include/AdapterConfig.h, 1127  
include/ArrivalEvent.h, 1129  
include/BasePtr.h, 1131  
include/Camera.h, 1133  
include/CameraBase.h, 1135  
include/CameraDefs.h, 1137  
include/CameraList.h, 1170  
include/CameraPtr.h, 1172  
include/ChunkData.h, 1174  
include/ChunkDataInference.h, 1176  
include/DeviceEvent.h, 1178  
include/Event.h, 1180  
include/Exception.h, 1181

include/Image.h, 1183  
include/ImageEvent.h, 1185  
include/ImagePtr.h, 1186  
include/ImageStatistics.h, 1188  
include/ImageUtility.h, 1190  
include/ImageUtilityHeatmap.h, 1190  
include/ImageUtilityPolarization.h, 1191  
include/Interface.h, 1191  
include/Interface/IArrivalEvent.h, 1193  
include/Interface/ICameraBase.h, 1195  
include/Interface/ICameraList.h, 1197  
include/Interface/IChunkData.h, 1199  
include/Interface/IDeviceEvent.h, 1201  
include/Interface/IImage.h, 1203  
include/Interface/IImageEvent.h, 1205  
include/Interface/IImageStatistics.h, 1206  
include/Interface/IInterface.h, 1208  
include/Interface/IInterfaceArrivalEvent.h, 1210  
include/Interface/IInterfaceEvent.h, 1211  
include/Interface/IInterfaceList.h, 1213  
include/Interface/IInterfaceRemovalEvent.h, 1214  
include/Interface/ILoggingEvent.h, 1216  
include/Interface/IRemovalEvent.h, 1217  
include/Interface/IStream.h, 1219  
include/Interface/ISystem.h, 1219  
include/Interface/ISystemEvent.h, 1221  
include/InterfaceArrivalEvent.h, 1222  
include/InterfaceEvent.h, 1223  
include/InterfaceList.h, 1225  
include/InterfacePtr.h, 1226  
include/InterfaceRemovalEvent.h, 1228  
include/LoggingEvent.h, 1229  
include/LoggingEventData.h, 1231  
include/LoggingEventDataPtr.h, 1233  
include/RemovalEvent.h, 1235  
include/SpinGenApi/Autovector.h, 1237  
include/SpinGenApi/Base.h, 1238  
include/SpinGenApi/BooleanNode.h, 1239  
include/SpinGenApi/CategoryNode.h, 1241  
include/SpinGenApi/ChunkAdapter.h, 1243  
include/SpinGenApi/ChunkAdapterDcam.h, 1245  
include/SpinGenApi/ChunkAdapterGEV.h, 1249  
include/SpinGenApi/ChunkAdapterGeneric.h, 1247  
include/SpinGenApi/ChunkAdapterU3V.h, 1251  
include/SpinGenApi/ChunkPort.h, 1253  
include/SpinGenApi/CommandNode.h, 1255  
include/SpinGenApi/Compatibility.h, 1258  
include/SpinGenApi/Container.h, 1259  
include/SpinGenApi/Counter.h, 1259  
include/SpinGenApi/EnumClasses.h, 1260  
include/SpinGenApi/EnumEntryNode.h, 1262  
include/SpinGenApi/EnumNode.h, 1264  
include/SpinGenApi/EnumNodeT.h, 1266  
include/SpinGenApi/EventAdapter.h, 1268  
include/SpinGenApi/EventAdapter1394.h, 1270  
include/SpinGenApi/EventAdapterGEV.h, 1274  
include/SpinGenApi/EventAdapterGeneric.h, 1272  
include/SpinGenApi/EventAdapterU3V.h, 1276  
include/SpinGenApi/EventPort.h, 1278  
include/SpinGenApi/Filestream.h, 1280  
include/SpinGenApi/FloatNode.h, 1282  
include/SpinGenApi/FloatRegNode.h, 1284  
include/SpinGenApi/GCBase.h, 1286  
include/SpinGenApi/GCString.h, 1287  
include/SpinGenApi/GCStringVector.h, 1289  
include/SpinGenApi/GCSynch.h, 1290  
include/SpinGenApi/GCTypes.h, 1291  
include/SpinGenApi/GCUtilities.h, 1294  
include/SpinGenApi/IBoolean.h, 1299  
include/SpinGenApi/ICategory.h, 1301  
include/SpinGenApi/IChunkPort.h, 1303  
include/SpinGenApi/ICommand.h, 1305  
include/SpinGenApi/IDestroy.h, 1307  
include/SpinGenApi/IDeviceInfo.h, 1308  
include/SpinGenApi/IEnumEntry.h, 1310  
include/SpinGenApi/IEnumeration.h, 1312  
include/SpinGenApi/IEnumerationT.h, 1313  
include/SpinGenApi/IFloat.h, 1315  
include/SpinGenApi/IInteger.h, 1317  
include/SpinGenApi/INode.h, 1319  
include/SpinGenApi/INodeMap.h, 1322  
include/SpinGenApi/INodeMapDyn.h, 1323  
include/SpinGenApi/IPort.h, 1329  
include/SpinGenApi/IPortConstruct.h, 1330  
include/SpinGenApi/IPortRecorder.h, 1332  
include/SpinGenApi/IRegister.h, 1334  
include/SpinGenApi/ISelector.h, 1335  
include/SpinGenApi/ISelectorDigit.h, 1336  
include/SpinGenApi/IString.h, 1338  
include/SpinGenApi/IValue.h, 1340  
include/SpinGenApi/IntRegNode.h, 1327  
include/SpinGenApi/IntegerNode.h, 1325  
include/SpinGenApi/Node.h, 1341  
include/SpinGenApi/NodeCallback.h, 1343  
include/SpinGenApi/NodeCallbackImpl.h, 1345  
include/SpinGenApi/NodeMap.h, 1346  
include/SpinGenApi/NodeMapFactory.h, 1348  
include/SpinGenApi/NodeMapRef.h, 1349  
include/SpinGenApi/Persistence.h, 1350  
include/SpinGenApi/Pointer.h, 1352  
include/SpinGenApi/PortImpl.h, 1355  
include/SpinGenApi/PortNode.h, 1356  
include/SpinGenApi/PortRecorder.h, 1358  
include/SpinGenApi/PortReplay.h, 1359  
include/SpinGenApi/PortWriteList.h, 1360  
include/SpinGenApi/Reference.h, 1362  
include/SpinGenApi/RegisterNode.h, 1363  
include/SpinGenApi/RegisterPortImpl.h, 1365  
include/SpinGenApi/SelectorSet.h, 1365  
include/SpinGenApi/SpinTestCamera.h, 1368  
include/SpinGenApi/SpinnakerGenApi.h, 1366  
include/SpinGenApi/StringNode.h, 1368  
include/SpinGenApi/StringRegNode.h, 1370  
include/SpinGenApi/StructPort.h, 1372  
include/SpinGenApi/Synch.h, 1372  
include/SpinGenApi/Types.h, 1373

include/SpinGenApi/ValueNode.h, 1377  
include/SpinUpdate.h, 1384  
include/SpinVideo.h, 1387  
include/SpinVideoDefs.h, 1387  
include/Spinnaker.h, 1379  
include/SpinnakerDefs.h, 1380  
include/SpinnakerPlatform.h, 1384  
include/System.h, 1388  
include/SystemEvent.h, 1390  
include/SystemPtr.h, 1391  
include/TransportLayerDefs.h, 1393  
include/TransportLayerDevice.h, 1395  
include/TransportLayerInterface.h, 1397  
include/TransportLayerStream.h, 1399  
include/TransportLayerSystem.h, 1401  
IncompatibleDeviceCount  
    Spinnaker::TransportLayerInterface, 1107  
IncompatibleDeviceID  
    Spinnaker::TransportLayerInterface, 1107  
IncompatibleDeviceModelName  
    Spinnaker::TransportLayerInterface, 1107  
IncompatibleDeviceSelector  
    Spinnaker::TransportLayerInterface, 1108  
IncompatibleDeviceVendorName  
    Spinnaker::TransportLayerInterface, 1108  
IncompatibleGevDeviceIPAddress  
    Spinnaker::TransportLayerInterface, 1108  
IncompatibleGevDeviceMACAddress  
    Spinnaker::TransportLayerInterface, 1108  
IncompatibleGevDeviceSubnetMask  
    Spinnaker::TransportLayerInterface, 1108  
Increasing  
    Types Enums, 373  
indexedColor\_8bit  
    Spinnaker::BMPOption, 457  
InferenceBoundingBox, 953  
InferenceBoundingBoxResult, 953  
    Chunk Data Inference Class, 162  
InferenceBoxCircle, 954  
InferenceBoxRect, 954  
InferenceBoxRotatedRect, 955  
InferenceBoxType  
    Spinnaker Classes, 34  
Init  
    Spinnaker::Camera, 492  
    Spinnaker::CameraBase, 622  
    Spinnaker::ICameraBase, 833  
InitChunkAdapter  
    Spinnaker::IDataStream, 853  
int64\_autovector\_t, 955  
    Spinnaker::GenApi::int64\_autovector\_t, 955, 956  
IntRegNode, 982  
    Spinnaker::GenApi::IntRegNode, 983, 984  
IntRegNode Class, 320  
IntegerNode, 958  
    Spinnaker::GenApi::IntegerNode, 960  
IntegerNode Class, 319  
    CIntegerRef, 319  
Interface, 964  
    Spinnaker::TransportLayerInterface, 1102  
interface  
    Types.h, 1376  
Interface Class, 176  
InterfaceArrivalEvent, 969  
    Spinnaker::InterfaceArrivalEvent, 970  
InterfaceArrivalEvent Class, 177  
InterfaceDisplayName  
    Spinnaker::TransportLayerInterface, 1108  
InterfaceEvent, 971  
    Spinnaker::InterfaceEvent, 972  
InterfaceEvent Class, 178  
InterfaceID  
    Spinnaker::TransportLayerInterface, 1109  
InterfaceImpl  
    Spinnaker::CameraBase, 627  
    Spinnaker::ICameraBase, 836  
    Spinnaker::ICameraList, 840  
InterfaceInternal  
    Spinnaker::IInterface, 883  
    Spinnaker::Interface, 969  
    Spinnaker::TransportLayerInterface, 1102  
InterfaceList, 974  
    Spinnaker::InterfaceList, 975  
InterfaceList Class, 179  
InterfacePtr, 977  
    Spinnaker::InterfacePtr, 978, 979  
InterfacePtr Class, 180  
InterfaceRemovalEvent, 979  
    Spinnaker::InterfaceRemovalEvent, 980  
InterfaceRemovalEvent Class, 181  
InterfaceType  
    Spinnaker::TransportLayerInterface, 1109  
interlaced  
    Spinnaker::PNGOption, 1040  
Invalidate  
    IPortRecorder Interface, 325  
InvalidateNode  
    INode Interface, 306  
    Spinnaker::GenApi::CChunkPort, 654  
    Spinnaker::GenApi::CEventPort, 675  
    Spinnaker::GenApi::CPortImpl, 736  
    Spinnaker::GenApi::Node, 1021  
InvalidateNodes  
    INodeMap Interface, 313  
    Spinnaker::GenApi::NodeMap, 1031  
InverseChunkLength  
    DCAM\_CHUNK\_TRAILER, 752  
Invisible  
    Types Enums, 374  
ios\_type  
    Spinnaker::GenApi::IDevFileStreamBase, 856  
    Spinnaker::GenApi::ODevFileStreamBase, 1035  
ipAddress  
    AdapterConfig::IpInfo, 985  
IpInfo, 984  
    AdapterConfig::IpInfo, 985

iplInfo  
     AdapterConfig::AdapterInfo, 445

is\_open  
     Spinnaker::GenApi::IDevFileStreamBase, 856  
     Spinnaker::GenApi::IDevFileStreamBuf, 858  
     Spinnaker::GenApi::ODevFileStreamBase, 1035  
     Spinnaker::GenApi::ODevFileStreamBuf, 1037

IsAccessModeCacheable  
     INode Interface, 306  
     Spinnaker::GenApi::Node, 1021

IsAvailable  
     INode Interface, 306  
     Pointer Class, 349

IsCRCCheckEnabled  
     Spinnaker::IDataStream, 853

IsCachable  
     INode Interface, 307  
     Spinnaker::GenApi::Node, 1022

IsCacheable  
     INode Interface, 307

IsCameraDescriptionFileDataReleased  
     Spinnaker::GenApi::CNodeMapFactory, 714

IsCompressed  
     Spinnaker::Image, 918

IsDeprecated  
     INode Interface, 307  
     Spinnaker::GenApi::Node, 1022

IsDone  
      ICommand Interface, 281  
     Spinnaker::GenApi::CommandNode, 726

IsEmpty  
     Spinnaker::GenApi::CNodeMapFactory, 714  
     Spinnaker::GenApi::CSelectorSet, 746

IsFeature  
     INode Interface, 307  
     Spinnaker::GenApi::Node, 1022

IsImageInUse  
     Spinnaker::IDataStream, 853

IsImplemented  
     INode Interface, 307, 308  
     Pointer Class, 349

IsInUse  
     Spinnaker::IImage, 871  
     Spinnaker::IInterface, 882  
     Spinnaker::ISystem, 990  
     Spinnaker::Image, 918  
     Spinnaker::Interface, 966  
     Spinnaker::System, 1077

IsIncomplete  
     Spinnaker::IImage, 871  
     Spinnaker::Image, 918

IsInitialized  
     Spinnaker::CameraBase, 623  
     Spinnaker::ICameraBase, 833

IsLoaded  
     Spinnaker::GenApi::CNodeMapFactory, 714

IsOnSameSubnet  
     AdapterConfig, 380

IsPreprocessed  
     Spinnaker::GenApi::CNodeMapFactory, 715

IsReadable  
     INode Interface, 308  
     Pointer Class, 349

IsSelector  
     Spinnaker::GenApi::Node, 1022

IsSelfClearing  
     IEnumEntry Interface, 287  
     Spinnaker::GenApi::EnumEntryNode, 767

IsStreamable  
     INode Interface, 308  
     Spinnaker::GenApi::Node, 1022

IsStreaming  
     Spinnaker::CameraBase, 623  
     Spinnaker::ICameraBase, 833  
     Spinnaker::IDataStream, 853

IsValid  
     Spinnaker::BasePtr, 454  
     Spinnaker::CameraBase, 623  
     Spinnaker::GenApi::CPointer, 731  
     Spinnaker::GenlCam::CGlobalLock, 683  
     Spinnaker::ICameraBase, 834  
     Spinnaker::IInterface, 882  
     Spinnaker::Interface, 966

IsValidIpAddress  
     AdapterConfig, 381

IsValidSubnetMask  
     AdapterConfig, 381

IsValueCacheValid  
     IValue Class, 334  
     Spinnaker::GenApi::ValueNode, 1124

IsVisible  
     INode Interface, 308

IsWritable  
     INode Interface, 309  
     Pointer Class, 349

IsZero  
     Spinnaker::GenApi::Counter, 728

IspEnable  
     Spinnaker::Camera, 581

istream\_type  
     Spinnaker::GenApi::IDevFileStreamBase, 856

Items  
     GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, 820  
     GVCP\_EVENT\_REQUEST, 819

JPEGOption, 996  
     Spinnaker::JPEGOption, 996

JPG2Option, 997  
     Spinnaker::JP2Option, 998

jumboPacketValidValues  
     AdapterConfig::AdapterInfo, 445

jumboPackets  
     AdapterConfig::AdapterInfo, 445

jumboPacketsRegKey  
     AdapterConfig::AdapterInfo, 445

KillBufferEvent

Spinnaker::IDataStream, 853  
LUTEnable  
    Spinnaker::Camera, 585  
LUTIndex  
    Spinnaker::Camera, 585  
LUTSelector  
    Spinnaker::Camera, 585  
LUTSelectorEnums  
    CameraDefs Class, 122  
LUTValue  
    Spinnaker::Camera, 585  
LUTValueAll  
    Spinnaker::Camera, 586  
Length  
    GVCP\_REQUEST\_HEADER, 823  
    IPort Interface, 322  
    U3V\_COMMAND\_HEADER, 1119  
length  
    Spinnaker::GenICam::gcstring, 810  
LibraryVersion, 999  
LineFilterWidth  
    Spinnaker::Camera, 581  
LineFormat  
    Spinnaker::Camera, 582  
LineFormatEnums  
    CameraDefs Class, 118  
LineInputFilterSelector  
    Spinnaker::Camera, 582  
LineInputFilterSelectorEnums  
    CameraDefs Class, 119  
LineInverter  
    Spinnaker::Camera, 582  
LineMode  
    Spinnaker::Camera, 582  
LineModeEnums  
    CameraDefs Class, 119  
LinePitch  
    Spinnaker::Camera, 582  
LineSelector  
    Spinnaker::Camera, 582  
LineSelectorEnums  
    CameraDefs Class, 119  
LineSource  
    Spinnaker::Camera, 583  
LineSourceEnums  
    CameraDefs Class, 120  
LineStatus  
    Spinnaker::Camera, 583  
LineStatusAll  
    Spinnaker::Camera, 583  
Linear  
    Types Enums, 372  
LinkErrorCount  
    Spinnaker::Camera, 583  
LinkUptime  
    Spinnaker::Camera, 583  
LoadAndInject  
    Spinnaker::GenApi::CNodeMapFactory, 715  
LoadFromBag  
    Spinnaker::GenApi::CFeatureBag, 677  
LoadXMLFromFile  
    INodeMapDyn Interface, 315  
    Spinnaker::GenApi::NodeMap, 1031  
LoadXMLFromFileInject  
    INodeMapDyn Interface, 315  
    Spinnaker::GenApi::NodeMap, 1031  
LoadXMLFromString  
    INodeMapDyn Interface, 316  
    Spinnaker::GenApi::NodeMap, 1031  
LoadXMLFromStringInject  
    INodeMapDyn Interface, 316  
    Spinnaker::GenApi::NodeMap, 1031  
LoadXMLFromZIPData  
    INodeMapDyn Interface, 316  
    Spinnaker::GenApi::NodeMap, 1032  
LoadXMLFromZIPFile  
    INodeMapDyn Interface, 316  
    Spinnaker::GenApi::NodeMap, 1032  
Lock  
    Spinnaker::GenApi::CLock, 701  
    Spinnaker::GenICam::CGlobalLock, 683  
    Spinnaker::GenICam::CLock, 699  
    Spinnaker::GenICam::LockableObject, 1002  
    Spinnaker::GenICam::LockableObject::Lock, 1000  
LockableObject< Object >, 1001  
LockableObject< Object >::Lock, 1000  
Logarithmic  
    Types Enums, 372  
Logging Event Class, 183  
LoggingEvent, 1003  
    Spinnaker::LoggingEvent, 1004  
LoggingEvent Class, 182  
LoggingEventData, 1005  
    Spinnaker::LoggingEventData, 1006  
LoggingEventDataPtr, 1008  
    Spinnaker::LoggingEventDataPtr, 1009  
LoggingEventDataPtr Class, 184  
LogicBlockLUTInputActivation  
    Spinnaker::Camera, 583  
LogicBlockLUTInputActivationEnums  
    CameraDefs Class, 120  
LogicBlockLUTInputSelector  
    Spinnaker::Camera, 584  
LogicBlockLUTInputSelectorEnums  
    CameraDefs Class, 121  
LogicBlockLUTInputSource  
    Spinnaker::Camera, 584  
LogicBlockLUTInputSourceEnums  
    CameraDefs Class, 121  
LogicBlockLUTOOutputValue  
    Spinnaker::Camera, 584  
LogicBlockLUTOOutputValueAll  
    Spinnaker::Camera, 584  
LogicBlockLUTRowIndex  
    Spinnaker::Camera, 584  
LogicBlockLUTSelector

Spinnaker::Camera, 584  
 LogicBlockLUTSelectorEnums  
     CameraDefs Class, 122  
 LogicBlockSelector  
     Spinnaker::Camera, 585  
 LogicBlockSelectorEnums  
     CameraDefs Class, 122  
  
 m\_BaseAddress  
     Spinnaker::GenApi::CTestPortStruct, 750  
 m\_CallbackType  
     Spinnaker::GenApi::CNodeCallback, 707  
 m\_Callbacks  
     Spinnaker::GenApi::Node, 1024  
 m\_DebugCount  
     Spinnaker::GenICam::CGlobalLock, 684  
 m\_Lock  
     Spinnaker::GenICam::CGlobalLockUnlocker, 686  
     Spinnaker::GenICam::LockableObject, 1002  
 m\_NumReads  
     Spinnaker::GenApi::CTestPortStruct, 750  
 m\_NumWrites  
     Spinnaker::GenApi::CTestPortStruct, 751  
 m\_bOwnLock  
     Spinnaker::GenApi::CLock, 702  
 m\_enabled  
     Spinnaker::GenICam::CGlobalLockUnlocker, 686  
 m\_lock  
     Spinnaker::GenApi::CLOCK, 702  
 m\_lockEx  
     Spinnaker::GenApi::ClockEx, 705  
 m\_pCameraBaseData  
     Spinnaker::ICameraBase, 836  
 m\_pCameraListData  
     Spinnaker::ICameraList, 840  
 m\_pChunkAdapter  
     Spinnaker::GenApi::CChunkAdapter, 640  
 m\_pChunkPort  
     Spinnaker::GenApi::CChunkPort, 655  
 m\_pEnumeration  
     Spinnaker::GenApi::EnumNode, 773  
 m\_pEventAdapter  
     Spinnaker::GenApi::CEventAdapter, 662  
 m\_pEventData  
     Spinnaker::Event, 781  
 m\_pEventPort  
     Spinnaker::GenApi::CEventPort, 675  
 m\_pInterfaceData  
     Spinnaker::IInterface, 884  
 m\_pInterfaceListData  
     Spinnaker::IInterfaceList, 891  
 m\_pNode  
     Spinnaker::GenApi::CEventPort, 676  
     Spinnaker::GenApi::CNodeCallback, 707  
 m\_pNodeData  
     Spinnaker::GenApi::Node, 1024  
 m\_pNodeMap  
     Spinnaker::GenApi::Node, 1024  
 m\_pPort

Spinnaker::GenApi::CChunkPort, 655  
 m\_pPortAdapter  
     Spinnaker::GenApi::CChunkPort, 655  
     Spinnaker::GenApi::CEventPort, 676  
 m\_pWriteList  
     Spinnaker::GenApi::CPortWriteList, 740  
 m\_pT  
     Spinnaker::BasePtr, 456  
     Spinnaker::GenApi::CPointer, 734  
 m\_ptrPort  
     Spinnaker::GenApi::CPortImpl, 737  
 MJPGOption, 1012  
     Spinnaker::Video::MJPGOption, 1012  
 Magic  
     GVCP\_REQUEST\_HEADER, 823  
 Major  
     Spinnaker::GenICam::Version\_t, 1125  
 major  
     Spinnaker::LibraryVersion, 999  
 make\_NodeCallback  
     NodeCallback Class, 337  
 max\_size  
     Spinnaker::GenICam::gcstring, 810  
 MaxDeviceResetTime  
     Spinnaker::Camera, 586  
 MemSet  
     Spinnaker::GenApi::CTestPortStruct, 749  
 Member\_NodeCallback  
     Spinnaker::GenApi::Member\_NodeCallback, 1011  
 Member\_NodeCallback< Client, Member >, 1010  
 MergeXMLFiles  
     INodeMapDyn Interface, 316  
 Minor  
     Spinnaker::GenICam::Version\_t, 1125  
 minor  
     Spinnaker::LibraryVersion, 999  
 NA  
     Types Enums, 368  
 NI  
     Types Enums, 368  
 No  
     Types Enums, 374  
 Node, 1014  
     Spinnaker::GenApi::Node, 1016  
 Node Class, 335  
 NodeCallback Class, 336  
     Deregister, 337  
     ECallbackType, 337  
     make\_NodeCallback, 337  
     Register, 338  
 NodeList\_t  
     Spinnaker GenApi Interfaces, 240  
 NodeMap, 1024  
     Spinnaker::GenApi::CLOCK, 702  
     Spinnaker::GenApi::NodeMap, 1026  
 NodeMap Class, 339  
 NodeMapFactory Class, 340  
     ECacheUsage\_t, 340

EContentType\_t, 341  
NodeMapRef Class, 342  
None  
    Types Enums, 373  
npos  
    Spinnaker::GenICam::gcstring, 814  
NumAttachedChunks  
    AttachStatistics\_t, 449  
NumChunkPorts  
    AttachStatistics\_t, 450  
NumChunks  
    AttachStatistics\_t, 450  
NumLinks  
    Spinnaker::GenApi::CNodeMapFactory::Node<→  
        Statistics\_t, 1033  
NumNodes  
    Spinnaker::GenApi::CNodeMapFactory::Node<→  
        Statistics\_t, 1033  
NumProperties  
    Spinnaker::GenApi::CNodeMapFactory::Node<→  
        Statistics\_t, 1033  
NumStrings  
    Spinnaker::GenApi::CNodeMapFactory::Node<→  
        Statistics\_t, 1033  
ODevFileStream  
    Spinnaker::GenApi, 435  
ODevFileStreamBase< CharType, Traits >, 1034  
ODevFileStreamBuf  
    Spinnaker::GenApi::ODevFileStreamBuf, 1037  
ODevFileStreamBuf< CharType, Traits >, 1036  
OffsetX  
    Spinnaker::Camera, 586  
OffsetY  
    Spinnaker::Camera, 586  
OnDeviceArrival  
    Spinnaker::ArrivalEvent, 448  
    Spinnaker::IArrivalEvent, 827  
    Spinnaker::IInterfaceEvent, 888  
    Spinnaker::InterfaceEvent, 973  
OnDeviceEvent  
    Spinnaker::DeviceEvent, 754  
    Spinnaker::IDeviceEvent, 861  
OnDeviceRemoval  
    Spinnaker::IInterfaceEvent, 888  
    Spinnaker::IRemovalEvent, 987  
    Spinnaker::InterfaceEvent, 973  
    Spinnaker::RemovalEvent, 1058  
OnImageEvent  
    Spinnaker::IImageEvent, 875  
    Spinnaker::ImageEvent, 926  
OnInterfaceArrival  
    Spinnaker::IInterfaceArrivalEvent, 886  
    Spinnaker::ISystemEvent, 995  
    Spinnaker::InterfaceArrivalEvent, 970  
    Spinnaker::SystemEvent, 1084  
OnInterfaceRemoval  
    Spinnaker::IInterfaceRemovalEvent, 893  
    Spinnaker::ISystemEvent, 995  
Spinnaker::InterfaceRemovalEvent, 981  
Spinnaker::SystemEvent, 1085  
OnLogEvent  
    Spinnaker::ILoggingEvent, 895  
    Spinnaker::LoggingEvent, 1004  
Open  
    Spinnaker::Video::SpinVideo, 1063, 1065  
open  
    Spinnaker::GenApi::IDevFileStreamBase, 856  
    Spinnaker::GenApi::IDevFileStreamBuf, 858  
    Spinnaker::GenApi::ODevFileStreamBase, 1035  
    Spinnaker::GenApi::ODevFileStreamBuf, 1037  
openFile  
    Spinnaker::GenApi::FileProtocolAdapter, 792  
operator bool  
    Spinnaker::BasePtr, 454  
    Spinnaker::GenApi::CPointer, 731  
operator const char \*  
    Spinnaker::GenICam::gcstring, 810  
operator delete  
    Spinnaker::GenApi::double\_automvector\_t, 756  
    Spinnaker::GenApi::int64\_automvector\_t, 956  
    Spinnaker::GenICam::gcstring, 810  
operator new  
    Spinnaker::GenApi::double\_automvector\_t, 756  
    Spinnaker::GenApi::int64\_automvector\_t, 956  
    Spinnaker::GenICam::gcstring, 810  
operator T\*  
    Spinnaker::BasePtr, 455  
    Spinnaker::GenApi::CPointer, 731  
operator unsigned int  
    Spinnaker::GenApi::Counter, 728  
operator!=  
    INode Interface, 309  
    Spinnaker::Exception, 787  
    Spinnaker::GenApi::CPointer, 731, 732  
    Spinnaker::GenApi::Node, 1022  
    Spinnaker::GenICam::gcstring, 811  
operator<  
    Spinnaker::GenICam::gcstring, 812  
operator<<  
    GCString.h, 1288  
    Spinnaker GenApi Classes, 236  
operator>  
    Spinnaker::GenICam::gcstring, 812  
operator>>  
    GCString.h, 1289  
    Spinnaker GenApi Classes, 237  
operator\*  
    IEnumerator Interface, 290  
    Spinnaker::GenApi::CPointer, 732  
    Spinnaker::GenApi::EnumNode, 772  
    Spinnaker::GenApi::FloatNode, 799  
    Spinnaker::GenApi::IntegerNode, 963  
    Spinnaker::GenApi::StringNode, 1069  
operator()  
    IBoolean Interface, 277  
    Spinnaker::GenApi::CEnumerationTRef, 658

Spinnaker::GenApi::CNodeCallback, 707  
 Spinnaker::GenApi::CPointer, 732  
 Spinnaker::GenApi::CommandNode, 727  
 Spinnaker::GenApi::FloatNode, 799  
 Spinnaker::GenApi::Function\_NodeCallback, 804  
 Spinnaker::GenApi::IntegerNode, 962  
 Spinnaker::GenApi::Member\_NodeCallback, 1012  
 Spinnaker::GenApi::StringNode, 1069  
 operator+  
     Spinnaker::GenICam::gcstring, 813  
 operator++  
     Spinnaker::GenApi::Counter, 728  
 operator+=  
     Spinnaker::GenICam::gcstring, 811  
 operator->  
     Spinnaker::BasePtr, 455  
     Spinnaker::GenApi::CPointer, 733  
 operator--  
     Spinnaker::GenApi::Counter, 728, 729  
 operator=  
     Chunk Data Inference Class, 162  
     IBoolean Interface, 277  
     IEnumeratorT Interface, 292, 293  
     IFloat Interface, 297  
     IInteger Interface, 298  
     Spinnaker GenApi Classes, 237  
     Spinnaker::ArrivalEvent, 449  
     Spinnaker::BasePtr, 455  
     Spinnaker::CameraBase, 624  
     Spinnaker::CameraList, 632  
     Spinnaker::DeviceEvent, 755  
     Spinnaker::Event, 780  
     Spinnaker::Exception, 787  
     Spinnaker::GenApi::BooleanNode, 460  
     Spinnaker::GenApi::CEnumerationTRef, 659  
     Spinnaker::GenApi::CFloatPtr, 680  
     Spinnaker::GenApi::CNodeMapFactory, 715  
     Spinnaker::GenApi::CNodeMapRef, 718  
     Spinnaker::GenApi::CPointer, 733  
     Spinnaker::GenApi::EnumNode, 772  
     Spinnaker::GenApi::FloatNode, 799  
     Spinnaker::GenApi::IntegerNode, 963  
     Spinnaker::GenApi::StringNode, 1069  
     Spinnaker::GenApi::double\_autovector\_t, 756  
     Spinnaker::GenApi::int64\_autovector\_t, 956  
     Spinnaker::GenICam::gcstring, 812  
     Spinnaker::IArrivalEvent, 827  
     Spinnaker::ICameraBase, 834  
     Spinnaker::ICameraList, 839  
     Spinnaker::IDeviceEvent, 861  
     Spinnaker::IImageEvent, 875  
     Spinnaker::IInterface, 882  
     Spinnaker::IInterfaceArrivalEvent, 886  
     Spinnaker::IInterfaceEvent, 888  
     Spinnaker::IInterfaceList, 890  
     Spinnaker::IInterfaceRemovalEvent, 893  
     Spinnaker::ILoggingEvent, 895  
     Spinnaker::IRemovalEvent, 987  
 Spinnaker::ISystem, 991  
 Spinnaker::ISystemEvent, 996  
 Spinnaker::ImageEvent, 926  
 Spinnaker::ImagePtr, 929  
 Spinnaker::ImageStatistics, 935  
 Spinnaker::InterfaceArrivalEvent, 971  
 Spinnaker::InterfaceEvent, 973  
 Spinnaker::InterfaceList, 976  
 Spinnaker::InterfaceRemovalEvent, 981  
 Spinnaker::LoggingEvent, 1005  
 Spinnaker::RemovalEvent, 1059  
 Spinnaker::SystemEvent, 1085  
 operator==  
     INode Interface, 309  
     Spinnaker::BasePtr, 456  
     Spinnaker::Exception, 787  
     Spinnaker::GenApi::CFeatureBag, 678  
     Spinnaker::GenApi::CPointer, 733  
     Spinnaker::GenApi::Node, 1023  
     Spinnaker::GenICam::gcstring, 812  
 operator[]  
     Spinnaker::CameraList, 632  
     Spinnaker::GenApi::double\_autovector\_t, 757  
     Spinnaker::GenApi::int64\_autovector\_t, 957  
     Spinnaker::ICameraList, 839  
     Spinnaker::IInterfaceList, 891  
     Spinnaker::InterfaceList, 977  
 ostream\_type  
     Spinnaker::GenApi::ODevFileStreamBase, 1035  
 overflow  
     Spinnaker::GenApi::ODevFileStreamBuf, 1038  
 PGMOOption, 1038  
     Spinnaker::PGMOOption, 1039  
 PMEMBERFUNC  
     Spinnaker::GenApi::Member\_NodeCallback, 1011  
 PNGOption, 1039  
     Spinnaker::PNGOption, 1040  
 POEStatus  
     Spinnaker::TransportLayerInterface, 1109  
 POEStatusEnum  
     TransportLayerDefs Class, 213  
 PPMOption, 1051  
     Spinnaker::PPMOption, 1052  
 PacketResendRequestCount  
     Spinnaker::Camera, 586  
 PayloadSize  
     Spinnaker::Camera, 587  
 PayloadTypeInfoIDs  
     Spinnaker Definitions, 197  
 pbackfail  
     Spinnaker::GenApi::IDevFileStreamBuf, 859  
 PersistFeature  
     Spinnaker::GenApi, 436  
     Spinnaker::GenApi::CFeatureBag, 678  
 Persistence Class, 343  
 PixelColorFilter  
     Spinnaker::Camera, 587  
 PixelColorFilterEnums

CameraDefs Class, 123  
PixelDynamicRangeMax  
    Spinnaker::Camera, 587  
PixelDynamicRangeMin  
    Spinnaker::Camera, 587  
PixelFormat  
    Spinnaker::Camera, 587  
PixelFormatEnums  
    CameraDefs Class, 123  
PixelFormatInfoID  
    Spinnaker::Camera, 588  
PixelFormatInfoSelector  
    Spinnaker::Camera, 588  
PixelFormatInfoSelectorEnums  
    CameraDefs Class, 129  
PixelFormatIntType  
    Spinnaker Definitions, 198  
PixelFormatNamespaceID  
    Spinnaker Definitions, 198  
PixelSize  
    Spinnaker::Camera, 588  
PixelSizeEnums  
    CameraDefs Class, 134  
Pointer Class, 344  
    CBasePtr, 345  
    CBooleanPtr, 345  
    CCategoryPtr, 346  
    CChunkPortPtr, 346  
    CCommandPtr, 346  
    CDeviceInfoPtr, 346  
    CEnumEntryPtr, 346  
    CEnumerationPtr, 346  
    CIntegerPtr, 347  
    CNodeMapDynPtr, 347  
    CNodeMapPtr, 347  
    CNodePtr, 347  
    CPortConstructPtr, 347  
    CPortPtr, 347  
    CPortRecorderPtr, 348  
    CPortReplayPtr, 348  
    CPortWriteListPtr, 348  
    CRegisterPtr, 348  
    CSelectorPtr, 348  
    CStringPtr, 348  
    CValuePtr, 349  
    GetInterfaceName, 349  
    IsAvailable, 349  
    IsImplemented, 349  
    IsReadable, 349  
    IsWritable, 349  
PolarizationQuadrant  
    Spinnaker::ImageUtilityPolarization, 946  
Poll  
    INodeMap Interface, 313  
    Spinnaker::GenApi::NodeMap, 1032  
PopulateAdapterImplInfo  
    AdapterConfig, 381  
PortImpl Class, 350  
    PortNode, 1041  
        Spinnaker::GenApi::PortNode, 1043  
    PortNode Class, 351  
        CPortRef, 351  
    PortRecorder, 1046  
        Spinnaker::GenApi::PortRecorder, 1047  
    PortRecorder Class, 352  
        CPortRecorderRef, 352  
    PortReplay, 1049  
        Spinnaker::GenApi::PortReplay, 1050  
    PortReplay Class, 353  
    PortWriteList Class, 354  
PowerSupplyCurrent  
    Spinnaker::Camera, 588  
PowerSupplyVoltage  
    Spinnaker::Camera, 588  
Prefix  
    U3V\_COMMAND\_HEADER, 1119  
Preprocess  
    Spinnaker::GenApi::CNodeMapFactory, 715  
PreprocessXMLFromFile  
    INodeMapDyn Interface, 317  
PreprocessXMLFromZIPFile  
    INodeMapDyn Interface, 317  
progressive  
    Spinnaker::JPEGOption, 997  
quality  
    Spinnaker::JPEGOption, 997  
    Spinnaker::JPG2Option, 998  
    Spinnaker::Video::MJPGOption, 1013  
radius  
    Chunk Data Inference Class, 164  
rdbuf  
    Spinnaker::GenApi::IDevFileStreamBase, 857  
    Spinnaker::GenApi::ODevFileStreamBase, 1036  
Read  
    Spinnaker::GenApi::CChunkPort, 654  
    Spinnaker::GenApi::CEventPort, 675  
    Spinnaker::GenApi::CPortImpl, 736  
    Spinnaker::GenApi::CRegisterPortImpl, 743  
    Spinnaker::GenApi::CTestPortStruct, 750  
    Spinnaker::GenApi::PortNode, 1044  
read  
    Spinnaker::GenApi::FileProtocolAdapter, 792  
ReadPort  
    Spinnaker::CameraBase, 624  
    Spinnaker::ICameraBase, 834  
ReadRegister  
    Spinnaker::GenApi::CRegisterPortImpl, 743  
receiveBuffers  
    AdapterConfig::AdapterInfo, 445  
receiveBuffersMax  
    AdapterConfig::AdapterInfo, 445  
receiveBuffersMin  
    AdapterConfig::AdapterInfo, 445  
receiveBuffersRegKey  
    AdapterConfig::AdapterInfo, 446

receiveBuffersStep  
     AdapterConfig::AdapterInfo, 446

rect  
     Chunk Data Inference Class, 164

Reference Interfaces, 355  
     SetNumEnums, 355

RegionDestination  
     Spinnaker::Camera, 588

RegionDestinationEnums  
     CameraDefs Class, 135

RegionMode  
     Spinnaker::Camera, 589

RegionModeEnums  
     CameraDefs Class, 135

RegionSelector  
     Spinnaker::Camera, 589

RegionSelectorEnums  
     CameraDefs Class, 136

Register  
     NodeCallback Class, 338

RegisterCallback  
     INode Interface, 309  
     Spinnaker::GenApi::Node, 1023

RegisterEvent  
     Spinnaker::CameraBase, 624, 625  
     Spinnaker::ICameraBase, 834  
     Spinnaker::IInterface, 883  
     Spinnaker::ISystem, 991  
     Spinnaker::Interface, 967  
     Spinnaker::System, 1078

RegisterImageEvent  
     Spinnaker::IDataStream, 853

RegisterInterfaceEvent  
     Spinnaker::ISystem, 991  
     Spinnaker::System, 1078

RegisterLoggingEvent  
     Spinnaker::ISystem, 991  
     Spinnaker::System, 1078

RegisterNode, 1053  
     Spinnaker::GenApi::RegisterNode, 1054, 1055

RegisterNode Class, 356  
     CRegisterRef, 356

RegisterPortImpl Class, 357

Release  
     Spinnaker::IImage, 871  
     Spinnaker::Image, 918

ReleaseCameraDescriptionFileData  
     Spinnaker::GenApi::CNodeMapFactory, 715

ReleaselImage  
     Spinnaker::IDataStream, 853

ReleaseInstance  
     Spinnaker::ISystem, 991  
     Spinnaker::System, 1079

RemovalEvent, 1057  
     Spinnaker::RemovalEvent, 1058

RemovalEvent Class, 185

RemoveByIndex  
     Spinnaker::CameraList, 633

Spinnaker::ICameraList, 839

RemoveBySerial  
     Spinnaker::CameraList, 633  
     Spinnaker::ICameraList, 839

ReplaceEnvironmentVariables  
     GCUtilities Utility, 273

Replay  
     IPortRecorder Interface, 324  
     Spinnaker::GenApi::CPortImpl, 736  
     Spinnaker::GenApi::CPortWriteList, 739  
     Spinnaker::GenApi::PortNode, 1044  
     Spinnaker::GenApi::PortReplay, 1051

ReqId  
     GVCP\_REQUEST\_HEADER, 823  
     U3V\_COMMAND\_HEADER, 1120

Reserved  
     U3V\_EVENT\_DATA, 1120

reserved  
     Spinnaker::BMPOption, 458  
     Spinnaker::JPEGOption, 997  
     Spinnaker::JPG2Option, 998  
     Spinnaker::PGMOption, 1039  
     Spinnaker::PNGOption, 1040  
     Spinnaker::PPMOption, 1052  
     Spinnaker::TIFFOption, 1089  
     Spinnaker::Video::AVIOption, 452  
     Spinnaker::Video::H264Option, 825  
     Spinnaker::Video::MJPGOption, 1013

ReservedOrEventSize  
     GVCP\_EVENT\_ITEM\_BASIC, 816  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 817  
     GVCP\_EVENT\_ITEM, 815

ResetImage  
     Spinnaker::IImage, 871  
     Spinnaker::Image, 919

ResetStatistics  
     Spinnaker::GenApi::CTestPortStruct, 750

resize  
     Spinnaker::GenICam::gcstring, 812

Restore  
     ISelectorDigit Interface, 330  
     Spinnaker::GenApi::CSelectorSet, 746

RetrieveAllAdapters  
     AdapterConfig, 381

ReverseX  
     Spinnaker::Camera, 589

ReverseY  
     Spinnaker::Camera, 589

RevokelImages  
     Spinnaker::IDataStream, 854

RgbTransformLightSource  
     Spinnaker::Camera, 589

RgbTransformLightSourceEnums  
     CameraDefs Class, 136

RO  
     Types Enums, 368

rotatedRect  
     Chunk Data Inference Class, 164

rotationAngle  
    Chunk Data Inference Class, 164

RW  
    Types Enums, 368

SET\_GUID  
    Spinnaker::GenApi, 436

SPINNAKER\_API\_ABSTRACT  
    Spinnaker Platform, 202

SPINNAKER\_API  
    Spinnaker Platform, 202

SPINNAKER\_LOCAL  
    Spinnaker Platform, 202

SPINUPDATE\_API  
    SpinUpdate.h, 1385

Saturation  
    Spinnaker::Camera, 590

SaturationEnable  
    Spinnaker::Camera, 590

Save  
    Spinnaker::IImage, 872, 873  
    Spinnaker::Image, 920–922

Scan3dAxisMax  
    Spinnaker::Camera, 590

Scan3dAxisMin  
    Spinnaker::Camera, 590

Scan3dCoordinateOffset  
    Spinnaker::Camera, 590

Scan3dCoordinateReferenceSelector  
    Spinnaker::Camera, 591

Scan3dCoordinateReferenceSelectorEnums  
    CameraDefs Class, 136

Scan3dCoordinateReferenceValue  
    Spinnaker::Camera, 591

Scan3dCoordinateScale  
    Spinnaker::Camera, 591

Scan3dCoordinateSelector  
    Spinnaker::Camera, 591

Scan3dCoordinateSelectorEnums  
    CameraDefs Class, 137

Scan3dCoordinateSystem  
    Spinnaker::Camera, 591

Scan3dCoordinateSystemEnums  
    CameraDefs Class, 137

Scan3dCoordinateSystemReference  
    Spinnaker::Camera, 591

Scan3dCoordinateSystemReferenceEnums  
    CameraDefs Class, 137

Scan3dCoordinateTransformSelector  
    Spinnaker::Camera, 592

Scan3dCoordinateTransformSelectorEnums  
    CameraDefs Class, 138

Scan3dDistanceUnit  
    Spinnaker::Camera, 592

Scan3dDistanceUnitEnums  
    CameraDefs Class, 138

Scan3dInvalidDataFlag  
    Spinnaker::Camera, 592

Scan3dInvalidHeaderValue

    Spinnaker::Camera, 592

Scan3dOutputMode  
    Spinnaker::Camera, 592

Scan3dOutputModeEnums  
    CameraDefs Class, 139

Scan3dTransformValue  
    Spinnaker::Camera, 592

SelectorSet Class, 358

SendActionCommand  
    Spinnaker::IInterface, 883  
    Spinnaker::ISystem, 991  
    Spinnaker::Interface, 967  
    Spinnaker::System, 1079

SensorDescription  
    Spinnaker::Camera, 593

SensorDigitizationTaps  
    Spinnaker::Camera, 593

SensorDigitizationTapsEnums  
    CameraDefs Class, 139

SensorHeight  
    Spinnaker::Camera, 593

SensorShutterMode  
    Spinnaker::Camera, 593

SensorShutterModeEnums  
    CameraDefs Class, 140

SensorTaps  
    Spinnaker::Camera, 593

SensorTapsEnums  
    CameraDefs Class, 140

SensorWidth  
    Spinnaker::Camera, 593

SequencerConfigurationMode  
    Spinnaker::Camera, 594

SequencerConfigurationModeEnums  
    CameraDefs Class, 141

SequencerConfigurationValid  
    Spinnaker::Camera, 594

SequencerConfigurationValidEnums  
    CameraDefs Class, 141

SequencerFeatureEnable  
    Spinnaker::Camera, 594

SequencerMode  
    Spinnaker::Camera, 594

SequencerModeEnums  
    CameraDefs Class, 141

SequencerPathSelector  
    Spinnaker::Camera, 594

SequencerSetActive  
    Spinnaker::Camera, 595

SequencerSetLoad  
    Spinnaker::Camera, 595

SequencerSetNext  
    Spinnaker::Camera, 595

SequencerSetSave  
    Spinnaker::Camera, 595

SequencerSetSelector  
    Spinnaker::Camera, 595

SequencerSetStart

Spinnaker::Camera, 596  
 SequencerSetValid  
     Spinnaker::Camera, 596  
 SequencerSetValidEnums  
     CameraDefs Class, 141  
 SequencerTriggerActivation  
     Spinnaker::Camera, 596  
 SequencerTriggerActivationEnums  
     CameraDefs Class, 142  
 SequencerTriggerSource  
     Spinnaker::Camera, 596  
 SequencerTriggerSourceEnums  
     CameraDefs Class, 142  
 SerialPortBaudRate  
     Spinnaker::Camera, 596  
 SerialPortBaudRateEnums  
     CameraDefs Class, 142  
 SerialPortDataBits  
     Spinnaker::Camera, 597  
 SerialPortParity  
     Spinnaker::Camera, 597  
 SerialPortParityEnums  
     CameraDefs Class, 143  
 SerialPortSelector  
     Spinnaker::Camera, 597  
 SerialPortSelectorEnums  
     CameraDefs Class, 143  
 SerialPortSource  
     Spinnaker::Camera, 597  
 SerialPortSourceEnums  
     CameraDefs Class, 144  
 SerialPortStopBits  
     Spinnaker::Camera, 597  
 SerialPortStopBitsEnums  
     CameraDefs Class, 144  
 SerialReceiveFramingErrorCount  
     Spinnaker::Camera, 597  
 SerialReceiveParityErrorCount  
     Spinnaker::Camera, 598  
 SerialReceiveQueueClear  
     Spinnaker::Camera, 598  
 SerialReceiveQueueCurrentCharacterCount  
     Spinnaker::Camera, 598  
 SerialReceiveQueueMaxCharacterCount  
     Spinnaker::Camera, 598  
 SerialTransmitQueueCurrentCharacterCount  
     Spinnaker::Camera, 598  
 SerialTransmitQueueMaxCharacterCount  
     Spinnaker::Camera, 598  
 Set  
     Spinnaker::GenApi::RegisterNode, 1056  
 SetBufferOwnership  
     Spinnaker::CameraBase, 625  
     Spinnaker::ICameraBase, 834  
 SetChannelStatus  
     Spinnaker::IImageStatistics, 879  
     Spinnaker::ImageStatistics, 935  
 SetChunks  
     Spinnaker::ChunkData, 698  
     Spinnaker::IChunkData, 848  
 SetCookie  
     IPortRecorder Interface, 325  
     Spinnaker::GenApi::CPortWriteList, 740  
 SetDefaultColorProcessing  
     Spinnaker::Image, 922  
 SetEnumReference  
     Spinnaker::GenApi::CEnumerationTRef, 659  
 SetEventPayload  
     Spinnaker::Event, 780  
 SetEventType  
     Spinnaker::Event, 780  
 SetFirst  
     Spinnaker::GenApi::CSelectorSet, 746  
 SetGenICamCLProtocolFolder  
     GCUtilities Utility, 274  
 SetGenICamCacheFolder  
     GCUtilities Utility, 273  
 SetGenICamLogConfig  
     GCUtilities Utility, 274  
 SetHeatmapColorGradient  
     Spinnaker::ImageUtilityHeatmap, 944  
 SetHeatmapRange  
     Spinnaker::ImageUtilityHeatmap, 944  
 SetInfo  
     Spinnaker::GenApi::CFeatureBag, 678  
 SetIntValue  
     IEnumeration Interface, 291  
     Spinnaker::GenApi::EnumNode, 772  
 SetLoggingEventPriorityLevel  
     Spinnaker::ISystem, 992  
     Spinnaker::System, 1080  
 SetMaximumFileSize  
     Spinnaker::Video::SpinVideo, 1065  
 SetMessageCallback  
     SpinUpdate.h, 1385  
 SetNext  
     ISelectorDigit Interface, 330  
     Spinnaker::GenApi::CSelectorSet, 746  
 SetNodeHandle  
     Spinnaker::GenApi::Node, 1023  
 SetNodeMap  
     Spinnaker::GenApi::Node, 1023  
 SetNumEnums  
     Reference Interfaces, 355  
     Spinnaker::GenApi::CEnumerationTRef, 659  
 SetPortImpl  
     Spinnaker::GenApi::CChunkPort, 654  
     Spinnaker::GenApi::CEventPort, 675  
     Spinnaker::GenApi::CPortImpl, 736  
     Spinnaker::GenApi::CRegisterPortImpl, 743  
     Spinnaker::GenApi::PortNode, 1044  
 SetProgressCallback  
     SpinUpdate.h, 1385  
 SetReference  
     Spinnaker::GenApi::BooleanNode, 460  
     Spinnaker::GenApi::CEnumerationTRef, 659

Spinnaker::GenApi::CategoryNode, 637  
Spinnaker::GenApi::CommandNode, 727  
Spinnaker::GenApi::EnumEntryNode, 767  
Spinnaker::GenApi::EnumNode, 772  
Spinnaker::GenApi::FloatNode, 799  
Spinnaker::GenApi::FloatRegNode, 802  
Spinnaker::GenApi::IntRegNode, 984  
Spinnaker::GenApi::IntegerNode, 963  
Spinnaker::GenApi::Node, 1023  
Spinnaker::GenApi::PortNode, 1045  
Spinnaker::GenApi::PortRecorder, 1048  
Spinnaker::GenApi::PortReplay, 1051  
Spinnaker::GenApi::RegisterNode, 1056  
Spinnaker::GenApi::StringNode, 1069  
Spinnaker::GenApi::StringRegNode, 1072  
Spinnaker::GenApi::ValueNode, 1124  
SetUserBuffers  
    Spinnaker::CameraBase, 625, 626  
    Spinnaker::ICameraBase, 835  
SetValue  
    Spinnaker::GenApi::BooleanNode, 461  
    Spinnaker::GenApi::CEnumerationTRef, 660  
    Spinnaker::GenApi::FloatNode, 799  
    Spinnaker::GenApi::IntegerNode, 963  
    Spinnaker::GenApi::StringNode, 1069  
Sharpening  
    Spinnaker::Camera, 599  
SharpeningAuto  
    Spinnaker::Camera, 599  
SharpeningEnable  
    Spinnaker::Camera, 599  
SharpeningThreshold  
    Spinnaker::Camera, 599  
Signed  
    Types Enums, 373  
SingleChunkData\_t, 1059  
    ChunkID, 1059  
    ChunkLength, 1059  
    ChunkOffset, 1059  
SingleChunkDataStr\_t, 1060  
    ChunkID, 1060  
    ChunkLength, 1060  
    ChunkOffset, 1060  
size  
    Spinnaker::GenApi::double\_automap\_t, 757  
    Spinnaker::GenApi::int64\_automap\_t, 957  
    Spinnaker::GenICam::gcstring, 812  
SoftwareSignalPulse  
    Spinnaker::Camera, 600  
SoftwareSignalSelector  
    Spinnaker::Camera, 600  
SoftwareSignalSelectorEnums  
    CameraDefs Class, 144  
SourceCount  
    Spinnaker::Camera, 600  
SourceDataRange  
    Spinnaker::ImageUtility, 937  
SourceSelector  
    Spinnaker::Camera, 600  
    SourceSelectorEnums  
        CameraDefs Class, 145  
    SpinTestCamera, 1061  
    SpinTestCamera Class, 359  
    SpinUpdate.h  
        GetErrorMessage, 1385  
        SPINUPDATE\_API, 1385  
        SetMessageCallback, 1385  
        SetProgressCallback, 1385  
        UpdateFirmware, 1385  
        UpdateFirmwareConsole, 1386  
        UpdateFirmwareGUI, 1386  
        UpdatorMessageCallback, 1386  
        UpdatorProgressCallback, 1386  
    SpinVideo, 1061  
        Spinnaker::Video::SpinVideo, 1062  
    Spinnaker, 382  
    Spinnaker Classes, 31  
        InferenceBoxType, 34  
    Spinnaker Definitions, 189  
        ActionCommandStatus, 193  
        BufferOwnership, 193  
        ColorProcessingAlgorithm, 193  
        DEPRECATED\_ENUM, 200  
        Error, 194  
        EventType, 195  
        HeatMapColor, 201  
        ImageFormat, 195  
        ImageStatus, 197  
        PayloadTypeInfoIDs, 197  
        PixelFormatIntType, 198  
        PixelFormatNamespaceID, 198  
        SpinnakerLogLevel, 199  
        StatisticsChannel, 199  
    Spinnaker Event Classes, 28  
    Spinnaker GenApi Classes, 227  
        \_ClearXMLCache, 233  
        \_Connect, 233, 234  
        \_Destroy, 234  
        \_GetDeviceName, 234  
        \_GetNode, 234  
        \_GetNodes, 234  
        \_GetSupportedSchemaVersions, 234  
        \_InvalidateNodes, 234  
        \_LoadXMLFromFile, 235  
        \_LoadXMLFromFileInject, 235  
        \_LoadXMLFromString, 235  
        \_LoadXMLFromStringInject, 235  
        \_LoadXMLFromZIPData, 235  
        \_LoadXMLFromZIPFile, 235  
        \_Poll, 235  
        ~CNodeMapRefT, 237  
    CNodeMapRef, 233  
    CNodeMapRefT, 236  
    CNodeRef, 233  
    CSelectorRef, 233  
    CastToIDestroy, 236

EatComments, 236  
 operator<<, 236  
 operator>>, 237  
 operator=, 237  
**Spinnaker GenApi Enums**, 364  
**Spinnaker GenApi Interfaces**, 239  
 CallbackHandleType, 240  
 NodeList\_t, 240  
**Spinnaker GenApi Utilities**, 269  
**Spinnaker Headers**, 186  
 EVENT\_TIMEOUT\_INFINITE, 187  
 EVENT\_TIMEOUT\_NONE, 187  
**Spinnaker Platform**, 202  
 SPINNAKER\_API\_ABSTRACT, 202  
 SPINNAKER\_API, 202  
 SPINNAKER\_LOCAL, 202  
**Spinnaker QuickSpin Classes**, 208  
**Spinnaker Video Class**, 203  
**Spinnaker Video Definitions**, 204  
**Spinnaker.h**, 188  
**Spinnaker::ActionCommandResult**  
 DeviceAddress, 441  
 Status, 441  
**Spinnaker::ArrivalEvent**  
 ~ArrivalEvent, 448  
 ArrivalEvent, 448  
 OnDeviceArrival, 448  
 operator=, 449  
**Spinnaker::BMPOption**  
 BMPOption, 457  
 indexedColor\_8bit, 457  
 reserved, 458  
**Spinnaker::BasePtr**  
 ~BasePtr, 454  
 BasePtr, 453, 454  
 get, 454  
 IsValid, 454  
 m\_pT, 456  
 operator bool, 454  
 operator T\*, 455  
 operator->, 455  
 operator=, 455  
 operator==, 456  
**Spinnaker::Camera**  
 ~Camera, 491  
 aPAUSEMACCtrlFramesReceived, 497  
 aPAUSEMACCtrlFramesTransmitted, 497  
 AasRoiEnable, 492  
 AasRoiHeight, 492  
 AasRoiOffsetX, 492  
 AasRoiOffsetY, 493  
 AasRoiWidth, 493  
 AcquisitionAbort, 493  
 AcquisitionArm, 493  
 AcquisitionBurstFrameCount, 493  
 AcquisitionFrameCount, 494  
 AcquisitionFrameRate, 494  
 AcquisitionFrameRateEnable, 494  
 AcquisitionLineRate, 494  
 AcquisitionMode, 494  
 AcquisitionResultingFrameRate, 495  
 AcquisitionStart, 495  
 AcquisitionStatus, 495  
 AcquisitionStatusSelector, 495  
 AcquisitionStop, 495  
 ActionDeviceKey, 495  
 ActionGroupKey, 496  
 ActionGroupMask, 496  
 ActionQueueSize, 496  
 ActionSelector, 496  
 ActionUnconditionalMode, 496  
 AdaptiveCompressionEnable, 496  
 AdcBitDepth, 497  
 AutoAlgorithmSelector, 497  
 AutoExposureControlLoopDamping, 497  
 AutoExposureControlPriority, 498  
 AutoExposureEVCompensation, 498  
 AutoExposureExposureTimeLowerLimit, 498  
 AutoExposureExposureTimeUpperLimit, 499  
 AutoExposureGainLowerLimit, 499  
 AutoExposureGainUpperLimit, 499  
 AutoExposureGreyValueLowerLimit, 499  
 AutoExposureGreyValueUpperLimit, 499  
 AutoExposureLightingMode, 500  
 AutoExposureMeteringMode, 500  
 AutoExposureTargetGreyValue, 500  
 AutoExposureTargetGreyValueAuto, 501  
 BalanceRatio, 501  
 BalanceRatioSelector, 501  
 BalanceWhiteAuto, 502  
 BalanceWhiteAutoDamping, 502  
 BalanceWhiteAutoLowerLimit, 502  
 BalanceWhiteAutoProfile, 502  
 BalanceWhiteAutoUpperLimit, 503  
 BinningHorizontal, 503  
 BinningHorizontalMode, 503  
 BinningSelector, 503  
 BinningVertical, 503  
 BinningVerticalMode, 504  
 BlackLevel, 504  
 BlackLevelAuto, 504  
 BlackLevelAutoBalance, 504  
 BlackLevelClampingEnable, 504  
 BlackLevelRaw, 505  
 BlackLevelSelector, 505  
 Camera, 491  
 ChunkBlackLevel, 505  
 ChunkBlackLevelSelector, 505  
 ChunkCRC, 506  
 ChunkCounterSelector, 505  
 ChunkCounterValue, 506  
 ChunkEnable, 506  
 ChunkEncoderSelector, 506  
 ChunkEncoderStatus, 506  
 ChunkEncoderValue, 506  
 ChunkExposureEndLineStatusAll, 507

ChunkExposureTime, 507  
ChunkExposureTimeSelector, 507  
ChunkFrameID, 507  
ChunkGain, 507  
ChunkGainSelector, 507  
ChunkHeight, 508  
ChunkImage, 508  
ChunkImageComponent, 508  
ChunkInferenceBoundingBoxResult, 508  
ChunkInferenceConfidence, 508  
ChunkInferenceResult, 508  
ChunkLinePitch, 509  
ChunkLineStatusAll, 509  
ChunkModeActive, 509  
ChunkOffsetX, 509  
ChunkOffsetY, 509  
ChunkPartSelector, 509  
ChunkPixelDynamicRangeMax, 510  
ChunkPixelDynamicRangeMin, 510  
ChunkPixelFormat, 510  
ChunkRegionID, 510  
ChunkScan3dAxisMax, 510  
ChunkScan3dAxisMin, 510  
ChunkScan3dCoordinateOffset, 511  
ChunkScan3dCoordinateReferenceSelector, 511  
ChunkScan3dCoordinateReferenceValue, 511  
ChunkScan3dCoordinateScale, 511  
ChunkScan3dCoordinateSelector, 511  
ChunkScan3dCoordinateSystem, 511  
ChunkScan3dCoordinateSystemReference, 512  
ChunkScan3dCoordinateTransformSelector, 512  
ChunkScan3dDistanceUnit, 512  
ChunkScan3dInvalidDataFlag, 512  
ChunkScan3dInvalidHeaderValue, 512  
ChunkScan3dOutputMode, 512  
ChunkScan3dTransformValue, 513  
ChunkScanLineSelector, 513  
ChunkSelector, 513  
ChunkSequencerSetActive, 513  
ChunkSerialData, 513  
ChunkSerialDataLength, 513  
ChunkSerialReceiveOverflow, 514  
ChunkSourceID, 514  
ChunkStreamChannelID, 514  
ChunkTimerSelector, 514  
ChunkTimerValue, 514  
ChunkTimestamp, 514  
ChunkTimestampLatchValue, 515  
ChunkTransferBlockID, 515  
ChunkTransferQueueCurrentBlockCount, 515  
ChunkTransferStreamID, 515  
ChunkWidth, 515  
CIConfiguration, 515  
CITimeSlotsCount, 516  
ColorTransformationEnable, 516  
ColorTransformationSelector, 516  
ColorTransformationValue, 516  
ColorTransformationValueSelector, 516  
CompressionRatio, 517  
CounterDelay, 517  
CounterDuration, 517  
CounterEventActivation, 517  
CounterEventSource, 517  
CounterReset, 517  
CounterResetActivation, 518  
CounterResetSource, 518  
CounterSelector, 518  
CounterStatus, 518  
CounterTriggerActivation, 518  
CounterTriggerSource, 518  
CounterValue, 519  
CounterValueAtReset, 519  
CxpConnectionSelector, 519  
CxpConnectionTestErrorCount, 519  
CxpConnectionTestMode, 519  
CxpConnectionTestPacketCount, 519  
CxpLinkConfiguration, 520  
CxpLinkConfigurationPreferred, 520  
CxpLinkConfigurationStatus, 520  
CxpPoCxpAuto, 520  
CxpPoCxpStatus, 520  
CxpPoCxpTripReset, 520  
CxpPoCxpTurnOff, 521  
DecimationHorizontal, 521  
DecimationHorizontalMode, 521  
DecimationSelector, 521  
DecimationVertical, 522  
DecimationVerticalMode, 522  
DefectCorrectStaticEnable, 522  
DefectCorrectionMode, 522  
DefectTableApply, 523  
DefectTableCoordinateX, 523  
DefectTableCoordinateY, 523  
DefectTableFactoryRestore, 523  
DefectTableIndex, 524  
DefectTablePixelCount, 524  
DefectTableSave, 524  
Deinterlacing, 524  
DeviceCharacterSet, 524  
DeviceClockFrequency, 525  
DeviceClockSelector, 525  
DeviceConnectionSelector, 525  
DeviceConnectionSpeed, 525  
DeviceConnectionStatus, 525  
DeviceEventChannelCount, 525  
DeviceFamilyName, 526  
DeviceFeaturePersistenceEnd, 526  
DeviceFeaturePersistenceStart, 526  
DeviceFirmwareVersion, 526  
DeviceGenCPVersionMajor, 526  
DeviceGenCPVersionMinor, 526  
DeviceID, 527  
DeviceIndicatorMode, 527  
DeviceLinkBandwidthReserve, 527  
DeviceLinkCommandTimeout, 527  
DeviceLinkConnectionCount, 527

DeviceLinkCurrentThroughput, 527  
DeviceLinkHeartbeatMode, 528  
DeviceLinkHeartbeatTimeout, 528  
DeviceLinkSelector, 528  
DeviceLinkSpeed, 528  
DeviceLinkThroughputLimit, 528  
DeviceLinkThroughputLimitMode, 529  
DeviceManifestEntrySelector, 529  
DeviceManifestPrimaryURL, 529  
DeviceManifestSchemaMajorVersion, 529  
DeviceManifestSchemaMinorVersion, 529  
DeviceManifestSecondaryURL, 530  
DeviceManifestXMLMajorVersion, 530  
DeviceManifestXMLMinorVersion, 530  
DeviceManifestXMLSubMinorVersion, 530  
DeviceManufacturerInfo, 530  
DeviceMaxThroughput, 530  
DeviceModelName, 531  
DevicePowerSupplySelector, 531  
DeviceRegistersCheck, 531  
DeviceRegistersEndianness, 531  
DeviceRegistersStreamingEnd, 531  
DeviceRegistersStreamingStart, 532  
DeviceRegistersValid, 532  
DeviceReset, 532  
DeviceSFNCVersionMajor, 533  
DeviceSFNCVersionMinor, 533  
DeviceSFNCVersionSubMinor, 533  
DeviceScanType, 532  
DeviceSerialNumber, 532  
DeviceSerialPortBaudRate, 532  
DeviceSerialPortSelector, 533  
DeviceStreamChannelCount, 533  
DeviceStreamChannelEndianness, 533  
DeviceStreamChannelLink, 534  
DeviceStreamChannelPacketSize, 534  
DeviceStreamChannelSelector, 534  
DeviceStreamChannelType, 534  
DeviceTLType, 535  
DeviceTLVersionMajor, 535  
DeviceTLVersionMinor, 535  
DeviceTLVersionSubMinor, 535  
DeviceTapGeometry, 534  
DeviceTemperature, 534  
DeviceTemperatureSelector, 535  
DeviceType, 536  
DeviceUptime, 536  
DeviceUserID, 536  
DeviceVendorName, 536  
DeviceVersion, 536  
EncoderDivider, 536  
EncoderMode, 537  
EncoderOutputMode, 537  
EncoderReset, 537  
EncoderResetActivation, 537  
EncoderResetSource, 537  
EncoderSelector, 537  
EncoderSourceA, 538  
EncoderSourceB, 538  
EncoderStatus, 538  
EncoderTimeout, 538  
EncoderValue, 538  
EncoderValueAtReset, 538  
EnumerationCount, 539  
EventAcquisitionEnd, 539  
EventAcquisitionEndFrameID, 539  
EventAcquisitionEndTimestamp, 539  
EventAcquisitionError, 539  
EventAcquisitionErrorFrameID, 539  
EventAcquisitionErrorTimestamp, 540  
EventAcquisitionStart, 540  
EventAcquisitionStartFrameID, 540  
EventAcquisitionStartTimestamp, 540  
EventAcquisitionTransferEnd, 540  
EventAcquisitionTransferEndFrameID, 540  
EventAcquisitionTransferEndTimestamp, 541  
EventAcquisitionTransferStart, 541  
EventAcquisitionTransferStartFrameID, 541  
EventAcquisitionTransferStartTimestamp, 541  
EventAcquisitionTrigger, 541  
EventAcquisitionTriggerFrameID, 541  
EventAcquisitionTriggerTimestamp, 542  
EventActionLate, 542  
EventActionLateFrameID, 542  
EventActionLateTimestamp, 542  
EventCounter0End, 542  
EventCounter0EndFrameID, 542  
EventCounter0EndTimestamp, 543  
EventCounter0Start, 543  
EventCounter0StartFrameID, 543  
EventCounter0StartTimestamp, 543  
EventCounter1End, 543  
EventCounter1EndFrameID, 543  
EventCounter1EndTimestamp, 544  
EventCounter1Start, 544  
EventCounter1StartFrameID, 544  
EventCounter1StartTimestamp, 544  
EventEncoder0Restarted, 544  
EventEncoder0RestartedFrameID, 544  
EventEncoder0RestartedTimestamp, 545  
EventEncoder0Stopped, 545  
EventEncoder0StoppedFrameID, 545  
EventEncoder0StoppedTimestamp, 545  
EventEncoder1Restarted, 545  
EventEncoder1RestartedFrameID, 545  
EventEncoder1RestartedTimestamp, 546  
EventEncoder1Stopped, 546  
EventEncoder1StoppedFrameID, 546  
EventEncoder1StoppedTimestamp, 546  
EventError, 546  
EventErrorCode, 546  
EventErrorFrameID, 547  
EventErrorTimestamp, 547  
EventExposureEnd, 547  
EventExposureEndFrameID, 547  
EventExposureEndTimestamp, 547

EventExposureStart, 547  
EventExposureStartFrameID, 548  
EventExposureStartTimestamp, 548  
EventFrameBurstEnd, 548  
EventFrameBurstEndFrameID, 548  
EventFrameBurstEndTimestamp, 548  
EventFrameBurstStart, 548  
EventFrameBurstStartFrameID, 549  
EventFrameBurstStartTimestamp, 549  
EventFrameEnd, 549  
EventFrameEndFrameID, 549  
EventFrameEndTimestamp, 549  
EventFrameStart, 549  
EventFrameStartFrameID, 550  
EventFrameStartTimestamp, 550  
EventFrameTransferEnd, 550  
EventFrameTransferEndFrameID, 550  
EventFrameTransferEndTimestamp, 550  
EventFrameTransferStart, 550  
EventFrameTransferStartFrameID, 551  
EventFrameTransferStartTimestamp, 551  
EventFrameTrigger, 551  
EventFrameTriggerFrameID, 551  
EventFrameTriggerTimestamp, 551  
EventLine0AnyEdge, 551  
EventLine0AnyEdgeFrameID, 552  
EventLine0AnyEdgeTimestamp, 552  
EventLine0FallingEdge, 552  
EventLine0FallingEdgeFrameID, 552  
EventLine0FallingEdgeTimestamp, 552  
EventLine0RisingEdge, 552  
EventLine0RisingEdgeFrameID, 553  
EventLine0RisingEdgeTimestamp, 553  
EventLine1AnyEdge, 553  
EventLine1AnyEdgeFrameID, 553  
EventLine1AnyEdgeTimestamp, 553  
EventLine1FallingEdge, 553  
EventLine1FallingEdgeFrameID, 554  
EventLine1FallingEdgeTimestamp, 554  
EventLine1RisingEdge, 554  
EventLine1RisingEdgeFrameID, 554  
EventLine1RisingEdgeTimestamp, 554  
EventLinkSpeedChange, 554  
EventLinkSpeedChangeFrameID, 555  
EventLinkSpeedChangeTimestamp, 555  
EventLinkTrigger0, 555  
EventLinkTrigger0FrameID, 555  
EventLinkTrigger0Timestamp, 555  
EventLinkTrigger1, 555  
EventLinkTrigger1FrameID, 556  
EventLinkTrigger1Timestamp, 556  
EventNotification, 556  
EventSelector, 556  
EventSequencerSetChange, 556  
EventSequencerSetChangeFrameID, 556  
EventSequencerSetChangeTimestamp, 557  
EventSerialData, 557  
EventSerialDataLength, 557  
EventSerialPortReceive, 557  
EventSerialPortReceiveTimestamp, 557  
EventSerialReceiveOverflow, 557  
EventStream0TransferBlockEnd, 558  
EventStream0TransferBlockEndFrameID, 558  
EventStream0TransferBlockEndTimestamp, 558  
EventStream0TransferBlockStart, 558  
EventStream0TransferBlockStartFrameID, 558  
EventStream0TransferBlockStartTimestamp, 558  
EventStream0TransferBlockTrigger, 559  
EventStream0TransferBlockTriggerFrameID, 559  
EventStream0TransferBlockTriggerTimestamp, 559  
EventStream0TransferBurstEnd, 559  
EventStream0TransferBurstEndFrameID, 559  
EventStream0TransferBurstEndTimestamp, 559  
EventStream0TransferBurstStart, 560  
EventStream0TransferBurstStartFrameID, 560  
EventStream0TransferBurstStartTimestamp, 560  
EventStream0TransferEnd, 560  
EventStream0TransferEndFrameID, 560  
EventStream0TransferEndTimestamp, 560  
EventStream0TransferOverflow, 561  
EventStream0TransferOverflowFrameID, 561  
EventStream0TransferOverflowTimestamp, 561  
EventStream0TransferPause, 561  
EventStream0TransferPauseFrameID, 561  
EventStream0TransferPauseTimestamp, 561  
EventStream0TransferResume, 562  
EventStream0TransferResumeFrameID, 562  
EventStream0TransferResumeTimestamp, 562  
EventStream0TransferStart, 562  
EventStream0TransferStartFrameID, 562  
EventStream0TransferStartTimestamp, 562  
EventTest, 563  
EventTestTimestamp, 563  
EventTimer0End, 563  
EventTimer0EndFrameID, 563  
EventTimer0EndTimestamp, 563  
EventTimer0Start, 563  
EventTimer0StartFrameID, 564  
EventTimer0StartTimestamp, 564  
EventTimer1End, 564  
EventTimer1EndFrameID, 564  
EventTimer1EndTimestamp, 564  
EventTimer1Start, 564  
EventTimer1StartFrameID, 565  
EventTimer1StartTimestamp, 565  
ExposureActiveMode, 565  
ExposureAuto, 565  
ExposureMode, 565  
ExposureTime, 565  
ExposureTimeMode, 566  
ExposureTimeSelector, 566  
FactoryReset, 566  
FileAccessBuffer, 566  
FileAccessLength, 566  
FileAccessOffset, 566  
FileOpenMode, 567

FileOperationExecute, 567  
 FileOperationResult, 567  
 FileOperationSelector, 567  
 FileOperationStatus, 567  
 FileSelector, 568  
 FileSize, 568  
 Gain, 568  
 GainAuto, 568  
 GainAutoBalance, 568  
 GainSelector, 569  
 Gamma, 569  
 GammaEnable, 569  
 GevActiveLinkCount, 569  
 GevCCP, 569  
 GevCurrentDefaultGateway, 569  
 GevCurrentIPAddress, 570  
 GevCurrentIPConfigurationDHCP, 570  
 GevCurrentIPConfigurationLLA, 570  
 GevCurrentIPConfigurationPersistentIP, 570  
 GevCurrentPhysicalLinkConfiguration, 570  
 GevCurrentSubnetMask, 570  
 GevDiscoveryAckDelay, 571  
 GevFirstURL, 571  
 GevGVCPExtendedStatusCodes, 571  
 GevGVCPExtendedStatusCodesSelector, 571  
 GevGVCPHeartbeatDisable, 571  
 GevGVCPPendingAck, 571  
 GevGVCPPendingTimeout, 572  
 GevGVSPExtendedIDMode, 572  
 GevHeartbeatTimeout, 572  
 GevIEEE1588, 572  
 GevIEEE1588ClockAccuracy, 572  
 GevIEEE1588Mode, 572  
 GevIEEE1588Status, 573  
 GevIPConfigurationStatus, 573  
 GevInterfaceSelector, 573  
 GevMACAddress, 573  
 GevMCDA, 573  
 GevMCPHostPort, 573  
 GevMCRC, 574  
 GevMCSP, 574  
 GevMCTT, 574  
 GevNumberOflnterfaces, 574  
 GevPAUSEFrameReception, 574  
 GevPAUSEFrameTransmission, 574  
 GevPersistentDefaultGateway, 575  
 GevPersistentIPAddress, 575  
 GevPersistentSubnetMask, 575  
 GevPhysicalLinkConfiguration, 575  
 GevPrimaryApplicationIPAddress, 575  
 GevPrimaryApplicationSocket, 575  
 GevPrimaryApplicationSwitchoverKey, 576  
 GevSCCFGAllInTransmission, 576  
 GevSCCFGExtendedChunkData, 576  
 GevSCCFGPacketResendDestination, 576  
 GevSCCFGUnconditionalStreaming, 576  
 GevSCDA, 576  
 GevSCPDDirection, 577  
 GevSCPHostPort, 577  
 GevSCPIInterfaceIndex, 577  
 GevSCPSBigEndian, 577  
 GevSCPSDoNotFragment, 577  
 GevSCPSFireTestPacket, 578  
 GevSCPSPacketSize, 578  
 GevSCPD, 577  
 GevSCSP, 578  
 GevSCZoneConfigurationLock, 578  
 GevSCZoneCount, 578  
 GevSCZoneDirectionAll, 578  
 GevSecondURL, 579  
 GevStreamChannelSelector, 579  
 GevSupportedOption, 579  
 GevSupportedOptionSelector, 579  
 GevTimestampTickFrequency, 579  
 GuiXmlManifestAddress, 579  
 Height, 580  
 HeightMax, 580  
 ImageComponentEnable, 580  
 ImageComponentSelector, 580  
 ImageCompressionBitrate, 580  
 ImageCompressionJPEGFormatOption, 580  
 ImageCompressionMode, 581  
 ImageCompressionQuality, 581  
 ImageCompressionRateOption, 581  
 Init, 492  
 IspEnable, 581  
 LUTEnable, 585  
 LUTIndex, 585  
 LUTSelector, 585  
 LUTValue, 585  
 LUTValueAll, 586  
 LineFilterWidth, 581  
 LineFormat, 582  
 LineInputFilterSelector, 582  
 LineInverter, 582  
 LineMode, 582  
 LinePitch, 582  
 LineSelector, 582  
 LineSource, 583  
 LineStatus, 583  
 LineStatusAll, 583  
 LinkErrorCount, 583  
 LinkUptime, 583  
 LogicBlockLUTInputActivation, 583  
 LogicBlockLUTInputSelector, 584  
 LogicBlockLUTInputSource, 584  
 LogicBlockLUTOutputValue, 584  
 LogicBlockLUTOutputValueAll, 584  
 LogicBlockLUTRowIndex, 584  
 LogicBlockLUTSelector, 584  
 LogicBlockSelector, 585  
 MaxDeviceResetTime, 586  
 OffsetX, 586  
 OffsetY, 586  
 PacketResendRequestCount, 586  
 PayloadSize, 587

PixelColorFilter, 587  
PixelDynamicRangeMax, 587  
PixelDynamicRangeMin, 587  
PixelFormat, 587  
PixelFormatInfoID, 588  
PixelFormatInfoSelector, 588  
PixelSize, 588  
PowerSupplyCurrent, 588  
PowerSupplyVoltage, 588  
RegionDestination, 588  
RegionMode, 589  
RegionSelector, 589  
ReverseX, 589  
ReverseY, 589  
RgbTransformLightSource, 589  
Saturation, 590  
SaturationEnable, 590  
Scan3dAxisMax, 590  
Scan3dAxisMin, 590  
Scan3dCoordinateOffset, 590  
Scan3dCoordinateReferenceSelector, 591  
Scan3dCoordinateReferenceValue, 591  
Scan3dCoordinateScale, 591  
Scan3dCoordinateSelector, 591  
Scan3dCoordinateSystem, 591  
Scan3dCoordinateSystemReference, 591  
Scan3dCoordinateTransformSelector, 592  
Scan3dDistanceUnit, 592  
Scan3dInvalidDataFlag, 592  
Scan3dInvalidHeaderValue, 592  
Scan3dOutputMode, 592  
Scan3dTransformValue, 592  
SensorDescription, 593  
SensorDigitizationTaps, 593  
SensorHeight, 593  
SensorShutterMode, 593  
SensorTaps, 593  
SensorWidth, 593  
SequencerConfigurationMode, 594  
SequencerConfigurationValid, 594  
SequencerFeatureEnable, 594  
SequencerMode, 594  
SequencerPathSelector, 594  
SequencerSetActive, 595  
SequencerSetLoad, 595  
SequencerSetNext, 595  
SequencerSetSave, 595  
SequencerSetSelector, 595  
SequencerSetStart, 596  
SequencerSetValid, 596  
SequencerTriggerActivation, 596  
SequencerTriggerSource, 596  
SerialPortBaudRate, 596  
SerialPortDataBits, 597  
SerialPortParity, 597  
SerialPortSelector, 597  
SerialPortSource, 597  
SerialPortStopBits, 597  
SerialReceiveFramingErrorCount, 597  
SerialReceiveParityErrorCount, 598  
SerialReceiveQueueClear, 598  
SerialReceiveQueueCurrentCharacterCount, 598  
SerialReceiveQueueMaxCharacterCount, 598  
SerialTransmitQueueCurrentCharacterCount, 598  
SerialTransmitQueueMaxCharacterCount, 598  
Sharpening, 599  
SharpeningAuto, 599  
SharpeningEnable, 599  
SharpeningThreshold, 599  
SoftwareSignalPulse, 600  
SoftwareSignalSelector, 600  
SourceCount, 600  
SourceSelector, 600  
TLParamsLocked, 603  
Test0001, 600  
TestEventGenerate, 601  
TestPattern, 601  
TestPatternGeneratorSelector, 601  
TestPendingAck, 601  
TimerDelay, 601  
TimerDuration, 602  
TimerReset, 602  
TimerSelector, 602  
TimerStatus, 602  
TimerTriggerActivation, 602  
TimerTriggerSource, 602  
TimerValue, 603  
Timestamp, 603  
TimestampLatch, 603  
TimestampLatchValue, 603  
TimestampReset, 603  
TransferAbort, 604  
TransferBlockCount, 604  
TransferBurstCount, 604  
TransferComponentSelector, 604  
TransferControlMode, 604  
TransferOperationMode, 604  
TransferPause, 605  
TransferQueueCurrentBlockCount, 605  
TransferQueueMaxBlockCount, 605  
TransferQueueMode, 605  
TransferQueueOverflowCount, 605  
TransferResume, 605  
TransferSelector, 606  
TransferStart, 606  
TransferStatus, 606  
TransferStatusSelector, 606  
TransferStop, 606  
TransferStreamChannel, 606  
TransferTriggerActivation, 607  
TransferTriggerMode, 607  
TransferTriggerSelector, 607  
TransferTriggerSource, 607  
TriggerActivation, 607  
TriggerDelay, 607  
TriggerDivider, 608

TriggerEventTest, 608  
 TriggerMode, 608  
 TriggerMultiplier, 608  
 TriggerOverlap, 608  
 TriggerSelector, 609  
 TriggerSoftware, 609  
 TriggerSource, 609  
 UserOutputSelector, 609  
 UserOutputValue, 609  
 UserOutputValueAll, 610  
 UserOutputValueAllMask, 610  
 UserSetDefault, 610  
 UserSetFeatureEnable, 610  
 UserSetLoad, 610  
 UserSetSave, 611  
 UserSetSelector, 611  
 V3\_3Enable, 611  
 WhiteClip, 611  
 WhiteClipSelector, 611  
 Width, 612  
 WidthMax, 612  
**Spinnaker::CameraBase**  
   ~CameraBase, 615  
   BeginAcquisition, 616  
   CameraBase, 615, 616  
   DelInit, 616  
   DiscoverMaxPacketSize, 616  
   EndAcquisition, 617  
   ForceIP, 617  
   GetAccessMode, 617  
   GetBufferOwnership, 618  
   GetGuiXml, 618  
   GetNextImage, 618  
   GetNodeMap, 619  
   GetNumDataStreams, 619  
   GetNumImagesInUse, 620  
   GetTLDeviceNodeMap, 620  
   GetTlStreamNodeMap, 620  
   GetUniqueId, 621  
    GetUserBufferCount, 621  
    GetUserBufferSize, 621  
    GetUserBufferTotalSize, 622  
   Init, 622  
   InterfaceImpl, 627  
   IsInitialized, 623  
   IsStreaming, 623  
   IsValid, 623  
   operator=, 624  
   ReadPort, 624  
   RegisterEvent, 624, 625  
   SetBufferOwnership, 625  
   SetUserBuffers, 625, 626  
   UnregisterEvent, 627  
   WritePort, 627  
**Spinnaker::CameraList**  
   ~CameraList, 629  
   Append, 630  
   CameraList, 629  
  
 Clear, 631  
 GetByIndex, 631  
 GetBySerial, 631  
 GetSize, 632  
 operator=, 632  
 operator[], 632  
 RemoveByIndex, 633  
 RemoveBySerial, 633  
**Spinnaker::ChunkData**  
   ~ChunkData, 689  
   ChunkData, 688  
   GetBlackLevel, 689  
   GetCRC, 689  
   GetCounterValue, 689  
   GetEncoderValue, 689  
   GetExposureEndLineStatusAll, 690  
   GetExposureTime, 690  
   GetFrameID, 690  
   GetGain, 690  
   GetHeight, 691  
   GetImage, 691  
   GetInferenceBoundingBoxResult, 691  
   GetInferenceConfidence, 691  
   GetInferenceResult, 692  
   GetLinePitch, 692  
   GetLineStatusAll, 692  
   GetOffsetX, 692  
   GetOffsetY, 693  
   GetPartSelector, 693  
   GetPixelDynamicRangeMax, 693  
   GetPixelDynamicRangeMin, 693  
   GetScan3dAxisMax, 694  
   GetScan3dAxisMin, 694  
   GetScan3dCoordinateOffset, 694  
   GetScan3dCoordinateReferenceValue, 694  
   GetScan3dCoordinateScale, 695  
   GetScan3dInvalidDataValue, 695  
   GetScan3dTransformValue, 695  
   GetScanLineSelector, 695  
   GetSequencerSetActive, 696  
   GetSerialDataLength, 696  
   GetStreamChannelID, 696  
   GetTimerValue, 696  
   GetTimestamp, 697  
   GetTimestampLatchValue, 697  
   GetTransferBlockID, 697  
   GetTransferQueueCurrentBlockCount, 697  
   GetWidth, 698  
   SetChunks, 698  
**Spinnaker::DeviceEvent**  
   ~DeviceEvent, 754  
   DeviceEvent, 753  
   GetDeviceEventId, 754  
   GetDeviceEventName, 754  
   OnDeviceEvent, 754  
   operator=, 755  
**Spinnaker::Event**  
   ~Event, 779

Event, 779  
EventProcessor, 781  
GetEventPayloadData, 779  
GetEventPayloadContentSize, 779  
GetEventType, 780  
IDataStream, 781  
m\_pEventData, 781  
operator=, 780  
SetEventPayload, 780  
SetEventType, 780  
Stream, 781  
Spinnaker::Exception  
  ~Exception, 785  
  Exception, 784, 785  
  GetBuildDate, 786  
  GetBuildTime, 786  
  GetError, 786  
  GetErrorMessage, 786  
  GetFileName, 786  
  GetFullErrorMessage, 786  
  GetFunctionName, 786  
  GetLineNumber, 787  
  operator!=, 787  
  operator=, 787  
  operator==, 787  
  what, 787  
Spinnaker::GenApi, 421  
  COMMAND\_MAGIC, 437  
  GENCP\_COMMAND\_HEADER\_SIZE, 437  
  GENCP\_EVENT\_BASIC\_SIZE, 437  
  GENCP\_EVENT\_CMD\_ID, 437  
  GVCP\_MESSAGE\_TAGS, 436  
  IDevFileStream, 435  
  IPersistScript, 437  
  ODevFileStream, 435  
  PersistFeature, 436  
  SET\_GUID, 436  
  U3V\_EVENT\_PREFIX, 437  
Spinnaker::GenApi::AutoLock  
  ~AutoLock, 451  
  AutoLock, 451  
Spinnaker::GenApi::BooleanNode  
  ~BooleanNode, 460  
  BooleanNode, 459, 460  
  GetValue, 460  
  operator=, 460  
  SetReference, 460  
  SetValue, 461  
Spinnaker::GenApi::CChunkAdapter  
  ~CChunkAdapter, 638  
  AttachBuffer, 639  
  AttachNodeMap, 639  
  CChunkAdapter, 638  
  CheckBufferLayout, 639  
  ClearCaches, 639  
  DetachBuffer, 640  
  DetachNodeMap, 640  
  m\_pChunkAdapter, 640  
    UpdateBuffer, 640  
Spinnaker::GenApi::CChunkAdapterDcam  
  ~CChunkAdapterDcam, 642  
  AttachBuffer, 642  
  CChunkAdapterDcam, 642  
  CheckBufferLayout, 642  
  CheckCRC, 643  
  HasCRC, 643  
Spinnaker::GenApi::CChunkAdapterGEV  
  ~CChunkAdapterGEV, 647  
  AttachBuffer, 647  
  CChunkAdapterGEV, 647  
  CheckBufferLayout, 647  
Spinnaker::GenApi::CChunkAdapterGeneric  
  ~CChunkAdapterGeneric, 644  
  AttachBuffer, 645  
  CChunkAdapterGeneric, 644  
  CheckBufferLayout, 645  
Spinnaker::GenApi::CChunkAdapterU3V  
  ~CChunkAdapterU3V, 649  
  AttachBuffer, 649  
  CChunkAdapterU3V, 649  
  CheckBufferLayout, 649  
Spinnaker::GenApi::CChunkPort  
  ~CChunkPort, 651  
  AttachChunk, 652  
  AttachPort, 652  
  CChunkPort, 651  
  CheckChunkID, 652  
  ClearCache, 652  
  DetachChunk, 653  
  DetachPort, 653  
  GetAccessMode, 653  
  GetChunkIDLength, 653  
  GetPrincipalInterfaceType, 653  
  GetSwapEndianess, 653  
  InvalidateNode, 654  
  m\_pChunkPort, 655  
  m\_pPort, 655  
  m\_pPortAdapter, 655  
  Read, 654  
  SetPortImpl, 654  
  UpdateBuffer, 654  
  Write, 654  
Spinnaker::GenApi::CEnumerationTRef  
  ~CEnumerationTRef, 657  
  CEnumerationTRef, 657  
  GetCurrentEntry, 657  
  GetEntry, 658  
  GetValue, 658  
  operator(), 658  
  operator=, 659  
  SetEnumReference, 659  
  SetNumEnums, 659  
  SetReference, 659  
  SetValue, 660  
Spinnaker::GenApi::CEventAdapter  
  ~CEventAdapter, 661

AttachNodeMap, 661  
 CEventAdapter, 661  
 DeliverMessage, 661  
 DetachNodeMap, 662  
 m\_pEventAdapter, 662  
 Spinnaker::GenApi::CEventAdapter1394  
   ~CEventAdapter1394, 663  
   CEventAdapter1394, 663  
   DeliverEventMessage, 664  
   DeliverMessage, 664  
 Spinnaker::GenApi::CEventAdapterGEV  
   ~CEventAdapterGEV, 668  
   CEventAdapterGEV, 668  
   DeliverEventMessage, 668  
   DeliverMessage, 668  
 Spinnaker::GenApi::CEventAdapterGeneric  
   ~CEventAdapterGeneric, 665  
   CEventAdapterGeneric, 665  
   DeliverMessage, 666  
 Spinnaker::GenApi::CEventAdapterU3V  
   ~CEventAdapterU3V, 670  
   CEventAdapterU3V, 670  
   DeliverEventMessage, 670  
   DeliverMessage, 670  
 Spinnaker::GenApi::CEventPort  
   ~CEventPort, 673  
   AttachEvent, 673  
   AttachNode, 673  
   CEventPort, 672  
   CheckEventID, 673  
   DetachEvent, 674  
   DetachNode, 674  
   GetAccessMode, 674  
   GetEventIDLength, 674  
   GetPrincipallInterfaceType, 674  
   GetSwapEndianess, 674  
   InvalidateNode, 675  
   m\_pEventPort, 675  
   m\_pNode, 676  
   m\_pPortAdapter, 676  
   Read, 675  
   SetPortImpl, 675  
   Write, 675  
 Spinnaker::GenApi::CFeatureBag  
   ~CFeatureBag, 677  
   CFeatureBag, 677  
   GetFeatureBagHandle, 677  
   LoadFromBag, 677  
   operator==, 678  
   PersistFeature, 678  
   SetInfo, 678  
   StoreToBag, 678  
 Spinnaker::GenApi::CFloatPtr  
   CFloatPtr, 680  
   GetEnumAlias, 680  
   GetIntAlias, 680  
   operator=, 680  
 Spinnaker::GenApi::CGeneric\_XMLLoaderParams  
   \_initialize, 681  
 Spinnaker::GenApi::CLock  
   ~CLock, 701  
   Clock, 701  
   Lock, 701  
   m\_bOwnLock, 702  
   m\_lock, 702  
   NodeMap, 702  
   TryLock, 702  
   Unlock, 702  
 Spinnaker::GenApi::CLockEx  
   m\_lockEx, 705  
 Spinnaker::GenApi::CNodeCallback  
   ~CNodeCallback, 706  
   CNodeCallback, 706  
   Destroy, 706  
   GetCallbackType, 706  
   GetNode, 707  
   m\_CallbackType, 707  
   m\_pNode, 707  
   operator(), 707  
 Spinnaker::GenApi::CNodeMapFactory  
   ~CNodeMapFactory, 710  
   AddInjectionData, 712  
   ApplyStyleSheet, 712  
   CNodeMapFactory, 710, 711  
   ClearCache, 712  
   CreateEmptyNodeMap, 713  
   CreateNodeDataFromNodeMap, 713  
   CreateNodeMap, 713  
   ExtractSubtree, 713  
   GetNodeStatistics, 714  
   GetSupportedSchemaVersions, 714  
   IsCameraDescriptionFileDataReleased, 714  
   IsEmpty, 714  
   IsLoaded, 714  
   IsPreprocessed, 715  
   LoadAndInject, 715  
   operator=, 715  
   Preprocess, 715  
   ReleaseCameraDescriptionFileData, 715  
   ToString, 716  
   ToXml, 716  
 Spinnaker::GenApi::CNodeMapFactory::NodeStatistics ←  
   \_t  
   NumLinks, 1033  
   NumNodes, 1033  
   NumProperties, 1033  
   NumStrings, 1033  
 Spinnaker::GenApi::CNodeMapRef  
   CNodeMapRef, 717, 718  
   operator=, 718  
 Spinnaker::GenApi::CNodeMapRefT  
   \_ClearXMLCache, 721  
   \_Connect, 721  
   \_GetDeviceName, 721  
   \_GetNode, 721  
   \_GetNodes, 722

\_GetSupportedSchemaVersions, 722  
\_InvalidateNodes, 722  
\_LoadXMLFromFile, 722  
\_LoadXMLFromFileInject, 722  
\_LoadXMLFromString, 723  
\_LoadXMLFromStringInject, 723  
\_LoadXMLFromZIPData, 723  
\_LoadXMLFromZIPFile, 723  
\_Poll, 723  
\_Ptr, 724  
Spinnaker::GenApi::CPointer  
~CPointer, 731  
CPointer, 730  
IsValid, 731  
m\_pT, 734  
operator bool, 731  
operator T\*, 731  
operator!=, 731, 732  
operator\*, 732  
operator(), 732  
operator->, 733  
operator=, 733  
operator==, 733  
Spinnaker::GenApi::CPortImpl  
~CPortImpl, 735  
CPortImpl, 735  
GetAccessMode, 736  
GetSwapEndianess, 736  
InvalidateNode, 736  
m\_ptrPort, 737  
Read, 736  
Replay, 736  
SetPortImpl, 736  
Write, 737  
Spinnaker::GenApi::CPortWriteList  
~CPortWriteList, 739  
CPortWriteList, 739  
GetCookie, 739  
GetPortWriteListHandle, 739  
m\_pWriteList, 740  
Replay, 739  
SetCookie, 740  
Write, 740  
Spinnaker::GenApi::CRegisterPortImpl  
~CRegisterPortImpl, 742  
CRegisterPortImpl, 742  
GetAccessMode, 742  
Read, 743  
ReadRegister, 743  
SetPortImpl, 743  
Write, 743  
WriteRegister, 744  
Spinnaker::GenApi::CSelectorSet  
~CSelectorSet, 746  
CSelectorSet, 745  
GetSelectorList, 746  
IsEmpty, 746  
Restore, 746  
SetFirst, 746  
SetNext, 746  
ToString, 747  
Spinnaker::GenApi::CTestPortStruct  
CTestPortStruct, 749  
GetAccessMode, 749  
GetNumReads, 749  
GetNumWrites, 749  
GetPrincipalInterfaceType, 749  
m\_BaseAddress, 750  
m\_NumReads, 750  
m\_NumWrites, 751  
MemSet, 749  
Read, 750  
ResetStatistics, 750  
Write, 750  
Spinnaker::GenApi::CategoryNode  
~CategoryNode, 637  
CategoryNode, 636  
GetFeatures, 637  
SetReference, 637  
Spinnaker::GenApi::CommandNode  
~CommandNode, 726  
CommandNode, 725, 726  
Execute, 726  
IsDone, 726  
operator(), 727  
SetReference, 727  
Spinnaker::GenApi::Counter  
Counter, 728  
GetValue, 728  
IsZero, 728  
operator unsigned int, 728  
operator++, 728  
operator--, 728, 729  
Spinnaker::GenApi::EAccessModeClass  
FromString, 758  
ToString, 758  
Spinnaker::GenApi::ECachingModeClass  
FromString, 759  
ToString, 759  
Spinnaker::GenApi::EDisplayNotationClass  
FromString, 760  
ToString, 760  
Spinnaker::GenApi::EEndianessClass  
FromString, 761  
ToString, 761  
Spinnaker::GenApi::EGenApiSchemaVersionClass  
FromString, 762  
ToString, 762  
Spinnaker::GenApi::EInputDirectionClass  
FromString, 763  
ToString, 763  
Spinnaker::GenApi::ENamespaceClass  
FromString, 764  
ToString, 764  
Spinnaker::GenApi::ERepresentationClass  
FromString, 773

ToString, 774  
**Spinnaker::GenApi::ESignClass**  
     FromString, 775  
     ToString, 775  
**Spinnaker::GenApi::ESlopeClass**  
     FromString, 776  
     ToString, 776  
**Spinnaker::GenApi::EStandardNameSpaceClass**  
     FromString, 777  
     ToString, 777  
**Spinnaker::GenApi::EVisibilityClass**  
     FromString, 782  
     ToString, 782  
**Spinnaker::GenApi::EYesNoClass**  
     FromString, 788  
     ToString, 788  
**Spinnaker::GenApi::EnumEntryNode**  
     ~EnumEntryNode, 766  
     EnumEntryNode, 766  
     GetNumericValue, 766  
     GetSymbolic, 767  
     GetValue, 767  
     IsSelfClearing, 767  
     SetReference, 767  
**Spinnaker::GenApi::EnumNode**  
     ~EnumNode, 770  
     EnumNode, 770  
     GetCurrentEntry, 770  
     GetEntries, 770  
     GetEntry, 771  
     GetEntryByName, 771  
     GetIntValue, 771  
     GetSymbolics, 771  
     m\_pEnumeration, 773  
     operator\*, 772  
     operator=, 772  
     SetIntValue, 772  
     SetReference, 772  
**Spinnaker::GenApi::FileProtocolAdapter**  
     ~FileProtocolAdapter, 789  
     attach, 790  
     closeFile, 791  
     deleteFile, 791  
     FileProtocolAdapter, 789  
     getBufSize, 791  
     openFile, 792  
     read, 792  
     write, 793  
**Spinnaker::GenApi::FloatNode**  
     ~FloatNode, 796  
     FloatNode, 796  
     GetDisplayNotation, 796  
     GetDisplayPrecision, 796  
     GetEnumAlias, 796  
     GetInc, 796  
     GetIncMode, 797  
     GetIntAlias, 797  
     GetListOfValidValues, 797  
     GetMax, 797  
     GetMin, 797  
     GetRepresentation, 797  
     GetUnit, 798  
     GetValue, 798  
     HasInc, 798  
     ImposeMax, 798  
     ImposeMin, 798  
     operator\*, 799  
     operator(), 799  
     operator=, 799  
     SetReference, 799  
     SetValue, 799  
**Spinnaker::GenApi::FloatRegNode**  
     ~FloatRegNode, 802  
     FloatRegNode, 801, 802  
     SetReference, 802  
**Spinnaker::GenApi::Function\_NodeCallback**  
     Destroy, 804  
     Function\_NodeCallback, 804  
     operator(), 804  
**Spinnaker::GenApi::IDevFileStreamBase**  
     close, 856  
     filebuf\_type, 856  
     ios\_type, 856  
     is\_open, 856  
     istream\_type, 856  
     open, 856  
     rdbuf, 857  
**Spinnaker::GenApi::IDevFileStreamBuf**  
     ~IDevFileStreamBuf, 858  
     close, 858  
     IDevFileStreamBuf, 858  
     is\_open, 858  
     open, 858  
     pbackfail, 859  
     underflow, 859  
**Spinnaker::GenApi::IntRegNode**  
     ~IntRegNode, 984  
     IntRegNode, 983, 984  
     SetReference, 984  
**Spinnaker::GenApi::IntegerNode**  
     ~IntegerNode, 960  
     GetFloatAlias, 960  
     GetInc, 961  
     GetIncMode, 961  
     GetListOfValidValues, 961  
     GetMax, 961  
     GetMin, 961  
     GetRepresentation, 961  
     GetUnit, 962  
     GetValue, 962  
     ImposeMax, 962  
     ImposeMin, 962  
     IntegerNode, 960  
     operator\*, 963  
     operator(), 962  
     operator=, 963

SetReference, 963  
SetValue, 963  
Spinnaker::GenApi::Member\_NodeCallback  
    Destroy, 1011  
    Member\_NodeCallback, 1011  
operator(), 1012  
PMEMBERFUNC, 1011  
Spinnaker::GenApi::Node  
    ~Node, 1017  
    DeregisterCallback, 1017  
    GetAccessMode, 1017  
    GetAlias, 1017  
    GetCachingMode, 1017  
    GetCastAlias, 1018  
    GetChildren, 1018  
    GetDescription, 1018  
    GetDeviceName, 1018  
    GetDisplayName, 1018  
    GetDocuURL, 1019  
    GetEventID, 1019  
    GetName, 1019  
    GetNameSpace, 1019  
    GetNodeHandle, 1019  
    GetNodeMap, 1019  
    GetParents, 1019  
    GetPollingTime, 1020  
    GetPrincipalInterfaceType, 1020  
    GetProperty, 1020  
    GetPropertyNames, 1020  
    GetSelectedFeatures, 1020  
    GetSelectingFeatures, 1020  
    GetToolTip, 1021  
    GetVisibility, 1021  
    ImposeAccessMode, 1021  
    ImposeVisibility, 1021  
    InvalidateNode, 1021  
    IsAccessModeCacheable, 1021  
    IsCachable, 1022  
    IsDeprecated, 1022  
    IsFeature, 1022  
    IsSelector, 1022  
    IsStreamable, 1022  
    m\_Callbacks, 1024  
    m\_pNodeData, 1024  
    m\_pNodeMap, 1024  
    Node, 1016  
    operator!=, 1022  
    operator==, 1023  
    RegisterCallback, 1023  
    SetNodeHandle, 1023  
    SetNodeMap, 1023  
    SetReference, 1023  
Spinnaker::GenApi::NodeMap  
    \_Ptr, 1032  
    ~NodeMap, 1027  
    ClearXMLCache, 1027  
    Connect, 1027  
    Destroy, 1027  
GetDeviceName, 1028  
GetDeviceVersion, 1028  
GetGenApiVersion, 1028  
GetLock, 1028  
GetmodelName, 1028  
GetNode, 1028  
GetNodeMapHandle, 1029  
GetNodes, 1029  
GetNumNodes, 1029  
GetProductGuid, 1029  
GetSchemaVersion, 1029  
GetStandardNameSpace, 1029  
GetSupportedSchemaVersions, 1030  
GetToolTip, 1030  
GetVendorName, 1030  
GetVersionGuid, 1030  
InvalidateNodes, 1031  
LoadXMLFromFile, 1031  
LoadXMLFromFileInject, 1031  
LoadXMLFromString, 1031  
LoadXMLFromStringInject, 1031  
LoadXMLFromZIPData, 1032  
LoadXMLFromZIPFile, 1032  
NodeMap, 1026  
Poll, 1032  
Spinnaker::GenApi::ODevFileStreamBase  
    close, 1035  
    filebuf\_type, 1035  
    ios\_type, 1035  
    is\_open, 1035  
    open, 1035  
    ostream\_type, 1035  
    rdbuf, 1036  
Spinnaker::GenApi::ODevFileStreamBuf  
    ~ODevFileStreamBuf, 1037  
    close, 1037  
    is\_open, 1037  
    ODevFileStreamBuf, 1037  
    open, 1037  
    overflow, 1038  
    sync, 1038  
    xputn, 1038  
Spinnaker::GenApi::PortNode  
    ~PortNode, 1043  
    CacheChunkData, 1043  
    GetChunkID, 1043  
    GetPortHandle, 1044  
    GetSwapEndianess, 1044  
    PortNode, 1043  
    Read, 1044  
    Replay, 1044  
    SetPortImpl, 1044  
    SetReference, 1045  
    StartRecording, 1045  
    StopRecording, 1045  
    Write, 1046  
Spinnaker::GenApi::PortRecorder  
    ~PortRecorder, 1048

GetAccessMode, 1048  
 PortRecorder, 1047  
 SetReference, 1048  
 StartRecording, 1048  
 StopRecording, 1048  
 Spinnaker::GenApi::PortReplay  
   ~PortReplay, 1050  
   GetPortReplayHandle, 1051  
   PortReplay, 1050  
   Replay, 1051  
   SetReference, 1051  
 Spinnaker::GenApi::RegisterNode  
   ~RegisterNode, 1055  
   Get, 1055  
   GetAddress, 1055  
   GetLength, 1056  
   RegisterNode, 1054, 1055  
   Set, 1056  
   SetReference, 1056  
 Spinnaker::GenApi::StringNode  
   ~StringNode, 1068  
   GetMaxLength, 1068  
   GetValue, 1068  
   operator\*, 1069  
   operator(), 1069  
   operator=, 1069  
   SetReference, 1069  
   SetValue, 1069  
   StringNode, 1068  
 Spinnaker::GenApi::StringRegNode  
   ~StringRegNode, 1072  
   SetReference, 1072  
   StringRegNode, 1071, 1072  
 Spinnaker::GenApi::ValueNode  
   ~ValueNode, 1123  
   FromString, 1124  
   GetNode, 1124  
   IsValueCacheValid, 1124  
   SetReference, 1124  
   ToString, 1124  
   ValueNode, 1123  
 Spinnaker::GenApi::double\_autovector\_t  
   \_pCount, 757  
   \_pv, 757  
   ~double\_autovector\_t, 756  
   double\_autovector\_t, 756  
   operator delete, 756  
   operator new, 756  
   operator=, 756  
   operator[], 757  
   size, 757  
 Spinnaker::GenApi::int64\_autovector\_t  
   \_pCount, 957  
   \_pv, 957  
   ~int64\_autovector\_t, 956  
   int64\_autovector\_t, 955, 956  
   operator delete, 956  
   operator new, 956  
   operator=, 956  
   operator[], 957  
   size, 957  
 operator[], 957  
 size, 957  
 Spinnaker::GenICam, 438  
   getline, 439  
   ThrowBadAlloc, 439  
 Spinnaker::GenICam::AutoLock  
   ~AutoLock, 450  
   AutoLock, 450  
 Spinnaker::GenICam::CGlobalLock  
   ~CGlobalLock, 683  
   CGlobalLock, 682, 683  
   IsValid, 683  
   Lock, 683  
   m\_DebugCount, 684  
   TryLock, 683  
   Unlock, 684  
 Spinnaker::GenICam::CGlobalLockUnlocker  
   ~CGlobalLockUnlocker, 685  
   CGlobalLockUnlocker, 685  
   m\_Lock, 686  
   m\_enabled, 686  
   UnlockEarly, 685  
 Spinnaker::GenICam::CLock  
   ~CLock, 699  
   CLock, 699  
   Lock, 699  
   TryLock, 699  
   Unlock, 700  
 Spinnaker::GenICam::LockableObject  
   GetLock, 1002  
   Lock, 1002  
   m\_Lock, 1002  
 Spinnaker::GenICam::LockableObject::Lock  
   ~Lock, 1000  
   Lock, 1000  
 Spinnaker::GenICam::Version\_t  
   Major, 1125  
   Minor, 1125  
   SubMinor, 1126  
 Spinnaker::GenICam::gcstring  
   npos, 807  
   ~gcstring, 807  
   append, 807  
   assign, 807, 808  
   c\_str, 808  
   compare, 808  
   empty, 808  
   find, 808, 809  
   find\_first\_not\_of, 809  
   find\_first\_of, 809  
   gcstring, 806  
   length, 810  
   max\_size, 810  
   npos, 814  
   operator const char \*, 810  
   operator delete, 810  
   operator new, 810

operator!=, 811  
operator<, 812  
operator>, 812  
operator+, 813  
operator+=, 811  
operator=, 812  
operator==, 812  
resize, 812  
size, 812  
substr, 813  
swap, 813  
Spinnaker::IArrivalEvent  
  ~IArrivalEvent, 827  
  IArrivalEvent, 827  
  OnDeviceArrival, 827  
  operator=, 827  
Spinnaker::ICameraBase  
  ~ICameraBase, 830  
  BeginAcquisition, 830  
  CameralInternal, 836  
  DelInit, 830  
  DiscoverMaxPacketSize, 830  
  EndAcquisition, 831  
  ForceIP, 831  
  GetAccessMode, 831  
  GetBufferOwnership, 831  
  GetGuiXml, 831  
  GetNextImage, 831  
  GetNodeMap, 832  
  GetNumDataStreams, 832  
  GetNumImagesInUse, 832  
  GetTLDeviceNodeMap, 832  
  GetTlStreamNodeMap, 832  
  GetUniqueId, 832  
   GetUserBufferCount, 833  
   GetUserBufferSize, 833  
   GetUserBufferTotalSize, 833  
  ICameraBase, 830  
  Init, 833  
  InterfaceImpl, 836  
  IsInitialized, 833  
  IsStreaming, 833  
  IsValid, 834  
  m\_pCameraBaseData, 836  
  operator=, 834  
  ReadPort, 834  
  RegisterEvent, 834  
  SetBufferOwnership, 834  
  SetUserBuffers, 835  
  TLDevice, 836  
  TlStream, 836  
  UnregisterEvent, 835  
  WritePort, 835  
Spinnaker::ICameraList  
  ~ICameraList, 838  
  Append, 838  
  CameraListImpl, 840  
  Clear, 838  
            GetByIndex, 838  
            GetBySerial, 839  
            GetSize, 839  
            ICameraList, 838  
            InterfaceImpl, 840  
            m\_pCameraListData, 840  
            operator=, 839  
            operator[], 839  
            RemoveByIndex, 839  
            RemoveBySerial, 839  
Spinnaker::IChunkData  
  ~IChunkData, 842  
  GetBlackLevel, 842  
  GetCRC, 842  
  GetCounterValue, 842  
  GetEncoderValue, 842  
  GetExposureEndLineStatusAll, 843  
  GetExposureTime, 843  
  GetFrameID, 843  
  GetGain, 843  
  GetHeight, 843  
  GetImage, 843  
  GetInferenceBoundingBoxResult, 844  
  GetInferenceConfidence, 844  
  GetInferenceResult, 844  
  GetLinePitch, 844  
  GetLineStatusAll, 844  
  GetOffsetX, 844  
  GetOffsetY, 845  
  GetPartSelector, 845  
  GetPixelDynamicRangeMax, 845  
  GetPixelDynamicRangeMin, 845  
  GetScan3dAxisMax, 845  
  GetScan3dAxisMin, 845  
  GetScan3dCoordinateOffset, 846  
  GetScan3dCoordinateReferenceValue, 846  
  GetScan3dCoordinateScale, 846  
  GetScan3dInvalidDataValue, 846  
  GetScan3dTransformValue, 846  
  GetScanLineSelector, 846  
  GetSequencerSetActive, 847  
  GetSerialDataLength, 847  
  GetStreamChannelID, 847  
  GetTimerValue, 847  
  GetTimestamp, 847  
  GetTimestampLatchValue, 847  
  GetTransferBlockID, 848  
  GetTransferQueueCurrentBlockCount, 848  
  GetWidth, 848  
  IChunkData, 842  
  SetChunks, 848  
Spinnaker::IDataStream  
  ~IDataStream, 849  
  AnnounceImage, 850  
  AttachBuffer, 850  
  CleanupChunkAdapter, 850  
  FlushQueueAllDiscard, 851  
  GetBufferChunkData, 851

GetBufferInfoBool8Type, 851  
 GetBufferInfoPtrType, 851  
 GetBufferInfoSizeType, 851  
 GetBufferInfoUInt64Type, 851  
 GetDeviceNodeMap, 852  
 GetNextImage, 852  
 GetNextImageInternal, 852  
 GetNodeMap, 852  
 GetNumImagesInUse, 852  
 GetPort, 852  
 GetStreamType, 852  
 IDataStream, 850  
 InitChunkAdapter, 853  
 IsCRCCheckEnabled, 853  
 IsImageInUse, 853  
 IsStreaming, 853  
 KillBufferEvent, 853  
 RegisterImageEvent, 853  
 ReleaseImage, 853  
 RevokelImages, 854  
 StartStream, 854  
 StopStream, 854  
 TransportLayerStreamInfo, 854  
 UnregisterImageEvent, 854  
 WaitOnImageEvent, 854  
**Spinnaker::IDeviceEvent**  
 ~IDeviceEvent, 860  
 GetDeviceEventId, 861  
 GetDeviceEventName, 861  
 IDeviceEvent, 860  
 OnDeviceEvent, 861  
 operator=, 861  
**Spinnaker::IImage**  
 ~IImage, 863  
 CalculateStatistics, 864  
 CheckCRC, 864  
 Convert, 864  
 DEPRECATED\_FUNC, 865  
 DeepCopy, 864  
 GetBitsPerPixel, 865  
 GetBufferSize, 865  
 GetChunkData, 866  
 GetChunkLayoutId, 866  
 GetColorProcessing, 866  
 GetData, 866  
 GetDataAbsoluteMax, 866  
 GetDataAbsoluteMin, 866  
 GetFrameID, 867  
 GetHeight, 867  
 GetID, 867  
 GetImageData, 867  
 GetImageSize, 867  
 GetImageStatus, 867  
 GetNumChannels, 868  
 GetPayloadType, 868  
 GetPixelFormat, 868  
 GetPixelFormatIntType, 868  
 GetPixelFormatName, 868  
 GetPrivateData, 868  
 GetStride, 869  
 GetTLPayloadType, 869  
 GetTLPixelFormat, 869  
 GetTLPixelFormatNamespace, 869  
 GetTimeStamp, 869  
 GetValidPayloadSize, 869  
 GetWidth, 870  
 GetXOffset, 870  
 GetXPadding, 870  
 GetYOffset, 870  
 GetYPadding, 870  
 HasCRC, 870  
 IImage, 864  
 IsInUse, 871  
 IsIncomplete, 871  
 Release, 871  
 ResetImage, 871  
 Save, 872, 873  
 Stream, 873  
**Spinnaker::IImageEvent**  
 ~IImageEvent, 875  
 IImageEvent, 875  
 OnImageEvent, 875  
 operator=, 875  
**Spinnaker::IImageStatistics**  
 ~IImageStatistics, 877  
 DisableAll, 877  
 EnableAll, 877  
 EnableGreyOnly, 877  
 EnableHSLOnly, 877  
 EnableRGBOnly, 878  
 GetChannelStatus, 878  
 GetHistogram, 878  
 GetMean, 878  
 GetNumPixelValues, 878  
 GetPixelValueRange, 879  
 GetRange, 879  
 GetStatistics, 879  
 IImageStatistics, 877  
 SetChannelStatus, 879  
**Spinnaker::IInterface**  
 ~IInterface, 881  
 GetCameras, 882  
 GetTLNodeMap, 882  
 IInterface, 881, 882  
 InterfaceInternal, 883  
 IsInUse, 882  
 IsValid, 882  
 m\_pInterfaceData, 884  
 operator=, 882  
 RegisterEvent, 883  
 SendActionCommand, 883  
 SystemImpl, 884  
 TLIInterface, 884  
 UnregisterEvent, 883  
 UpdateCameras, 883  
**Spinnaker::IInterfaceArrivalEvent**

~IInterfaceArrivalEvent, 885  
IInterfaceArrivalEvent, 885  
OnInterfaceArrival, 886  
operator=, 886  
Spinnaker::IInterfaceEvent  
  ~IInterfaceEvent, 887  
  IInterfaceEvent, 887  
  OnDeviceArrival, 888  
  OnDeviceRemoval, 888  
  operator=, 888  
Spinnaker::IInterfaceList  
  ~IInterfaceList, 889  
  Clear, 890  
  GetByIndex, 890  
  GetSize, 890  
  IInterfaceList, 890  
  m\_pInterfaceListData, 891  
  operator=, 890  
  operator[], 891  
Spinnaker::IInterfaceRemovalEvent  
  ~IInterfaceRemovalEvent, 892  
  IInterfaceRemovalEvent, 892  
  OnInterfaceRemoval, 893  
  operator=, 893  
Spinnaker::ILoggingEvent  
  ~ILoggingEvent, 894  
  ILoggingEvent, 894  
  OnLogEvent, 895  
  operator=, 895  
Spinnaker::IRemovalEvent  
  ~IRemovalEvent, 987  
  IRemovalEvent, 987  
  OnDeviceRemoval, 987  
  operator=, 987  
Spinnaker::ISystem  
  ~ISystem, 989  
  GetCameras, 990  
  GetInterfaces, 990  
  GetLibraryVersion, 990  
  GetLoggingEventPriorityLevel, 990  
  GetTLNodeMap, 990  
  ISystem, 989  
  IsInUse, 990  
  operator=, 991  
  RegisterEvent, 991  
  RegisterInterfaceEvent, 991  
  RegisterLoggingEvent, 991  
  ReleaseInstance, 991  
  SendActionCommand, 991  
  SetLoggingEventPriorityLevel, 992  
  SystemPtrInternal, 993  
  TLSYSTEM, 993  
  UnregisterAllLoggingEvent, 992  
  UnregisterEvent, 992  
  UnregisterInterfaceEvent, 992  
  UnregisterLoggingEvent, 992  
  UpdateCameras, 993  
  UpdateInterfaceList, 993  
Spinnaker::ISystemEvent  
  ~ISystemEvent, 995  
  ISystemEvent, 995  
  OnInterfaceArrival, 995  
  OnInterfaceRemoval, 995  
  operator=, 996  
Spinnaker::Image  
  ~Image, 899  
  CalculateStatistics, 900  
  CheckCRC, 901  
  Convert, 901, 902  
  Create, 902, 903  
  CreateShared, 903  
  DEPRECATED\_FUNC, 904–906  
  DeepCopy, 903  
  ExtractPolarization, 906  
  GetBitsPerPixel, 907  
  GetBufferSize, 907  
  GetChunkData, 907  
  GetChunkLayoutId, 907  
  GetColorProcessing, 908  
  GetData, 908  
  GetDataAbsoluteMax, 908  
  GetDataAbsoluteMin, 909  
  GetDefaultColorProcessing, 909  
  GetFrameID, 909  
  GetHeight, 910  
  GetID, 910  
  GetImageData, 910  
  GetImageSize, 910  
  GetImageStatus, 911  
  GetImageStatusDescription, 911  
  GetNumChannels, 911  
  GetPayloadType, 911  
  GetPixelFormat, 912  
  GetPixelFormatIntType, 912  
  GetPixelFormatName, 912  
  GetPolarizationAlgorithm, 913  
  GetPolarizationValues, 913  
  GetPrivateData, 913  
  GetStride, 914  
  GetTLPayloadType, 914  
  GetTLPixelFormat, 915  
  GetTLPixelFormatNamespace, 915  
  GetTimeStamp, 914  
  GetValidPayloadSize, 915  
  GetWidth, 916  
  GetXOffset, 916  
  GetXPadding, 916  
  GetYOffset, 917  
  GetYPadding, 917  
  HasCRC, 917  
  IDataStream, 923  
  Image, 900  
  ImageConverter, 923  
  ImageFiler, 923  
  ImageStatsCalculator, 923  
  ImageUtilityImpl, 923

**ImageUtilityPolarizationImpl**, 923  
**IsCompressed**, 918  
**IsInUse**, 918  
**IsIncomplete**, 918  
**Release**, 918  
**ResetImage**, 919  
**Save**, 920–922  
**SetDefaultColorProcessing**, 922  
**Stream**, 924  
**Spinnaker::ImageEvent**  
    ~**ImageEvent**, 926  
    **ImageEvent**, 925  
    **OnImageEvent**, 926  
    **operator=**, 926  
**Spinnaker::ImagePtr**  
    ~**ImagePtr**, 928  
    **ImagePtr**, 928  
    **operator=**, 929  
**Spinnaker::ImageStatistics**  
    ~**ImageStatistics**, 931  
    **DisableAll**, 931  
    **EnableAll**, 931  
    **EnableGreyOnly**, 931  
    **EnableHSLOnly**, 932  
    **EnableRGBOnly**, 932  
    **GetChannelStatus**, 932  
    **GetHistogram**, 932  
    **GetMean**, 933  
    **GetNumPixelValues**, 933  
    **GetPixelValueRange**, 933  
    **GetRange**, 934  
    **GetStatistics**, 934  
    **ImageStatistics**, 931  
    **ImageStatsCalculator**, 936  
    **operator=**, 935  
    **SetChannelStatus**, 935  
**Spinnaker::ImageUtility**  
    **CreateNormalized**, 938–940  
    **CreateScaled**, 940  
    **ImageScalingAlgorithm**, 937  
    **SourceDataRange**, 937  
**Spinnaker::ImageUtilityHeatmap**  
    **CreateHeatmap**, 942, 943  
    **GetHeatmapColorGradient**, 943  
    **GetHeatmapRange**, 944  
    **HeatmapColor**, 942  
    **SetHeatmapColorGradient**, 944  
    **SetHeatmapRange**, 944  
**Spinnaker::ImageUtilityPolarization**  
    **CreateAolp**, 947  
    **CreateDolp**, 947, 948  
    **CreateGlareReduced**, 948, 949  
    **CreateStokesS0**, 949  
    **CreateStokesS1**, 950  
    **CreateStokesS2**, 951  
    **ExtractPolarQuadrant**, 952  
    **PolarizationQuadrant**, 946  
**Spinnaker::Interface**  
    ~**Interface**, 965  
    **GetCameras**, 965  
    **GetTLNodeMap**, 966  
    **InterfaceInternal**, 969  
    **IsInUse**, 966  
    **IsValid**, 966  
    **RegisterEvent**, 967  
    **SendActionCommand**, 967  
    **UnregisterEvent**, 968  
    **UpdateCameras**, 968  
**Spinnaker::InterfaceArrivalEvent**  
    ~**InterfaceArrivalEvent**, 970  
    **InterfaceArrivalEvent**, 970  
    **OnInterfaceArrival**, 970  
    **operator=**, 971  
**Spinnaker::InterfaceEvent**  
    ~**InterfaceEvent**, 973  
    **InterfaceEvent**, 972  
    **OnDeviceArrival**, 973  
    **OnDeviceRemoval**, 973  
    **operator=**, 973  
**Spinnaker::InterfaceList**  
    ~**InterfaceList**, 975  
    **Clear**, 975  
    **GetByIndex**, 976  
    **GetSize**, 976  
    **InterfaceList**, 975  
    **operator=**, 976  
    **operator[]**, 977  
    **SystemImpl**, 977  
**Spinnaker::InterfacePtr**  
    **InterfacePtr**, 978, 979  
**Spinnaker::InterfaceRemovalEvent**  
    ~**InterfaceRemovalEvent**, 981  
    **InterfaceRemovalEvent**, 980  
    **OnInterfaceRemoval**, 981  
    **operator=**, 981  
**Spinnaker::JPEGOption**  
    **JPEGOption**, 996  
    **progressive**, 997  
    **quality**, 997  
    **reserved**, 997  
**Spinnaker::JPQ2Option**  
    **JPG2Option**, 998  
    **quality**, 998  
    **reserved**, 998  
**Spinnaker::LibraryVersion**  
    **build**, 999  
    **major**, 999  
    **minor**, 999  
    **type**, 999  
**Spinnaker::LoggingEvent**  
    ~**LoggingEvent**, 1004  
    **LoggingEvent**, 1004  
    **OnLogEvent**, 1004  
    **operator=**, 1005  
**Spinnaker::LoggingEventData**  
    ~**LoggingEventData**, 1006

GetCategoryName, 1006  
GetLogMessage, 1006  
GetNDC, 1006  
GetPriority, 1007  
GetPriorityName, 1007  
GetThreadName, 1007  
GetTimestamp, 1007  
LoggingEventData, 1006  
SystemImpl, 1008  
Spinnaker::LoggingEventDataPtr  
LoggingEventDataPtr, 1009  
Spinnaker::PGMOption  
binaryFile, 1039  
PGMOption, 1039  
reserved, 1039  
Spinnaker::PNGOption  
compressionLevel, 1040  
interlaced, 1040  
PNGOption, 1040  
reserved, 1040  
Spinnaker::PPMOption  
binaryFile, 1052  
PPMOption, 1052  
reserved, 1052  
Spinnaker::RemovalEvent  
~RemovalEvent, 1058  
OnDeviceRemoval, 1058  
operator=, 1059  
RemovalEvent, 1058  
Spinnaker::System  
~System, 1075  
GetCameras, 1075  
GetInstance, 1076  
GetInterfaces, 1076  
GetLibraryVersion, 1076  
GetLoggingEventPriorityLevel, 1077  
GetTLNodeMap, 1077  
IsInUse, 1077  
RegisterEvent, 1078  
RegisterInterfaceEvent, 1078  
RegisterLoggingEvent, 1078  
ReleaseInstance, 1079  
SendActionCommand, 1079  
SetLoggingEventPriorityLevel, 1080  
System, 1075  
UnregisterAllLoggingEvent, 1080  
UnregisterEvent, 1081  
UnregisterInterfaceEvent, 1081  
UnregisterLoggingEvent, 1081  
UpdateCameras, 1082  
UpdateInterfaceList, 1082  
Spinnaker::SystemEvent  
~SystemEvent, 1084  
OnInterfaceArrival, 1084  
OnInterfaceRemoval, 1085  
operator=, 1085  
SystemEvent, 1084  
Spinnaker::SystemPtr  
~SystemPtr, 1087  
SystemPtr, 1086, 1087  
Spinnaker::TIFFOption  
compression, 1089  
CompressionMethod, 1088  
reserved, 1089  
TIFFOption, 1088  
Spinnaker::TransportLayerDevice  
~TransportLayerDevice, 1091  
CameraBase, 1092  
CameraInternal, 1092  
DeviceAccessStatus, 1092  
DeviceCurrentSpeed, 1092  
DeviceDisplayName, 1093  
DeviceDriverVersion, 1093  
DeviceEndianessMechanism, 1093  
DeviceID, 1093  
DeviceInstanceld, 1093  
DeviceUpdater, 1093  
DeviceLinkSpeed, 1094  
DeviceLocation, 1094  
DeviceModelName, 1094  
DeviceMulticastMonitorMode, 1094  
DeviceSerialNumber, 1094  
DeviceType, 1094  
DeviceU3VProtocol, 1095  
DeviceUserID, 1095  
DeviceVendorName, 1095  
DeviceVersion, 1095  
GUIXMLLocation, 1099  
GUIXMLPath, 1099  
GenICamXMLLocation, 1095  
GenICamXMLPath, 1095  
GevCCP, 1096  
GevDeviceDiscoverMaximumPacketSize, 1096  
GevDeviceForceGateway, 1096  
GevDeviceForceIPAddress, 1096  
GevDeviceForceIPEx, 1096  
GevDeviceForceIP, 1096  
GevDeviceForceSubnetMask, 1097  
GevDeviceGateway, 1097  
GevDeviceIPAddress, 1097  
GevDeviceIsWrongSubnet, 1097  
GevDeviceMACAddress, 1097  
GevDeviceMaximumPacketSize, 1097  
GevDeviceMaximumRetryCount, 1098  
GevDeviceModelsBigEndian, 1098  
GevDevicePort, 1098  
GevDeviceReadAndWriteTimeout, 1098  
GevDeviceSubnetMask, 1098  
GevVersionMajor, 1098  
GevVersionMinor, 1099  
ICameraBase, 1092  
TransportLayerDevice, 1091, 1092  
Spinnaker::TransportLayerInterface  
~TransportLayerInterface, 1102  
ActionCommand, 1102  
AutoForceIP, 1103

DeviceAccessStatus, 1103  
 DeviceCount, 1103  
 DeviceID, 1103  
 DeviceModelName, 1103  
 DeviceSelector, 1103  
 DeviceUnlock, 1104  
 DeviceUpdateList, 1104  
 DeviceVendorName, 1104  
 FilterDriverStatus, 1104  
 GevActionDeviceKey, 1104  
 GevActionGroupKey, 1104  
 GevActionGroupMask, 1105  
 GevActionTime, 1105  
 GevDeviceIPAddress, 1105  
 GevDeviceMACAddress, 1105  
 GevDeviceSubnetMask, 1105  
 GevInterfaceGateway, 1105  
 GevInterfaceIPAddress, 1106  
 GevInterfaceMACAddress, 1106  
 GevInterfaceMTU, 1106  
 GevInterfaceReceiveLinkSpeed, 1106  
 GevInterfaceSubnetMask, 1106  
 GevInterfaceTransmitLinkSpeed, 1106  
 HostAdapterDriverVersion, 1107  
 HostAdapterName, 1107  
 HostAdapterVendor, 1107  
 IInterface, 1102  
 IncompatibleDeviceCount, 1107  
 IncompatibleDeviceID, 1107  
 IncompatibleDeviceModelName, 1107  
 IncompatibleDeviceSelector, 1108  
 IncompatibleDeviceVendorName, 1108  
 IncompatibleGevDeviceIPAddress, 1108  
 IncompatibleGevDeviceMACAddress, 1108  
 IncompatibleGevDeviceSubnetMask, 1108  
 Interface, 1102  
 InterfaceDisplayName, 1108  
 InterfaceID, 1109  
 InterfaceInternal, 1102  
 InterfaceType, 1109  
 POEStatus, 1109  
 TransportLayerInterface, 1102  
 Spinnaker::TransportLayerStream  
   ~TransportLayerStream, 1111  
   CameraBase, 1111  
   CameralInternal, 1111  
   GevFailedPacketCount, 1112  
   GevMaximumNumberResendBuffers, 1112  
   GevMaximumNumberResendRequests, 1112  
   GevPacketResendMode, 1112  
   GevPacketResendTimeout, 1112  
   GevResendPacketCount, 1113  
   GevResendRequestCount, 1113  
   GevTotalPacketCount, 1113  
   ICameraBase, 1112  
   StreamBlockTransferSize, 1113  
   StreamBufferCountManual, 1113  
   StreamBufferCountMax, 1113  
 StreamBufferCountMode, 1114  
 StreamBufferCountResult, 1114  
 StreamBufferHandlingMode, 1114  
 StreamBufferUnderrunCount, 1114  
 StreamCRCCheckEnable, 1114  
 StreamDefaultBufferCount, 1114  
 StreamDefaultBufferCountMax, 1115  
 StreamDefaultBufferCountMode, 1115  
 StreamFailedBufferCount, 1115  
 StreamID, 1115  
 StreamTotalBufferCount, 1115  
 StreamType, 1115  
 TransportLayerStream, 1111  
 Spinnaker::TransportLayerSystem  
   ~TransportLayerSystem, 1117  
   AutoForceIP, 1117  
   EnumerateGEVInterfaces, 1118  
   ISystem, 1117  
   System, 1117  
   SystemPtrInternal, 1117  
   TransportLayerSystem, 1116, 1117  
 Spinnaker::Video, 440  
 Spinnaker::Video::AVIOption  
   AVIOption, 452  
   frameRate, 452  
   reserved, 452  
 Spinnaker::Video::H264Option  
   bitrate, 824  
   frameRate, 824  
   H264Option, 824  
   height, 825  
   reserved, 825  
   width, 825  
 Spinnaker::Video::MJPEGOption  
   frameRate, 1013  
   MJPEGOption, 1012  
   quality, 1013  
   reserved, 1013  
 Spinnaker::Video::SpinVideo  
   ~SpinVideo, 1062  
   Append, 1062  
   Close, 1063  
   Open, 1063, 1065  
   SetMaximumFileSize, 1065  
   SpinVideo, 1062  
 SpinnakerLogLevel  
   Spinnaker Definitions, 199  
 Standard  
   Types Enums, 372  
 StartRecording  
   Spinnaker::GenApi::PortNode, 1045  
   Spinnaker::GenApi::PortRecorder, 1048  
 StartStream  
   Spinnaker::IDataStream, 854  
 StatisticsChannel  
   Spinnaker Definitions, 199  
 Status  
   Spinnaker::ActionCommandResult, 441

StopRecording  
  IPortRecorder Interface, 325  
  Spinnaker::GenApi::PortNode, 1045  
  Spinnaker::GenApi::PortRecorder, 1048  
StopStream  
  Spinnaker::IDataStream, 854  
StoreToBag  
  Spinnaker::GenApi::CFeatureBag, 678  
Stream  
  Spinnaker::Event, 781  
  Spinnaker::Image, 873  
  Spinnaker::Image, 924  
StreamBlockTransferSize  
  Spinnaker::TransportLayerStream, 1113  
StreamBufferCountManual  
  Spinnaker::TransportLayerStream, 1113  
StreamBufferCountMax  
  Spinnaker::TransportLayerStream, 1113  
StreamBufferCountMode  
  Spinnaker::TransportLayerStream, 1114  
StreamBufferCountModeEnum  
  TransportLayerDefs Class, 214  
StreamBufferCountResult  
  Spinnaker::TransportLayerStream, 1114  
StreamBufferHandlingMode  
  Spinnaker::TransportLayerStream, 1114  
StreamBufferHandlingModeEnum  
  TransportLayerDefs Class, 214  
StreamBufferUnderrunCount  
  Spinnaker::TransportLayerStream, 1114  
StreamCRCCheckEnable  
  Spinnaker::TransportLayerStream, 1114  
StreamChannelId  
  GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 818  
  GVCP\_EVENT\_ITEM, 815  
StreamDefaultBufferCount  
  Spinnaker::TransportLayerStream, 1114  
StreamDefaultBufferCountMax  
  Spinnaker::TransportLayerStream, 1115  
StreamDefaultBufferCountMode  
  Spinnaker::TransportLayerStream, 1115  
StreamDefaultBufferCountModeEnum  
  TransportLayerDefs Class, 215  
StreamFailedBufferCount  
  Spinnaker::TransportLayerStream, 1115  
StreamID  
  Spinnaker::TransportLayerStream, 1115  
StreamTotalBufferCount  
  Spinnaker::TransportLayerStream, 1115  
StreamType  
  Spinnaker::TransportLayerStream, 1115  
StreamTypeEnum  
  TransportLayerDefs Class, 215  
StringList\_t  
  Types Enums, 367  
StringNode, 1066  
  Spinnaker::GenApi::StringNode, 1068  
TreeNode Class, 360  
  CStringRef, 360  
StringRegNode, 1070  
  Spinnaker::GenApi::StringRegNode, 1071, 1072  
StringRegNode Class, 361  
StructPort Class, 362  
SubMinor  
  Spinnaker::GenICam::Version\_t, 1126  
subnetLength  
  AdapterConfig::IpInfo, 985  
subnetMask  
  AdapterConfig::IpInfo, 985  
substr  
  Spinnaker::GenICam::gcstring, 813  
swap  
  Spinnaker::GenICam::gcstring, 813  
sync  
  Spinnaker::GenApi::ODevFileStreamBuf, 1038  
Synch Class, 363  
System, 1073  
  Spinnaker::System, 1075  
  Spinnaker::TransportLayerSystem, 1117  
System Class, 205  
System.h  
  FLIR\_SPINNAKER\_VERSION\_BUILD, 1389  
  FLIR\_SPINNAKER\_VERSION\_MAJOR, 1390  
  FLIR\_SPINNAKER\_VERSION\_MINOR, 1390  
  FLIR\_SPINNAKER\_VERSION\_TYPE, 1390  
SystemEvent, 1083  
  Spinnaker::SystemEvent, 1084  
SystemEvent Class, 206  
SystemImpl  
  Spinnaker::IInterface, 884  
  Spinnaker::InterfaceList, 977  
  Spinnaker::LoggingEventData, 1008  
SystemPtr, 1085  
  Spinnaker::SystemPtr, 1086, 1087  
SystemPtr Class, 207  
SystemPtrInternal  
  Spinnaker::ISystem, 993  
  Spinnaker::TransportLayerSystem, 1117  
TIFFOption, 1087  
  Spinnaker::TIFFOption, 1088  
TLDevice  
  Spinnaker::ICameraBase, 836  
TLInterface  
  Spinnaker::IInterface, 884  
TLPParamsLocked  
  Spinnaker::Camera, 603  
TLStream  
  Spinnaker::ICameraBase, 836  
TLSysyem  
  Spinnaker::ISystem, 993  
Test0001  
  Spinnaker::Camera, 600  
TestEventGenerate  
  Spinnaker::Camera, 601  
TestPattern  
  Spinnaker::Camera, 601

TestPatternEnums  
   CameraDefs Class, 145

TestPatternGeneratorSelector  
   Spinnaker::Camera, 601

TestPatternGeneratorSelectorEnums  
   CameraDefs Class, 145

TestPendingAck  
   Spinnaker::Camera, 601

ThrowBadAlloc  
   Spinnaker::GenICam, 439

TimerDelay  
   Spinnaker::Camera, 601

TimerDuration  
   Spinnaker::Camera, 602

TimerReset  
   Spinnaker::Camera, 602

TimerSelector  
   Spinnaker::Camera, 602

TimerSelectorEnums  
   CameraDefs Class, 146

TimerStatus  
   Spinnaker::Camera, 602

TimerStatusEnums  
   CameraDefs Class, 146

TimerTriggerActivation  
   Spinnaker::Camera, 602

TimerTriggerActivationEnums  
   CameraDefs Class, 146

TimerTriggerSource  
   Spinnaker::Camera, 602

TimerTriggerSourceEnums  
   CameraDefs Class, 147

TimerValue  
   Spinnaker::Camera, 603

Timestamp  
   Spinnaker::Camera, 603  
   U3V\_EVENT\_DATA, 1120

TimestampHigh  
   GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 818  
   GVCP\_EVENT\_ITEM, 815

TimestampLatch  
   Spinnaker::Camera, 603

TimestampLatchValue  
   Spinnaker::Camera, 603

TimestampLow  
   GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 818  
   GVCP\_EVENT\_ITEM, 816

TimestampReset  
   Spinnaker::Camera, 603

ToString  
   ISelectorDigit Interface, 330  
   IValue Class, 334  
   Spinnaker::GenApi::CNodeMapFactory, 716  
   Spinnaker::GenApi::CSelectorSet, 747  
   Spinnaker::GenApi::EAccessModeClass, 758  
   Spinnaker::GenApi::ECachingModeClass, 759  
   Spinnaker::GenApi::EDisplayNotationClass, 760  
   Spinnaker::GenApi::EEndianessClass, 761

Spinnaker::GenApi::EGenApiSchemaVersion←  
   Class, 762

Spinnaker::GenApi::EInputDirectionClass, 763

Spinnaker::GenApi::ENamespaceClass, 764

Spinnaker::GenApi::ERepresentationClass, 774

Spinnaker::GenApi::ESignClass, 775

Spinnaker::GenApi::ESlopeClass, 776

Spinnaker::GenApi::EStandardNameSpaceClass,  
   777

Spinnaker::GenApi::EVisibilityClass, 782

Spinnaker::GenApi::EYesNoClass, 788

Spinnaker::GenApi::ValueNode, 1124

ToXml  
   Spinnaker::GenApi::CNodeMapFactory, 716

Tokenize  
   GCUtilities Utility, 274

topLeftXCoord  
   Chunk Data Inference Class, 164, 165

topLeftYCoord  
   Chunk Data Inference Class, 165

TransferAbort  
   Spinnaker::Camera, 604

TransferBlockCount  
   Spinnaker::Camera, 604

TransferBurstCount  
   Spinnaker::Camera, 604

TransferComponentSelector  
   Spinnaker::Camera, 604

TransferComponentSelectorEnums  
   CameraDefs Class, 148

TransferControlMode  
   Spinnaker::Camera, 604

TransferControlModeEnums  
   CameraDefs Class, 148

TransferOperationMode  
   Spinnaker::Camera, 604

TransferOperationModeEnums  
   CameraDefs Class, 149

TransferPause  
   Spinnaker::Camera, 605

TransferQueueCurrentBlockCount  
   Spinnaker::Camera, 605

TransferQueueMaxBlockCount  
   Spinnaker::Camera, 605

TransferQueueMode  
   Spinnaker::Camera, 605

TransferQueueModeEnums  
   CameraDefs Class, 149

TransferQueueOverflowCount  
   Spinnaker::Camera, 605

TransferResume  
   Spinnaker::Camera, 605

TransferSelector  
   Spinnaker::Camera, 606

TransferSelectorEnums  
   CameraDefs Class, 149

TransferStart  
   Spinnaker::Camera, 606

TransferStatus  
    Spinnaker::Camera, 606

TransferStatusSelector  
    Spinnaker::Camera, 606

TransferStatusSelectorEnums  
    CameraDefs Class, 150

TransferStop  
    Spinnaker::Camera, 606

TransferStreamChannel  
    Spinnaker::Camera, 606

TransferTriggerActivation  
    Spinnaker::Camera, 607

TransferTriggerActivationEnums  
    CameraDefs Class, 150

TransferTriggerMode  
    Spinnaker::Camera, 607

TransferTriggerModeEnums  
    CameraDefs Class, 150

TransferTriggerSelector  
    Spinnaker::Camera, 607

TransferTriggerSelectorEnums  
    CameraDefs Class, 151

TransferTriggerSource  
    Spinnaker::Camera, 607

TransferTriggerSourceEnums  
    CameraDefs Class, 151

transmitBuffers  
    AdapterConfig::AdapterInfo, 446

transmitBuffersMax  
    AdapterConfig::AdapterInfo, 446

transmitBuffersMin  
    AdapterConfig::AdapterInfo, 446

transmitBuffersRegKey  
    AdapterConfig::AdapterInfo, 446

transmitBuffersStep  
    AdapterConfig::AdapterInfo, 446

TransportLayerDefs Class, 209  
    DeviceAccessStatusEnum, 210  
    DeviceCurrentSpeedEnum, 211  
    DeviceEndianessMechanismEnum, 211  
    DeviceTypeEnum, 212  
    FilterDriverStatusEnum, 212  
    GUIXMLLocationEnum, 213  
    GenICamXMLLocationEnum, 212  
    GevCCPEnum, 213  
    POEStatusEnum, 213  
    StreamBufferCountModeEnum, 214  
    StreamBufferHandlingModeEnum, 214  
    StreamDefaultBufferCountModeEnum, 215  
    StreamTypeEnum, 215

TransportLayerDevice, 1089  
    Spinnaker::TransportLayerDevice, 1091, 1092

TransportLayerDevice Class, 216

TransportLayerInterface, 1099  
    Spinnaker::TransportLayerInterface, 1102

TransportLayerInterface Class, 217

TransportLayerStream, 1109  
    Spinnaker::TransportLayerStream, 1111

TransportLayerStream Class, 218

TransportLayerStreamInfo  
    Spinnaker::IDataStream, 854

TransportLayerSystem, 1116  
    Spinnaker::TransportLayerSystem, 1116, 1117

TransportLayerSystem Class, 219

TriggerActivation  
    Spinnaker::Camera, 607

TriggerActivationEnums  
    CameraDefs Class, 152

TriggerDelay  
    Spinnaker::Camera, 607

TriggerDivider  
    Spinnaker::Camera, 608

TriggerEventTest  
    Spinnaker::Camera, 608

TriggerMode  
    Spinnaker::Camera, 608

TriggerModeEnums  
    CameraDefs Class, 153

TriggerMultiplier  
    Spinnaker::Camera, 608

TriggerOverlap  
    Spinnaker::Camera, 608

TriggerOverlapEnums  
    CameraDefs Class, 153

TriggerSelector  
    Spinnaker::Camera, 609

TriggerSelectorEnums  
    CameraDefs Class, 153

TriggerSoftware  
    Spinnaker::Camera, 609

TriggerSource  
    Spinnaker::Camera, 609

TriggerSourceEnums  
    CameraDefs Class, 153

TryLock  
    Spinnaker::GenApi::CLOCK, 702  
    Spinnaker::GenICam::CGlobalLock, 683  
    Spinnaker::GenICam::Clock, 699

type  
    Spinnaker::LibraryVersion, 999

Types Enums, 365  
    \_UndefinedRepresentation, 367  
    Automatic, 373  
    Beginner, 374  
    Boolean, 372  
    CL, 373  
    Custom, 372  
    Decreasing, 373  
    EAccessMode, 368  
    ECachingMode, 368  
    EDisplayNotation, 368  
    EEndianess, 370  
    EGenApiSchemaVersion, 370  
    EIncMode, 370  
    EInputDirection, 371  
    EInterfaceType, 371

ELinkType, 371  
 ENameSpace, 372  
 ERepresentation, 372  
 ESign, 372  
 ESlope, 373  
 EStandardNameSpace, 373  
 EVisibility, 373  
 EXMLValidation, 374  
 EYesNo, 374  
 Expert, 374  
 Guru, 374  
 Increasing, 373  
 Invisible, 374  
 Linear, 372  
 Logarithmic, 372  
 NA, 368  
 NI, 368  
 No, 374  
 None, 373  
 RO, 368  
 RW, 368  
 Signed, 373  
 Standard, 372  
 StringList\_t, 367  
 Unsigned, 373  
 Varying, 373  
 WO, 368  
 Yes, 374  
**Types.h**  
 interface, 1376  
**U3V\_CHUNK\_TRAILER**, 1118  
 ChunkID, 1118  
 ChunkLength, 1118  
**U3V\_COMMAND\_HEADER**, 1119  
 CommandId, 1119  
 Flags, 1119  
 Length, 1119  
 Prefix, 1119  
 ReqId, 1120  
**U3V\_EVENT\_DATA**, 1120  
 EventId, 1120  
 Reserved, 1120  
 Timestamp, 1120  
**U3V\_EVENT\_MESSAGE**, 1121  
 CommandHeader, 1121  
 EventData, 1121  
**U3V\_EVENT\_PREFIX**  
 Spinnaker::GenApi, 437  
**USE\_TEMP\_CACHE\_FILE**  
 GCUtilities.h, 1298  
 underflow  
 Spinnaker::GenApi::IDevFileStreamBuf, 859  
 Unlock  
 Spinnaker::GenApi::CLock, 702  
 Spinnaker::GenICam::CGlobalLock, 684  
 Spinnaker::GenICam::CLock, 700  
 UnlockEarly  
 Spinnaker::GenICam::CGlobalLockUnlocker, 685  
 UnregisterAllLoggingEvent  
 Spinnaker::ISystem, 992  
 Spinnaker::System, 1080  
 UnregisterEvent  
 Spinnaker::CameraBase, 627  
 Spinnaker::ICameraBase, 835  
 Spinnaker::IInterface, 883  
 Spinnaker::ISystem, 992  
 Spinnaker::Interface, 968  
 Spinnaker::System, 1081  
 UnregisterImageEvent  
 Spinnaker::IDataStream, 854  
 UnregisterInterfaceEvent  
 Spinnaker::ISystem, 992  
 Spinnaker::System, 1081  
 UnregisterLoggingEvent  
 Spinnaker::ISystem, 992  
 Spinnaker::System, 1081  
 Unsigned  
 Types Enums, 373  
 UpdateBuffer  
 Spinnaker::GenApi::CChunkAdapter, 640  
 Spinnaker::GenApi::CChunkPort, 654  
 UpdateCameras  
 Spinnaker::IInterface, 883  
 Spinnaker::ISystem, 993  
 Spinnaker::Interface, 968  
 Spinnaker::System, 1082  
 UpdateFirmware  
 SpinUpdate.h, 1385  
 UpdateFirmwareConsole  
 SpinUpdate.h, 1386  
 UpdateFirmwareGUI  
 SpinUpdate.h, 1386  
 UpdateInterfaceList  
 Spinnaker::ISystem, 993  
 Spinnaker::System, 1082  
 UpdaterMessageCallback  
 SpinUpdate.h, 1386  
 UpdaterProgressCallback  
 SpinUpdate.h, 1386  
 UrlDecode  
 GCUtilities Utility, 274  
 UrlEncode  
 GCUtilities Utility, 275  
 UserOutputSelector  
 Spinnaker::Camera, 609  
 UserOutputSelectorEnums  
 CameraDefs Class, 154  
 UserOutputValue  
 Spinnaker::Camera, 609  
 UserOutputValueAll  
 Spinnaker::Camera, 610  
 UserOutputValueAllMask  
 Spinnaker::Camera, 610  
 UserSetDefault  
 Spinnaker::Camera, 610  
 UserSetDefaultEnums

CameraDefs Class, 154  
UserSetFeatureEnable  
    Spinnaker::Camera, 610  
UserSetLoad  
    Spinnaker::Camera, 610  
UserSetSave  
    Spinnaker::Camera, 611  
UserSetSelector  
    Spinnaker::Camera, 611  
UserSetSelectorEnums  
    CameraDefs Class, 155

V3\_3Enable  
    Spinnaker::Camera, 611  
ValidateIpAddress  
    AdapterConfig, 381  
ValueNode, 1122  
    Spinnaker::GenApi::ValueNode, 1123  
ValueNode Class, 375  
    CValueRef, 375  
Varying  
    Types Enums, 373  
Verify  
    IBoolean Interface, 277  
Version\_t, 1125

WaitOnImageEvent  
    Spinnaker::IDataStream, 854  
what  
    Spinnaker::Exception, 787  
WhiteClip  
    Spinnaker::Camera, 611  
WhiteClipSelector  
    Spinnaker::Camera, 611  
WhiteClipSelectorEnums  
    CameraDefs Class, 155  
Width  
    Spinnaker::Camera, 612  
width  
    Spinnaker::Video::H264Option, 825  
WidthMax  
    Spinnaker::Camera, 612  
WO  
    Types Enums, 368  
Write  
    IPort Interface, 321  
    Spinnaker::GenApi::CChunkPort, 654  
    Spinnaker::GenApi::CEventPort, 675  
    Spinnaker::GenApi::CPortImpl, 737  
    Spinnaker::GenApi::CPortWriteList, 740  
    Spinnaker::GenApi::CRegisterPortImpl, 743  
    Spinnaker::GenApi::CTestPortStruct, 750  
    Spinnaker::GenApi::PortNode, 1046

write  
    Spinnaker::GenApi::FileProtocolAdapter, 793  
WritePort  
    Spinnaker::CameraBase, 627  
    Spinnaker::ICameraBase, 835  
WriteRegister