Spinnaker C++

2.0.0.0

Generated by Doxygen 1.8.13

Contents

1 Getting Started **1**

2 Programmer's Guide **3**

3 Benefits of Spinnaker **5**

4 FlyCapture2 Feature Comparison with Spinnaker **7**

5 Working with GenICam GenTL Devices **9**

5.1 GenTL Overview 9

5.2 Installation 9

5.3 Troubleshooting 10

5.3.1 Enable FLIR GenTL Logging 10

5.3.2 USB3 Device Image Tearing 10

6 Software Licensing Information **11**

7 Module Index **13**

7.1 Modules 13

8 Namespace Index **17**

8.1 Namespace List 17

9 Hierarchical Index **19**

9.1 Class Hierarchy 19

10 Class Index **25**

10.1 Class List 25

11 File Index 33

11.1 File List . . . 33

12 Module Documentation 39

12.1 Spinnaker Classes . . . 40

12.1.1 Detailed Description . . . 43

12.1.2 Enumeration Type Documentation . . . 43

12.1.2.1 InferenceBoxType . . . 43

12.2 AVI Recorder Class . . . 44

12.2.1 Detailed Description . . . 44

12.2.2 Function Documentation . . . 44

12.2.2.1 DEPRECATED_CLASS() . . . 44

12.3 BasePtr Class . . . 47

12.3.1 Detailed Description . . . 47

12.3.2 Function Documentation . . . 47

12.3.2.1 operator==() . . . 47

12.4 Camera Class . . . 48

12.4.1 Detailed Description . . . 48

12.5 Camera Base Class . . . 49

12.5.1 Detailed Description . . . 49

12.6 CameraDefs Class . . . 50

12.6.1 Detailed Description . . . 82

12.6.2 Enumeration Type Documentation . . . 82

12.6.2.1 AcquisitionModeEnums . . . 82

12.6.2.2 AcquisitionStatusSelectorEnums . . . 82

12.6.2.3 ActionUnconditionalModeEnums . . . 83

12.6.2.4 AdcBitDepthEnums . . . 83

12.6.2.5 AutoAlgorithmSelectorEnums . . . 83

12.6.2.6 AutoExposureControlPriorityEnums . . . 84

12.6.2.7 AutoExposureLightingModeEnums . . . 84

12.6.2.8 AutoExposureMeteringModeEnums . . . 84

12.6.2.9 AutoExposureTargetGreyValueAutoEnums 85
12.6.2.10 BalanceRatioSelectorEnums 85
12.6.2.11 BalanceWhiteAutoEnums 86
12.6.2.12 BalanceWhiteAutoProfileEnums 86
12.6.2.13 BinningHorizontalModeEnums 86
12.6.2.14 BinningSelectorEnums 87
12.6.2.15 BinningVerticalModeEnums 87
12.6.2.16 BlackLevelAutoBalanceEnums 87
12.6.2.17 BlackLevelAutoEnums 88
12.6.2.18 BlackLevelSelectorEnums 88
12.6.2.19 ChunkBlackLevelSelectorEnums 88
12.6.2.20 ChunkCounterSelectorEnums 89
12.6.2.21 ChunkEncoderSelectorEnums 89
12.6.2.22 ChunkEncoderStatusEnums 89
12.6.2.23 ChunkExposureTimeSelectorEnums 89
12.6.2.24 ChunkGainSelectorEnums 90
12.6.2.25 ChunkImageComponentEnums 90
12.6.2.26 ChunkPixelFormatEnums 91
12.6.2.27 ChunkRegionIDEnums 91
12.6.2.28 ChunkScan3dCoordinateReferenceSelectorEnums 92
12.6.2.29 ChunkScan3dCoordinateSelectorEnums 92
12.6.2.30 ChunkScan3dCoordinateSystemEnums 92
12.6.2.31 ChunkScan3dCoordinateSystemReferenceEnums 93
12.6.2.32 ChunkScan3dCoordinateTransformSelectorEnums 93
12.6.2.33 ChunkScan3dDistanceUnitEnums 93
12.6.2.34 ChunkScan3dOutputModeEnums 94
12.6.2.35 ChunkSelectorEnums 95
12.6.2.36 ChunkSourceIDEnums 95
12.6.2.37 ChunkTimerSelectorEnums 95
12.6.2.38 ChunkTransferStreamIDEnums 96

12.6.2.39 ClConfigurationEnums 96
12.6.2.40 ClTimeSlotsCountEnums 97
12.6.2.41 ColorTransformationSelectorEnums 97
12.6.2.42 ColorTransformationValueSelectorEnums 97
12.6.2.43 CounterEventActivationEnums 98
12.6.2.44 CounterEventSourceEnums 98
12.6.2.45 CounterResetActivationEnums 99
12.6.2.46 CounterResetSourceEnums 99
12.6.2.47 CounterSelectorEnums 100
12.6.2.48 CounterStatusEnums 100
12.6.2.49 CounterTriggerActivationEnums 100
12.6.2.50 CounterTriggerSourceEnums 101
12.6.2.51 CxpConnectionTestModeEnums 101
12.6.2.52 CxpLinkConfigurationEnums 101
12.6.2.53 CxpLinkConfigurationPreferredEnums 102
12.6.2.54 CxpLinkConfigurationStatusEnums 103
12.6.2.55 CxpPoCxpStatusEnums 104
12.6.2.56 DecimationHorizontalModeEnums 105
12.6.2.57 DecimationSelectorEnums 105
12.6.2.58 DecimationVerticalModeEnums 105
12.6.2.59 DefectCorrectionModeEnums 106
12.6.2.60 DeinterlacingEnums 106
12.6.2.61 DeviceCharacterSetEnums 106
12.6.2.62 DeviceClockSelectorEnums 107
12.6.2.63 DeviceConnectionStatusEnums 107
12.6.2.64 DeviceIndicatorModeEnums 107
12.6.2.65 DeviceLinkHeartbeatModeEnums 107
12.6.2.66 DeviceLinkThroughputLimitModeEnums 109
12.6.2.67 DevicePowerSupplySelectorEnums 109
12.6.2.68 DeviceRegistersEndiannessEnums 109

12.6.2.69 DeviceScanTypeEnums 110
12.6.2.70 DeviceSerialPortBaudRateEnums 110
12.6.2.71 DeviceSerialPortSelectorEnums 110
12.6.2.72 DeviceStreamChannelEndiannessEnums 111
12.6.2.73 DeviceStreamChannelTypeEnums 111
12.6.2.74 DeviceTapGeometryEnums 111
12.6.2.75 DeviceTemperatureSelectorEnums 112
12.6.2.76 DeviceTLTypeEnums 113
12.6.2.77 DeviceTypeEnums 113
12.6.2.78 EncoderModeEnums 113
12.6.2.79 EncoderOutputModeEnums 114
12.6.2.80 EncoderResetActivationEnums 114
12.6.2.81 EncoderResetSourceEnums 115
12.6.2.82 EncoderSelectorEnums 116
12.6.2.83 EncoderSourceAEnums 116
12.6.2.84 EncoderSourceBEnums 116
12.6.2.85 EncoderStatusEnums 117
12.6.2.86 EventNotificationEnums 117
12.6.2.87 EventSelectorEnums 117
12.6.2.88 ExposureActiveModeEnums 118
12.6.2.89 ExposureAutoEnums 118
12.6.2.90 ExposureModeEnums 118
12.6.2.91 ExposureTimeModeEnums 119
12.6.2.92 ExposureTimeSelectorEnums 119
12.6.2.93 FileOpenModeEnums 120
12.6.2.94 FileOperationSelectorEnums 120
12.6.2.95 FileOperationStatusEnums 120
12.6.2.96 FileSelectorEnums 121
12.6.2.97 GainAutoBalanceEnums 121
12.6.2.98 GainAutoEnums 121

12.6.2.99 GainSelectorEnums 122
12.6.2.100 GevCCPEnums 122
12.6.2.101 GevCurrentPhysicalLinkConfigurationEnums 122
12.6.2.102 GevGVCPExtendedStatusCodesSelectorEnums 123
12.6.2.103 GevGVSPExtendedIDModeEnums 123
12.6.2.104 GevIEEE1588ClockAccuracyEnums 123
12.6.2.105 GevIEEE1588ModeEnums 123
12.6.2.106 GevIEEE1588StatusEnums 124
12.6.2.107 GevIPConfigurationStatusEnums 124
12.6.2.108 GevPhysicalLinkConfigurationEnums 125
12.6.2.109 GevSupportedOptionSelectorEnums 125
12.6.2.110 ImageComponentSelectorEnums 126
12.6.2.111 ImageCompressionJPEGFormatOptionEnums 126
12.6.2.112 ImageCompressionModeEnums 127
12.6.2.113 ImageCompressionRateOptionEnums 127
12.6.2.114 LineFormatEnums 127
12.6.2.115 LineInputFilterSelectorEnums 128
12.6.2.116 LineModeEnums 128
12.6.2.117 LineSelectorEnums 128
12.6.2.118 LineSourceEnums 129
12.6.2.119 LogicBlockLUTInputActivationEnums 129
12.6.2.120 LogicBlockLUTInputSelectorEnums 130
12.6.2.121 LogicBlockLUTInputSourceEnums 130
12.6.2.122 LogicBlockLUTSelectorEnums 131
12.6.2.123 LogicBlockSelectorEnums 131
12.6.2.124 LUTSelectorEnums 131
12.6.2.125 PixelColorFilterEnums 132
12.6.2.126 PixelFormatEnums 132
12.6.2.127 PixelFormatInfoSelectorEnums 138
12.6.2.128 PixelSizeEnums 143

12.6.2.129 RegionDestinationEnums 144
12.6.2.130 RegionModeEnums 144
12.6.2.131 RegionSelectorEnums 145
12.6.2.132 RgbTransformLightSourceEnums 145
12.6.2.133 Scan3dCoordinateReferenceSelectorEnums 146
12.6.2.134 Scan3dCoordinateSelectorEnums 146
12.6.2.135 Scan3dCoordinateSystemEnums 146
12.6.2.136 Scan3dCoordinateSystemReferenceEnums 147
12.6.2.137 Scan3dCoordinateTransformSelectorEnums 147
12.6.2.138 Scan3dDistanceUnitEnums 147
12.6.2.139 Scan3dOutputModeEnums 148
12.6.2.140 SensorDigitizationTapsEnums 148
12.6.2.141 SensorShutterModeEnums 149
12.6.2.142 SensorTapsEnums 149
12.6.2.143 SequencerConfigurationModeEnums 150
12.6.2.144 SequencerConfigurationValidEnums 150
12.6.2.145 SequencerModeEnums 150
12.6.2.146 SequencerSetValidEnums 150
12.6.2.147 SequencerTriggerActivationEnums 151
12.6.2.148 SequencerTriggerSourceEnums 151
12.6.2.149 SerialPortBaudRateEnums 151
12.6.2.150 SerialPortParityEnums 152
12.6.2.151 SerialPortSelectorEnums 152
12.6.2.152 SerialPortSourceEnums 153
12.6.2.153 SerialPortStopBitsEnums 153
12.6.2.154 SoftwareSignalSelectorEnums 153
12.6.2.155 SourceSelectorEnums 154
12.6.2.156 TestPatternEnums 154
12.6.2.157 TestPatternGeneratorSelectorEnums 154
12.6.2.158 TimerSelectorEnums 155

12.6.2.159 TimerStatusEnums . . . 155
12.6.2.160 TimerTriggerActivationEnums . . . 155
12.6.2.161 TimerTriggerSourceEnums . . . 156
12.6.2.162 TransferComponentSelectorEnums . . . 157
12.6.2.163 TransferControlModeEnums . . . 157
12.6.2.164 TransferOperationModeEnums . . . 158
12.6.2.165 TransferQueueModeEnums . . . 158
12.6.2.166 TransferSelectorEnums . . . 158
12.6.2.167 TransferStatusSelectorEnums . . . 159
12.6.2.168 TransferTriggerActivationEnums . . . 159
12.6.2.169 TransferTriggerModeEnums . . . 159
12.6.2.170 TransferTriggerSelectorEnums . . . 160
12.6.2.171 TransferTriggerSourceEnums . . . 160
12.6.2.172 TriggerActivationEnums . . . 161
12.6.2.173 TriggerModeEnums . . . 162
12.6.2.174 TriggerOverlapEnums . . . 162
12.6.2.175 TriggerSelectorEnums . . . 162
12.6.2.176 TriggerSourceEnums . . . 163
12.6.2.177 UserOutputSelectorEnums . . . 163
12.6.2.178 UserSetDefaultEnums . . . 163
12.6.2.179 UserSetSelectorEnums . . . 164
12.6.2.180 WhiteClipSelectorEnums . . . 164
12.7 Camera List Class . . . 165
12.7.1 Detailed Description . . . 165
12.8 CameraPtr Class . . . 166
12.8.1 Detailed Description . . . 166
12.8.2 Function Documentation . . . 166
12.8.2.1 CameraPtr() [1/4] . . . 166
12.8.2.2 CameraPtr() [2/4] . . . 166
12.8.2.3 CameraPtr() [3/4] . . . 167

12.8.2.4 CameraPtr() [4/4] . . . 167
12.9 ChunkData Class . . . 168
12.9.1 Detailed Description . . . 168
12.10 Chunk Data Inference Class . . . 169
12.10.1 Detailed Description . . . 170
12.10.2 Function Documentation . . . 170
12.10.2.1 GetBoxAt() . . . 170
12.10.2.2 GetBoxCount() . . . 170
12.10.2.3 GetBoxSize() . . . 170
12.10.2.4 GetVersion() . . . 171
12.10.2.5 InferenceBoundingBoxResult() [1/3] . . . 171
12.10.2.6 InferenceBoundingBoxResult() [2/3] . . . 171
12.10.2.7 InferenceBoundingBoxResult() [3/3] . . . 171
12.10.2.8 operator=() . . . 171
12.10.2.9 ∼InferenceBoundingBoxResult() . . . 172
12.10.3 Variable Documentation . . . 172
12.10.3.1 bottomRightXCoord [1/2] . . . 172
12.10.3.2 bottomRightXCoord [2/2] . . . 172
12.10.3.3 bottomRightYCoord [1/2] . . . 172
12.10.3.4 bottomRightYCoord [2/2] . . . 172
12.10.3.5 boxType . . . 172
12.10.3.6 centerXCoord . . . 172
12.10.3.7 centerYCoord . . . 173
12.10.3.8 circle . . . 173
12.10.3.9 classId . . . 173
12.10.3.10 confidence . . . 173
12.10.3.11 radius . . . 173
12.10.3.12 rect . . . 173
12.10.3.13 rotatedRect . . . 173
12.10.3.14 rotationAngle . . . 173

12.10.3.15 topLeftXCoord [1/2] . . . 174
12.10.3.16 topLeftXCoord [2/2] . . . 174
12.10.3.17 topLeftYCoord [1/2] . . . 174
12.10.3.18 topLeftYCoord [2/2] . . . 174
12.11 Spinnaker EventHandler Classes . . . 175
12.11.1 Detailed Description . . . 176
12.12 DeviceArrivalEventHandler Class . . . 177
12.12.1 Detailed Description . . . 177
12.13 DeviceEventHandler Class . . . 178
12.13.1 Detailed Description . . . 178
12.14 DeviceRemovalEventHandler Class . . . 179
12.14.1 Detailed Description . . . 179
12.15 EventHandler Class . . . 180
12.15.1 Detailed Description . . . 180
12.16 Exception Class . . . 181
12.16.1 Detailed Description . . . 181
12.17 Image Class . . . 182
12.17.1 Detailed Description . . . 182
12.18 ImageEventHandler Class . . . 183
12.18.1 Detailed Description . . . 183
12.19 ImagePtr Class . . . 184
12.19.1 Detailed Description . . . 184
12.20 ImageStatistics Class . . . 185
12.20.1 Detailed Description . . . 185
12.21 Image Utility Class . . . 186
12.21.1 Detailed Description . . . 186
12.22 Image Utility Heatmap Class . . . 187
12.22.1 Detailed Description . . . 187
12.23 Image Utility Polarization Class . . . 188
12.23.1 Detailed Description . . . 188

12.24 Interface Class . . . 189
12.24.1 Detailed Description . . . 189
12.25 InterfaceArrivalEventHandler Class . . . 190
12.25.1 Detailed Description . . . 190
12.26 InterfaceEventHandler Class . . . 191
12.26.1 Detailed Description . . . 191
12.27 InterfaceList Class . . . 192
12.27.1 Detailed Description . . . 192
12.28 InterfacePtr Class . . . 193
12.28.1 Detailed Description . . . 193
12.29 InterfaceRemovalEventHandler Class . . . 194
12.29.1 Detailed Description . . . 194
12.30 Logging EventHandler Class . . . 195
12.30.1 Detailed Description . . . 195
12.31 LoggingEventDataPtr Class . . . 196
12.31.1 Detailed Description . . . 196
12.32 LoggingEventHandler Class . . . 197
12.32.1 Detailed Description . . . 197
12.33 Spinnaker Headers . . . 198
12.33.1 Detailed Description . . . 199
12.33.2 Variable Documentation . . . 199
12.33.2.1 EVENT_TIMEOUT_INFINITE . . . 199
12.33.2.2 EVENT_TIMEOUT_NONE . . . 199
12.34 Spinnaker.h . . . 200
12.35 Spinnaker Definitions . . . 201
12.35.1 Detailed Description . . . 204
12.35.2 Enumeration Type Documentation . . . 204
12.35.2.1 ActionCommandStatus . . . 204
12.35.2.2 BufferOwnership . . . 205
12.35.2.3 ColorProcessingAlgorithm . . . 205

12.35.2.4 Error . . . 206
12.35.2.5 EventType . . . 207
12.35.2.6 ImageFileFormat . . . 207
12.35.2.7 ImageStatus . . . 208
12.35.2.8 PayloadTypeInfoIDs . . . 208
12.35.2.9 PixelFormatIntType . . . 209
12.35.2.10 PixelFormatNamespaceID . . . 209
12.35.2.11 SpinnakerLogLevel . . . 210
12.35.2.12 StatisticsChannel . . . 210
12.36 Spinnaker Platform . . . 211
12.36.1 Detailed Description . . . 211
12.36.2 Macro Definition Documentation . . . 211
12.36.2.1 SPINNAKER_API . . . 211
12.36.2.2 SPINNAKER_API_ABSTRACT . . . 211
12.36.2.3 SPINNAKER_LOCAL . . . 211
12.37 Spinnaker Video Class . . . 212
12.37.1 Detailed Description . . . 212
12.38 Spinnaker Video Definitions . . . 213
12.39 System Class . . . 214
12.39.1 Detailed Description . . . 214
12.40 SystemEventHandler Class . . . 215
12.40.1 Detailed Description . . . 215
12.41 SystemPtr Class . . . 216
12.41.1 Detailed Description . . . 216
12.42 Spinnaker QuickSpin Classes . . . 217
12.42.1 Detailed Description . . . 217
12.43 TransportLayerDefs Class . . . 218
12.43.1 Detailed Description . . . 219
12.43.2 Enumeration Type Documentation . . . 219
12.43.2.1 DeviceAccessStatusEnum . . . 220

12.43.2.2 DeviceCurrentSpeedEnum . . . 220
12.43.2.3 DeviceEndianessMechanismEnum . . . 220
12.43.2.4 DeviceTypeEnum . . . 221
12.43.2.5 FilterDriverStatusEnum . . . 221
12.43.2.6 GenICamXMLLocationEnum . . . 221
12.43.2.7 GevCCPEnum . . . 222
12.43.2.8 GUIXMLLocationEnum . . . 222
12.43.2.9 InterfaceTypeEnum . . . 222
12.43.2.10 POEStatusEnum . . . 223
12.43.2.11 StreamBufferCountModeEnum . . . 223
12.43.2.12 StreamBufferHandlingModeEnum . . . 223
12.43.2.13 StreamTypeEnum . . . 224
12.43.2.14 TLTypeEnum . . . 224
12.44 TransportLayerDevice Class . . . 226
12.44.1 Detailed Description . . . 226
12.45 TransportLayerInterface Class . . . 227
12.45.1 Detailed Description . . . 227
12.46 TransportLayerStream Class . . . 228
12.46.1 Detailed Description . . . 228
12.47 TransportLayerSystem Class . . . 229
12.47.1 Detailed Description . . . 229
12.48 Camera Base Interface Class . . . 230
12.48.1 Detailed Description . . . 230
12.49 IChunkData Class . . . 231
12.49.1 Detailed Description . . . 231
12.50 IImage Class . . . 232
12.50.1 Detailed Description . . . 232
12.51 IImageStatistics Class . . . 233
12.51.1 Detailed Description . . . 233
12.52 IInterface Class . . . 234

12.52.1 Detailed Description . . . 234
12.53 IInterfaceList Class . . . 235
12.53.1 Detailed Description . . . 235
12.54 ISystem Class . . . 236
12.54.1 Detailed Description . . . 236
12.55 Spinnaker GenApi Classes . . . 237
12.55.1 Detailed Description . . . 243
12.55.2 Typedef Documentation . . . 243
12.55.2.1 CNodeMapRef . . . 243
12.55.2.2 CNodeRef . . . 243
12.55.2.3 CSelectorRef . . . 243
12.55.3 Function Documentation . . . 243
12.55.3.1 _ClearXMLCache() . . . 243
12.55.3.2 _Connect() [1/2] . . . 244
12.55.3.3 _Connect() [2/2] . . . 244
12.55.3.4 _Destroy() . . . 244
12.55.3.5 _GetDeviceName() . . . 244
12.55.3.6 _GetNode() . . . 244
12.55.3.7 _GetNodes() . . . 244
12.55.3.8 _GetSupportedSchemaVersions() . . . 244
12.55.3.9 _InvalidateNodes() . . . 245
12.55.3.10 _LoadXMLFromFile() . . . 245
12.55.3.11 _LoadXMLFromFileInject() . . . 245
12.55.3.12 _LoadXMLFromString() . . . 245
12.55.3.13 _LoadXMLFromStringInject() . . . 245
12.55.3.14 _LoadXMLFromZIPData() . . . 245
12.55.3.15 _LoadXMLFromZIPFile() . . . 245
12.55.3.16 _Poll() . . . 246
12.55.3.17 CastToIDestroy() . . . 246
12.55.3.18 CNodeMapRefT() [1/3] . . . 246

12.55.3.19 CNodeMapRefT() [2/3] . . . 246
12.55.3.20 CNodeMapRefT() [3/3] . . . 246
12.55.3.21 EatComments() . . . 246
12.55.3.22 operator<<() . . . 247
12.55.3.23 operator=() [1/2] . . . 247
12.55.3.24 operator=() [2/2] . . . 247
12.55.3.25 operator>>() . . . 247
12.55.3.26 ~CNodeMapRefT() . . . 247
12.56 AutoVector Class . . . 248
12.56.1 Detailed Description . . . 248
12.57 Spinnaker GenApi Interfaces . . . 249
12.57.1 Detailed Description . . . 250
12.57.2 Typedef Documentation . . . 250
12.57.2.1 CallbackHandleType . . . 250
12.57.2.2 NodeList_t . . . 250
12.58 IBase Interface . . . 251
12.58.1 Detailed Description . . . 251
12.58.2 Variable Documentation . . . 251
12.58.2.1 IBase . . . 251
12.59 BooleanNode Class . . . 252
12.59.1 Detailed Description . . . 252
12.59.2 Typedef Documentation . . . 252
12.59.2.1 CBooleanRef . . . 252
12.60 CategoryNode Class . . . 253
12.60.1 Detailed Description . . . 253
12.60.2 Typedef Documentation . . . 253
12.60.2.1 CCategoryRef . . . 253
12.61 ChunkAdapter Class . . . 254
12.61.1 Detailed Description . . . 254
12.62 ChunkAdapterDcam Class . . . 255

12.62.1 Detailed Description . . . 255
12.63 ChunkAdapterGeneric Class . . . 256
12.63.1 Detailed Description . . . 256
12.64 ChunkAdapterGEV Class . . . 257
12.64.1 Detailed Description . . . 257
12.65 ChunkPort Class . . . 258
12.65.1 Detailed Description . . . 258
12.66 CommandNode Class . . . 259
12.66.1 Detailed Description . . . 259
12.66.2 Typedef Documentation . . . 259
12.66.2.1 CCommandRef . . . 259
12.67 Container Class . . . 260
12.68 Counter Class . . . 261
12.68.1 Detailed Description . . . 261
12.69 EnumClasses Class . . . 262
12.69.1 Detailed Description . . . 263
12.70 EnumEntryNode Class . . . 264
12.70.1 Detailed Description . . . 264
12.70.2 Typedef Documentation . . . 264
12.70.2.1 CEnumEntryRef . . . 264
12.71 EnumNode Class . . . 265
12.71.1 Detailed Description . . . 265
12.71.2 Typedef Documentation . . . 265
12.71.2.1 CEnumerationRef . . . 265
12.72 EnumNodeT Class . . . 266
12.72.1 Detailed Description . . . 266
12.73 EventAdapter Class . . . 267
12.73.1 Detailed Description . . . 267
12.74 EventAdapter1394 Class . . . 268
12.74.1 Detailed Description . . . 268

12.75 EventAdapterGeneric Class . . . 269
12.75.1 Detailed Description . . . 269
12.76 EventAdapterGEV Class . . . 270
12.76.1 Detailed Description . . . 270
12.77 EventAdapterU3V Class . . . 271
12.77.1 Detailed Description . . . 271
12.78 EventPort Class . . . 272
12.78.1 Detailed Description . . . 272
12.79 Filestream Class . . . 273
12.79.1 Detailed Description . . . 273
12.80 FloatNode Class . . . 274
12.80.1 Detailed Description . . . 274
12.80.2 Typedef Documentation . . . 274
12.80.2.1 CFloatRef . . . 274
12.81 FloatRegNode Class . . . 275
12.81.1 Detailed Description . . . 275
12.82 GCString Class . . . 276
12.82.1 Detailed Description . . . 276
12.83 GCSynch Class . . . 277
12.83.1 Detailed Description . . . 277
12.84 GCTypes Class . . . 278
12.84.1 Detailed Description . . . 278
12.84.2 Typedef Documentation . . . 278
12.84.2.1 float32_t . . . 278
12.84.2.2 float64_t . . . 278
12.85 Spinnaker GenApi Utilities . . . 279
12.85.1 Detailed Description . . . 279
12.86 GCUtilities Utility . . . 280
12.86.1 Detailed Description . . . 281
12.86.2 Function Documentation . . . 281

12.86.2.1 DoesEnvironmentVariableExist() . . . 281
12.86.2.2 GetFiles() . . . 281
12.86.2.3 GetGenICamCacheFolder() . . . 281
12.86.2.4 GetGenICamCLProtocolFolder() . . . 282
12.86.2.5 GetGenICamLogConfig() . . . 282
12.86.2.6 GetModulePathFromFunction() . . . 282
12.86.2.7 GetValueOfEnvironmentVariable() [1/2] . . . 282
12.86.2.8 GetValueOfEnvironmentVariable() [2/2] . . . 283
12.86.2.9 INTEGRAL_CAST() . . . 283
12.86.2.10 INTEGRAL_CAST2() . . . 283
12.86.2.11 ReplaceEnvironmentVariables() . . . 283
12.86.2.12 SetGenICamCacheFolder() . . . 284
12.86.2.13 SetGenICamCLProtocolFolder() . . . 284
12.86.2.14 SetGenICamLogConfig() . . . 284
12.86.2.15 Tokenize() . . . 284
12.86.2.16 UrlDecode() . . . 284
12.86.2.17 UrlEncode() . . . 285
12.87 IBoolean Interface . . . 286
12.87.1 Detailed Description . . . 286
12.87.2 Function Documentation . . . 286
12.87.2.1 GetValue() . . . 286
12.87.2.2 operator()() . . . 287
12.87.2.3 operator=() . . . 287
12.87.3 Variable Documentation . . . 287
12.87.3.1 IBoolean . . . 287
12.87.3.2 Verify . . . 287
12.88 ICategory Interfaces . . . 288
12.88.1 Detailed Description . . . 288
12.88.2 Variable Documentation . . . 288
12.88.2.1 ICategory . . . 288

12.89 IChunkPort Interface . . . 289
12.89.1 Detailed Description . . . 289
12.89.2 Macro Definition Documentation . . . 289
12.89.2.1 CHUNK_BASE_ADDRESS_REGISTER . . . 289
12.89.2.2 CHUNK_BASE_ADDRESS_REGISTER_LEN . . . 290
12.89.2.3 CHUNK_LENGTH_REGISTER . . . 290
12.89.2.4 CHUNK_LENGTH_REGISTER_LEN . . . 290
12.89.3 Function Documentation . . . 290
12.89.3.1 CacheChunkData() . . . 290
12.89.4 Variable Documentation . . . 290
12.89.4.1 IChunkPort . . . 290
12.90 ICommand Interface . . . 291
12.90.1 Detailed Description . . . 291
12.90.2 Function Documentation . . . 291
12.90.2.1 IsDone() . . . 291
12.90.3 Variable Documentation . . . 291
12.90.3.1 ICommand . . . 292
12.91 IDestroy Interface . . . 293
12.91.1 Detailed Description . . . 293
12.91.2 Variable Documentation . . . 293
12.91.2.1 IDestroy . . . 293
12.92 IDeviceInfo Interface . . . 294
12.92.1 Detailed Description . . . 294
12.92.2 Function Documentation . . . 294
12.92.2.1 GetDeviceVersion() . . . 294
12.92.2.2 GetGenApiVersion() . . . 295
12.92.2.3 GetProductGuid() . . . 295
12.92.2.4 GetSchemaVersion() . . . 295
12.92.2.5 GetStandardNameSpace() . . . 295
12.92.2.6 GetToolTip() . . . 295

12.92.2.7 GetVendorName() . . . 295
12.92.2.8 GetVersionGuid() . . . 296
12.92.3 Variable Documentation . . . 296
12.92.3.1 IDeviceInfo . . . 296
12.93 IEnumEntry Interface . . . 297
12.93.1 Detailed Description . . . 297
12.93.2 Function Documentation . . . 297
12.93.2.1 GetNumericValue() . . . 297
12.93.2.2 GetSymbolic() . . . 297
12.93.2.3 IsSelfClearing() . . . 298
12.93.3 Variable Documentation . . . 298
12.93.3.1 IEnumEntry . . . 298
12.94 IEnumeration Interface . . . 299
12.94.1 Detailed Description . . . 299
12.94.2 Function Documentation . . . 299
12.94.2.1 GetCurrentEntry() . . . 299
12.94.2.2 GetEntries() . . . 300
12.94.2.3 GetEntry() . . . 300
12.94.2.4 GetEntryByName() . . . 300
12.94.2.5 GetIntValue() . . . 300
12.94.2.6 operator∗() . . . 300
12.94.2.7 SetIntValue() . . . 301
12.94.3 Variable Documentation . . . 301
12.94.3.1 IEnumeration . . . 301
12.95 IEnumerationT Interface . . . 302
12.95.1 Detailed Description . . . 302
12.95.2 Function Documentation . . . 302
12.95.2.1 GetEntry() . . . 302
12.95.2.2 operator=() [1/2] . . . 303
12.95.2.3 operator=() [2/2] . . . 303

12.95.3 Variable Documentation . . . 303
12.95.3.1 IEnumerationT . . . 303
12.95.3.2 IEnumReference . . . 303
12.96 IFloat Interface . . . 304
12.96.1 Detailed Description . . . 305
12.96.2 Function Documentation . . . 305
12.96.2.1 GetDisplayNotation() . . . 305
12.96.2.2 GetDisplayPrecision() . . . 305
12.96.2.3 GetInc() . . . 305
12.96.2.4 GetIncMode() . . . 305
12.96.2.5 GetListOfValidValues() . . . 305
12.96.2.6 GetMax() . . . 306
12.96.2.7 GetMin() . . . 306
12.96.2.8 GetRepresentation() . . . 306
12.96.2.9 GetUnit() . . . 306
12.96.2.10 HasInc() . . . 306
12.96.2.11 ImposeMax() . . . 306
12.96.2.12 ImposeMin() . . . 307
12.96.2.13 operator=() . . . 307
12.96.3 Variable Documentation . . . 307
12.96.3.1 IFloat . . . 307
12.97 IInteger Interface . . . 308
12.97.1 Detailed Description . . . 308
12.97.2 Function Documentation . . . 308
12.97.2.1 ImposeMax() . . . 308
12.97.2.2 ImposeMin() . . . 308
12.97.2.3 operator=() . . . 309
12.97.3 Variable Documentation . . . 309
12.97.3.1 IInteger . . . 309
12.98 INode Interface . . . 310

12.98.1 Detailed Description 312
12.98.2 Function Documentation 312
12.98.2.1 Combine() [1/3] 312
12.98.2.2 Combine() [2/3] 312
12.98.2.3 Combine() [3/3] 312
12.98.2.4 DeregisterCallback() 313
12.98.2.5 GetAlias() 313
12.98.2.6 GetCachingMode() 313
12.98.2.7 GetCastAlias() 313
12.98.2.8 GetChildren() 313
12.98.2.9 GetDescription() 314
12.98.2.10 GetDisplayName() 314
12.98.2.11 GetDocuURL() 314
12.98.2.12 GetEventID() 314
12.98.2.13 GetNameSpace() 314
12.98.2.14 GetNodeMap() 314
12.98.2.15 GetParents() 314
12.98.2.16 GetPollingTime() 315
12.98.2.17 GetPrincipalInterfaceType() 315
12.98.2.18 GetProperty() 315
12.98.2.19 GetPropertyNames() 315
12.98.2.20 GetVisibility() 315
12.98.2.21 ImposeAccessMode() 316
12.98.2.22 ImposeVisibility() 316
12.98.2.23 InvalidateNode() 316
12.98.2.24 IsAccessModeCacheable() 316
12.98.2.25 IsAvailable() [1/3] 316
12.98.2.26 IsAvailable() [2/3] 316
12.98.2.27 IsAvailable() [3/3] 317
12.98.2.28 IsCachable() 317

12.98.2.29 IsCacheable() . . . 317
12.98.2.30 IsDeprecated() . . . 317
12.98.2.31 IsFeature() . . . 317
12.98.2.32 IsImplemented() [1/3] . . . 317
12.98.2.33 IsImplemented() [2/3] . . . 318
12.98.2.34 IsImplemented() [3/3] . . . 318
12.98.2.35 IsReadable() [1/3] . . . 318
12.98.2.36 IsReadable() [2/3] . . . 318
12.98.2.37 IsReadable() [3/3] . . . 318
12.98.2.38 IsStreamable() . . . 318
12.98.2.39 IsVisible() . . . 319
12.98.2.40 IsWritable() [1/3] . . . 319
12.98.2.41 IsWritable() [2/3] . . . 319
12.98.2.42 IsWritable() [3/3] . . . 319
12.98.2.43 operator"!=() . . . 319
12.98.2.44 operator==() . . . 319
12.98.2.45 RegisterCallback() . . . 320
12.98.3 Variable Documentation . . . 320
12.98.3.1 INode . . . 320
12.98.3.2 IReference . . . 320
12.99 INodeMap Interface . . . 321
12.99.1 Detailed Description . . . 321
12.99.2 Function Documentation . . . 321
12.99.2.1 Connect() [1/2] . . . 322
12.99.2.2 Connect() [2/2] . . . 322
12.99.2.3 GetDeviceName() . . . 322
12.99.2.4 GetLock() . . . 322
12.99.2.5 GetNode() . . . 322
12.99.2.6 GetNumNodes() . . . 323
12.99.2.7 InvalidateNodes() . . . 323

12.99.2.8 Poll() . . . 323
12.99.3 Variable Documentation . . . 323
12.99.3.1 INodeMap . . . 323
12.100 INodeMapDyn Interface . . . 324
12.100.1 Detailed Description . . . 325
12.100.2 Function Documentation . . . 325
12.100.2.1 ExtractIndependentSubtree() . . . 325
12.100.2.2 GetSupportedSchemaVersions() . . . 325
12.100.2.3 LoadXMLFromFile() . . . 325
12.100.2.4 LoadXMLFromFileInject() . . . 326
12.100.2.5 LoadXMLFromString() . . . 326
12.100.2.6 LoadXMLFromStringInject() . . . 326
12.100.2.7 LoadXMLFromZIPData() . . . 326
12.100.2.8 LoadXMLFromZIPFile() . . . 326
12.100.2.9 MergeXMLFiles() . . . 326
12.100.2.10 PreprocessXMLFromFile() . . . 327
12.100.2.11 PreprocessXMLFromZIPFile() . . . 327
12.100.3 Variable Documentation . . . 328
12.100.3.1 IINodeMapDyn . . . 328
12.101 IntegerNode Class . . . 329
12.101.1 Detailed Description . . . 329
12.101.2 Typedef Documentation . . . 329
12.101.2.1 CIntegerRef . . . 329
12.102 IntRegNode Class . . . 330
12.102.1 Detailed Description . . . 330
12.103 IPort Interface . . . 331
12.103.1 Detailed Description . . . 331
12.103.2 Function Documentation . . . 331
12.103.2.1 Write() . . . 331
12.103.3 Variable Documentation . . . 331

12.103.3.1 Address . . . 332
12.103.3.2 Port . . . 332
12.103.3.3 Length . . . 332
12.104 IPortConstruct Interface . . . 333
12.104.1 Detailed Description . . . 333
12.104.2 Function Documentation . . . 333
12.104.2.1 GetSwapEndianess() . . . 333
12.104.3 Variable Documentation . . . 333
12.104.3.1 IPortConstruct . . . 333
12.105 IPortRecorder Interface . . . 334
12.105.1 Detailed Description . . . 334
12.105.2 Function Documentation . . . 334
12.105.2.1 GetCookie() . . . 334
12.105.2.2 Replay() . . . 335
12.105.2.3 SetCookie() . . . 335
12.105.2.4 StopRecording() . . . 335
12.105.3 Variable Documentation . . . 335
12.105.3.1 Invalidate . . . 335
12.105.3.2 IPortRecorder . . . 335
12.105.3.3 IPortReplay . . . 335
12.105.3.4 IPortWriteList . . . 335
12.106 IRegister Interfaces . . . 336
12.106.1 Detailed Description . . . 336
12.106.2 Function Documentation . . . 336
12.106.2.1 Get() . . . 336
12.106.2.2 GetAddress() . . . 337
12.106.2.3 GetLength() . . . 337
12.106.3 Variable Documentation . . . 337
12.106.3.1 IRegister . . . 337
12.107 ISelector Interface . . . 338

12.107.1 Detailed Description . . . 338
12.107.2 Function Documentation . . . 338
12.107.2.1 GetSelectedFeatures() . . . 338
12.107.2.2 GetSelectingFeatures() . . . 338
12.107.3 Variable Documentation . . . 338
12.107.3.1 IlSelector . . . 338
12.108 ISelectorDigit Interface . . . 339
12.108.1 Detailed Description . . . 339
12.108.2 Function Documentation . . . 339
12.108.2.1 GetSelectorList() . . . 339
12.108.2.2 Restore() . . . 340
12.108.2.3 SetNext() . . . 340
12.108.2.4 ToString() . . . 340
12.108.3 Variable Documentation . . . 340
12.108.3.1 IlSelectorDigit . . . 341
12.109 IString Class . . . 342
12.109.1 Detailed Description . . . 342
12.109.2 Function Documentation . . . 342
12.109.2.1 GetMaxLength() . . . 342
12.109.3 Variable Documentation . . . 342
12.109.3.1 IlString . . . 342
12.110 IValue Class . . . 343
12.110.1 Detailed Description . . . 343
12.110.2 Function Documentation . . . 343
12.110.2.1 FromString() . . . 343
12.110.2.2 IsValueCacheValid() . . . 344
12.110.2.3 ToString() . . . 344
12.110.3 Variable Documentation . . . 344
12.110.3.1 IlValue . . . 344
12.111 INode Class . . . 345

12.111.1 Detailed Description . . . 345
12.112 NodeCallback Class . . . 346
12.112.1 Detailed Description . . . 347
12.112.2 Enumeration Type Documentation . . . 347
12.112.2.1 ECallbackType . . . 347
12.112.3 Function Documentation . . . 347
12.112.3.1 Deregister() . . . 347
12.112.3.2 make_NodeCallback() [1/2] . . . 347
12.112.3.3 make_NodeCallback() [2/2] . . . 348
12.112.3.4 Register() [1/2] . . . 348
12.112.3.5 Register() [2/2] . . . 348
12.113 NodeMap Class . . . 349
12.113.1 Detailed Description . . . 349
12.114 NodeMapFactory Class . . . 350
12.114.1 Detailed Description . . . 350
12.114.2 Enumeration Type Documentation . . . 350
12.114.2.1 ECacheUsage_t . . . 350
12.114.2.2 EContentType_t . . . 351
12.115 NodeMapRef Class . . . 352
12.115.1 Detailed Description . . . 352
12.116 Persistence Class . . . 353
12.116.1 Detailed Description . . . 353
12.117 Pointer Class . . . 354
12.117.1 Detailed Description . . . 355
12.117.2 Typedef Documentation . . . 355
12.117.2.1 CBasePtr . . . 355
12.117.2.2 CBooleanPtr . . . 356
12.117.2.3 CCategoryPtr . . . 356
12.117.2.4 CChunkPortPtr . . . 356
12.117.2.5 CCommandPtr . . . 356

12.117.2.6 CDeviceInfoPtr . . . 356
12.117.2.7 CEnumEntryPtr . . . 356
12.117.2.8 CEnumerationPtr . . . 357
12.117.2.9 CIntegerPtr . . . 357
12.117.2.10 CNodeMapDynPtr . . . 357
12.117.2.11 CNodeMapPtr . . . 357
12.117.2.12 CNodePtr . . . 357
12.117.2.13 CPortConstructPtr . . . 357
12.117.2.14 CPortPtr . . . 358
12.117.2.15 CPortRecorderPtr . . . 358
12.117.2.16 CPortReplayPtr . . . 358
12.117.2.17 CPortWriteListPtr . . . 358
12.117.2.18 CRegisterPtr . . . 358
12.117.2.19 CSelectorPtr . . . 358
12.117.2.20 CStringPtr . . . 359
12.117.2.21 CValuePtr . . . 359
12.117.3 Function Documentation . . . 359
12.117.3.1 GetInterfaceName() . . . 359
12.117.3.2 IsAvailable() . . . 359
12.117.3.3 IsImplemented() . . . 359
12.117.3.4 IsReadable() . . . 359
12.117.3.5 IsWritable() . . . 359
12.118 PortImpl Class . . . 360
12.118.1 Detailed Description . . . 360
12.119 PortNode Class . . . 361
12.119.1 Detailed Description . . . 361
12.119.2 Typedef Documentation . . . 361
12.119.2.1 CPortRef . . . 361
12.120 PortRecorder Class . . . 362
12.120.1 Detailed Description . . . 362

12.120.2 Typedef Documentation . . . 362
12.120.2.1 CPortRecorderRef . . . 362
12.121 PortReplay Class . . . 363
12.121.1 Detailed Description . . . 363
12.122 PortWriteList Class . . . 364
12.122.1 Detailed Description . . . 364
12.123 Reference Interfaces . . . 365
12.123.1 Detailed Description . . . 365
12.123.2 Function Documentation . . . 365
12.123.2.1 SetNumEnums() . . . 365
12.124 RegisterNode Class . . . 366
12.124.1 Detailed Description . . . 366
12.124.2 Typedef Documentation . . . 366
12.124.2.1 CRegisterRef . . . 366
12.125 RegisterPortImpl Class . . . 367
12.125.1 Detailed Description . . . 367
12.126 SelectorSet Class . . . 368
12.126.1 Detailed Description . . . 368
12.127 SpinTestCamera Class . . . 369
12.127.1 Detailed Description . . . 369
12.128 StringNode Class . . . 370
12.128.1 Detailed Description . . . 370
12.128.2 Typedef Documentation . . . 370
12.128.2.1 CStringRef . . . 370
12.129 StringRegNode Class . . . 371
12.129.1 Detailed Description . . . 371
12.130 StructPort Class . . . 372
12.130.1 Detailed Description . . . 372
12.131 Synch Class . . . 373
12.131.1 Detailed Description . . . 373

12.132 Spinnaker GenApi Enums . . . 374
12.132.1 Detailed Description . . . 374
12.133 Types Enums . . . 375
12.133.1 Detailed Description . . . 377
12.133.2 Macro Definition Documentation . . . 377
12.133.2.1 _UndefindedRepresentation . . . 377
12.133.3 Typedef Documentation . . . 377
12.133.3.1 StringList_t . . . 378
12.133.4 Enumeration Type Documentation . . . 378
12.133.4.1 EAccessMode . . . 378
12.133.4.2 ECachingMode . . . 378
12.133.4.3 EDisplayNotation . . . 378
12.133.4.4 EEndianess . . . 380
12.133.4.5 EGenApiSchemaVersion . . . 380
12.133.4.6 EIncMode . . . 380
12.133.4.7 EInputDirection . . . 381
12.133.4.8 EInterfaceType . . . 381
12.133.4.9 ELinkType . . . 381
12.133.4.10 ENameSpace . . . 382
12.133.4.11 ERepresentation . . . 382
12.133.4.12 ESign . . . 382
12.133.4.13 ESlope . . . 383
12.133.4.14 EStandardNameSpace . . . 383
12.133.4.15 EVisibility . . . 383
12.133.4.16 EXMLValidation . . . 384
12.133.4.17 EYesNo . . . 384
12.134 ValueNode Class . . . 385
12.134.1 Detailed Description . . . 385
12.134.2 Typedef Documentation . . . 385
12.134.2.1 CValueRef . . . 385
12.135 ChunkAdapterU3V Class . . . 386
12.135.1 Detailed Description . . . 386

13 Namespace Documentation 387

13.1 AdapterConfig Namespace Reference . . . 387

13.1.1 Enumeration Type Documentation . . . 388

13.1.1.1 AdapterConfigErr . . . 388

13.1.2 Function Documentation . . . 388

13.1.2.1 AutoPopulateAdapterInfo() . . . 388

13.1.2.2 AutoPopulateAdvancedProperties() . . . 388

13.1.2.3 ConfigureAdapter() . . . 389

13.1.2.4 GetAuto10GDesc() . . . 389

13.1.2.5 GetAutoGigabitDesc() . . . 389

13.1.2.6 GetAutoStartIp() . . . 389

13.1.2.7 GetAutoSubnetMask() . . . 389

13.1.2.8 GetAutoSubnetMaskLength() . . . 389

13.1.2.9 GetConfigLogFileName() . . . 389

13.1.2.10 GetEnumerationLogFileName() . . . 390

13.1.2.11 GetMaxIpAddress() . . . 390

13.1.2.12 GetMinIpAddress() . . . 390

13.1.2.13 GetSubnetMaskLength() . . . 390

13.1.2.14 IsOnSameSubnet() . . . 390

13.1.2.15 IsValidIpAddress() . . . 390

13.1.2.16 IsValidSubnetMask() . . . 390

13.1.2.17 PopulateAdapterIpInfo() . . . 391

13.1.2.18 RetrieveAllAdapters() . . . 391

13.1.2.19 ValidateIpAddress() . . . 391

13.2 Conversion Namespace Reference . . . 391

13.2.1 Function Documentation . . . 391

13.2.1.1 NumToCString() [1/3] . . . 391

13.2.1.2 NumToCString() [2/3] . . . 391

13.2.1.3 NumToCString() [3/3] . . . 392

13.3 CpuUtil Namespace Reference . . . 392

13.3.1 Function Documentation . . . 392
13.3.1.1 GetCpuStats() . . . 392
13.3.1.2 StartCpuTracing() . . . 392
13.3.1.3 StopCpuTracing() . . . 392
13.4 PerformanceCounter Namespace Reference . . . 392
13.4.1 Function Documentation . . . 393
13.4.1.1 GetPerformanceCounter() . . . 393
13.4.1.2 StartPerformanceCounter() . . . 393
13.4.2 Variable Documentation . . . 393
13.4.2.1 CounterStart . . . 393
13.4.2.2 PCFreq . . . 393
13.5 SecondsCounter Namespace Reference . . . 393
13.5.1 Function Documentation . . . 394
13.5.1.1 GetSecondsCounter() . . . 394
13.5.1.2 StartSecondsCounter() . . . 394
13.5.2 Variable Documentation . . . 394
13.5.2.1 endTime . . . 394
13.5.2.2 startTime . . . 394
13.5.2.3 timeDiff . . . 394
13.6 Spinnaker Namespace Reference . . . 395
13.7 Spinnaker::GenApi Namespace Reference . . . 434
13.7.1 Typedef Documentation . . . 448
13.7.1.1 IDevFileStream . . . 448
13.7.1.2 ODevFileStream . . . 449
13.7.2 Enumeration Type Documentation . . . 449
13.7.2.1 GVCP_MESSAGE_TAGS . . . 449
13.7.3 Function Documentation . . . 449
13.7.3.1 PersistFeature() . . . 449
13.7.3.2 SET_GUID() . . . 449
13.7.4 Variable Documentation . . . 450
13.7.4.1 COMMAND_MAGIC . . . 450
13.7.4.2 GENCP_COMMAND_HEADER_SIZE . . . 450
13.7.4.3 GENCP_EVENT_BASIC_SIZE . . . 450
13.7.4.4 GENCP_EVENT_CMD_ID . . . 450
13.7.4.5 IPersistScript . . . 450
13.7.4.6 U3V_EVENT_PREFIX . . . 450
13.8 Spinnaker::GenICam Namespace Reference . . . 451
13.8.1 Function Documentation . . . 452
13.8.1.1 getline() [1/2] . . . 452
13.8.1.2 getline() [2/2] . . . 452
13.8.1.3 ThrowBadAlloc() . . . 452
13.9 Spinnaker::Video Namespace Reference . . . 453

14 Class Documentation 455

14.1 ActionCommandResult Struct Reference . . . 455

14.1.1 Detailed Description . . . 455

14.1.2 Member Data Documentation . . . 455

14.1.2.1 DeviceAddress . . . 455

14.1.2.2 Status . . . 455

14.2 AdapterConfigException Class Reference . . . 456

14.2.1 Constructor & Destructor Documentation . . . 456

14.2.1.1 AdapterConfigException() [1/2] . . . 456

14.2.1.2 AdapterConfigException() [2/2] . . . 457

14.2.2 Member Function Documentation . . . 457

14.2.2.1 ErrCode() . . . 457

14.2.2.2 GetParamStr() . . . 457

14.3 AdapterInfo Struct Reference . . . 457

14.3.1 Constructor & Destructor Documentation . . . 458

14.3.1.1 AdapterInfo() . . . 458

14.3.2 Member Data Documentation . . . 458

14.3.2.1 adapterDescription . . . 458

14.3.2.2 adapterGUID . . . 458

14.3.2.3 adapterMACAddress . . . 459

14.3.2.4 adapterName . . . 459

14.3.2.5 dhcpEnabled . . . 459

14.3.2.6 ipInfo . . . 459

14.3.2.7 jumboPackets . . . 459

14.3.2.8 jumboPacketsRegKey . . . 459

14.3.2.9 jumboPacketValidValues . . . 459

14.3.2.10 receiveBuffers . . . 459

14.3.2.11 receiveBuffersMax . . . 460

14.3.2.12 receiveBuffersMin . . . 460

14.3.2.13 receiveBuffersRegKey . . . 460

14.3.2.14 receiveBuffersStep . . . 460
14.3.2.15 transmitBuffers . . . 460
14.3.2.16 transmitBuffersMax . . . 460
14.3.2.17 transmitBuffersMin . . . 460
14.3.2.18 transmitBuffersRegKey . . . 460
14.3.2.19 transmitBuffersStep . . . 461
14.4 AttachStatistics_t Struct Reference . . . 461
14.4.1 Detailed Description . . . 461
14.4.2 Member Data Documentation . . . 461
14.4.2.1 NumAttachedChunks . . . 461
14.4.2.2 NumChunkPorts . . . 461
14.4.2.3 NumChunks . . . 462
14.5 AutoLock Class Reference . . . 462
14.5.1 Constructor & Destructor Documentation . . . 462
14.5.1.1 AutoLock() . . . 462
14.5.1.2 ∼AutoLock() . . . 462
14.6 AutoLock Class Reference . . . 462
14.6.1 Constructor & Destructor Documentation . . . 463
14.6.1.1 AutoLock() . . . 463
14.6.1.2 ∼AutoLock() . . . 463
14.7 AVIOption Struct Reference . . . 463
14.7.1 Detailed Description . . . 463
14.7.2 Constructor & Destructor Documentation . . . 463
14.7.2.1 AVIOption() . . . 464
14.7.3 Member Data Documentation . . . 464
14.7.3.1 frameRate . . . 464
14.7.3.2 reserved . . . 464
14.8 BasePtr< T, B > Class Template Reference . . . 464
14.8.1 Detailed Description . . . 465
14.8.2 Constructor & Destructor Documentation . . . 465

14.8.2.1 BasePtr() [1/2] . . . 465
14.8.2.2 ~BasePtr() . . . 465
14.8.2.3 BasePtr() [2/2] . . . 466
14.8.3 Member Function Documentation . . . 466
14.8.3.1 get() . . . 466
14.8.3.2 IsValid() . . . 466
14.8.3.3 operator bool() . . . 466
14.8.3.4 operator T*() . . . 466
14.8.3.5 operator->() . . . 467
14.8.3.6 operator=() [1/4] . . . 467
14.8.3.7 operator=() [2/4] . . . 467
14.8.3.8 operator=() [3/4] . . . 467
14.8.3.9 operator=() [4/4] . . . 467
14.8.3.10 operator==() [1/4] . . . 467
14.8.3.11 operator==() [2/4] . . . 468
14.8.3.12 operator==() [3/4] . . . 468
14.8.3.13 operator==() [4/4] . . . 468
14.8.4 Member Data Documentation . . . 468
14.8.4.1 m_pT . . . 468
14.9 BMPOption Struct Reference . . . 468
14.9.1 Detailed Description . . . 469
14.9.2 Constructor & Destructor Documentation . . . 469
14.9.2.1 BMPOption() . . . 469
14.9.3 Member Data Documentation . . . 469
14.9.3.1 indexedColor_8bit . . . 469
14.9.3.2 reserved . . . 469
14.10 BooleanNode Class Reference . . . 470
14.10.1 Detailed Description . . . 471
14.10.2 Constructor & Destructor Documentation . . . 471
14.10.2.1 BooleanNode() [1/2] . . . 471

14.10.2.2 BooleanNode() [2/2] . . . 471
14.10.2.3 ~BooleanNode() . . . 471
14.10.3 Member Function Documentation . . . 471
14.10.3.1 GetValue() . . . 471
14.10.3.2 operator=() . . . 472
14.10.3.3 SetReference() . . . 472
14.10.3.4 SetValue() . . . 472
14.11 Camera Class Reference . . . 473
14.11.1 Detailed Description . . . 503
14.11.2 Constructor & Destructor Documentation . . . 503
14.11.2.1 ~Camera() . . . 503
14.11.2.2 Camera() . . . 503
14.11.3 Member Function Documentation . . . 503
14.11.3.1 Init() . . . 503
14.11.4 Member Data Documentation . . . 503
14.11.4.1 AasRoiEnable . . . 503
14.11.4.2 AasRoiHeight . . . 504
14.11.4.3 AasRoiOffsetX . . . 504
14.11.4.4 AasRoiOffsetY . . . 504
14.11.4.5 AasRoiWidth . . . 504
14.11.4.6 AcquisitionAbort . . . 505
14.11.4.7 AcquisitionArm . . . 505
14.11.4.8 AcquisitionBurstFrameCount . . . 505
14.11.4.9 AcquisitionFrameCount . . . 505
14.11.4.10 AcquisitionFrameRate . . . 505
14.11.4.11 AcquisitionFrameRateEnable . . . 506
14.11.4.12 AcquisitionLineRate . . . 506
14.11.4.13 AcquisitionMode . . . 506
14.11.4.14 AcquisitionResultingFrameRate . . . 506
14.11.4.15 AcquisitionStart . . . 506

14.11.4.16AcquisitionStatus 506
14.11.4.17AcquisitionStatusSelector 507
14.11.4.18AcquisitionStop 507
14.11.4.19ActionDeviceKey 507
14.11.4.20ActionGroupKey 507
14.11.4.21ActionGroupMask 507
14.11.4.22ActionQueueSize 507
14.11.4.23ActionSelector 508
14.11.4.24ActionUnconditionalMode 508
14.11.4.25AdaptiveCompressionEnable 508
14.11.4.26AdcBitDepth 508
14.11.4.27aPAUSEMACCtrlFramesReceived 508
14.11.4.28aPAUSEMACCtrlFramesTransmitted 509
14.11.4.29AutoAlgorithmSelector 509
14.11.4.30AutoExposureControlLoopDamping 509
14.11.4.31AutoExposureControlPriority 509
14.11.4.32AutoExposureEVCompensation 510
14.11.4.33AutoExposureExposureTimeLowerLimit 510
14.11.4.34AutoExposureExposureTimeUpperLimit 510
14.11.4.35AutoExposureGainLowerLimit 510
14.11.4.36AutoExposureGainUpperLimit 511
14.11.4.37AutoExposureGreyValueLowerLimit 511
14.11.4.38AutoExposureGreyValueUpperLimit 511
14.11.4.39AutoExposureLightingMode 511
14.11.4.40AutoExposureMeteringMode 512
14.11.4.41AutoExposureTargetGreyValue 512
14.11.4.42AutoExposureTargetGreyValueAuto 512
14.11.4.43BalanceRatio 513
14.11.4.44BalanceRatioSelector 513
14.11.4.45BalanceWhiteAuto 513

14.11.4.46 BalanceWhiteAutoDamping 513
14.11.4.47 BalanceWhiteAutoLowerLimit 514
14.11.4.48 BalanceWhiteAutoProfile 514
14.11.4.49 BalanceWhiteAutoUpperLimit 514
14.11.4.50 BinningHorizontal 514
14.11.4.51 BinningHorizontalMode 515
14.11.4.52 BinningSelector 515
14.11.4.53 BinningVertical 515
14.11.4.54 BinningVerticalMode 515
14.11.4.55 BlackLevel 515
14.11.4.56 BlackLevelAuto 516
14.11.4.57 BlackLevelAutoBalance 516
14.11.4.58 BlackLevelClampingEnable 516
14.11.4.59 BlackLevelRaw 516
14.11.4.60 BlackLevelSelector 516
14.11.4.61 ChunkBlackLevel 517
14.11.4.62 ChunkBlackLevelSelector 517
14.11.4.63 ChunkCounterSelector 517
14.11.4.64 ChunkCounterValue 517
14.11.4.65 ChunkCRC 517
14.11.4.66 ChunkEnable 517
14.11.4.67 ChunkEncoderSelector 518
14.11.4.68 ChunkEncoderStatus 518
14.11.4.69 ChunkEncoderValue 518
14.11.4.70 ChunkExposureEndLineStatusAll 518
14.11.4.71 ChunkExposureTime 518
14.11.4.72 ChunkExposureTimeSelector 518
14.11.4.73 ChunkFrameID 519
14.11.4.74 ChunkGain 519
14.11.4.75 ChunkGainSelector 519

14.11.4.76 ChunkHeight 519
14.11.4.77 ChunkImage 519
14.11.4.78 ChunkImageComponent 519
14.11.4.79 ChunkInferenceBoundingBoxResult 520
14.11.4.80 ChunkInferenceConfidence 520
14.11.4.81 ChunkInferenceFrameId 520
14.11.4.82 ChunkInferenceResult 520
14.11.4.83 ChunkLinePitch 520
14.11.4.84 ChunkLineStatusAll 520
14.11.4.85 ChunkModeActive 521
14.11.4.86 ChunkOffsetX 521
14.11.4.87 ChunkOffsetY 521
14.11.4.88 ChunkPartSelector 521
14.11.4.89 ChunkPixelDynamicRangeMax 521
14.11.4.90 ChunkPixelDynamicRangeMin 521
14.11.4.91 ChunkPixelFormat 522
14.11.4.92 ChunkRegionID 522
14.11.4.93 ChunkScan3dAxisMax 522
14.11.4.94 ChunkScan3dAxisMin 522
14.11.4.95 ChunkScan3dCoordinateOffset 522
14.11.4.96 ChunkScan3dCoordinateReferenceSelector 522
14.11.4.97 ChunkScan3dCoordinateReferenceValue 523
14.11.4.98 ChunkScan3dCoordinateScale 523
14.11.4.99 ChunkScan3dCoordinateSelector 523
14.11.4.100 ChunkScan3dCoordinateSystem 523
14.11.4.101 ChunkScan3dCoordinateSystemReference 523
14.11.4.102 ChunkScan3dCoordinateTransformSelector 523
14.11.4.103 ChunkScan3dDistanceUnit 524
14.11.4.104 ChunkScan3dInvalidDataFlag 524
14.11.4.105 ChunkScan3dInvalidDataValue 524

14.11.4.106 ChunkScan3dOutputMode 524
14.11.4.107 ChunkScan3dTransformValue 524
14.11.4.108 ChunkScanLineSelector 524
14.11.4.109 ChunkSelector 525
14.11.4.110 ChunkSequencerSetActive 525
14.11.4.111 ChunkSerialData 525
14.11.4.112 ChunkSerialDataLength 525
14.11.4.113 ChunkSerialReceiveOverflow 525
14.11.4.114 ChunkSourceID 525
14.11.4.115 ChunkStreamChannelID 526
14.11.4.116 ChunkTimerSelector 526
14.11.4.117 ChunkTimerValue 526
14.11.4.118 ChunkTimestamp 526
14.11.4.119 ChunkTimestampLatchValue 526
14.11.4.120 ChunkTransferBlockID 526
14.11.4.121 ChunkTransferQueueCurrentBlockCount 527
14.11.4.122 ChunkTransferStreamID 527
14.11.4.123 ChunkWidth 527
14.11.4.124 ClConfiguration 527
14.11.4.125 ClTimeSlotsCount 527
14.11.4.126 ColorTransformationEnable 528
14.11.4.127 ColorTransformationSelector 528
14.11.4.128 ColorTransformationValue 528
14.11.4.129 ColorTransformationValueSelector 528
14.11.4.130 CompressionRatio 528
14.11.4.131 CounterDelay 529
14.11.4.132 CounterDuration 529
14.11.4.133 CounterEventActivation 529
14.11.4.134 CounterEventSource 529
14.11.4.135 CounterReset 529

14.11.4.136 CounterResetActivation . . . 529
14.11.4.137 CounterResetSource . . . 530
14.11.4.138 CounterSelector . . . 530
14.11.4.139 CounterStatus . . . 530
14.11.4.140 CounterTriggerActivation . . . 530
14.11.4.141 CounterTriggerSource . . . 530
14.11.4.142 CounterValue . . . 530
14.11.4.143 CounterValueAtReset . . . 531
14.11.4.144 CxpConnectionSelector . . . 531
14.11.4.145 CxpConnectionTestErrorCount . . . 531
14.11.4.146 CxpConnectionTestMode . . . 531
14.11.4.147 CxpConnectionTestPacketCount . . . 531
14.11.4.148 CxpLinkConfiguration . . . 531
14.11.4.149 CxpLinkConfigurationPreferred . . . 532
14.11.4.150 CxpLinkConfigurationStatus . . . 532
14.11.4.151 CxpPoCxpAuto . . . 532
14.11.4.152 CxpPoCxpStatus . . . 532
14.11.4.153 CxpPoCxpTripReset . . . 532
14.11.4.154 CxpPoCxpTurnOff . . . 532
14.11.4.155 DecimationHorizontal . . . 533
14.11.4.156 DecimationHorizontalMode . . . 533
14.11.4.157 DecimationSelector . . . 533
14.11.4.158 DecimationVertical . . . 533
14.11.4.159 DecimationVerticalMode . . . 534
14.11.4.160 DefectCorrectionMode . . . 534
14.11.4.161 DefectCorrectStaticEnable . . . 534
14.11.4.162 DefectTableApply . . . 534
14.11.4.163 DefectTableCoordinateX . . . 534
14.11.4.164 DefectTableCoordinateY . . . 535
14.11.4.165 DefectTableFactoryRestore . . . 535

14.11.4.166 DefectTableIndex 535
14.11.4.167 DefectTablePixelCount 535
14.11.4.168 DefectTableSave 536
14.11.4.169 Deinterlacing 536
14.11.4.170 DeviceCharacterSet 536
14.11.4.171 DeviceClockFrequency 536
14.11.4.172 DeviceClockSelector 536
14.11.4.173 DeviceConnectionSelector 537
14.11.4.174 DeviceConnectionSpeed 537
14.11.4.175 DeviceConnectionStatus 537
14.11.4.176 DeviceEventChannelCount 537
14.11.4.177 DeviceFamilyName 537
14.11.4.178 DeviceFeaturePersistenceEnd 537
14.11.4.179 DeviceFeaturePersistenceStart 538
14.11.4.180 DeviceFirmwareVersion 538
14.11.4.181 DeviceGenCPVersionMajor 538
14.11.4.182 DeviceGenCPVersionMinor 538
14.11.4.183 DeviceID 538
14.11.4.184 DeviceIndicatorMode 538
14.11.4.185 DeviceLinkBandwidthReserve 539
14.11.4.186 DeviceLinkCommandTimeout 539
14.11.4.187 DeviceLinkConnectionCount 539
14.11.4.188 DeviceLinkCurrentThroughput 539
14.11.4.189 DeviceLinkHeartbeatMode 539
14.11.4.190 DeviceLinkHeartbeatTimeout 539
14.11.4.191 DeviceLinkSelector 540
14.11.4.192 DeviceLinkSpeed 540
14.11.4.193 DeviceLinkThroughputLimit 540
14.11.4.194 DeviceLinkThroughputLimitMode 540
14.11.4.195 DeviceManifestEntrySelector 541

14.11.4.196 DeviceManifestPrimaryURL 541
14.11.4.197 DeviceManifestSchemaMajorVersion 541
14.11.4.198 DeviceManifestSchemaMinorVersion 541
14.11.4.199 DeviceManifestSecondaryURL 541
14.11.4.200 DeviceManifestXMLMajorVersion 541
14.11.4.201 DeviceManifestXMLMinorVersion 542
14.11.4.202 DeviceManifestXMLSubMinorVersion 542
14.11.4.203 DeviceManufacturerInfo 542
14.11.4.204 DeviceMaxThroughput 542
14.11.4.205 DeviceModelName 542
14.11.4.206 DevicePowerSupplySelector 543
14.11.4.207 DeviceRegistersCheck 543
14.11.4.208 DeviceRegistersEndianness 543
14.11.4.209 DeviceRegistersStreamingEnd 543
14.11.4.210 DeviceRegistersStreamingStart 543
14.11.4.211 DeviceRegistersValid 543
14.11.4.212 DeviceReset 544
14.11.4.213 DeviceScanType 544
14.11.4.214 DeviceSerialNumber 544
14.11.4.215 DeviceSerialPortBaudRate 544
14.11.4.216 DeviceSerialPortSelector 544
14.11.4.217 DeviceSFNCVersionMajor 545
14.11.4.218 DeviceSFNCVersionMinor 545
14.11.4.219 DeviceSFNCVersionSubMinor 545
14.11.4.220 DeviceStreamChannelCount 545
14.11.4.221 DeviceStreamChannelEndianness 545
14.11.4.222 DeviceStreamChannelLink 546
14.11.4.223 DeviceStreamChannelPacketSize 546
14.11.4.224 DeviceStreamChannelSelector 546
14.11.4.225 DeviceStreamChannelType 546

14.11.4.226 DeviceTapGeometry 546
14.11.4.227 DeviceTemperature 546
14.11.4.228 DeviceTemperatureSelector 547
14.11.4.229 DeviceTLType 547
14.11.4.230 DeviceTLVersionMajor 547
14.11.4.231 DeviceTLVersionMinor 547
14.11.4.232 DeviceTLVersionSubMinor 547
14.11.4.233 DeviceType 548
14.11.4.234 DeviceUptime 548
14.11.4.235 DeviceUserID 548
14.11.4.236 DeviceVendorName 548
14.11.4.237 DeviceVersion 548
14.11.4.238 EncoderDivider 548
14.11.4.239 EncoderMode 549
14.11.4.240 EncoderOutputMode 549
14.11.4.241 EncoderReset 549
14.11.4.242 EncoderResetActivation 549
14.11.4.243 EncoderResetSource 549
14.11.4.244 EncoderSelector 549
14.11.4.245 EncoderSourceA 550
14.11.4.246 EncoderSourceB 550
14.11.4.247 EncoderStatus 550
14.11.4.248 EncoderTimeout 550
14.11.4.249 EncoderValue 550
14.11.4.250 EncoderValueAtReset 550
14.11.4.251 EnumerationCount 551
14.11.4.252 EventAcquisitionEnd 551
14.11.4.253 EventAcquisitionEndFrameID 551
14.11.4.254 EventAcquisitionEndTimestamp 551
14.11.4.255 EventAcquisitionError 551

14.11.4.256 EventAcquisitionErrorFrameID 551
14.11.4.257 EventAcquisitionErrorTimestamp 552
14.11.4.258 EventAcquisitionStart 552
14.11.4.259 EventAcquisitionStartFrameID 552
14.11.4.260 EventAcquisitionStartTimestamp 552
14.11.4.261 EventAcquisitionTransferEnd 552
14.11.4.262 EventAcquisitionTransferEndFrameID 552
14.11.4.263 EventAcquisitionTransferEndTimestamp 553
14.11.4.264 EventAcquisitionTransferStart 553
14.11.4.265 EventAcquisitionTransferStartFrameID 553
14.11.4.266 EventAcquisitionTransferStartTimestamp 553
14.11.4.267 EventAcquisitionTrigger 553
14.11.4.268 EventAcquisitionTriggerFrameID 553
14.11.4.269 EventAcquisitionTriggerTimestamp 554
14.11.4.270 EventActionLate 554
14.11.4.271 EventActionLateFrameID 554
14.11.4.272 EventActionLateTimestamp 554
14.11.4.273 EventCounter0End 554
14.11.4.274 EventCounter0EndFrameID 554
14.11.4.275 EventCounter0EndTimestamp 555
14.11.4.276 EventCounter0Start 555
14.11.4.277 EventCounter0StartFrameID 555
14.11.4.278 EventCounter0StartTimestamp 555
14.11.4.279 EventCounter1End 555
14.11.4.280 EventCounter1EndFrameID 555
14.11.4.281 EventCounter1EndTimestamp 556
14.11.4.282 EventCounter1Start 556
14.11.4.283 EventCounter1StartFrameID 556
14.11.4.284 EventCounter1StartTimestamp 556
14.11.4.285 EventEncoder0Restarted 556

14.11.4.286 EventEncoder0RestartedFrameID 556
14.11.4.287 EventEncoder0RestartedTimestamp 557
14.11.4.288 EventEncoder0Stopped 557
14.11.4.289 EventEncoder0StoppedFrameID 557
14.11.4.290 EventEncoder0StoppedTimestamp 557
14.11.4.291 EventEncoder1Restarted 557
14.11.4.292 EventEncoder1RestartedFrameID 557
14.11.4.293 EventEncoder1RestartedTimestamp 558
14.11.4.294 EventEncoder1Stopped 558
14.11.4.295 EventEncoder1StoppedFrameID 558
14.11.4.296 EventEncoder1StoppedTimestamp 558
14.11.4.297 EventError 558
14.11.4.298 EventErrorCode 558
14.11.4.299 EventErrorFrameID 559
14.11.4.300 EventErrorTimestamp 559
14.11.4.301 EventExposureEnd 559
14.11.4.302 EventExposureEndFrameID 559
14.11.4.303 EventExposureEndTimestamp 559
14.11.4.304 EventExposureStart 559
14.11.4.305 EventExposureStartFrameID 560
14.11.4.306 EventExposureStartTimestamp 560
14.11.4.307 EventFrameBurstEnd 560
14.11.4.308 EventFrameBurstEndFrameID 560
14.11.4.309 EventFrameBurstEndTimestamp 560
14.11.4.310 EventFrameBurstStart 560
14.11.4.311 EventFrameBurstStartFrameID 561
14.11.4.312 EventFrameBurstStartTimestamp 561
14.11.4.313 EventFrameEnd 561
14.11.4.314 EventFrameEndFrameID 561
14.11.4.315 EventFrameEndTimestamp 561

14.11.4.316 EventFrameStart 561
14.11.4.317 EventFrameStartFrameID 562
14.11.4.318 EventFrameStartTimestamp 562
14.11.4.319 EventFrameTransferEnd 562
14.11.4.320 EventFrameTransferEndFrameID 562
14.11.4.321 EventFrameTransferEndTimestamp 562
14.11.4.322 EventFrameTransferStart 562
14.11.4.323 EventFrameTransferStartFrameID 563
14.11.4.324 EventFrameTransferStartTimestamp 563
14.11.4.325 EventFrameTrigger 563
14.11.4.326 EventFrameTriggerFrameID 563
14.11.4.327 EventFrameTriggerTimestamp 563
14.11.4.328 EventLine0AnyEdge 563
14.11.4.329 EventLine0AnyEdgeFrameID 564
14.11.4.330 EventLine0AnyEdgeTimestamp 564
14.11.4.331 EventLine0FallingEdge 564
14.11.4.332 EventLine0FallingEdgeFrameID 564
14.11.4.333 EventLine0FallingEdgeTimestamp 564
14.11.4.334 EventLine0RisingEdge 564
14.11.4.335 EventLine0RisingEdgeFrameID 565
14.11.4.336 EventLine0RisingEdgeTimestamp 565
14.11.4.337 EventLine1AnyEdge 565
14.11.4.338 EventLine1AnyEdgeFrameID 565
14.11.4.339 EventLine1AnyEdgeTimestamp 565
14.11.4.340 EventLine1FallingEdge 565
14.11.4.341 EventLine1FallingEdgeFrameID 566
14.11.4.342 EventLine1FallingEdgeTimestamp 566
14.11.4.343 EventLine1RisingEdge 566
14.11.4.344 EventLine1RisingEdgeFrameID 566
14.11.4.345 EventLine1RisingEdgeTimestamp 566

14.11.4.346 EventLinkSpeedChange 566
14.11.4.347 EventLinkSpeedChangeFrameID 567
14.11.4.348 EventLinkSpeedChangeTimestamp 567
14.11.4.349 EventLinkTrigger0 567
14.11.4.350 EventLinkTrigger0FrameID 567
14.11.4.351 EventLinkTrigger0Timestamp 567
14.11.4.352 EventLinkTrigger1 567
14.11.4.353 EventLinkTrigger1FrameID 568
14.11.4.354 EventLinkTrigger1Timestamp 568
14.11.4.355 EventNotification 568
14.11.4.356 EventSelector 568
14.11.4.357 EventSequencerSetChange 568
14.11.4.358 EventSequencerSetChangeFrameID 568
14.11.4.359 EventSequencerSetChangeTimestamp 569
14.11.4.360 EventSerialData 569
14.11.4.361 EventSerialDataLength 569
14.11.4.362 EventSerialPortReceive 569
14.11.4.363 EventSerialPortReceiveTimestamp 569
14.11.4.364 EventSerialReceiveOverflow 569
14.11.4.365 EventStream0TransferBlockEnd 570
14.11.4.366 EventStream0TransferBlockEndFrameID 570
14.11.4.367 EventStream0TransferBlockEndTimestamp 570
14.11.4.368 EventStream0TransferBlockStart 570
14.11.4.369 EventStream0TransferBlockStartFrameID 570
14.11.4.370 EventStream0TransferBlockStartTimestamp 570
14.11.4.371 EventStream0TransferBlockTrigger 571
14.11.4.372 EventStream0TransferBlockTriggerFrameID 571
14.11.4.373 EventStream0TransferBlockTriggerTimestamp 571
14.11.4.374 EventStream0TransferBurstEnd 571
14.11.4.375 EventStream0TransferBurstEndFrameID 571

14.11.4.376 EventStream0TransferBurstEndTimestamp 571
14.11.4.377 EventStream0TransferBurstStart 572
14.11.4.378 EventStream0TransferBurstStartFrameID 572
14.11.4.379 EventStream0TransferBurstStartTimestamp 572
14.11.4.380 EventStream0TransferEnd 572
14.11.4.381 EventStream0TransferEndFrameID 572
14.11.4.382 EventStream0TransferEndTimestamp 572
14.11.4.383 EventStream0TransferOverflow 573
14.11.4.384 EventStream0TransferOverflowFrameID 573
14.11.4.385 EventStream0TransferOverflowTimestamp 573
14.11.4.386 EventStream0TransferPause 573
14.11.4.387 EventStream0TransferPauseFrameID 573
14.11.4.388 EventStream0TransferPauseTimestamp 573
14.11.4.389 EventStream0TransferResume 574
14.11.4.390 EventStream0TransferResumeFrameID 574
14.11.4.391 EventStream0TransferResumeTimestamp 574
14.11.4.392 EventStream0TransferStart 574
14.11.4.393 EventStream0TransferStartFrameID 574
14.11.4.394 EventStream0TransferStartTimestamp 574
14.11.4.395 EventTest 575
14.11.4.396 EventTestTimestamp 575
14.11.4.397 EventTimer0End 575
14.11.4.398 EventTimer0EndFrameID 575
14.11.4.399 EventTimer0EndTimestamp 575
14.11.4.400 EventTimer0Start 575
14.11.4.401 EventTimer0StartFrameID 576
14.11.4.402 EventTimer0StartTimestamp 576
14.11.4.403 EventTimer1End 576
14.11.4.404 EventTimer1EndFrameID 576
14.11.4.405 EventTimer1EndTimestamp 576

14.11.4.406 EventTimer1Start . . . 576
14.11.4.407 EventTimer1StartFrameID . . . 577
14.11.4.408 EventTimer1StartTimestamp . . . 577
14.11.4.409 ExposureActiveMode . . . 577
14.11.4.410 ExposureAuto . . . 577
14.11.4.411 ExposureMode . . . 577
14.11.4.412 ExposureTime . . . 577
14.11.4.413 ExposureTimeMode . . . 578
14.11.4.414 ExposureTimeSelector . . . 578
14.11.4.415 FactoryReset . . . 578
14.11.4.416 FileAccessBuffer . . . 578
14.11.4.417 FileAccessLength . . . 578
14.11.4.418 FileAccessOffset . . . 578
14.11.4.419 FileOpenMode . . . 579
14.11.4.420 FileOperationExecute . . . 579
14.11.4.421 FileOperationResult . . . 579
14.11.4.422 FileOperationSelector . . . 579
14.11.4.423 FileOperationStatus . . . 579
14.11.4.424 FileSelector . . . 580
14.11.4.425 FileSize . . . 580
14.11.4.426 Gain . . . 580
14.11.4.427 GainAuto . . . 580
14.11.4.428 GainAutoBalance . . . 581
14.11.4.429 GainSelector . . . 581
14.11.4.430 Gamma . . . 581
14.11.4.431 GammaEnable . . . 581
14.11.4.432 GevActiveLinkCount . . . 581
14.11.4.433 GevCCP . . . 581
14.11.4.434 GevCurrentDefaultGateway . . . 582
14.11.4.435 GevCurrentIPAddress . . . 582

14.11.4.436 GevCurrentIPConfigurationDHCP 582
14.11.4.437 GevCurrentIPConfigurationLLA 582
14.11.4.438 GevCurrentIPConfigurationPersistentIP 582
14.11.4.439 GevCurrentPhysicalLinkConfiguration 582
14.11.4.440 GevCurrentSubnetMask 583
14.11.4.441 GevDiscoveryAckDelay 583
14.11.4.442 GevFirstURL 583
14.11.4.443 GevGVCPExtendedStatusCodes 583
14.11.4.444 GevGVCPExtendedStatusCodesSelector 583
14.11.4.445 GevGVCPHeartbeatDisable 583
14.11.4.446 GevGVCPPendingAck 584
14.11.4.447 GevGVCPPendingTimeout 584
14.11.4.448 GevGVSPExtendedIDMode 584
14.11.4.449 GevHeartbeatTimeout 584
14.11.4.450 GevIEEE1588 584
14.11.4.451 GevIEEE1588ClockAccuracy 584
14.11.4.452 GevIEEE1588Mode 585
14.11.4.453 GevIEEE1588Status 585
14.11.4.454 GevInterfaceSelector 585
14.11.4.455 GevIPConfigurationStatus 585
14.11.4.456 GevMACAddress 585
14.11.4.457 GevMCDA 585
14.11.4.458 GevMCPHostPort 586
14.11.4.459 GevMCRC 586
14.11.4.460 GevMCSP 586
14.11.4.461 GevMCTT 586
14.11.4.462 GevNumberOfInterfaces 586
14.11.4.463 GevPAUSEFrameReception 586
14.11.4.464 GevPAUSEFrameTransmission 587
14.11.4.465 GevPersistentDefaultGateway 587

14.11.4.466 GevPersistentIPAddress 587
14.11.4.467 GevPersistentSubnetMask 587
14.11.4.468 GevPhysicalLinkConfiguration 587
14.11.4.469 GevPrimaryApplicationIPAddress 587
14.11.4.470 GevPrimaryApplicationSocket 588
14.11.4.471 GevPrimaryApplicationSwitchoverKey 588
14.11.4.472 GevSCCFGAllInTransmission 588
14.11.4.473 GevSCCFGExtendedChunkData 588
14.11.4.474 GevSCCFGPacketResendDestination 588
14.11.4.475 GevSCCFGUnconditionalStreaming 588
14.11.4.476 GevSCDA 589
14.11.4.477 GevSCPD 589
14.11.4.478 GevSCPDirection 589
14.11.4.479 GevSCPHostPort 589
14.11.4.480 GevSCPInterfaceIndex 589
14.11.4.481 GevSCPSBigEndian 589
14.11.4.482 GevSCPSDoNotFragment 590
14.11.4.483 GevSCPSFireTestPacket 590
14.11.4.484 GevSCPSPacketSize 590
14.11.4.485 GevSCSP 590
14.11.4.486 GevSCZoneConfigurationLock 590
14.11.4.487 GevSCZoneCount 590
14.11.4.488 GevSCZoneDirectionAll 591
14.11.4.489 GevSecondURL 591
14.11.4.490 GevStreamChannelSelector 591
14.11.4.491 GevSupportedOption 591
14.11.4.492 GevSupportedOptionSelector 591
14.11.4.493 GevTimestampTickFrequency 591
14.11.4.494 GuiXmlManifestAddress 592
14.11.4.495 Height 592

14.11.4.496 HeightMax . . . 592
14.11.4.497 ImageComponentEnable . . . 592
14.11.4.498 ImageComponentSelector . . . 592
14.11.4.499 ImageCompressionBitrate . . . 592
14.11.4.500 ImageCompressionJPEGFormatOption . . . 593
14.11.4.501 ImageCompressionMode . . . 593
14.11.4.502 ImageCompressionQuality . . . 593
14.11.4.503 ImageCompressionRateOption . . . 593
14.11.4.504 IspEnable . . . 593
14.11.4.505 LineFilterWidth . . . 594
14.11.4.506 LineFormat . . . 594
14.11.4.507 LineInputFilterSelector . . . 594
14.11.4.508 LineInverter . . . 594
14.11.4.509 LineMode . . . 594
14.11.4.510 LinePitch . . . 594
14.11.4.511 LineSelector . . . 595
14.11.4.512 LineSource . . . 595
14.11.4.513 LineStatus . . . 595
14.11.4.514 LineStatusAll . . . 595
14.11.4.515 LinkErrorCount . . . 595
14.11.4.516 LinkUptime . . . 595
14.11.4.517 LogicBlockLUTInputActivation . . . 596
14.11.4.518 LogicBlockLUTInputSelector . . . 596
14.11.4.519 LogicBlockLUTInputSource . . . 596
14.11.4.520 LogicBlockLUTOutputValue . . . 596
14.11.4.521 LogicBlockLUTOutputValueAll . . . 596
14.11.4.522 LogicBlockLUTRowIndex . . . 596
14.11.4.523 LogicBlockLUTSelector . . . 597
14.11.4.524 LogicBlockSelector . . . 597
14.11.4.525 LUTEnable . . . 597

14.11.4.526 LUTIndex . . . 597
14.11.4.527 LUTSelector . . . 597
14.11.4.528 LUTValue . . . 598
14.11.4.529 LUTValueAll . . . 598
14.11.4.530 MaxDeviceResetTime . . . 598
14.11.4.531 OffsetX . . . 598
14.11.4.532 OffsetY . . . 598
14.11.4.533 PacketResendRequestCount . . . 599
14.11.4.534 PayloadSize . . . 599
14.11.4.535 PixelColorFilter . . . 599
14.11.4.536 PixelDynamicRangeMax . . . 599
14.11.4.537 PixelDynamicRangeMin . . . 599
14.11.4.538 PixelFormat . . . 600
14.11.4.539 PixelFormatInfoID . . . 600
14.11.4.540 PixelFormatInfoSelector . . . 600
14.11.4.541 PixelSize . . . 600
14.11.4.542 PowerSupplyCurrent . . . 600
14.11.4.543 PowerSupplyVoltage . . . 600
14.11.4.544 RegionDestination . . . 601
14.11.4.545 RegionMode . . . 601
14.11.4.546 RegionSelector . . . 601
14.11.4.547 ReverseX . . . 601
14.11.4.548 ReverseY . . . 601
14.11.4.549 RgbTransformLightSource . . . 602
14.11.4.550 Saturation . . . 602
14.11.4.551 SaturationEnable . . . 602
14.11.4.552 Scan3dAxisMax . . . 602
14.11.4.553 Scan3dAxisMin . . . 602
14.11.4.554 Scan3dCoordinateOffset . . . 603
14.11.4.555 Scan3dCoordinateReferenceSelector . . . 603

14.11.4.556 Scan3dCoordinateReferenceValue 603
14.11.4.557 Scan3dCoordinateScale 603
14.11.4.558 Scan3dCoordinateSelector 603
14.11.4.559 Scan3dCoordinateSystem 603
14.11.4.560 Scan3dCoordinateSystemReference 604
14.11.4.561 Scan3dCoordinateTransformSelector 604
14.11.4.562 Scan3dDistanceUnit 604
14.11.4.563 Scan3dInvalidDataFlag 604
14.11.4.564 Scan3dInvalidDataValue 604
14.11.4.565 Scan3dOutputMode 604
14.11.4.566 Scan3dTransformValue 605
14.11.4.567 SensorDescription 605
14.11.4.568 SensorDigitizationTaps 605
14.11.4.569 SensorHeight 605
14.11.4.570 SensorShutterMode 605
14.11.4.571 SensorTaps 605
14.11.4.572 SensorWidth 606
14.11.4.573 SequencerConfigurationMode 606
14.11.4.574 SequencerConfigurationValid 606
14.11.4.575 SequencerFeatureEnable 606
14.11.4.576 SequencerMode 606
14.11.4.577 SequencerPathSelector 607
14.11.4.578 SequencerSetActive 607
14.11.4.579 SequencerSetLoad 607
14.11.4.580 SequencerSetNext 607
14.11.4.581 SequencerSetSave 607
14.11.4.582 SequencerSetSelector 608
14.11.4.583 SequencerSetStart 608
14.11.4.584 SequencerSetValid 608
14.11.4.585 SequencerTriggerActivation 608

14.11.4.586 SequencerTriggerSource 608
14.11.4.587 SerialPortBaudRate 609
14.11.4.588 SerialPortDataBits 609
14.11.4.589 SerialPortParity 609
14.11.4.590 SerialPortSelector 609
14.11.4.591 SerialPortSource 609
14.11.4.592 SerialPortStopBits 609
14.11.4.593 SerialReceiveFramingErrorCount 610
14.11.4.594 SerialReceiveParityErrorCount 610
14.11.4.595 SerialReceiveQueueClear 610
14.11.4.596 SerialReceiveQueueCurrentCharacterCount 610
14.11.4.597 SerialReceiveQueueMaxCharacterCount 610
14.11.4.598 SerialTransmitQueueCurrentCharacterCount 610
14.11.4.599 SerialTransmitQueueMaxCharacterCount 611
14.11.4.600 Sharpening 611
14.11.4.601 SharpeningAuto 611
14.11.4.602 SharpeningEnable 611
14.11.4.603 SharpeningThreshold 612
14.11.4.604 SoftwareSignalPulse 612
14.11.4.605 SoftwareSignalSelector 612
14.11.4.606 SourceCount 612
14.11.4.607 SourceSelector 612
14.11.4.608 Test0001 613
14.11.4.609 TestEventGenerate 613
14.11.4.610 TestPattern 613
14.11.4.611 TestPatternGeneratorSelector 613
14.11.4.612 TestPendingAck 613
14.11.4.613 TimerDelay 614
14.11.4.614 TimerDuration 614
14.11.4.615 TimerReset 614

14.11.4.616 TimerSelector 614
14.11.4.617 TimerStatus 614
14.11.4.618 TimerTriggerActivation 614
14.11.4.619 TimerTriggerSource 615
14.11.4.620 TimerValue 615
14.11.4.621 Timestamp 615
14.11.4.622 TimestampLatch 615
14.11.4.623 TimestampLatchValue 615
14.11.4.624 TimestampReset 615
14.11.4.625 TLParamsLocked 616
14.11.4.626 TransferAbort 616
14.11.4.627 TransferBlockCount 616
14.11.4.628 TransferBurstCount 616
14.11.4.629 TransferComponentSelector 616
14.11.4.630 TransferControlMode 616
14.11.4.631 TransferOperationMode 617
14.11.4.632 TransferPause 617
14.11.4.633 TransferQueueCurrentBlockCount 617
14.11.4.634 TransferQueueMaxBlockCount 617
14.11.4.635 TransferQueueMode 617
14.11.4.636 TransferQueueOverflowCount 617
14.11.4.637 TransferResume 618
14.11.4.638 TransferSelector 618
14.11.4.639 TransferStart 618
14.11.4.640 TransferStatus 618
14.11.4.641 TransferStatusSelector 618
14.11.4.642 TransferStop 618
14.11.4.643 TransferStreamChannel 619
14.11.4.644 TransferTriggerActivation 619
14.11.4.645 TransferTriggerMode 619

14.11.4.646 TransferTriggerSelector . . . 619
14.11.4.647 TransferTriggerSource . . . 619
14.11.4.648 TriggerActivation . . . 619
14.11.4.649 TriggerDelay . . . 620
14.11.4.650 TriggerDivider . . . 620
14.11.4.651 TriggerEventTest . . . 620
14.11.4.652 TriggerMode . . . 620
14.11.4.653 TriggerMultiplier . . . 620
14.11.4.654 TriggerOverlap . . . 621
14.11.4.655 TriggerSelector . . . 621
14.11.4.656 TriggerSoftware . . . 621
14.11.4.657 TriggerSource . . . 621
14.11.4.658 UserOutputSelector . . . 621
14.11.4.659 UserOutputValue . . . 622
14.11.4.660 UserOutputValueAll . . . 622
14.11.4.661 UserOutputValueAllMask . . . 622
14.11.4.662 UserSetDefault . . . 622
14.11.4.663 UserSetFeatureEnable . . . 622
14.11.4.664 UserSetLoad . . . 623
14.11.4.665 UserSetSave . . . 623
14.11.4.666 UserSetSelector . . . 623
14.11.4.667 U3_3Enable . . . 623
14.11.4.668 WhiteClip . . . 623
14.11.4.669 WhiteClipSelector . . . 624
14.11.4.670 Width . . . 624
14.11.4.671 WidthMax . . . 624
14.12 CameraBase Class Reference . . . 625
14.12.1 Detailed Description . . . 627
14.12.2 Constructor & Destructor Documentation . . . 627
14.12.2.1 ~CameraBase() . . . 627

14.12.2.2 CameraBase() [1/2] 628
14.12.2.3 CameraBase() [2/2] 628
14.12.3 Member Function Documentation 628
14.12.3.1 BeginAcquisition() 628
14.12.3.2 DeInit() 628
14.12.3.3 DiscoverMaxPacketSize() 629
14.12.3.4 EndAcquisition() 629
14.12.3.5 ForceIP() 629
14.12.3.6 GetAccessMode() 630
14.12.3.7 GetBufferOwnership() 630
14.12.3.8 GetGuiXml() 630
14.12.3.9 GetNextImage() 631
14.12.3.10 GetNodeMap() 631
14.12.3.11 GetNumDataStreams() 632
14.12.3.12 GetNumImagesInUse() 632
14.12.3.13 GetTLDeviceNodeMap() 632
14.12.3.14 GetTLStreamNodeMap() 633
14.12.3.15 GetUniqueID() 633
14.12.3.16 GetUserBufferCount() 633
14.12.3.17 GetUserBufferSize() 634
14.12.3.18 GetUserBufferTotalSize() 634
14.12.3.19 Init() 635
14.12.3.20 IsInitialized() 635
14.12.3.21 IsStreaming() 635
14.12.3.22 IsValid() 636
14.12.3.23 operator=() 636
14.12.3.24 ReadPort() 636
14.12.3.25 RegisterEventHandler() [1/2] 636
14.12.3.26 RegisterEventHandler() [2/2] 637
14.12.3.27 SetBufferOwnership() 637

14.12.3.28 SetUserBuffers() [1/2] . . . 638
14.12.3.29 SetUserBuffers() [2/2] . . . 638
14.12.3.30 UnregisterEventHandler() . . . 639
14.12.3.31 WritePort() . . . 639
14.12.4 Friends And Related Function Documentation . . . 639
14.12.4.1 InterfaceImpl . . . 639
14.13 CameraList Class Reference . . . 640
14.13.1 Detailed Description . . . 641
14.13.2 Constructor & Destructor Documentation . . . 641
14.13.2.1 CameraList() [1/2] . . . 641
14.13.2.2 ~CameraList() . . . 641
14.13.2.3 CameraList() [2/2] . . . 642
14.13.3 Member Function Documentation . . . 642
14.13.3.1 Append() . . . 642
14.13.3.2 Clear() . . . 642
14.13.3.3 GetByDeviceID() . . . 642
14.13.3.4 GetByIndex() . . . 643
14.13.3.5 GetBySerial() . . . 643
14.13.3.6 GetSize() . . . 644
14.13.3.7 operator=() . . . 644
14.13.3.8 operator[]() . . . 644
14.13.3.9 RemoveByDeviceID() . . . 644
14.13.3.10 RemoveByIndex() . . . 645
14.13.3.11 RemoveBySerial() . . . 645
14.14 CameraPtr Class Reference . . . 645
14.14.1 Detailed Description . . . 646
14.15 CategoryNode Class Reference . . . 647
14.15.1 Detailed Description . . . 648
14.15.2 Constructor & Destructor Documentation . . . 648
14.15.2.1 CategoryNode() [1/2] . . . 648

14.15.2.2 CategoryNode() [2/2] . . . 648
14.15.2.3 ~CategoryNode() . . . 648
14.15.3 Member Function Documentation . . . 648
14.15.3.1 GetFeatures() . . . 649
14.15.3.2 SetReference() . . . 649
14.16 CChunkAdapter Class Reference . . . 649
14.16.1 Detailed Description . . . 650
14.16.2 Constructor & Destructor Documentation . . . 650
14.16.2.1 ~CChunkAdapter() . . . 650
14.16.2.2 CChunkAdapter() . . . 650
14.16.3 Member Function Documentation . . . 650
14.16.3.1 AttachBuffer() . . . 650
14.16.3.2 AttachNodeMap() . . . 651
14.16.3.3 CheckBufferLayout() . . . 651
14.16.3.4 ClearCaches() . . . 651
14.16.3.5 DetachBuffer() . . . 651
14.16.3.6 DetachNodeMap() . . . 651
14.16.3.7 UpdateBuffer() . . . 651
14.16.4 Member Data Documentation . . . 652
14.16.4.1 m_pChunkAdapter . . . 652
14.17 CChunkAdapterDcam Class Reference . . . 652
14.17.1 Detailed Description . . . 653
14.17.2 Constructor & Destructor Documentation . . . 653
14.17.2.1 CChunkAdapterDcam() . . . 653
14.17.2.2 ~CChunkAdapterDcam() . . . 653
14.17.3 Member Function Documentation . . . 653
14.17.3.1 AttachBuffer() . . . 654
14.17.3.2 CheckBufferLayout() . . . 654
14.17.3.3 CheckCRC() . . . 654
14.17.3.4 HasCRC() . . . 654

14.18 CChunkAdapterGeneric Class Reference . . . 655
14.18.1 Constructor & Destructor Documentation . . . 655
14.18.1.1 CChunkAdapterGeneric() . . . 656
14.18.1.2 ~CChunkAdapterGeneric() . . . 656
14.18.2 Member Function Documentation . . . 656
14.18.2.1 AttachBuffer() [1/3] . . . 656
14.18.2.2 AttachBuffer() [2/3] . . . 656
14.18.2.3 AttachBuffer() [3/3] . . . 656
14.18.2.4 CheckBufferLayout() . . . 657
14.19 CChunkAdapterGEV Class Reference . . . 657
14.19.1 Detailed Description . . . 658
14.19.2 Constructor & Destructor Documentation . . . 658
14.19.2.1 CChunkAdapterGEV() . . . 658
14.19.2.2 ~CChunkAdapterGEV() . . . 658
14.19.3 Member Function Documentation . . . 658
14.19.3.1 AttachBuffer() . . . 659
14.19.3.2 CheckBufferLayout() . . . 659
14.20 CChunkAdapterU3V Class Reference . . . 659
14.20.1 Detailed Description . . . 660
14.20.2 Constructor & Destructor Documentation . . . 660
14.20.2.1 CChunkAdapterU3V() . . . 660
14.20.2.2 ~CChunkAdapterU3V() . . . 661
14.20.3 Member Function Documentation . . . 661
14.20.3.1 AttachBuffer() . . . 661
14.20.3.2 CheckBufferLayout() . . . 661
14.21 CChunkPort Class Reference . . . 662
14.21.1 Detailed Description . . . 663
14.21.2 Constructor & Destructor Documentation . . . 663
14.21.2.1 CChunkPort() . . . 663
14.21.2.2 ~CChunkPort() . . . 664

14.21.3 Member Function Documentation . . . 664
14.21.3.1 AttachChunk() . . . 664
14.21.3.2 AttachPort() . . . 664
14.21.3.3 CheckChunkID() [1/2] . . . 664
14.21.3.4 CheckChunkID() [2/2] . . . 664
14.21.3.5 ClearCache() . . . 665
14.21.3.6 DetachChunk() . . . 665
14.21.3.7 DetachPort() . . . 665
14.21.3.8 GetAccessMode() . . . 665
14.21.3.9 GetChunkIDLength() . . . 665
14.21.3.10 GetPrincipalInterfaceType() . . . 665
14.21.3.11 GetSwapEndianess() . . . 666
14.21.3.12 InvalidateNode() . . . 666
14.21.3.13 Read() . . . 666
14.21.3.14 SetPortImpl() . . . 666
14.21.3.15 UpdateBuffer() . . . 666
14.21.3.16 Write() . . . 666
14.21.4 Member Data Documentation . . . 667
14.21.4.1 m_pChunkPort . . . 667
14.21.4.2 m_pPort . . . 667
14.21.4.3 m_pPortAdapter . . . 667
14.22 CEnumerationTRef< EnumT > Class Template Reference . . . 667
14.22.1 Detailed Description . . . 669
14.22.2 Constructor & Destructor Documentation . . . 669
14.22.2.1 CEnumerationTRef() [1/2] . . . 669
14.22.2.2 CEnumerationTRef() [2/2] . . . 669
14.22.2.3 ~CEnumerationTRef() . . . 669
14.22.3 Member Function Documentation . . . 669
14.22.3.1 GetCurrentEntry() . . . 670
14.22.3.2 GetEntry() [1/2] . . . 670

14.22.3.3 GetEntry() [2/2] . . . 670
14.22.3.4 GetValue() . . . 670
14.22.3.5 operator()() . . . 671
14.22.3.6 operator=() [1/2] . . . 671
14.22.3.7 operator=() [2/2] . . . 671
14.22.3.8 SetEnumReference() . . . 671
14.22.3.9 SetNumEnums() . . . 671
14.22.3.10 SetReference() . . . 672
14.22.3.11 SetValue() . . . 672
14.23 CEventAdapter Class Reference . . . 672
14.23.1 Detailed Description . . . 673
14.23.2 Constructor & Destructor Documentation . . . 673
14.23.2.1 CEventAdapter() . . . 673
14.23.2.2 ~CEventAdapter() . . . 673
14.23.3 Member Function Documentation . . . 673
14.23.3.1 AttachNodeMap() . . . 673
14.23.3.2 DeliverMessage() . . . 674
14.23.3.3 DetachNodeMap() . . . 674
14.23.4 Member Data Documentation . . . 674
14.23.4.1 m_pEventAdapter . . . 674
14.24 CEventAdapter1394 Class Reference . . . 674
14.24.1 Detailed Description . . . 675
14.24.2 Constructor & Destructor Documentation . . . 675
14.24.2.1 CEventAdapter1394() . . . 675
14.24.2.2 ~CEventAdapter1394() . . . 675
14.24.3 Member Function Documentation . . . 676
14.24.3.1 DeliverEventMessage() . . . 676
14.24.3.2 DeliverMessage() . . . 676
14.25 CEventAdapterGeneric Class Reference . . . 676
14.25.1 Detailed Description . . . 677

14.25.2 Constructor & Destructor Documentation . . . 677
14.25.2.1 CEventAdapterGeneric() . . . 677
14.25.2.2 ~CEventAdapterGeneric() . . . 678
14.25.3 Member Function Documentation . . . 678
14.25.3.1 DeliverMessage() [1/3] . . . 678
14.25.3.2 DeliverMessage() [2/3] . . . 678
14.25.3.3 DeliverMessage() [3/3] . . . 678
14.26 CEventAdapterGEV Class Reference . . . 679
14.26.1 Detailed Description . . . 680
14.26.2 Constructor & Destructor Documentation . . . 680
14.26.2.1 CEventAdapterGEV() . . . 680
14.26.2.2 ~CEventAdapterGEV() . . . 680
14.26.3 Member Function Documentation . . . 680
14.26.3.1 DeliverEventMessage() [1/2] . . . 680
14.26.3.2 DeliverEventMessage() [2/2] . . . 680
14.26.3.3 DeliverMessage() . . . 681
14.27 CEventAdapterU3V Class Reference . . . 681
14.27.1 Detailed Description . . . 682
14.27.2 Constructor & Destructor Documentation . . . 682
14.27.2.1 CEventAdapterU3V() . . . 682
14.27.2.2 ~CEventAdapterU3V() . . . 682
14.27.3 Member Function Documentation . . . 682
14.27.3.1 DeliverEventMessage() . . . 682
14.27.3.2 DeliverMessage() . . . 683
14.28 CEventPort Class Reference . . . 683
14.28.1 Detailed Description . . . 684
14.28.2 Constructor & Destructor Documentation . . . 684
14.28.2.1 CEventPort() . . . 685
14.28.2.2 ~CEventPort() . . . 685
14.28.3 Member Function Documentation . . . 685

14.28.3.1 AttachEvent() . . . 685
14.28.3.2 AttachNode() . . . 685
14.28.3.3 CheckEventID() [1/2] . . . 685
14.28.3.4 CheckEventID() [2/2] . . . 686
14.28.3.5 DetachEvent() . . . 686
14.28.3.6 DetachNode() . . . 686
14.28.3.7 GetAccessMode() . . . 686
14.28.3.8 GetEventIDLength() . . . 686
14.28.3.9 GetPrincipalInterfaceType() . . . 686
14.28.3.10 GetSwapEndianess() . . . 687
14.28.3.11 InvalidateNode() . . . 687
14.28.3.12 Read() . . . 687
14.28.3.13 SetPortImpl() . . . 687
14.28.3.14 Write() . . . 687
14.28.4 Member Data Documentation . . . 687
14.28.4.1 m_pEventPort . . . 688
14.28.4.2 m_pNode . . . 688
14.28.4.3 m_pPortAdapter . . . 688
14.29 CFeatureBag Class Reference . . . 688
14.29.1 Detailed Description . . . 689
14.29.2 Constructor & Destructor Documentation . . . 689
14.29.2.1 CFeatureBag() . . . 689
14.29.2.2 ~CFeatureBag() . . . 689
14.29.3 Member Function Documentation . . . 689
14.29.3.1 GetFeatureBagHandle() . . . 689
14.29.3.2 LoadFromBag() . . . 689
14.29.3.3 operator==() . . . 690
14.29.3.4 PersistFeature() . . . 690
14.29.3.5 SetInfo() . . . 690
14.29.3.6 StoreToBag() . . . 690

14.30 CFloatPtr Class Reference . . . 691
14.30.1 Detailed Description . . . 692
14.30.2 Constructor & Destructor Documentation . . . 692
14.30.2.1 CFloatPtr() [1/2] . . . 692
14.30.2.2 CFloatPtr() [2/2] . . . 692
14.30.3 Member Function Documentation . . . 692
14.30.3.1 GetEnumAlias() . . . 692
14.30.3.2 GetIntAlias() . . . 692
14.30.3.3 operator=() . . . 693
14.31 CGeneric_XMLLoaderParams Class Reference . . . 693
14.31.1 Detailed Description . . . 693
14.31.2 Member Function Documentation . . . 693
14.31.2.1 _Initialize() . . . 694
14.32 CGlobalLock Class Reference . . . 694
14.32.1 Detailed Description . . . 694
14.32.2 Constructor & Destructor Documentation . . . 694
14.32.2.1 CGlobalLock() [1/2] . . . 695
14.32.2.2 CGlobalLock() [2/2] . . . 695
14.32.2.3 ~CGlobalLock() . . . 695
14.32.3 Member Function Documentation . . . 695
14.32.3.1 IsValid() . . . 695
14.32.3.2 Lock() . . . 695
14.32.3.3 TryLock() . . . 696
14.32.3.4 Unlock() . . . 696
14.32.4 Member Data Documentation . . . 696
14.32.4.1 m_DebugCount . . . 696
14.33 CGlobalLockUnlocker Class Reference . . . 696
14.33.1 Detailed Description . . . 697
14.33.2 Constructor & Destructor Documentation . . . 697
14.33.2.1 CGlobalLockUnlocker() . . . 697

14.33.2.2 ~CGlobalLockUnlocker() . . . 697
14.33.3 Member Function Documentation . . . 697
14.33.3.1 UnlockEarly() . . . 697
14.33.4 Member Data Documentation . . . 698
14.33.4.1 m_enabled . . . 698
14.33.4.2 m_Lock . . . 698
14.34 ChunkData Class Reference . . . 698
14.34.1 Detailed Description . . . 700
14.34.2 Constructor & Destructor Documentation . . . 700
14.34.2.1 ChunkData() [1/2] . . . 700
14.34.2.2 ChunkData() [2/2] . . . 701
14.34.2.3 ~ChunkData() . . . 701
14.34.3 Member Function Documentation . . . 701
14.34.3.1 GetBlackLevel() . . . 701
14.34.3.2 GetCounterValue() . . . 701
14.34.3.3 GetCRC() . . . 701
14.34.3.4 GetEncoderValue() . . . 702
14.34.3.5 GetExposureEndLineStatusAll() . . . 702
14.34.3.6 GetExposureTime() . . . 702
14.34.3.7 GetFrameID() . . . 702
14.34.3.8 GetGain() . . . 703
14.34.3.9 GetHeight() . . . 703
14.34.3.10 GetImage() . . . 703
14.34.3.11 GetInferenceBoundingBoxResult() . . . 703
14.34.3.12 GetInferenceConfidence() . . . 704
14.34.3.13 GetInferenceFrameId() . . . 704
14.34.3.14 GetInferenceResult() . . . 704
14.34.3.15 GetLinePitch() . . . 704
14.34.3.16 GetLineStatusAll() . . . 705
14.34.3.17 GetOffsetX() . . . 705

14.34.3.18 GetOffsetY() . . . 705
14.34.3.19 GetPartSelector() . . . 705
14.34.3.20 GetPixelDynamicRangeMax() . . . 706
14.34.3.21 GetPixelDynamicRangeMin() . . . 706
14.34.3.22 GetScan3dAxisMax() . . . 706
14.34.3.23 GetScan3dAxisMin() . . . 706
14.34.3.24 GetScan3dCoordinateOffset() . . . 707
14.34.3.25 GetScan3dCoordinateReferenceValue() . . . 707
14.34.3.26 GetScan3dCoordinateScale() . . . 707
14.34.3.27 GetScan3dInvalidDataValue() . . . 707
14.34.3.28 GetScan3dTransformValue() . . . 708
14.34.3.29 GetScanLineSelector() . . . 708
14.34.3.30 GetSequencerSetActive() . . . 708
14.34.3.31 GetSerialDataLength() . . . 708
14.34.3.32 GetStreamChannelID() . . . 709
14.34.3.33 GetTimerValue() . . . 709
14.34.3.34 GetTimestamp() . . . 709
14.34.3.35 GetTimestampLatchValue() . . . 709
14.34.3.36 GetTransferBlockID() . . . 710
14.34.3.37 GetTransferQueueCurrentBlockCount() . . . 710
14.34.3.38 GetWidth() . . . 710
14.34.3.39 SetChunks() . . . 710
14.35 CLock Class Reference . . . 711
14.35.1 Detailed Description . . . 711
14.35.2 Constructor & Destructor Documentation . . . 711
14.35.2.1 CLock() . . . 711
14.35.2.2 ~CLock() . . . 712
14.35.3 Member Function Documentation . . . 712
14.35.3.1 Lock() . . . 712
14.35.3.2 TryLock() . . . 712

14.35.3.3 Unlock() . . . 712
14.36 CLock Class Reference . . . 712
14.36.1 Detailed Description . . . 713
14.36.2 Constructor & Destructor Documentation . . . 713
14.36.2.1 CLock() [1/2] . . . 713
14.36.2.2 CLock() [2/2] . . . 713
14.36.2.3 ~CLock() . . . 714
14.36.3 Member Function Documentation . . . 714
14.36.3.1 Lock() . . . 714
14.36.3.2 TryLock() . . . 714
14.36.3.3 Unlock() . . . 714
14.36.4 Friends And Related Function Documentation . . . 714
14.36.4.1 NodeMap . . . 714
14.36.5 Member Data Documentation . . . 714
14.36.5.1 m_bOwnLock . . . 715
14.36.5.2 m_lock . . . 715
14.37 CLockEx Class Reference . . . 715
14.37.1 Detailed Description . . . 716
14.38 CLockEx Class Reference . . . 716
14.38.1 Detailed Description . . . 717
14.38.2 Member Data Documentation . . . 717
14.38.2.1 m_lockEx . . . 717
14.39 CNodeCallback Class Reference . . . 717
14.39.1 Detailed Description . . . 718
14.39.2 Constructor & Destructor Documentation . . . 718
14.39.2.1 CNodeCallback() . . . 718
14.39.2.2 ~CNodeCallback() . . . 718
14.39.3 Member Function Documentation . . . 718
14.39.3.1 Destroy() . . . 719
14.39.3.2 GetCallbackType() . . . 719

14.39.3.3 GetNode() 719
14.39.3.4 operator()() 719
14.39.4 Member Data Documentation 719
14.39.4.1 m_CallbackType 719
14.39.4.2 m_pNode 720
14.40 CNodeMapFactory Class Reference 720
14.40.1 Detailed Description 721
14.40.2 Constructor & Destructor Documentation 722
14.40.2.1 CNodeMapFactory() [1/5] 722
14.40.2.2 ~CNodeMapFactory() 722
14.40.2.3 CNodeMapFactory() [2/5] 722
14.40.2.4 CNodeMapFactory() [3/5] 723
14.40.2.5 CNodeMapFactory() [4/5] 723
14.40.2.6 CNodeMapFactory() [5/5] 724
14.40.3 Member Function Documentation 724
14.40.3.1 AddInjectionData() 724
14.40.3.2 ApplyStyleSheet() 725
14.40.3.3 ClearCache() 725
14.40.3.4 CreateEmptyNodeMap() 725
14.40.3.5 CreateNodeDataFromNodeMap() 725
14.40.3.6 CreateNodeMap() [1/2] 725
14.40.3.7 CreateNodeMap() [2/2] 726
14.40.3.8 ExtractSubtree() 726
14.40.3.9 GetNodeStatistics() 726
14.40.3.10 GetSupportedSchemaVersions() 727
14.40.3.11 IsCameraDescriptionFileDataReleased() 727
14.40.3.12 IsEmpty() 727
14.40.3.13 IsLoaded() 727
14.40.3.14 IsPreprocessed() 727
14.40.3.15 LoadAndInject() 727

14.40.3.16 operator=() . . . 728
14.40.3.17 Preprocess() . . . 728
14.40.3.18 ReleaseCameraDescriptionFileData() . . . 728
14.40.3.19 ToString() . . . 728
14.40.3.20 ToXml() . . . 728
14.41 CNodeMapRef Class Reference . . . 729
14.41.1 Detailed Description . . . 730
14.41.2 Constructor & Destructor Documentation . . . 730
14.41.2.1 CNodeMapRef() [1/3] . . . 730
14.41.2.2 CNodeMapRef() [2/3] . . . 730
14.41.2.3 CNodeMapRef() [3/3] . . . 730
14.41.3 Member Function Documentation . . . 731
14.41.3.1 operator=() [1/2] . . . 731
14.41.3.2 operator=() [2/2] . . . 731
14.42 CNodeMapRefT< TCameraParams > Class Template Reference . . . 731
14.42.1 Detailed Description . . . 733
14.42.2 Member Function Documentation . . . 733
14.42.2.1 _ClearXMLCache() . . . 733
14.42.2.2 _Connect() [1/2] . . . 734
14.42.2.3 _Connect() [2/2] . . . 734
14.42.2.4 _GetDeviceName() . . . 734
14.42.2.5 _GetNode() . . . 734
14.42.2.6 _GetNodes() . . . 734
14.42.2.7 _GetSupportedSchemaVersions() . . . 735
14.42.2.8 _InvalidateNodes() . . . 735
14.42.2.9 _LoadXMLFromFile() . . . 735
14.42.2.10 _LoadXMLFromFileInject() . . . 735
14.42.2.11 _LoadXMLFromString() . . . 735
14.42.2.12 _LoadXMLFromStringInject() . . . 736
14.42.2.13 _LoadXMLFromZIPData() . . . 736

14.42.2.14 LoadXMLFromZIPFile() . . . 736
14.42.2.15 Poll() . . . 736
14.42.3 Member Data Documentation . . . 736
14.42.3.1 _Ptr . . . 736
14.43 CommandNode Class Reference . . . 737
14.43.1 Detailed Description . . . 738
14.43.2 Constructor & Destructor Documentation . . . 738
14.43.2.1 CommandNode() [1/2] . . . 738
14.43.2.2 CommandNode() [2/2] . . . 738
14.43.2.3 ~CommandNode() . . . 738
14.43.3 Member Function Documentation . . . 738
14.43.3.1 Execute() . . . 738
14.43.3.2 IsDone() . . . 739
14.43.3.3 operator()() . . . 739
14.43.3.4 SetReference() . . . 739
14.44 Counter Class Reference . . . 739
14.44.1 Detailed Description . . . 740
14.44.2 Constructor & Destructor Documentation . . . 740
14.44.2.1 Counter() . . . 740
14.44.3 Member Function Documentation . . . 740
14.44.3.1 GetValue() . . . 740
14.44.3.2 IsZero() . . . 740
14.44.3.3 operator unsigned int() . . . 740
14.44.3.4 operator++() [1/2] . . . 741
14.44.3.5 operator++() [2/2] . . . 741
14.44.3.6 operator--() [1/2] . . . 741
14.44.3.7 operator--() [2/2] . . . 741
14.45 CPointer< T, B > Class Template Reference . . . 741
14.45.1 Detailed Description . . . 742
14.45.2 Constructor & Destructor Documentation . . . 743

14.45.2.1 CPointer() [1/2] . . . 743
14.45.2.2 CPointer() [2/2] . . . 743
14.45.2.3 ∼CPointer() . . . 743
14.45.3 Member Function Documentation . . . 743
14.45.3.1 IsValid() . . . 743
14.45.3.2 operator bool() . . . 743
14.45.3.3 operator T∗() . . . 744
14.45.3.4 operator"!=() [1/5] . . . 744
14.45.3.5 operator"!=() [2/5] . . . 744
14.45.3.6 operator"!=() [3/5] . . . 744
14.45.3.7 operator"!=() [4/5] . . . 744
14.45.3.8 operator"!=() [5/5] . . . 744
14.45.3.9 operator()() . . . 745
14.45.3.10operator∗() . . . 745
14.45.3.11operator->() . . . 745
14.45.3.12operator=() . . . 745
14.45.3.13operator==() [1/3] . . . 745
14.45.3.14operator==() [2/3] . . . 745
14.45.3.15operator==() [3/3] . . . 746
14.45.4 Member Data Documentation . . . 746
14.45.4.1 m_pT . . . 746
14.46CPortImpl Class Reference . . . 746
14.46.1 Detailed Description . . . 747
14.46.2 Constructor & Destructor Documentation . . . 747
14.46.2.1 CPortImpl() . . . 748
14.46.2.2 ∼CPortImpl() . . . 748
14.46.3 Member Function Documentation . . . 748
14.46.3.1 GetAccessMode() . . . 748
14.46.3.2 GetSwapEndianess() . . . 748
14.46.3.3 InvalidateNode() . . . 748

14.46.3.4 Read() . . . 749
14.46.3.5 Replay() . . . 749
14.46.3.6 SetPortImpl() . . . 749
14.46.3.7 Write() . . . 749
14.46.4 Member Data Documentation . . . 749
14.46.4.1 m_ptrPort . . . 750
14.47 CPortWriteList Class Reference . . . 750
14.47.1 Detailed Description . . . 751
14.47.2 Constructor & Destructor Documentation . . . 751
14.47.2.1 CPortWriteList() . . . 751
14.47.2.2 ~CPortWriteList() . . . 751
14.47.3 Member Function Documentation . . . 751
14.47.3.1 GetCookie() . . . 752
14.47.3.2 GetPortWriteListHandle() . . . 752
14.47.3.3 Replay() . . . 752
14.47.3.4 SetCookie() . . . 752
14.47.3.5 Write() . . . 752
14.47.4 Member Data Documentation . . . 752
14.47.4.1 m_pWriteList . . . 753
14.48 CpuUsageInfo Struct Reference . . . 753
14.48.1 Member Data Documentation . . . 753
14.48.1.1 dummy . . . 753
14.49 CRegisterPortImpl Class Reference . . . 753
14.49.1 Detailed Description . . . 754
14.49.2 Constructor & Destructor Documentation . . . 754
14.49.2.1 CRegisterPortImpl() . . . 755
14.49.2.2 ~CRegisterPortImpl() . . . 755
14.49.3 Member Function Documentation . . . 755
14.49.3.1 GetAccessMode() . . . 755
14.49.3.2 Read() . . . 755

14.49.3.3 ReadRegister() . . . 756
14.49.3.4 SetPortImpl() . . . 756
14.49.3.5 Write() . . . 756
14.49.3.6 WriteRegister() . . . 756
14.50 CSelectorSet Class Reference . . . 757
14.50.1 Detailed Description . . . 758
14.50.2 Constructor & Destructor Documentation . . . 758
14.50.2.1 CSelectorSet() . . . 758
14.50.2.2 ∼CSelectorSet() . . . 758
14.50.3 Member Function Documentation . . . 758
14.50.3.1 GetSelectorList() . . . 758
14.50.3.2 IsEmpty() . . . 759
14.50.3.3 Restore() . . . 759
14.50.3.4 SetFirst() . . . 759
14.50.3.5 SetNext() . . . 759
14.50.3.6 ToString() . . . 759
14.51 CTestPortStruct< CDataStruct > Class Template Reference . . . 760
14.51.1 Detailed Description . . . 761
14.51.2 Constructor & Destructor Documentation . . . 761
14.51.2.1 CTestPortStruct() . . . 761
14.51.3 Member Function Documentation . . . 761
14.51.3.1 GetAccessMode() . . . 761
14.51.3.2 GetNumReads() . . . 762
14.51.3.3 GetNumWrites() . . . 762
14.51.3.4 GetPrincipalInterfaceType() . . . 762
14.51.3.5 MemSet() . . . 762
14.51.3.6 Read() . . . 762
14.51.3.7 ResetStatistics() . . . 762
14.51.3.8 Write() . . . 763
14.51.4 Member Data Documentation . . . 763

14.51.4.1 m_BaseAddress . . . 763
14.51.4.2 m_NumReads . . . 763
14.51.4.3 m_NumWrites . . . 763
14.52 DCAM_CHECKSUM Struct Reference . . . 763
14.52.1 Member Data Documentation . . . 764
14.52.1.1 CRCChecksum . . . 764
14.53 DCAM_CHUNK_TRAILER Struct Reference . . . 764
14.53.1 Member Data Documentation . . . 764
14.53.1.1 ChunkID . . . 764
14.53.1.2 ChunkLength . . . 764
14.53.1.3 InverseChunkLength . . . 764
14.54 DeviceArrivalEventHandler Class Reference . . . 765
14.54.1 Detailed Description . . . 766
14.54.2 Constructor & Destructor Documentation . . . 766
14.54.2.1 DeviceArrivalEventHandler() . . . 766
14.54.2.2 ~DeviceArrivalEventHandler() . . . 766
14.54.3 Member Function Documentation . . . 766
14.54.3.1 OnDeviceArrival() . . . 766
14.54.3.2 operator=() . . . 767
14.55 DeviceEventHandler Class Reference . . . 767
14.55.1 Detailed Description . . . 768
14.55.2 Constructor & Destructor Documentation . . . 768
14.55.2.1 DeviceEventHandler() . . . 769
14.55.2.2 ~DeviceEventHandler() . . . 769
14.55.3 Member Function Documentation . . . 769
14.55.3.1 GetDeviceEventId() . . . 769
14.55.3.2 GetDeviceEventName() . . . 769
14.55.3.3 OnDeviceEvent() . . . 769
14.55.3.4 operator=() . . . 770
14.56 DeviceEventHandlerImpl Class Reference . . . 770

14.56.1 Constructor & Destructor Documentation . . . 771
14.56.1.1 DeviceEventHandlerImpl() . . . 771
14.56.1.2 ~DeviceEventHandlerImpl() . . . 771
14.56.2 Member Function Documentation . . . 772
14.56.2.1 OnDeviceEvent() . . . 772
14.57 DeviceRemovalEventHandler Class Reference . . . 772
14.57.1 Detailed Description . . . 773
14.57.2 Constructor & Destructor Documentation . . . 773
14.57.2.1 DeviceRemovalEventHandler() . . . 774
14.57.2.2 ~DeviceRemovalEventHandler() . . . 774
14.57.3 Member Function Documentation . . . 774
14.57.3.1 OnDeviceRemoval() . . . 774
14.57.3.2 operator=() . . . 774
14.58 double_autovector_t Class Reference . . . 774
14.58.1 Detailed Description . . . 775
14.58.2 Constructor & Destructor Documentation . . . 775
14.58.2.1 double_autovector_t() [1/3] . . . 775
14.58.2.2 double_autovector_t() [2/3] . . . 775
14.58.2.3 double_autovector_t() [3/3] . . . 775
14.58.2.4 ~double_autovector_t() . . . 776
14.58.3 Member Function Documentation . . . 776
14.58.3.1 operator delete() . . . 776
14.58.3.2 operator new() . . . 776
14.58.3.3 operator=() . . . 776
14.58.3.4 operator[]() [1/2] . . . 776
14.58.3.5 operator[]() [2/2] . . . 776
14.58.3.6 size() . . . 777
14.58.4 Member Data Documentation . . . 777
14.58.4.1 _pCount . . . 777
14.58.4.2 _pv . . . 777

14.59 EAccessModeClass Class Reference . . . 777
14.59.1 Detailed Description . . . 777
14.59.2 Member Function Documentation . . . 777
14.59.2.1 FromString() . . . 778
14.59.2.2 ToString() [1/2] . . . 778
14.59.2.3 ToString() [2/2] . . . 778
14.60 ECachingModeClass Class Reference . . . 778
14.60.1 Detailed Description . . . 778
14.60.2 Member Function Documentation . . . 779
14.60.2.1 FromString() . . . 779
14.60.2.2 ToString() [1/2] . . . 779
14.60.2.3 ToString() [2/2] . . . 779
14.61 EDisplayNotationClass Class Reference . . . 779
14.61.1 Detailed Description . . . 780
14.61.2 Member Function Documentation . . . 780
14.61.2.1 FromString() . . . 780
14.61.2.2 ToString() [1/2] . . . 780
14.61.2.3 ToString() [2/2] . . . 780
14.62 EEndianessClass Class Reference . . . 780
14.62.1 Detailed Description . . . 781
14.62.2 Member Function Documentation . . . 781
14.62.2.1 FromString() . . . 781
14.62.2.2 ToString() [1/2] . . . 781
14.62.2.3 ToString() [2/2] . . . 781
14.63 EGenApiSchemaVersionClass Class Reference . . . 782
14.63.1 Detailed Description . . . 782
14.63.2 Member Function Documentation . . . 782
14.63.2.1 FromString() . . . 782
14.63.2.2 ToString() [1/2] . . . 782
14.63.2.3 ToString() [2/2] . . . 782

14.64 EInputDirectionClass Class Reference . . . 783
14.64.1 Detailed Description . . . 783
14.64.2 Member Function Documentation . . . 783
14.64.2.1 FromString() . . . 783
14.64.2.2 ToString() [1/2] . . . 783
14.64.2.3 ToString() [2/2] . . . 783
14.65 ENameSpaceClass Class Reference . . . 784
14.65.1 Detailed Description . . . 784
14.65.2 Member Function Documentation . . . 784
14.65.2.1 FromString() . . . 784
14.65.2.2 ToString() [1/2] . . . 784
14.65.2.3 ToString() [2/2] . . . 784
14.66 EnumEntryNode Class Reference . . . 785
14.66.1 Detailed Description . . . 786
14.66.2 Constructor & Destructor Documentation . . . 786
14.66.2.1 EnumEntryNode() [1/2] . . . 786
14.66.2.2 EnumEntryNode() [2/2] . . . 786
14.66.2.3 ~EnumEntryNode() . . . 786
14.66.3 Member Function Documentation . . . 786
14.66.3.1 GetNumericValue() . . . 787
14.66.3.2 GetSymbolic() . . . 787
14.66.3.3 GetValue() . . . 787
14.66.3.4 IsSelfClearing() . . . 787
14.66.3.5 SetReference() . . . 787
14.67 EnumNode Class Reference . . . 788
14.67.1 Detailed Description . . . 790
14.67.2 Constructor & Destructor Documentation . . . 790
14.67.2.1 EnumNode() [1/2] . . . 790
14.67.2.2 EnumNode() [2/2] . . . 790
14.67.2.3 ~EnumNode() . . . 790

14.67.3 Member Function Documentation . . . 790
14.67.3.1 GetCurrentEntry() . . . 790
14.67.3.2 GetEntries() . . . 791
14.67.3.3 GetEntry() . . . 791
14.67.3.4 GetEntryByName() . . . 791
14.67.3.5 GetIntValue() . . . 791
14.67.3.6 GetSymbolics() . . . 792
14.67.3.7 operator∗() . . . 792
14.67.3.8 operator=() . . . 792
14.67.3.9 SetIntValue() . . . 792
14.67.3.10 SetReference() . . . 792
14.67.4 Member Data Documentation . . . 793
14.67.4.1 m_pEnumeration . . . 793
14.68 ERepresentationClass Class Reference . . . 793
14.68.1 Detailed Description . . . 793
14.68.2 Member Function Documentation . . . 793
14.68.2.1 FromString() . . . 794
14.68.2.2 ToString() [1/2] . . . 794
14.68.2.3 ToString() [2/2] . . . 794
14.69 ESignClass Class Reference . . . 794
14.69.1 Detailed Description . . . 794
14.69.2 Member Function Documentation . . . 795
14.69.2.1 FromString() . . . 795
14.69.2.2 ToString() [1/2] . . . 795
14.69.2.3 ToString() [2/2] . . . 795
14.70 ESlopeClass Class Reference . . . 795
14.70.1 Detailed Description . . . 796
14.70.2 Member Function Documentation . . . 796
14.70.2.1 FromString() . . . 796
14.70.2.2 ToString() [1/2] . . . 796

14.70.2.3 ToString() [2/2] . . . 796
14.71 EStandardNameSpaceClass Class Reference . . . 796
14.71.1 Detailed Description . . . 797
14.71.2 Member Function Documentation . . . 797
14.71.2.1 FromString() . . . 797
14.71.2.2 ToString() [1/2] . . . 797
14.71.2.3 ToString() [2/2] . . . 797
14.72 EventHandler Class Reference . . . 798
14.72.1 Detailed Description . . . 799
14.72.2 Constructor & Destructor Documentation . . . 799
14.72.2.1 ∼EventHandler() . . . 799
14.72.2.2 EventHandler() . . . 799
14.72.3 Member Function Documentation . . . 799
14.72.3.1 GetEventPayloadData() . . . 799
14.72.3.2 GetEventPayloadDataSize() . . . 799
14.72.3.3 GetEventType() . . . 800
14.72.3.4 operator=() . . . 800
14.72.3.5 SetEventPayload() . . . 800
14.72.3.6 SetEventType() . . . 800
14.72.4 Friends And Related Function Documentation . . . 800
14.72.4.1 EventProcessor . . . 800
14.72.4.2 IDataStream . . . 801
14.72.4.3 Stream . . . 801
14.72.5 Member Data Documentation . . . 801
14.72.5.1 m_pEventData . . . 801
14.73 EVisibilityClass Class Reference . . . 801
14.73.1 Detailed Description . . . 801
14.73.2 Member Function Documentation . . . 801
14.73.2.1 FromString() . . . 802
14.73.2.2 ToString() [1/2] . . . 802

14.73.2.3 ToString() [2/2] . . . 802
14.74 Exception Class Reference . . . 802
14.74.1 Detailed Description . . . 803
14.74.2 Constructor & Destructor Documentation . . . 804
14.74.2.1 Exception() [1/4] . . . 804
14.74.2.2 Exception() [2/4] . . . 804
14.74.2.3 Exception() [3/4] . . . 804
14.74.2.4 Exception() [4/4] . . . 805
14.74.2.5 ~Exception() . . . 805
14.74.3 Member Function Documentation . . . 805
14.74.3.1 GetBuildDate() . . . 805
14.74.3.2 GetBuildTime() . . . 805
14.74.3.3 GetError() . . . 805
14.74.3.4 GetErrorMessage() . . . 806
14.74.3.5 GetFileName() . . . 806
14.74.3.6 GetFullErrorMessage() . . . 806
14.74.3.7 GetFunctionName() . . . 806
14.74.3.8 GetLineNumber() . . . 806
14.74.3.9 operator"!=() . . . 806
14.74.3.10 operator=() . . . 807
14.74.3.11 operator==() . . . 807
14.74.3.12 what() . . . 807
14.75 EYesNoClass Class Reference . . . 807
14.75.1 Detailed Description . . . 807
14.75.2 Member Function Documentation . . . 808
14.75.2.1 FromString() . . . 808
14.75.2.2 ToString() [1/2] . . . 808
14.75.2.3 ToString() [2/2] . . . 808
14.76 FileProtocolAdapter Class Reference . . . 808
14.76.1 Detailed Description . . . 809

14.76.2 Constructor & Destructor Documentation . . . 809
14.76.2.1 FileProtocolAdapter() . . . 809
14.76.2.2 ~FileProtocolAdapter() . . . 809
14.76.3 Member Function Documentation . . . 809
14.76.3.1 attach() . . . 809
14.76.3.2 closeFile() . . . 810
14.76.3.3 deleteFile() . . . 810
14.76.3.4 getBufSize() . . . 810
14.76.3.5 openFile() . . . 811
14.76.3.6 read() . . . 811
14.76.3.7 write() . . . 812
14.77 FloatNode Class Reference . . . 812
14.77.1 Detailed Description . . . 814
14.77.2 Constructor & Destructor Documentation . . . 815
14.77.2.1 FloatNode() [1/2] . . . 815
14.77.2.2 FloatNode() [2/2] . . . 815
14.77.2.3 ~FloatNode() . . . 815
14.77.3 Member Function Documentation . . . 815
14.77.3.1 GetDisplayNotation() . . . 815
14.77.3.2 GetDisplayPrecision() . . . 815
14.77.3.3 GetEnumAlias() . . . 815
14.77.3.4 GetInc() . . . 816
14.77.3.5 GetIncMode() . . . 816
14.77.3.6 GetIntAlias() . . . 816
14.77.3.7 GetListOfValidValues() . . . 816
14.77.3.8 GetMax() . . . 816
14.77.3.9 GetMin() . . . 816
14.77.3.10 GetRepresentation() . . . 817
14.77.3.11 GetUnit() . . . 817
14.77.3.12 GetValue() . . . 817

14.77.3.13 HasInc() . . . 817
14.77.3.14 ImposeMax() . . . 817
14.77.3.15 ImposeMin() . . . 818
14.77.3.16 operator()() . . . 818
14.77.3.17 operator∗() . . . 818
14.77.3.18 operator=() . . . 818
14.77.3.19 SetReference() . . . 818
14.77.3.20 SetValue() . . . 818
14.78 FloatRegNode Class Reference . . . 819
14.78.1 Detailed Description . . . 820
14.78.2 Constructor & Destructor Documentation . . . 820
14.78.2.1 FloatRegNode() [1/2] . . . 821
14.78.2.2 FloatRegNode() [2/2] . . . 821
14.78.2.3 ∼FloatRegNode() . . . 821
14.78.3 Member Function Documentation . . . 821
14.78.3.1 SetReference() . . . 821
14.79 Function_NodeCallback< Function > Class Template Reference . . . 822
14.79.1 Detailed Description . . . 822
14.79.2 Constructor & Destructor Documentation . . . 823
14.79.2.1 Function_NodeCallback() . . . 823
14.79.3 Member Function Documentation . . . 823
14.79.3.1 Destroy() . . . 823
14.79.3.2 operator()() . . . 823
14.80 gcstring Class Reference . . . 824
14.80.1 Constructor & Destructor Documentation . . . 825
14.80.1.1 gcstring() [1/5] . . . 825
14.80.1.2 gcstring() [2/5] . . . 825
14.80.1.3 gcstring() [3/5] . . . 825
14.80.1.4 gcstring() [4/5] . . . 825
14.80.1.5 gcstring() [5/5] . . . 826

14.80.1.6 ~gcstring() . . . 826
14.80.2 Member Function Documentation . . . 826
14.80.2.1 _npos() . . . 826
14.80.2.2 append() [1/2] . . . 826
14.80.2.3 append() [2/2] . . . 826
14.80.2.4 assign() [1/4] . . . 826
14.80.2.5 assign() [2/4] . . . 827
14.80.2.6 assign() [3/4] . . . 827
14.80.2.7 assign() [4/4] . . . 827
14.80.2.8 c_str() . . . 827
14.80.2.9 compare() . . . 827
14.80.2.10 empty() . . . 827
14.80.2.11 find() [1/5] . . . 828
14.80.2.12 find() [2/5] . . . 828
14.80.2.13 find() [3/5] . . . 828
14.80.2.14 find() [4/5] . . . 828
14.80.2.15 find() [5/5] . . . 828
14.80.2.16 find_first_not_of() . . . 828
14.80.2.17 find_first_of() . . . 829
14.80.2.18 length() . . . 829
14.80.2.19 max_size() . . . 829
14.80.2.20 operator const char *() . . . 829
14.80.2.21 operator delete() [1/2] . . . 829
14.80.2.22 operator delete() [2/2] . . . 829
14.80.2.23 operator new() [1/2] . . . 829
14.80.2.24 operator new() [2/2] . . . 830
14.80.2.25 operator"!=() [1/2] . . . 830
14.80.2.26 operator"!=() [2/2] . . . 830
14.80.2.27 operator+=() [1/5] . . . 830
14.80.2.28 operator+=() [2/5] . . . 830

14.80.2.29 operator+=() [3/5] . . . 830
14.80.2.30 operator+=() [4/5] . . . 830
14.80.2.31 operator+=() [5/5] . . . 831
14.80.2.32 operator<() . . . 831
14.80.2.33 operator=() . . . 831
14.80.2.34 operator==() [1/2] . . . 831
14.80.2.35 operator==() [2/2] . . . 831
14.80.2.36 operator>() . . . 831
14.80.2.37 resize() . . . 831
14.80.2.38 size() . . . 832
14.80.2.39 substr() . . . 832
14.80.2.40 swap() . . . 832
14.80.3 Friends And Related Function Documentation . . . 832
14.80.3.1 operator+ [1/3] . . . 832
14.80.3.2 operator+ [2/3] . . . 832
14.80.3.3 operator+ [3/3] . . . 832
14.80.4 Member Data Documentation . . . 833
14.80.4.1 npos . . . 833
14.81 GrabInfo Struct Reference . . . 833
14.81.1 Constructor & Destructor Documentation . . . 833
14.81.1.1 GrabInfo() . . . 833
14.81.2 Member Data Documentation . . . 833
14.81.2.1 imageEventHandler . . . 833
14.81.2.2 numImagesGrabbed . . . 834
14.81.2.3 numIncompleteImages . . . 834
14.81.2.4 numRemovals . . . 834
14.82 GVCP_CHUNK_TRAILER Struct Reference . . . 834
14.82.1 Detailed Description . . . 834
14.82.2 Member Data Documentation . . . 834
14.82.2.1 ChunkID . . . 834

14.82.2.2 ChunkLength . . . 835
14.83 GVCP_EVENT_ITEM Struct Reference . . . 835
14.83.1 Detailed Description . . . 835
14.83.2 Member Data Documentation . . . 835
14.83.2.1 BlockId . . . 835
14.83.2.2 EventId . . . 835
14.83.2.3 ReservedOrEventSize . . . 836
14.83.2.4 StreamChannelId . . . 836
14.83.2.5 TimestampHigh . . . 836
14.83.2.6 TimestampLow . . . 836
14.84 GVCP_EVENT_ITEM_BASIC Struct Reference . . . 836
14.84.1 Detailed Description . . . 836
14.84.2 Member Data Documentation . . . 836
14.84.2.1 EventId . . . 837
14.84.2.2 ReservedOrEventSize . . . 837
14.85 GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference . . . 837
14.85.1 Detailed Description . . . 837
14.85.2 Member Data Documentation . . . 837
14.85.2.1 BlockId . . . 837
14.85.2.2 BlockId64High . . . 838
14.85.2.3 BlockId64Low . . . 838
14.85.2.4 EventId . . . 838
14.85.2.5 ReservedOrEventSize . . . 838
14.85.2.6 StreamChannelId . . . 838
14.85.2.7 TimestampHigh . . . 838
14.85.2.8 TimestampLow . . . 838
14.86 GVCP_EVENT_REQUEST Struct Reference . . . 839
14.86.1 Detailed Description . . . 839
14.86.2 Member Data Documentation . . . 839
14.86.2.1 Header . . . 839

14.86.2.2 Items . . . 839
14.87 GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference . . . 840
14.87.1 Detailed Description . . . 840
14.87.2 Member Data Documentation . . . 840
14.87.2.1 Header . . . 840
14.87.2.2 Items . . . 840
14.88 GVCP_EVENTDATA_REQUEST Struct Reference . . . 841
14.88.1 Detailed Description . . . 841
14.88.2 Member Data Documentation . . . 841
14.88.2.1 Data . . . 841
14.88.2.2 Event . . . 841
14.88.2.3 Header . . . 842
14.89 GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference . . . 842
14.89.1 Detailed Description . . . 842
14.89.2 Member Data Documentation . . . 842
14.89.2.1 Data . . . 842
14.89.2.2 Event . . . 843
14.89.2.3 Header . . . 843
14.90 GVCP_REQUEST_HEADER Struct Reference . . . 843
14.90.1 Detailed Description . . . 843
14.90.2 Member Data Documentation . . . 843
14.90.2.1 Command . . . 843
14.90.2.2 Flags . . . 843
14.90.2.3 Length . . . 844
14.90.2.4 Magic . . . 844
14.90.2.5 ReqId . . . 844
14.91 H264Option Struct Reference . . . 844
14.91.1 Detailed Description . . . 844
14.91.2 Constructor & Destructor Documentation . . . 845
14.91.2.1 H264Option() . . . 845

14.91.3 Member Data Documentation . . . 845
14.91.3.1 bitrate . . . 845
14.91.3.2 frameRate . . . 845
14.91.3.3 height . . . 845
14.91.3.4 reserved . . . 845
14.91.3.5 width . . . 846
14.92 ICameraBase Class Reference . . . 846
14.92.1 Detailed Description . . . 848
14.92.2 Constructor & Destructor Documentation . . . 848
14.92.2.1 ~ICameraBase() . . . 848
14.92.2.2 ICameraBase() [1/2] . . . 848
14.92.2.3 ICameraBase() [2/2] . . . 848
14.92.3 Member Function Documentation . . . 848
14.92.3.1 BeginAcquisition() . . . 848
14.92.3.2 DeInit() . . . 849
14.92.3.3 DiscoverMaxPacketSize() . . . 849
14.92.3.4 EndAcquisition() . . . 849
14.92.3.5 ForceIP() . . . 849
14.92.3.6 GetAccessMode() . . . 849
14.92.3.7 GetBufferOwnership() . . . 849
14.92.3.8 GetGuiXml() . . . 850
14.92.3.9 GetNextImage() . . . 850
14.92.3.10 GetNodeMap() . . . 850
14.92.3.11 GetNumDataStreams() . . . 850
14.92.3.12 GetNumImagesInUse() . . . 850
14.92.3.13 GetTLDeviceNodeMap() . . . 850
14.92.3.14 GetTLStreamNodeMap() . . . 851
14.92.3.15 GetUniqueID() . . . 851
14.92.3.16 GetUserBufferCount() . . . 851
14.92.3.17 GetUserBufferSize() . . . 851

14.92.3.18 GetUserBufferTotalSize() . . . 851
14.92.3.19 Init() . . . 851
14.92.3.20 IsInitialized() . . . 852
14.92.3.21 IsStreaming() . . . 852
14.92.3.22 IsValid() . . . 852
14.92.3.23 operator=() . . . 852
14.92.3.24 ReadPort() . . . 852
14.92.3.25 RegisterEventHandler() [1/2] . . . 852
14.92.3.26 RegisterEventHandler() [2/2] . . . 853
14.92.3.27 SetBufferOwnership() . . . 853
14.92.3.28 SetUserBuffers() [1/2] . . . 853
14.92.3.29 SetUserBuffers() [2/2] . . . 853
14.92.3.30 UnregisterEventHandler() . . . 853
14.92.3.31 WritePort() . . . 854
14.92.4 Friends And Related Function Documentation . . . 854
14.92.4.1 CameraInternal . . . 854
14.92.4.2 InterfaceImpl . . . 854
14.92.5 Member Data Documentation . . . 854
14.92.5.1 m_pCameraBaseData . . . 854
14.92.5.2 TLDevice . . . 854
14.92.5.3 TLStream . . . 855
14.93 ICameraList Class Reference . . . 855
14.93.1 Detailed Description . . . 856
14.93.2 Constructor & Destructor Documentation . . . 856
14.93.2.1 ~ICameraList() . . . 856
14.93.2.2 ICameraList() [1/2] . . . 856
14.93.2.3 ICameraList() [2/2] . . . 856
14.93.3 Member Function Documentation . . . 856
14.93.3.1 Append() . . . 856
14.93.3.2 Clear() . . . 857

14.93.3.3 GetByDeviceID() . . . 857
14.93.3.4 GetByIndex() . . . 857
14.93.3.5 GetBySerial() . . . 857
14.93.3.6 GetSize() . . . 857
14.93.3.7 operator=() . . . 857
14.93.3.8 operator[]() . . . 858
14.93.3.9 RemoveByDeviceID() . . . 858
14.93.3.10 RemoveByIndex() . . . 858
14.93.3.11 RemoveBySerial() . . . 858
14.93.4 Friends And Related Function Documentation . . . 858
14.93.4.1 CameraListImpl . . . 858
14.93.4.2 InterfaceImpl . . . 858
14.93.5 Member Data Documentation . . . 859
14.93.5.1 m_pCameraListData . . . 859
14.94 IChunkData Class Reference . . . 859
14.94.1 Detailed Description . . . 860
14.94.2 Constructor & Destructor Documentation . . . 860
14.94.2.1 ~IChunkData() . . . 860
14.94.2.2 IChunkData() . . . 860
14.94.3 Member Function Documentation . . . 861
14.94.3.1 GetBlackLevel() . . . 861
14.94.3.2 GetCounterValue() . . . 861
14.94.3.3 GetCRC() . . . 861
14.94.3.4 GetEncoderValue() . . . 861
14.94.3.5 GetExposureEndLineStatusAll() . . . 861
14.94.3.6 GetExposureTime() . . . 862
14.94.3.7 GetFrameID() . . . 862
14.94.3.8 GetGain() . . . 862
14.94.3.9 GetHeight() . . . 862
14.94.3.10 GetImage() . . . 862

14.94.3.11 GetInferenceBoundingBoxResult() . . . 862
14.94.3.12 GetInferenceConfidence() . . . 863
14.94.3.13 GetInferenceFrameId() . . . 863
14.94.3.14 GetInferenceResult() . . . 863
14.94.3.15 GetLinePitch() . . . 863
14.94.3.16 GetLineStatusAll() . . . 863
14.94.3.17 GetOffsetX() . . . 863
14.94.3.18 GetOffsetY() . . . 864
14.94.3.19 GetPartSelector() . . . 864
14.94.3.20 GetPixelDynamicRangeMax() . . . 864
14.94.3.21 GetPixelDynamicRangeMin() . . . 864
14.94.3.22 GetScan3dAxisMax() . . . 864
14.94.3.23 GetScan3dAxisMin() . . . 864
14.94.3.24 GetScan3dCoordinateOffset() . . . 865
14.94.3.25 GetScan3dCoordinateReferenceValue() . . . 865
14.94.3.26 GetScan3dCoordinateScale() . . . 865
14.94.3.27 GetScan3dInvalidDataValue() . . . 865
14.94.3.28 GetScan3dTransformValue() . . . 865
14.94.3.29 GetScanLineSelector() . . . 865
14.94.3.30 GetSequencerSetActive() . . . 866
14.94.3.31 GetSerialDataLength() . . . 866
14.94.3.32 GetStreamChannelID() . . . 866
14.94.3.33 GetTimerValue() . . . 866
14.94.3.34 GetTimestamp() . . . 866
14.94.3.35 GetTimestampLatchValue() . . . 866
14.94.3.36 GetTransferBlockID() . . . 867
14.94.3.37 GetTransferQueueCurrentBlockCount() . . . 867
14.94.3.38 GetWidth() . . . 867
14.94.3.39 SetChunks() . . . 867
14.95 IDataStream Class Reference . . . 868

14.95.1 Constructor & Destructor Documentation . . . 868
14.95.1.1 ~IDataStream() . . . 869
14.95.1.2 IDataStream() . . . 869
14.95.2 Member Function Documentation . . . 869
14.95.2.1 AnnounceImage() [1/3] . . . 869
14.95.2.2 AnnounceImage() [2/3] . . . 869
14.95.2.3 AnnounceImage() [3/3] . . . 869
14.95.2.4 AttachBuffer() . . . 869
14.95.2.5 CleanupChunkAdapter() . . . 870
14.95.2.6 FlushQueueAllDiscard() . . . 870
14.95.2.7 GetBufferChunkData() . . . 870
14.95.2.8 GetBufferInfoBool8Type() . . . 870
14.95.2.9 GetBufferInfoPtrType() . . . 870
14.95.2.10 GetBufferInfoSizeType() . . . 870
14.95.2.11 GetBufferInfoUInt64Type() . . . 871
14.95.2.12 GetDeviceNodeMap() . . . 871
14.95.2.13 GetNextImage() . . . 871
14.95.2.14 GetNextImageInternal() . . . 871
14.95.2.15 GetNodeMap() . . . 871
14.95.2.16 GetNumImagesInUse() . . . 871
14.95.2.17 GetPort() . . . 871
14.95.2.18 GetStreamInfoBool8Type() . . . 872
14.95.2.19 GetStreamInfoSizeType() . . . 872
14.95.2.20 GetStreamType() . . . 872
14.95.2.21 InitChunkAdapter() . . . 872
14.95.2.22 IsCRCCheckEnabled() . . . 872
14.95.2.23 IsImageInUse() . . . 872
14.95.2.24 IsStreaming() . . . 872
14.95.2.25 KillBufferEvent() . . . 873
14.95.2.26 RegisterImageEventHandler() . . . 873

14.95.2.27 ReleaseImage() . . . 873
14.95.2.28 ResetStreamHandle() . . . 873
14.95.2.29 RevokeImages() . . . 873
14.95.2.30 StartStream() . . . 873
14.95.2.31 StopStream() . . . 873
14.95.2.32 TransportLayerStreamInfo() . . . 874
14.95.2.33 UnregisterImageEventHandler() . . . 874
14.95.2.34 WaitOnImageEvent() . . . 874
14.96 IDevFileStreamBase< CharType, Traits > Class Template Reference . . . 874
14.96.1 Member Typedef Documentation . . . 875
14.96.1.1 filebuf_type . . . 875
14.96.1.2 ios_type . . . 875
14.96.1.3 istream_type . . . 876
14.96.2 Member Function Documentation . . . 876
14.96.2.1 close() . . . 876
14.96.2.2 is_open() . . . 876
14.96.2.3 open() . . . 876
14.96.2.4 rdbuf() . . . 876
14.97 IDevFileStreamBuf< CharType, Traits > Class Template Reference . . . 877
14.97.1 Constructor & Destructor Documentation . . . 878
14.97.1.1 IDevFileStreamBuf() . . . 878
14.97.1.2 ~IDevFileStreamBuf() . . . 878
14.97.2 Member Function Documentation . . . 878
14.97.2.1 close() . . . 878
14.97.2.2 is_open() . . . 878
14.97.2.3 open() . . . 878
14.97.2.4 pbackfail() . . . 878
14.97.2.5 underflow() . . . 879
14.98 IDeviceArrivalEventHandler Class Reference . . . 879
14.98.1 Constructor & Destructor Documentation . . . 880

14.98.1.1 ~IDeviceArrivalEventHandler() . . . 880
14.98.1.2 IDeviceArrivalEventHandler() [1/2] . . . 880
14.98.1.3 IDeviceArrivalEventHandler() [2/2] . . . 880
14.98.2 Member Function Documentation . . . 880
14.98.2.1 OnDeviceArrival() . . . 880
14.98.2.2 operator=() . . . 881
14.99 IDeviceEventHandler Class Reference . . . 881
14.99.1 Constructor & Destructor Documentation . . . 882
14.99.1.1 ~IDeviceEventHandler() . . . 882
14.99.1.2 IDeviceEventHandler() [1/2] . . . 882
14.99.1.3 IDeviceEventHandler() [2/2] . . . 882
14.99.2 Member Function Documentation . . . 882
14.99.2.1 GetDeviceEventId() . . . 882
14.99.2.2 GetDeviceEventName() . . . 883
14.99.2.3 OnDeviceEvent() . . . 883
14.99.2.4 operator=() . . . 883
14.100 IDeviceRemovalEventHandler Class Reference . . . 883
14.100.1 Constructor & Destructor Documentation . . . 884
14.100.1.1 ~IDeviceRemovalEventHandler() . . . 884
14.100.1.2 IDeviceRemovalEventHandler() [1/2] . . . 884
14.100.1.3 IDeviceRemovalEventHandler() [2/2] . . . 885
14.100.2 Member Function Documentation . . . 885
14.100.2.1 OnDeviceRemoval() . . . 885
14.100.2.2 operator=() . . . 885
14.101 IImage Class Reference . . . 885
14.101.1 Detailed Description . . . 887
14.101.2 Constructor & Destructor Documentation . . . 887
14.101.2.1 ~IImage() . . . 887
14.101.2.2 IImage() . . . 887
14.101.3 Member Function Documentation . . . 887

14.101.3.1 CalculateStatistics() . . . 887
14.101.3.2 CheckCRC() . . . 887
14.101.3.3 Convert() [1/2] . . . 888
14.101.3.4 Convert() [2/2] . . . 888
14.101.3.5 DeepCopy() . . . 888
14.101.3.6 GetBitsPerPixel() . . . 888
14.101.3.7 GetBufferSize() . . . 888
14.101.3.8 GetChunkData() . . . 889
14.101.3.9 GetChunkLayoutId() . . . 889
14.101.3.10 GetColorProcessing() . . . 889
14.101.3.11 GetData() . . . 889
14.101.3.12 GetDataAbsoluteMax() . . . 889
14.101.3.13 GetDataAbsoluteMin() . . . 889
14.101.3.14 GetFrameID() . . . 890
14.101.3.15 GetHeight() . . . 890
14.101.3.16 GetID() . . . 890
14.101.3.17 GetImageData() . . . 890
14.101.3.18 GetImageSize() . . . 890
14.101.3.19 GetImageStatus() . . . 890
14.101.3.20 GetNumChannels() . . . 891
14.101.3.21 GetPayloadType() . . . 891
14.101.3.22 GetPixelFormat() . . . 891
14.101.3.23 GetPixelFormatIntType() . . . 891
14.101.3.24 GetPixelFormatName() . . . 891
14.101.3.25 GetPrivateData() . . . 891
14.101.3.26 GetStride() . . . 892
14.101.3.27 GetTimeStamp() . . . 892
14.101.3.28 GetTLPayloadType() . . . 892
14.101.3.29 GetTLPixelFormat() . . . 892
14.101.3.30 GetTLPixelFormatNamespace() . . . 892

14.101.3.31 GetValidPayloadSize() . . . 892
14.101.3.32 GetWidth() . . . 893
14.101.3.33 GetXOffset() . . . 893
14.101.3.34 GetXPadding() . . . 893
14.101.3.35 GetYOffset() . . . 893
14.101.3.36 GetYPadding() . . . 893
14.101.3.37 HasCRC() . . . 893
14.101.3.38 IsIncomplete() . . . 894
14.101.3.39 IsInUse() . . . 894
14.101.3.40 Release() . . . 894
14.101.3.41 ResetImage() [1/2] . . . 894
14.101.3.42 ResetImage() [2/2] . . . 894
14.101.3.43 Save() [1/8] . . . 895
14.101.3.44 Save() [2/8] . . . 895
14.101.3.45 Save() [3/8] . . . 895
14.101.3.46 Save() [4/8] . . . 895
14.101.3.47 Save() [5/8] . . . 895
14.101.3.48 Save() [6/8] . . . 896
14.101.3.49 Save() [7/8] . . . 896
14.101.3.50 Save() [8/8] . . . 896
14.101.4 Friends And Related Function Documentation . . . 896
14.101.4.1 Stream . . . 896
14.102 IImageEventHandler Class Reference . . . 897
14.102.1 Constructor & Destructor Documentation . . . 898
14.102.1.1 ~IImageEventHandler() . . . 898
14.102.1.2 IImageEventHandler() [1/2] . . . 898
14.102.1.3 IImageEventHandler() [2/2] . . . 898
14.102.2 Member Function Documentation . . . 898
14.102.2.1 OnImageEvent() . . . 898
14.102.2.2 operator=() . . . 898

14.103 IImageStatistics Class Reference . . . 899
14.103.1 Detailed Description . . . 899
14.103.2 Constructor & Destructor Documentation . . . 900
14.103.2.1 ~IImageStatistics() . . . 900
14.103.2.2 IImageStatistics() [1/2] . . . 900
14.103.2.3 IImageStatistics() [2/2] . . . 900
14.103.3 Member Function Documentation . . . 900
14.103.3.1 DisableAll() . . . 900
14.103.3.2 EnableAll() . . . 900
14.103.3.3 EnableGreyOnly() . . . 900
14.103.3.4 EnableHSLOnly() . . . 901
14.103.3.5 EnableRGBOnly() . . . 901
14.103.3.6 GetChannelStatus() . . . 901
14.103.3.7 GetHistogram() . . . 901
14.103.3.8 GetMean() . . . 901
14.103.3.9 GetNumPixelValues() . . . 902
14.103.3.10 GetPixelValueRange() . . . 902
14.103.3.11 GetRange() . . . 902
14.103.3.12 GetStatistics() . . . 902
14.103.3.13 SetChannelStatus() . . . 903
14.104 IInterface Class Reference . . . 903
14.104.1 Detailed Description . . . 904
14.104.2 Constructor & Destructor Documentation . . . 904
14.104.2.1 ~IInterface() . . . 904
14.104.2.2 IInterface() [1/2] . . . 905
14.104.2.3 IInterface() [2/2] . . . 905
14.104.3 Member Function Documentation . . . 905
14.104.3.1 GetCameras() . . . 905
14.104.3.2 GetTLNodeMap() . . . 905
14.104.3.3 IsInUse() . . . 905

14.104.3.4 IsValid() . . . 905
14.104.3.5 operator=() . . . 906
14.104.3.6 RegisterEventHandler() . . . 906
14.104.3.7 SendActionCommand() . . . 906
14.104.3.8 UnregisterEventHandler() . . . 906
14.104.3.9 UpdateCameras() . . . 906
14.104.4 Friends And Related Function Documentation . . . 906
14.104.4.1 IInterfaceInternal . . . 907
14.104.4.2 ProducerImpl . . . 907
14.104.5 Member Data Documentation . . . 907
14.104.5.1 m_pInterfaceData . . . 907
14.104.5.2 TLInterface . . . 907
14.105 IInterfaceArrivalEventHandler Class Reference . . . 908
14.105.1 Constructor & Destructor Documentation . . . 909
14.105.1.1 ~IInterfaceArrivalEventHandler() . . . 909
14.105.1.2 IInterfaceArrivalEventHandler() [1/2] . . . 909
14.105.1.3 IInterfaceArrivalEventHandler() [2/2] . . . 909
14.105.2 Member Function Documentation . . . 909
14.105.2.1 OnInterfaceArrival() . . . 909
14.105.2.2 operator=() . . . 910
14.106 IInterfaceEventHandler Class Reference . . . 910
14.106.1 Constructor & Destructor Documentation . . . 911
14.106.1.1 ~IInterfaceEventHandler() . . . 911
14.106.1.2 IInterfaceEventHandler() [1/2] . . . 911
14.106.1.3 IInterfaceEventHandler() [2/2] . . . 912
14.106.2 Member Function Documentation . . . 912
14.106.2.1 OnDeviceArrival() . . . 912
14.106.2.2 OnDeviceRemoval() . . . 912
14.106.2.3 operator=() . . . 912
14.107 IInterfaceList Class Reference . . . 913

14.107.1 Detailed Description . . . 913
14.107.2 Constructor & Destructor Documentation . . . 914
14.107.2.1 ~IInterfaceList() . . . 914
14.107.2.2 IInterfaceList() [1/2] . . . 914
14.107.2.3 IInterfaceList() [2/2] . . . 914
14.107.3 Member Function Documentation . . . 914
14.107.3.1 Append() . . . 914
14.107.3.2 Clear() . . . 914
14.107.3.3 GetByIndex() . . . 915
14.107.3.4 GetSize() . . . 915
14.107.3.5 operator=() . . . 915
14.107.3.6 operator[]() . . . 915
14.107.4 Friends And Related Function Documentation . . . 915
14.107.4.1 IInterfaceListImpl . . . 915
14.107.5 Member Data Documentation . . . 915
14.107.5.1 m_pInterfaceListData . . . 916
14.108 IInterfaceRemovalEventHandler Class Reference . . . 916
14.108.1 Constructor & Destructor Documentation . . . 917
14.108.1.1 ~IInterfaceRemovalEventHandler() . . . 917
14.108.1.2 IInterfaceRemovalEventHandler() [1/2] . . . 917
14.108.1.3 IInterfaceRemovalEventHandler() [2/2] . . . 917
14.108.2 Member Function Documentation . . . 917
14.108.2.1 OnInterfaceRemoval() . . . 917
14.108.2.2 operator=() . . . 918
14.109 ILoggingEventHandler Class Reference . . . 918
14.109.1 Constructor & Destructor Documentation . . . 919
14.109.1.1 ~ILoggingEventHandler() . . . 919
14.109.1.2 ILoggingEventHandler() [1/2] . . . 919
14.109.1.3 ILoggingEventHandler() [2/2] . . . 919
14.109.2 Member Function Documentation . . . 919

14.109.2.1 OnLogEvent() . . . 919
14.109.2.2 operator=() . . . 920
14.110 Image Class Reference . . . 920
14.110.1 Detailed Description . . . 923
14.110.2 Constructor & Destructor Documentation . . . 924
14.110.2.1 ~Image() . . . 924
14.110.2.2 Image() [1/3] . . . 924
14.110.2.3 Image() [2/3] . . . 924
14.110.2.4 Image() [3/3] . . . 924
14.110.3 Member Function Documentation . . . 924
14.110.3.1 CalculateStatistics() . . . 924
14.110.3.2 CheckCRC() . . . 925
14.110.3.3 Convert() [1/3] . . . 925
14.110.3.4 Convert() [2/3] . . . 926
14.110.3.5 Convert() [3/3] . . . 926
14.110.3.6 Create() [1/3] . . . 926
14.110.3.7 Create() [2/3] . . . 926
14.110.3.8 Create() [3/3] . . . 927
14.110.3.9 CreateShared() . . . 927
14.110.3.10 DeepCopy() [1/2] . . . 927
14.110.3.11 DeepCopy() [2/2] . . . 928
14.110.3.12 GetBitsPerPixel() . . . 928
14.110.3.13 GetBufferSize() . . . 928
14.110.3.14 GetChunkData() . . . 928
14.110.3.15 GetChunkLayoutId() . . . 929
14.110.3.16 GetColorProcessing() . . . 929
14.110.3.17 GetData() . . . 929
14.110.3.18 GetDataAbsoluteMax() . . . 930
14.110.3.19 GetDataAbsoluteMin() . . . 930
14.110.3.20 GetDefaultColorProcessing() . . . 930

14.110.3.21 GetFrameID() . . . 931
14.110.3.22 GetHeight() . . . 931
14.110.3.23 GetID() . . . 931
14.110.3.24 GetImageData() . . . 931
14.110.3.25 GetImageSize() . . . 932
14.110.3.26 GetImageStatus() . . . 932
14.110.3.27 GetImageStatusDescription() . . . 932
14.110.3.28 GetNumChannels() . . . 932
14.110.3.29 GetPayloadType() . . . 933
14.110.3.30 GetPixelFormat() . . . 933
14.110.3.31 GetPixelFormatIntType() . . . 933
14.110.3.32 GetPixelFormatName() . . . 934
14.110.3.33 GetPrivateData() . . . 934
14.110.3.34 GetStride() . . . 934
14.110.3.35 GetTimeStamp() . . . 935
14.110.3.36 GetTLPayloadType() . . . 935
14.110.3.37 GetTLPixelFormat() . . . 935
14.110.3.38 GetTLPixelFormatNamespace() . . . 936
14.110.3.39 GetValidPayloadSize() . . . 936
14.110.3.40 GetWidth() . . . 936
14.110.3.41 GetXOffset() . . . 937
14.110.3.42 GetXPadding() . . . 937
14.110.3.43 GetYOffset() . . . 937
14.110.3.44 GetYPadding() . . . 938
14.110.3.45 HasCRC() . . . 938
14.110.3.46 IsCompressed() . . . 938
14.110.3.47 IsIncomplete() . . . 938
14.110.3.48 IsInUse() . . . 939
14.110.3.49 Release() . . . 939
14.110.3.50 ResetImage() [1/2] . . . 939

14.110.3.51 ResetImage() [2/2] . . . 939
14.110.3.52 Save() [1/8] . . . 940
14.110.3.53 Save() [2/8] . . . 940
14.110.3.54 Save() [3/8] . . . 941
14.110.3.55 Save() [4/8] . . . 941
14.110.3.56 Save() [5/8] . . . 941
14.110.3.57 Save() [6/8] . . . 942
14.110.3.58 Save() [7/8] . . . 942
14.110.3.59 Save() [8/8] . . . 942
14.110.3.60 SetDefaultColorProcessing() . . . 943
14.110.4 Friends And Related Function Documentation . . . 943
14.110.4.1 IDataStream . . . 943
14.110.4.2 ImageConverter . . . 943
14.110.4.3 ImageFiler . . . 943
14.110.4.4 ImageStatsCalculator . . . 944
14.110.4.5 ImageUtilityImpl . . . 944
14.110.4.6 ImageUtilityPolarizationImpl . . . 944
14.110.4.7 Stream . . . 944
14.111 ImageEventHandler Class Reference . . . 944
14.111.1 Detailed Description . . . 945
14.111.2 Constructor & Destructor Documentation . . . 945
14.111.2.1 ImageEventHandler() . . . 946
14.111.2.2 ~ImageEventHandler() . . . 946
14.111.3 Member Function Documentation . . . 946
14.111.3.1 OnImageEvent() . . . 946
14.111.3.2 operator=() . . . 946
14.112 ImageEventHandlerImpl Class Reference . . . 947
14.112.1 Constructor & Destructor Documentation . . . 948
14.112.1.1 ImageEventHandlerImpl() [1/2] . . . 948
14.112.1.2 ~ImageEventHandlerImpl() [1/2] . . . 948

14.112.1.3 ImageEventHandlerImpl() [2/2] . . . 948
14.112.1.4 ~ImageEventHandlerImpl() [2/2] . . . 948
14.112.2 Member Function Documentation . . . 948
14.112.2.1 getImageCount() . . . 949
14.112.2.2 getMaxImages() . . . 949
14.112.2.3 OnImageEvent() [1/2] . . . 949
14.112.2.4 OnImageEvent() [2/2] . . . 949
14.113 ImagePtr Class Reference . . . 950
14.113.1 Detailed Description . . . 951
14.113.2 Constructor & Destructor Documentation . . . 951
14.113.2.1 ImagePtr() [1/4] . . . 951
14.113.2.2 ImagePtr() [2/4] . . . 951
14.113.2.3 ImagePtr() [3/4] . . . 951
14.113.2.4 ImagePtr() [4/4] . . . 951
14.113.2.5 ~ImagePtr() . . . 952
14.113.3 Member Function Documentation . . . 952
14.113.3.1 operator=() . . . 952
14.114 ImageStatistics Class Reference . . . 952
14.114.1 Detailed Description . . . 953
14.114.2 Constructor & Destructor Documentation . . . 954
14.114.2.1 ImageStatistics() [1/2] . . . 954
14.114.2.2 ~ImageStatistics() . . . 954
14.114.2.3 ImageStatistics() [2/2] . . . 954
14.114.3 Member Function Documentation . . . 954
14.114.3.1 DisableAll() . . . 954
14.114.3.2 EnableAll() . . . 954
14.114.3.3 EnableGreyOnly() . . . 955
14.114.3.4 EnableHSLOnly() . . . 955
14.114.3.5 EnableRGBOnly() . . . 955
14.114.3.6 GetChannelStatus() . . . 955

14.114.3.7 GetHistogram() . . . 956
14.114.3.8 GetMean() . . . 956
14.114.3.9 GetNumPixelValues() . . . 956
14.114.3.10 GetPixelValueRange() . . . 957
14.114.3.11 GetRange() . . . 957
14.114.3.12 GetStatistics() . . . 957
14.114.3.13 operator=() . . . 958
14.114.3.14 SetChannelStatus() . . . 958
14.114.4 Friends And Related Function Documentation . . . 959
14.114.4.1 ImageStatsCalculator . . . 959
14.115 ImageUtility Class Reference . . . 959
14.115.1 Detailed Description . . . 960
14.115.2 Member Enumeration Documentation . . . 960
14.115.2.1 ImageScalingAlgorithm . . . 960
14.115.2.2 SourceDataRange . . . 960
14.115.3 Member Function Documentation . . . 961
14.115.3.1 CreateNormalized() [1/5] . . . 961
14.115.3.2 CreateNormalized() [2/5] . . . 961
14.115.3.3 CreateNormalized() [3/5] . . . 962
14.115.3.4 CreateNormalized() [4/5] . . . 962
14.115.3.5 CreateNormalized() [5/5] . . . 963
14.115.3.6 CreateScaled() [1/2] . . . 963
14.115.3.7 CreateScaled() [2/2] . . . 964
14.116 ImageUtilityHeatmap Class Reference . . . 964
14.116.1 Detailed Description . . . 965
14.116.2 Member Enumeration Documentation . . . 965
14.116.2.1 HeatmapColor . . . 965
14.116.3 Member Function Documentation . . . 965
14.116.3.1 CreateHeatmap() [1/2] . . . 965
14.116.3.2 CreateHeatmap() [2/2] . . . 966

14.116.3.3 GetHeatmapColorGradient() . . . 966
14.116.3.4 GetHeatmapRange() . . . 967
14.116.3.5 SetHeatmapColorGradient() . . . 967
14.116.3.6 SetHeatmapRange() . . . 968
14.117 ImageUtilityPolarization Class Reference . . . 968
14.117.1 Detailed Description . . . 969
14.117.2 Member Enumeration Documentation . . . 969
14.117.2.1 PolarizationQuadrant . . . 969
14.117.3 Member Function Documentation . . . 970
14.117.3.1 CreateAolp() [1/2] . . . 970
14.117.3.2 CreateAolp() [2/2] . . . 970
14.117.3.3 CreateDolp() [1/2] . . . 971
14.117.3.4 CreateDolp() [2/2] . . . 971
14.117.3.5 CreateGlareReduced() [1/2] . . . 971
14.117.3.6 CreateGlareReduced() [2/2] . . . 972
14.117.3.7 CreateStokesS0() [1/2] . . . 972
14.117.3.8 CreateStokesS0() [2/2] . . . 973
14.117.3.9 CreateStokesS1() [1/2] . . . 973
14.117.3.10 CreateStokesS1() [2/2] . . . 973
14.117.3.11 CreateStokesS2() [1/2] . . . 974
14.117.3.12 CreateStokesS2() [2/2] . . . 974
14.117.3.13 ExtractPolarQuadrant() [1/2] . . . 975
14.117.3.14 ExtractPolarQuadrant() [2/2] . . . 975
14.118 InferenceBoundingBox Struct Reference . . . 976
14.118.1 Detailed Description . . . 976
14.119 InferenceBoundingBoxResult Class Reference . . . 976
14.119.1 Detailed Description . . . 977
14.120 InferenceBoxCircle Struct Reference . . . 977
14.121 InferenceBoxRect Struct Reference . . . 977
14.121.1 Detailed Description . . . 977

14.122 InferenceBoxRotatedRect Struct Reference . . . 978
14.123 int64_autovector_t Class Reference . . . 978
14.123.1 Detailed Description . . . 978
14.123.2 Constructor & Destructor Documentation . . . 978
14.123.2.1 int64_autovector_t() [1/3] . . . 979
14.123.2.2 int64_autovector_t() [2/3] . . . 979
14.123.2.3 int64_autovector_t() [3/3] . . . 979
14.123.2.4 ~int64_autovector_t() . . . 979
14.123.3 Member Function Documentation . . . 979
14.123.3.1 operator delete() . . . 979
14.123.3.2 operator new() . . . 979
14.123.3.3 operator=() . . . 980
14.123.3.4 operator[]() [1/2] . . . 980
14.123.3.5 operator[]() [2/2] . . . 980
14.123.3.6 size() . . . 980
14.123.4 Member Data Documentation . . . 980
14.123.4.1 _pCount . . . 980
14.123.4.2 _pv . . . 980
14.124 IntegerNode Class Reference . . . 981
14.124.1 Detailed Description . . . 983
14.124.2 Constructor & Destructor Documentation . . . 983
14.124.2.1 IntegerNode() [1/2] . . . 983
14.124.2.2 IntegerNode() [2/2] . . . 983
14.124.2.3 ~IntegerNode() . . . 983
14.124.3 Member Function Documentation . . . 983
14.124.3.1 GetFloatAlias() . . . 984
14.124.3.2 GetInc() . . . 984
14.124.3.3 GetIncMode() . . . 984
14.124.3.4 GetListOfValidValues() . . . 984
14.124.3.5 GetMax() . . . 984

14.124.3.6 GetMin() . . . 984
14.124.3.7 GetRepresentation() . . . 985
14.124.3.8 GetUnit() . . . 985
14.124.3.9 GetValue() . . . 985
14.124.3.10 ImposeMax() . . . 985
14.124.3.11 ImposeMin() . . . 985
14.124.3.12 operator()() . . . 986
14.124.3.13 operator∗() . . . 986
14.124.3.14 operator=() . . . 986
14.124.3.15 SetReference() . . . 986
14.124.3.16 SetValue() . . . 986
14.125 Interface Class Reference . . . 987
14.125.1 Detailed Description . . . 988
14.125.2 Constructor & Destructor Documentation . . . 988
14.125.2.1 ∼Interface() . . . 988
14.125.3 Member Function Documentation . . . 988
14.125.3.1 GetCameras() . . . 989
14.125.3.2 GetTLNodeMap() . . . 989
14.125.3.3 IsInUse() . . . 989
14.125.3.4 IsValid() . . . 990
14.125.3.5 RegisterEventHandler() . . . 990
14.125.3.6 SendActionCommand() . . . 990
14.125.3.7 UnregisterEventHandler() . . . 991
14.125.3.8 UpdateCameras() . . . 991
14.125.4 Friends And Related Function Documentation . . . 992
14.125.4.1 InterfaceInternal . . . 992
14.126 InterfaceArrivalEventHandler Class Reference . . . 992
14.126.1 Detailed Description . . . 993
14.126.2 Constructor & Destructor Documentation . . . 993
14.126.2.1 InterfaceArrivalEventHandler() . . . 994

14.126.2.2 ~InterfaceArrivalEventHandler() . . . 994
14.126.3 Member Function Documentation . . . 994
14.126.3.1 OnInterfaceArrival() . . . 994
14.126.3.2 operator=() . . . 994
14.127 InterfaceEventHandler Class Reference . . . 995
14.127.1 Detailed Description . . . 996
14.127.2 Constructor & Destructor Documentation . . . 996
14.127.2.1 InterfaceEventHandler() . . . 996
14.127.2.2 ~InterfaceEventHandler() . . . 996
14.127.3 Member Function Documentation . . . 996
14.127.3.1 OnDeviceArrival() . . . 997
14.127.3.2 OnDeviceRemoval() . . . 997
14.127.3.3 operator=() . . . 997
14.128 InterfaceEventHandlerImpl Class Reference . . . 998
14.128.1 Constructor & Destructor Documentation . . . 999
14.128.1.1 InterfaceEventHandlerImpl() [1/3] . . . 1000
14.128.1.2 ~InterfaceEventHandlerImpl() [1/2] . . . 1000
14.128.1.3 InterfaceEventHandlerImpl() [2/3] . . . 1000
14.128.1.4 InterfaceEventHandlerImpl() [3/3] . . . 1000
14.128.1.5 ~InterfaceEventHandlerImpl() [2/2] . . . 1000
14.128.2 Member Function Documentation . . . 1000
14.128.2.1 GetInterfaceId() . . . 1000
14.128.2.2 OnDeviceArrival() [1/2] . . . 1001
14.128.2.3 OnDeviceArrival() [2/2] . . . 1001
14.128.2.4 OnDeviceRemoval() [1/2] . . . 1001
14.128.2.5 OnDeviceRemoval() [2/2] . . . 1001
14.128.2.6 PrintGenericHandlerMessage() . . . 1002
14.129 InterfaceList Class Reference . . . 1002
14.129.1 Detailed Description . . . 1003
14.129.2 Constructor & Destructor Documentation . . . 1003

14.129.2.1 InterfaceList() [1/2] . . . 1003
14.129.2.2 ~InterfaceList() . . . 1003
14.129.2.3 InterfaceList() [2/2] . . . 1004
14.129.3 Member Function Documentation . . . 1004
14.129.3.1 Append() . . . 1004
14.129.3.2 Clear() . . . 1004
14.129.3.3 GetByIndex() . . . 1004
14.129.3.4 GetSize() . . . 1005
14.129.3.5 operator=() . . . 1005
14.129.3.6 operator[]() . . . 1005
14.129.4 Friends And Related Function Documentation . . . 1005
14.129.4.1 ProducerImpl . . . 1006
14.129.4.2 SystemImpl . . . 1006
14.130 InterfacePtr Class Reference . . . 1006
14.130.1 Detailed Description . . . 1007
14.130.2 Constructor & Destructor Documentation . . . 1007
14.130.2.1 InterfacePtr() [1/4] . . . 1007
14.130.2.2 InterfacePtr() [2/4] . . . 1007
14.130.2.3 InterfacePtr() [3/4] . . . 1007
14.130.2.4 InterfacePtr() [4/4] . . . 1008
14.131 InterfaceRemovalEventHandler Class Reference . . . 1008
14.131.1 Detailed Description . . . 1009
14.131.2 Constructor & Destructor Documentation . . . 1009
14.131.2.1 InterfaceRemovalEventHandler() . . . 1009
14.131.2.2 ~InterfaceRemovalEventHandler() . . . 1009
14.131.3 Member Function Documentation . . . 1009
14.131.3.1 OnInterfaceRemoval() . . . 1009
14.131.3.2 operator=() . . . 1010
14.132 IntRegNode Class Reference . . . 1010
14.132.1 Detailed Description . . . 1011

14.132.2 Constructor & Destructor Documentation . . . 1011
14.132.2.1 IntRegNode() [1/2] . . . 1012
14.132.2.2 IntRegNode() [2/2] . . . 1012
14.132.2.3 ~IntRegNode() . . . 1012
14.132.3 Member Function Documentation . . . 1012
14.132.3.1 SetReference() . . . 1012
14.133 IpInfo Struct Reference . . . 1012
14.133.1 Constructor & Destructor Documentation . . . 1013
14.133.1.1 IpInfo() . . . 1013
14.133.2 Member Data Documentation . . . 1013
14.133.2.1 gateway . . . 1013
14.133.2.2 ipAddress . . . 1013
14.133.2.3 subnetLength . . . 1013
14.133.2.4 subnetMask . . . 1013
14.134 ISystem Class Reference . . . 1014
14.134.1 Detailed Description . . . 1015
14.134.2 Constructor & Destructor Documentation . . . 1015
14.134.2.1 ~ISystem() . . . 1015
14.134.2.2 ISystem() [1/2] . . . 1015
14.134.2.3 ISystem() [2/2] . . . 1016
14.134.3 Member Function Documentation . . . 1016
14.134.3.1 GetCameras() . . . 1016
14.134.3.2 GetInterfaces() . . . 1016
14.134.3.3 GetLibraryVersion() . . . 1016
14.134.3.4 GetLoggingEventPriorityLevel() . . . 1016
14.134.3.5 GetTLNodeMap() . . . 1017
14.134.3.6 IsInUse() . . . 1017
14.134.3.7 operator=() . . . 1017
14.134.3.8 RegisterEventHandler() . . . 1017
14.134.3.9 RegisterInterfaceEventHandler() . . . 1017

14.134.3.18 RegisterLoggingEventHandler() . . . 1017
14.134.3.19 ReleaseInstance() . . . 1018
14.134.3.20 SendActionCommand() . . . 1018
14.134.3.21 SetLoggingEventPriorityLevel() . . . 1018
14.134.3.22 UnregisterAllLoggingEventHandlers() . . . 1018
14.134.3.23 UnregisterEventHandler() . . . 1018
14.134.3.24 UnregisterInterfaceEventHandler() . . . 1019
14.134.3.25 UnregisterLoggingEventHandler() . . . 1019
14.134.3.26 UpdateCameras() . . . 1019
14.134.3.27 UpdateInterfaceList() . . . 1019
14.134.4 Friends And Related Function Documentation . . . 1019
14.134.4.1 SystemPtrInternal . . . 1019
14.134.5 Member Data Documentation . . . 1019
14.134.5.1 TLSystem . . . 1020
14.135 ISystemEventHandler Class Reference . . . 1020
14.135.1 Constructor & Destructor Documentation . . . 1021
14.135.1.1 ~ISystemEventHandler() . . . 1021
14.135.1.2 ISystemEventHandler() [1/2] . . . 1021
14.135.1.3 ISystemEventHandler() [2/2] . . . 1021
14.135.2 Member Function Documentation . . . 1021
14.135.2.1 OnInterfaceArrival() . . . 1021
14.135.2.2 OnInterfaceRemoval() . . . 1022
14.135.2.3 operator=() . . . 1022
14.136 JPEGOption Struct Reference . . . 1022
14.136.1 Detailed Description . . . 1022
14.136.2 Constructor & Destructor Documentation . . . 1022
14.136.2.1 JPEGOption() . . . 1023
14.136.3 Member Data Documentation . . . 1023
14.136.3.1 progressive . . . 1023
14.136.3.2 quality . . . 1023

14.136.3.3 reserved . . . 1023
14.137 JPG2Option Struct Reference . . . 1023
14.137.1 Detailed Description . . . 1024
14.137.2 Constructor & Destructor Documentation . . . 1024
14.137.2.1 JPG2Option() . . . 1024
14.137.3 Member Data Documentation . . . 1024
14.137.3.1 quality . . . 1024
14.137.3.2 reserved . . . 1024
14.138 LibraryVersion Struct Reference . . . 1025
14.138.1 Detailed Description . . . 1025
14.138.2 Member Data Documentation . . . 1025
14.138.2.1 build . . . 1025
14.138.2.2 major . . . 1025
14.138.2.3 minor . . . 1025
14.138.2.4 type . . . 1026
14.139 LockableObject< Object >::Lock Class Reference . . . 1026
14.139.1 Detailed Description . . . 1026
14.139.2 Constructor & Destructor Documentation . . . 1026
14.139.2.1 Lock() . . . 1026
14.139.2.2 ~Lock() . . . 1026
14.140 LockableObject< Object > Class Template Reference . . . 1027
14.140.1 Detailed Description . . . 1027
14.140.2 Member Function Documentation . . . 1028
14.140.2.1 GetLock() . . . 1028
14.140.3 Friends And Related Function Documentation . . . 1028
14.140.3.1 Lock . . . 1028
14.140.4 Member Data Documentation . . . 1028
14.140.4.1 m_Lock . . . 1028
14.141 LoggingEventData Class Reference . . . 1028
14.141.1 Detailed Description . . . 1029

14.141.2 Constructor & Destructor Documentation . . . 1029
14.141.2.1 ~LoggingEventData() . . . 1029
14.141.2.2 LoggingEventData() . . . 1030
14.141.3 Member Function Documentation . . . 1030
14.141.3.1 GetCategoryName() . . . 1030
14.141.3.2 GetLogMessage() . . . 1030
14.141.3.3 GetNDC() . . . 1030
14.141.3.4 GetPriority() . . . 1031
14.141.3.5 GetPriorityName() . . . 1031
14.141.3.6 GetThreadName() . . . 1031
14.141.3.7 GetTimestamp() . . . 1031
14.141.4 Friends And Related Function Documentation . . . 1031
14.141.4.1 SystemImpl . . . 1032
14.142 LoggingEventDataPtr Class Reference . . . 1032
14.142.1 Detailed Description . . . 1033
14.142.2 Constructor & Destructor Documentation . . . 1033
14.142.2.1 LoggingEventDataPtr() [1/4] . . . 1033
14.142.2.2 LoggingEventDataPtr() [2/4] . . . 1033
14.142.2.3 LoggingEventDataPtr() [3/4] . . . 1033
14.142.2.4 LoggingEventDataPtr() [4/4] . . . 1033
14.143 LoggingEventHandler Class Reference . . . 1034
14.143.1 Detailed Description . . . 1035
14.143.2 Constructor & Destructor Documentation . . . 1035
14.143.2.1 LoggingEventHandler() . . . 1035
14.143.2.2 ~LoggingEventHandler() . . . 1035
14.143.3 Member Function Documentation . . . 1035
14.143.3.1 OnLogEvent() . . . 1035
14.143.3.2 operator=() . . . 1036
14.144 LoggingEventHandlerImpl Class Reference . . . 1036
14.145 Member_NodeCallback< Client, Member > Class Template Reference . . . 1037

14.145.1 Detailed Description . . . 1038
14.145.2 Member Typedef Documentation . . . 1038
14.145.2.1 PMEMBERFUNC . . . 1038
14.145.3 Constructor & Destructor Documentation . . . 1039
14.145.3.1 Member_NodeCallback() . . . 1039
14.145.4 Member Function Documentation . . . 1039
14.145.4.1 Destroy() . . . 1039
14.145.4.2 operator()() . . . 1039
14.146 MJPGOption Struct Reference . . . 1039
14.146.1 Detailed Description . . . 1040
14.146.2 Constructor & Destructor Documentation . . . 1040
14.146.2.1 MJPGOption() . . . 1040
14.146.3 Member Data Documentation . . . 1040
14.146.3.1 frameRate . . . 1040
14.146.3.2 quality . . . 1040
14.146.3.3 reserved . . . 1041
14.147 Node Class Reference . . . 1041
14.147.1 Detailed Description . . . 1043
14.147.2 Constructor & Destructor Documentation . . . 1044
14.147.2.1 Node() [1/2] . . . 1044
14.147.2.2 Node() [2/2] . . . 1044
14.147.2.3 ~Node() . . . 1044
14.147.3 Member Function Documentation . . . 1044
14.147.3.1 DeregisterCallback() . . . 1044
14.147.3.2 GetAccessMode() . . . 1045
14.147.3.3 GetAlias() . . . 1045
14.147.3.4 GetCachingMode() . . . 1045
14.147.3.5 GetCastAlias() . . . 1045
14.147.3.6 GetChildren() . . . 1045
14.147.3.7 GetDescription() . . . 1046

14.147.3.8 GetDeviceName() . . . 1046
14.147.3.9 GetDisplayName() . . . 1046
14.147.3.10 GetDocuURL() . . . 1046
14.147.3.11 GetEventID() . . . 1046
14.147.3.12 GetName() . . . 1046
14.147.3.13 GetNameSpace() . . . 1047
14.147.3.14 GetNodeHandle() . . . 1047
14.147.3.15 GetNodeMap() . . . 1047
14.147.3.16 GetParents() . . . 1047
14.147.3.17 GetPollingTime() . . . 1047
14.147.3.18 GetPrincipalInterfaceType() . . . 1048
14.147.3.19 GetProperty() . . . 1048
14.147.3.20 GetPropertyNames() . . . 1048
14.147.3.21 GetSelectedFeatures() . . . 1048
14.147.3.22 GetSelectingFeatures() . . . 1048
14.147.3.23 GetToolTip() . . . 1049
14.147.3.24 GetVisibility() . . . 1049
14.147.3.25 ImposeAccessMode() . . . 1049
14.147.3.26 ImposeVisibility() . . . 1049
14.147.3.27 InvalidateNode() . . . 1049
14.147.3.28 IsAccessModeCacheable() . . . 1049
14.147.3.29 IsCachable() . . . 1050
14.147.3.30 IsDeprecated() . . . 1050
14.147.3.31 IsFeature() . . . 1050
14.147.3.32 IsSelector() . . . 1050
14.147.3.33 IsStreamable() . . . 1050
14.147.3.34 operator"!=() . . . 1050
14.147.3.35 operator==() . . . 1051
14.147.3.36 RegisterCallback() . . . 1051
14.147.3.37 SetNodeHandle() . . . 1051

14.147.3.38 SetNodeMap() . . . 1051
14.147.3.39 SetReference() [1/2] . . . 1051
14.147.3.40 SetReference() [2/2] . . . 1051
14.147.4 Member Data Documentation . . . 1052
14.147.4.1 m_Callbacks . . . 1052
14.147.4.2 m_pNodeData . . . 1052
14.147.4.3 m_pNodeMap . . . 1052
14.148 NodeMap Class Reference . . . 1052
14.148.1 Detailed Description . . . 1054
14.148.2 Constructor & Destructor Documentation . . . 1054
14.148.2.1 NodeMap() . . . 1055
14.148.2.2 ~NodeMap() . . . 1055
14.148.3 Member Function Documentation . . . 1055
14.148.3.1 ClearXMLCache() . . . 1055
14.148.3.2 Connect() [1/2] . . . 1055
14.148.3.3 Connect() [2/2] . . . 1055
14.148.3.4 Destroy() . . . 1056
14.148.3.5 GetDeviceName() . . . 1056
14.148.3.6 GetDeviceVersion() . . . 1056
14.148.3.7 GetGenApiVersion() . . . 1056
14.148.3.8 GetLock() . . . 1056
14.148.3.9 GetModelName() . . . 1056
14.148.3.10 GetNode() . . . 1057
14.148.3.11 GetNodeMapHandle() . . . 1057
14.148.3.12 GetNodes() . . . 1057
14.148.3.13 GetNumNodes() . . . 1057
14.148.3.14 GetProductGuid() . . . 1057
14.148.3.15 GetSchemaVersion() . . . 1057
14.148.3.16 GetStandardNameSpace() . . . 1058
14.148.3.17 GetSupportedSchemaVersions() . . . 1058

14.148.3.18 GetToolTip() . . . 1058
14.148.3.19 GetVendorName() . . . 1058
14.148.3.20 GetVersionGuid() . . . 1059
14.148.3.21 InvalidateNodes() . . . 1059
14.148.3.22 LoadXMLFromFile() . . . 1059
14.148.3.23 LoadXMLFromFileInject() . . . 1059
14.148.3.24 LoadXMLFromString() . . . 1059
14.148.3.25 LoadXMLFromStringInject() . . . 1060
14.148.3.26 LoadXMLFromZIPData() . . . 1060
14.148.3.27 LoadXMLFromZIPFile() . . . 1060
14.148.3.28 Poll() . . . 1060
14.148.4 Member Data Documentation . . . 1060
14.148.4.1 _Ptr . . . 1060
14.149 CNodeMapFactory::NodeStatistics_t Struct Reference . . . 1061
14.149.1 Member Data Documentation . . . 1061
14.149.1.1 NumLinks . . . 1061
14.149.1.2 NumNodes . . . 1061
14.149.1.3 NumProperties . . . 1061
14.149.1.4 NumStrings . . . 1061
14.150 COutDevFileStreamBase< CharType, Traits > Class Template Reference . . . 1062
14.150.1 Member Typedef Documentation . . . 1063
14.150.1.1 filebuf_type . . . 1063
14.150.1.2 ios_type . . . 1063
14.150.1.3 ostream_type . . . 1063
14.150.2 Member Function Documentation . . . 1063
14.150.2.1 close() . . . 1063
14.150.2.2 is_open() . . . 1063
14.150.2.3 open() . . . 1063
14.150.2.4 rdbuf() . . . 1064
14.151 COutDevFileStreamBuf< CharType, Traits > Class Template Reference . . . 1064

14.151.1 Constructor & Destructor Documentation . . . 1065
14.151.1.1 ODevFileStreamBuf() . . . 1065
14.151.1.2 ~ODevFileStreamBuf() . . . 1065
14.151.2 Member Function Documentation . . . 1065
14.151.2.1 close() . . . 1065
14.151.2.2 is_open() . . . 1065
14.151.2.3 open() . . . 1066
14.151.2.4 overflow() . . . 1066
14.151.2.5 sync() . . . 1066
14.151.2.6 xsputn() . . . 1066
14.152 PGMOption Struct Reference . . . 1066
14.152.1 Detailed Description . . . 1067
14.152.2 Constructor & Destructor Documentation . . . 1067
14.152.2.1 PGMOption() . . . 1067
14.152.3 Member Data Documentation . . . 1067
14.152.3.1 binaryFile . . . 1067
14.152.3.2 reserved . . . 1067
14.153 PNGOption Struct Reference . . . 1067
14.153.1 Detailed Description . . . 1068
14.153.2 Constructor & Destructor Documentation . . . 1068
14.153.2.1 PNGOption() . . . 1068
14.153.3 Member Data Documentation . . . 1068
14.153.3.1 compressionLevel . . . 1068
14.153.3.2 interlaced . . . 1068
14.153.3.3 reserved . . . 1069
14.154 PortNode Class Reference . . . 1069
14.154.1 Detailed Description . . . 1071
14.154.2 Constructor & Destructor Documentation . . . 1071
14.154.2.1 PortNode() [1/2] . . . 1071
14.154.2.2 PortNode() [2/2] . . . 1071

14.154.2.3 ~PortNode() . . . 1071
14.154.3 Member Function Documentation . . . 1071
14.154.3.1 CacheChunkData() . . . 1071
14.154.3.2 GetChunkID() . . . 1072
14.154.3.3 GetPortHandle() . . . 1072
14.154.3.4 GetSwapEndianess() . . . 1072
14.154.3.5 Read() . . . 1072
14.154.3.6 Replay() . . . 1072
14.154.3.7 SetPortImpl() . . . 1073
14.154.3.8 SetReference() [1/3] . . . 1073
14.154.3.9 SetReference() [2/3] . . . 1073
14.154.3.10 SetReference() [3/3] . . . 1073
14.154.3.11 StartRecording() . . . 1073
14.154.3.12 StopRecording() . . . 1074
14.154.3.13 Write() . . . 1074
14.155 PortRecorder Class Reference . . . 1074
14.155.1 Detailed Description . . . 1075
14.155.2 Constructor & Destructor Documentation . . . 1076
14.155.2.1 PortRecorder() . . . 1076
14.155.2.2 ~PortRecorder() . . . 1076
14.155.3 Member Function Documentation . . . 1076
14.155.3.1 GetAccessMode() . . . 1076
14.155.3.2 Read() . . . 1076
14.155.3.3 Replay() . . . 1077
14.155.3.4 SetReference() . . . 1077
14.155.3.5 StartRecording() . . . 1077
14.155.3.6 StopRecording() . . . 1077
14.155.3.7 Write() . . . 1078
14.156 PortReplay Class Reference . . . 1078
14.156.1 Detailed Description . . . 1079

14.156.2 Constructor & Destructor Documentation . . . 1079
14.156.2.1 PortReplay() . . . 1080
14.156.2.2 ~PortReplay() . . . 1080
14.156.3 Member Function Documentation . . . 1080
14.156.3.1 GetAccessMode() . . . 1080
14.156.3.2 GetPortReplayHandle() . . . 1080
14.156.3.3 Read() . . . 1080
14.156.3.4 Replay() . . . 1081
14.156.3.5 SetReference() . . . 1081
14.156.3.6 Write() . . . 1081
14.157 PPMOption Struct Reference . . . 1081
14.157.1 Detailed Description . . . 1082
14.157.2 Constructor & Destructor Documentation . . . 1082
14.157.2.1 PPMOption() . . . 1082
14.157.3 Member Data Documentation . . . 1082
14.157.3.1 binaryFile . . . 1082
14.157.3.2 reserved . . . 1082
14.158 RegisterNode Class Reference . . . 1083
14.158.1 Detailed Description . . . 1084
14.158.2 Constructor & Destructor Documentation . . . 1084
14.158.2.1 RegisterNode() [1/2] . . . 1085
14.158.2.2 RegisterNode() [2/2] . . . 1085
14.158.2.3 ~RegisterNode() . . . 1085
14.158.3 Member Function Documentation . . . 1085
14.158.3.1 Get() . . . 1085
14.158.3.2 GetAddress() . . . 1085
14.158.3.3 GetLength() . . . 1086
14.158.3.4 Set() . . . 1086
14.158.3.5 SetReference() . . . 1086
14.159 SingleChunkData_t Struct Reference . . . 1086

14.159.1 Member Data Documentation . . . 1087
14.159.1.1 ChunkID . . . 1087
14.159.1.2 ChunkLength . . . 1087
14.159.1.3 ChunkOffset . . . 1087
14.160 SingleChunkDataStr_t Struct Reference . . . 1087
14.160.1 Member Data Documentation . . . 1087
14.160.1.1 ChunkID . . . 1087
14.160.1.2 ChunkLength . . . 1088
14.160.1.3 ChunkOffset . . . 1088
14.161 SpinTestCamera Class Reference . . . 1088
14.162 SpinVideo Class Reference . . . 1089
14.162.1 Detailed Description . . . 1089
14.162.2 Constructor & Destructor Documentation . . . 1089
14.162.2.1 SpinVideo() . . . 1089
14.162.2.2 ~SpinVideo() . . . 1090
14.162.3 Member Function Documentation . . . 1090
14.162.3.1 Append() . . . 1090
14.162.3.2 Close() . . . 1090
14.162.3.3 Open() [1/3] . . . 1090
14.162.3.4 Open() [2/3] . . . 1091
14.162.3.5 Open() [3/3] . . . 1091
14.162.3.6 SetMaximumFileSize() . . . 1092
14.163 StringNode Class Reference . . . 1092
14.163.1 Detailed Description . . . 1094
14.163.2 Constructor & Destructor Documentation . . . 1094
14.163.2.1 StringNode() [1/2] . . . 1094
14.163.2.2 StringNode() [2/2] . . . 1094
14.163.2.3 ~StringNode() . . . 1094
14.163.3 Member Function Documentation . . . 1095
14.163.3.1 GetMaxLength() . . . 1095

14.163.3.2 GetValue() . . . 1095
14.163.3.3 operator()() . . . 1095
14.163.3.4 operator∗() . . . 1095
14.163.3.5 operator=() . . . 1096
14.163.3.6 SetReference() . . . 1096
14.163.3.7 SetValue() . . . 1096
14.164 StringRegNode Class Reference . . . 1096
14.164.1 Detailed Description . . . 1098
14.164.2 Constructor & Destructor Documentation . . . 1098
14.164.2.1 StringRegNode() [1/2] . . . 1099
14.164.2.2 StringRegNode() [2/2] . . . 1099
14.164.2.3 ∼StringRegNode() . . . 1099
14.164.3 Member Function Documentation . . . 1099
14.164.3.1 SetReference() . . . 1099
14.165 System Class Reference . . . 1100
14.165.1 Detailed Description . . . 1101
14.165.2 Constructor & Destructor Documentation . . . 1102
14.165.2.1 ∼System() . . . 1102
14.165.2.2 System() . . . 1102
14.165.3 Member Function Documentation . . . 1102
14.165.3.1 GetCameras() . . . 1102
14.165.3.2 GetInstance() . . . 1103
14.165.3.3 GetInterfaces() . . . 1103
14.165.3.4 GetLibraryVersion() . . . 1104
14.165.3.5 GetLoggingEventPriorityLevel() . . . 1104
14.165.3.6 GetTLNodeMap() . . . 1104
14.165.3.7 IsInUse() . . . 1105
14.165.3.8 RegisterEventHandler() . . . 1105
14.165.3.9 RegisterInterfaceEventHandler() . . . 1105
14.165.3.10 RegisterLoggingEventHandler() . . . 1106

14.165.3.12 ReleaseInstance() . . . 1106
14.165.3.13 SendActionCommand() . . . 1106
14.165.3.14 SetLoggingEventPriorityLevel() . . . 1107
14.165.3.15 UnregisterAllLoggingEventHandlers() . . . 1108
14.165.3.16 UnregisterEventHandler() . . . 1108
14.165.3.17 UnregisterInterfaceEventHandler() . . . 1108
14.165.3.18 UnregisterLoggingEventHandler() . . . 1109
14.165.3.19 UpdateCameras() . . . 1109
14.165.3.20 UpdateInterfaceList() . . . 1109
14.166 SystemEventHandler Class Reference . . . 1110
14.166.1 Detailed Description . . . 1111
14.166.2 Constructor & Destructor Documentation . . . 1111
14.166.2.1 SystemEventHandler() . . . 1112
14.166.2.2 ~SystemEventHandler() . . . 1112
14.166.3 Member Function Documentation . . . 1112
14.166.3.1 OnInterfaceArrival() . . . 1112
14.166.3.2 OnInterfaceRemoval() . . . 1112
14.166.3.3 operator=() . . . 1113
14.167 SystemEventHandlerImpl Class Reference . . . 1113
14.167.1 Constructor & Destructor Documentation . . . 1114
14.167.1.1 SystemEventHandlerImpl() . . . 1114
14.167.1.2 ~SystemEventHandlerImpl() . . . 1115
14.167.2 Member Function Documentation . . . 1115
14.167.2.1 LockEventHandlerMutex() . . . 1115
14.167.2.2 OnInterfaceArrival() . . . 1115
14.167.2.3 OnInterfaceRemoval() . . . 1115
14.167.2.4 RegisterAllInterfaceEvents() . . . 1116
14.167.2.5 RegisterInterfaceEventToSystem() . . . 1116
14.167.2.6 UnlockEventHandlerMutex() . . . 1116
14.167.2.7 UnregisterAllInterfaceEvents() . . . 1116

14.167.2.8 UnregisterInterfaceEventFromSystem() . . . 1116
14.168 SystemPtr Class Reference . . . 1117
14.168.1 Detailed Description . . . 1118
14.168.2 Constructor & Destructor Documentation . . . 1118
14.168.2.1 SystemPtr() [1/4] . . . 1118
14.168.2.2 SystemPtr() [2/4] . . . 1118
14.168.2.3 SystemPtr() [3/4] . . . 1118
14.168.2.4 SystemPtr() [4/4] . . . 1118
14.168.2.5 ~SystemPtr() . . . 1119
14.169 TIFFOption Struct Reference . . . 1119
14.169.1 Detailed Description . . . 1119
14.169.2 Member Enumeration Documentation . . . 1119
14.169.2.1 CompressionMethod . . . 1119
14.169.3 Constructor & Destructor Documentation . . . 1120
14.169.3.1 TIFFOption() . . . 1120
14.169.4 Member Data Documentation . . . 1120
14.169.4.1 compression . . . 1120
14.169.4.2 reserved . . . 1120
14.170 TransportLayerDevice Class Reference . . . 1121
14.170.1 Detailed Description . . . 1123
14.170.2 Constructor & Destructor Documentation . . . 1123
14.170.2.1 TransportLayerDevice() [1/2] . . . 1123
14.170.2.2 ~TransportLayerDevice() . . . 1123
14.170.2.3 TransportLayerDevice() [2/2] . . . 1123
14.170.3 Friends And Related Function Documentation . . . 1123
14.170.3.1 CameraBase . . . 1123
14.170.3.2 CameraInternal . . . 1124
14.170.3.3 ICameraBase . . . 1124
14.170.4 Member Data Documentation . . . 1124
14.170.4.1 DeviceAccessStatus . . . 1124

14.170.4.2 DeviceCurrentSpeed 1124
14.170.4.3 DeviceDisplayName 1124
14.170.4.4 DeviceDriverVersion 1124
14.170.4.5 DeviceEndianessMechanism 1125
14.170.4.6 DeviceID 1125
14.170.4.7 DeviceInstanceId 1125
14.170.4.8 DeviceIsUpdater 1125
14.170.4.9 DeviceLinkSpeed 1125
14.170.4.10 DeviceLocation 1125
14.170.4.11 DeviceModelName 1126
14.170.4.12 DeviceMulticastMonitorMode 1126
14.170.4.13 DeviceSerialNumber 1126
14.170.4.14 DeviceType 1126
14.170.4.15 DeviceU3VProtocol 1126
14.170.4.16 DeviceUserID 1126
14.170.4.17 DeviceVendorName 1127
14.170.4.18 DeviceVersion 1127
14.170.4.19 GenICamXMLLocation 1127
14.170.4.20 GenICamXMLPath 1127
14.170.4.21 GevCCP 1127
14.170.4.22 GevDeviceAutoForceIP 1127
14.170.4.23 GevDeviceDiscoverMaximumPacketSize 1128
14.170.4.24 GevDeviceForceGateway 1128
14.170.4.25 GevDeviceForceIP 1128
14.170.4.26 GevDeviceForceIPAddress 1128
14.170.4.27 GevDeviceForceSubnetMask 1128
14.170.4.28 GevDeviceGateway 1128
14.170.4.29 GevDeviceIPAddress 1129
14.170.4.30 GevDeviceIsWrongSubnet 1129
14.170.4.31 GevDeviceMACAddress 1129

14.170.4.32 GevDeviceMaximumPacketSize . . . 1129
14.170.4.33 GevDeviceMaximumRetryCount . . . 1129
14.170.4.34 GevDeviceModeIsBigEndian . . . 1129
14.170.4.35 GevDevicePort . . . 1130
14.170.4.36 GevDeviceReadAndWriteTimeout . . . 1130
14.170.4.37 GevDeviceSubnetMask . . . 1130
14.170.4.38 GevVersionMajor . . . 1130
14.170.4.39 GevVersionMinor . . . 1130
14.170.4.40 GUIXMLLocation . . . 1130
14.170.4.41 GUIXMLPath . . . 1131
14.171 TransportLayerInterface Class Reference . . . 1131
14.171.1 Detailed Description . . . 1133
14.171.2 Constructor & Destructor Documentation . . . 1133
14.171.2.1 TransportLayerInterface() [1/2] . . . 1133
14.171.2.2 ~TransportLayerInterface() . . . 1134
14.171.2.3 TransportLayerInterface() [2/2] . . . 1134
14.171.3 Friends And Related Function Documentation . . . 1134
14.171.3.1 IIInterface . . . 1134
14.171.3.2 Interface . . . 1134
14.171.3.3 InterfaceInternal . . . 1134
14.171.4 Member Data Documentation . . . 1134
14.171.4.1 ActionCommand . . . 1134
14.171.4.2 DeviceAccessStatus . . . 1135
14.171.4.3 DeviceCount . . . 1135
14.171.4.4 DeviceID . . . 1135
14.171.4.5 DeviceModelName . . . 1135
14.171.4.6 DeviceSelector . . . 1135
14.171.4.7 DeviceSerialNumber . . . 1135
14.171.4.8 DeviceUnlock . . . 1136
14.171.4.9 DeviceUpdateList . . . 1136

14.171.4.11 DeviceVendorName . . . 1136
14.171.4.12 FilterDriverStatus . . . 1136
14.171.4.13 GevActionDeviceKey . . . 1136
14.171.4.14 GevActionGroupKey . . . 1136
14.171.4.15 GevActionGroupMask . . . 1137
14.171.4.16 GevActionTime . . . 1137
14.171.4.17 GevDeviceAutoForceIP . . . 1137
14.171.4.18 GevDeviceForceGateway . . . 1137
14.171.4.19 GevDeviceForceIP . . . 1137
14.171.4.20 GevDeviceForceIPAddress . . . 1137
14.171.4.21 GevDeviceForceSubnetMask . . . 1138
14.171.4.22 GevDeviceGateway . . . 1138
14.171.4.23 GevDeviceIPAddress . . . 1138
14.171.4.24 GevDeviceMACAddress . . . 1138
14.171.4.25 GevDeviceSubnetMask . . . 1138
14.171.4.26 GevInterfaceGateway . . . 1138
14.171.4.27 GevInterfaceGatewaySelector . . . 1139
14.171.4.28 GevInterfaceMACAddress . . . 1139
14.171.4.29 GevInterfaceMTU . . . 1139
14.171.4.30 GevInterfaceReceiveLinkSpeed . . . 1139
14.171.4.31 GevInterfaceSubnetIPAddress . . . 1139
14.171.4.32 GevInterfaceSubnetMask . . . 1139
14.171.4.33 GevInterfaceSubnetSelector . . . 1140
14.171.4.34 GevInterfaceTransmitLinkSpeed . . . 1140
14.171.4.35 HostAdapterDriverVersion . . . 1140
14.171.4.36 HostAdapterName . . . 1140
14.171.4.37 HostAdapterVendor . . . 1140
14.171.4.38 IncompatibleDeviceCount . . . 1140
14.171.4.39 IncompatibleDeviceID . . . 1141
14.171.4.40 IncompatibleDeviceModelName . . . 1141

14.171.4.40 IncompatibleDeviceSelector . . . 1141
14.171.4.41 IncompatibleDeviceVendorName . . . 1141
14.171.4.42 IncompatibleGevDeviceIPAddress . . . 1141
14.171.4.43 IncompatibleGevDeviceMACAddress . . . 1141
14.171.4.44 IncompatibleGevDeviceSubnetMask . . . 1142
14.171.4.45 InterfaceDisplayName . . . 1142
14.171.4.46 InterfaceID . . . 1142
14.171.4.47 InterfaceType . . . 1142
14.171.4.48 POEStatus . . . 1142
14.172 TransportLayerStream Class Reference . . . 1142
14.172.1 Detailed Description . . . 1144
14.172.2 Constructor & Destructor Documentation . . . 1144
14.172.2.1 TransportLayerStream() [1/2] . . . 1144
14.172.2.2 ~TransportLayerStream() . . . 1144
14.172.2.3 TransportLayerStream() [2/2] . . . 1144
14.172.3 Friends And Related Function Documentation . . . 1145
14.172.3.1 CameraBase . . . 1145
14.172.3.2 CameraInternal . . . 1145
14.172.3.3 ICameraBase . . . 1145
14.172.4 Member Data Documentation . . . 1145
14.172.4.1 GevFailedPacketCount . . . 1145
14.172.4.2 GevMaximumNumberResendRequests . . . 1145
14.172.4.3 GevPacketResendMode . . . 1145
14.172.4.4 GevPacketResendTimeout . . . 1146
14.172.4.5 GevResendPacketCount . . . 1146
14.172.4.6 GevResendRequestCount . . . 1146
14.172.4.7 GevTotalPacketCount . . . 1146
14.172.4.8 StreamAnnounceBufferMinimum . . . 1146
14.172.4.9 StreamAnnouncedBufferCount . . . 1146
14.172.4.10 StreamBlockTransferSize . . . 1147

14.172.4.13 StreamBufferAlignment . . . 1147
14.172.4.14 StreamBufferCountManual . . . 1147
14.172.4.15 StreamBufferCountMax . . . 1147
14.172.4.16 StreamBufferCountMode . . . 1147
14.172.4.17 StreamBufferCountResult . . . 1147
14.172.4.18 StreamBufferHandlingMode . . . 1148
14.172.4.19 StreamChunkCountMaximum . . . 1148
14.172.4.20 StreamCRCCheckEnable . . . 1148
14.172.4.21 StreamDeliveredFrameCount . . . 1148
14.172.4.22 StreamFailedBufferCount . . . 1148
14.172.4.23 StreamID . . . 1148
14.172.4.24 StreamInputBufferCount . . . 1149
14.172.4.25 StreamIsGrabbing . . . 1149
14.172.4.26 StreamLostFrameCount . . . 1149
14.172.4.27 StreamOutputBufferCount . . . 1149
14.172.4.28 StreamStartedFrameCount . . . 1149
14.172.4.29 StreamType . . . 1150
14.173 TransportLayerSystem Class Reference . . . 1150
14.173.1 Detailed Description . . . 1151
14.173.2 Constructor & Destructor Documentation . . . 1151
14.173.2.1 TransportLayerSystem() [1/2] . . . 1152
14.173.2.2 ~TransportLayerSystem() . . . 1152
14.173.2.3 TransportLayerSystem() [2/2] . . . 1152
14.173.3 Friends And Related Function Documentation . . . 1152
14.173.3.1 IISystem . . . 1152
14.173.3.2 System . . . 1152
14.173.3.3 SystemPtrInternal . . . 1152
14.173.4 Member Data Documentation . . . 1152
14.173.4.1 EnumerateGEVInterfaces . . . 1153
14.173.4.2 GenTLSFNCVersionMajor . . . 1153

14.173.4.3 GenTLSFNCVersionMinor . . . 1153
14.173.4.4 GenTLSFNCVersionSubMinor . . . 1153
14.173.4.5 GenTLVersionMajor . . . 1153
14.173.4.6 GenTLVersionMinor . . . 1153
14.173.4.7 GevInterfaceDefaultGateway . . . 1154
14.173.4.8 GevInterfaceDefaultIPAddress . . . 1154
14.173.4.9 GevInterfaceDefaultSubnetMask . . . 1154
14.173.4.10 GevInterfaceMACAddress . . . 1154
14.173.4.11 GevVersionMajor . . . 1154
14.173.4.12 GevVersionMinor . . . 1154
14.173.4.13 InterfaceDisplayName . . . 1155
14.173.4.14 InterfaceID . . . 1155
14.173.4.15 InterfaceSelector . . . 1155
14.173.4.16 InterfaceUpdateList . . . 1155
14.173.4.17 TLDisplayName . . . 1155
14.173.4.18 TLFileName . . . 1155
14.173.4.19 TLID . . . 1156
14.173.4.20 TLModelName . . . 1156
14.173.4.21 TLPath . . . 1156
14.173.4.22 TLType . . . 1156
14.173.4.23 TLVendorName . . . 1156
14.173.4.24 TLVersion . . . 1156
14.174 U3V_CHUNK_TRAILER Struct Reference . . . 1157
14.174.1 Detailed Description . . . 1157
14.174.2 Member Data Documentation . . . 1157
14.174.2.1 ChunkID . . . 1157
14.174.2.2 ChunkLength . . . 1157
14.175 U3V_COMMAND_HEADER Struct Reference . . . 1157
14.175.1 Detailed Description . . . 1158
14.175.2 Member Data Documentation . . . 1158

14.175.2.1 CommandId . . . 1158
14.175.2.2 Flags . . . 1158
14.175.2.3 Length . . . 1158
14.175.2.4 Prefix . . . 1158
14.175.2.5 ReqId . . . 1158
14.176 U3V_EVENT_DATA Struct Reference . . . 1158
14.176.1 Detailed Description . . . 1159
14.176.2 Member Data Documentation . . . 1159
14.176.2.1 EventId . . . 1159
14.176.2.2 Reserved . . . 1159
14.176.2.3 Timestamp . . . 1159
14.177 U3V_EVENT_MESSAGE Struct Reference . . . 1159
14.177.1 Detailed Description . . . 1160
14.177.2 Member Data Documentation . . . 1160
14.177.2.1 CommandHeader . . . 1160
14.177.2.2 EventData . . . 1160
14.178 ValueNode Class Reference . . . 1160
14.178.1 Detailed Description . . . 1161
14.178.2 Constructor & Destructor Documentation . . . 1161
14.178.2.1 ValueNode() [1/2] . . . 1162
14.178.2.2 ValueNode() [2/2] . . . 1162
14.178.2.3 ~ValueNode() . . . 1162
14.178.3 Member Function Documentation . . . 1162
14.178.3.1 FromString() . . . 1162
14.178.3.2 GetNode() . . . 1162
14.178.3.3 IsValueCacheValid() . . . 1163
14.178.3.4 SetReference() . . . 1163
14.178.3.5 ToString() . . . 1163
14.179 Version_t Struct Reference . . . 1163
14.179.1 Detailed Description . . . 1164
14.179.2 Member Data Documentation . . . 1164
14.179.2.1 Major . . . 1164
14.179.2.2 Minor . . . 1164
14.179.2.3 SubMinor . . . 1164

15 File Documentation 1165

15.1 doc/spindocs/C++/GettingStarted.dox File Reference . . . 1165

15.2 doc/spindocs/C++/ProgrammerGuide.dox File Reference . . . 1165

15.3 doc/spindocs/shared/Benefits.dox File Reference . . . 1165

15.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference . . . 1165

15.5 doc/spindocs/shared/GenICamGenTL.dox File Reference . . . 1165

15.6 doc/spindocs/shared/Licensing.dox File Reference . . . 1165

15.7 include/AdapterConfig.h File Reference . . . 1165

15.7.1 Macro Definition Documentation . . . 1167

15.7.1.1 ADAPTERCONFIG_API . . . 1167

15.8 include/AVIRecorder.h File Reference . . . 1167

15.9 include/BasePtr.h File Reference . . . 1168

15.10 include/Camera.h File Reference . . . 1169

15.11 include/CameraBase.h File Reference . . . 1169

15.12 include/CameraDefs.h File Reference . . . 1170

15.13 include/CameraList.h File Reference . . . 1202

15.14 include/CameraPtr.h File Reference . . . 1203

15.15 include/ChunkData.h File Reference . . . 1203

15.16 include/ChunkDataInference.h File Reference . . . 1204

15.17 include/DeviceArrivalEventHandler.h File Reference . . . 1205

15.18 include/DeviceEventHandler.h File Reference . . . 1206

15.19 include/DeviceRemovalEventHandler.h File Reference . . . 1207

15.20 include/EventHandler.h File Reference . . . 1208

15.21 include/Exception.h File Reference . . . 1208

15.22 include/Image.h File Reference . . . 1209

15.23 include/ImageEventHandler.h File Reference . . . 1210

15.24 include/ImagePtr.h File Reference . . . 1210

15.25 include/ImageStatistics.h File Reference . . . 1211

15.26 include/ImageUtility.h File Reference . . . 1212

15.27 include/ImageUtilityHeatmap.h File Reference . . . 1213

15.28 include/ImageUtilityPolarization.h File Reference . . . 1214
15.29 include/Interface.h File Reference . . . 1215
15.30 include/Interface/ICameraBase.h File Reference . . . 1215
15.31 include/Interface/ICameraList.h File Reference . . . 1216
15.32 include/Interface/IChunkData.h File Reference . . . 1217
15.33 include/Interface/IDeviceArrivalEventHandler.h File Reference . . . 1218
15.34 include/Interface/IDeviceEventHandler.h File Reference . . . 1219
15.35 include/Interface/IDeviceRemovalEventHandler.h File Reference . . . 1220
15.36 include/Interface/IImage.h File Reference . . . 1221
15.37 include/Interface/IImageEventHandler.h File Reference . . . 1221
15.38 include/Interface/IImageStatistics.h File Reference . . . 1222
15.39 include/Interface/IInterface.h File Reference . . . 1223
15.40 include/Interface/IInterfaceArrivalEventHandler.h File Reference . . . 1224
15.41 include/Interface/IInterfaceEventHandler.h File Reference . . . 1225
15.42 include/Interface/IInterfaceList.h File Reference . . . 1225
15.43 include/Interface/IInterfaceRemovalEventHandler.h File Reference . . . 1226
15.44 include/Interface/ILoggingEventHandler.h File Reference . . . 1227
15.45 include/Interface/IStream.h File Reference . . . 1227
15.46 include/Interface/ISystem.h File Reference . . . 1228
15.47 include/Interface/ISystemEventHandler.h File Reference . . . 1229
15.48 include/InterfaceArrivalEventHandler.h File Reference . . . 1230
15.49 include/InterfaceEventHandler.h File Reference . . . 1231
15.50 include/InterfaceList.h File Reference . . . 1232
15.51 include/InterfacePtr.h File Reference . . . 1232
15.52 include/InterfaceRemovalEventHandler.h File Reference . . . 1233
15.53 include/LoggingEventData.h File Reference . . . 1234
15.54 include/LoggingEventDataPtr.h File Reference . . . 1234
15.55 include/LoggingEventHandler.h File Reference . . . 1235
15.56 include/SpinGenApi/Autovector.h File Reference . . . 1236
15.57 include/SpinGenApi/Base.h File Reference . . . 1237

15.58 include/SpinGenApi/BooleanNode.h File Reference . . . 1238
15.59 include/SpinGenApi/CategoryNode.h File Reference . . . 1239
15.60 include/SpinGenApi/ChunkAdapter.h File Reference . . . 1240
15.61 include/SpinGenApi/ChunkAdapterDcam.h File Reference . . . 1241
15.62 include/SpinGenApi/ChunkAdapterGeneric.h File Reference . . . 1242
15.63 include/SpinGenApi/ChunkAdapterGEV.h File Reference . . . 1243
15.64 include/SpinGenApi/ChunkAdapterU3V.h File Reference . . . 1244
15.65 include/SpinGenApi/ChunkPort.h File Reference . . . 1245
15.66 include/SpinGenApi/CommandNode.h File Reference . . . 1245
15.67 include/SpinGenApi/Compatibility.h File Reference . . . 1246
15.67.1 Macro Definition Documentation . . . 1246
15.67.1.1 FMT_I64 . . . 1246
15.68 include/SpinGenApi/Container.h File Reference . . . 1247
15.69 include/SpinGenApi/Counter.h File Reference . . . 1247
15.70 include/SpinGenApi/EnumClasses.h File Reference . . . 1248
15.71 include/SpinGenApi/EnumEntryNode.h File Reference . . . 1249
15.72 include/SpinGenApi/EnumNode.h File Reference . . . 1250
15.73 include/SpinGenApi/EnumNodeT.h File Reference . . . 1251
15.74 include/SpinGenApi/EventAdapter.h File Reference . . . 1251
15.75 include/SpinGenApi/EventAdapter1394.h File Reference . . . 1252
15.76 include/SpinGenApi/EventAdapterGeneric.h File Reference . . . 1253
15.77 include/SpinGenApi/EventAdapterGEV.h File Reference . . . 1253
15.78 include/SpinGenApi/EventAdapterU3V.h File Reference . . . 1254
15.79 include/SpinGenApi/EventPort.h File Reference . . . 1255
15.80 include/SpinGenApi/Filestream.h File Reference . . . 1256
15.81 include/SpinGenApi/FloatNode.h File Reference . . . 1257
15.82 include/SpinGenApi/FloatRegNode.h File Reference . . . 1258
15.83 include/SpinGenApi/GCBase.h File Reference . . . 1259
15.84 include/SpinGenApi/GCString.h File Reference . . . 1259
15.84.1 Macro Definition Documentation . . . 1260

15.84.1.1 GCSTRING_NPOS . . . 1260
15.84.2 Function Documentation . . . 1260
15.84.2.1 operator<<() . . . 1261
15.84.2.2 operator>>() . . . 1261
15.85 include/SpinGenApi/GCStringVector.h File Reference . . . 1261
15.86 include/SpinGenApi/GCSynch.h File Reference . . . 1262
15.87 include/SpinGenApi/GCTypes.h File Reference . . . 1263
15.87.1 Macro Definition Documentation . . . 1264
15.87.1.1 __STDC_CONSTANT_MACROS . . . 1264
15.87.1.2 __STDC_LIMIT_MACROS . . . 1264
15.87.1.3 GC_INT32_MAX . . . 1264
15.87.1.4 GC_INT32_MIN . . . 1264
15.87.1.5 GC_INT64_MAX . . . 1264
15.87.1.6 GC_INT64_MIN . . . 1265
15.87.1.7 GC_INT8_MAX . . . 1265
15.87.1.8 GC_INT8_MIN . . . 1265
15.87.1.9 GC_UINT32_MAX . . . 1265
15.87.1.10 GC_UINT64_MAX . . . 1265
15.87.1.11 GC_UINT8_MAX . . . 1265
15.88 include/SpinGenApi/GCUtilities.h File Reference . . . 1266
15.88.1 Macro Definition Documentation . . . 1267
15.88.1.1 __ERR__ . . . 1268
15.88.1.2 __LINE_STR__ . . . 1268
15.88.1.3 __LOCATION__ . . . 1268
15.88.1.4 __OUTPUT_FORMATER__ . . . 1268
15.88.1.5 __TODO__ . . . 1268
15.88.1.6 __WARN__ . . . 1268
15.88.1.7 _TO_STRING . . . 1268
15.88.1.8 EXPAND_TO_STRINGISE . . . 1269
15.88.1.9 GC_COUNTOF . . . 1269

15.88.1.10 GENICAM_DEPRECATED . . . 1269
15.88.1.11 GENICAM_UNUSED . . . 1269
15.88.1.12 USE_TEMP_CACHE_FILE [1/2] . . . 1269
15.88.1.13 USE_TEMP_CACHE_FILE [2/2] . . . 1269
15.89 include/SpinGenApi/IBoolean.h File Reference . . . 1270
15.90 include/SpinGenApi/ICategory.h File Reference . . . 1271
15.91 include/SpinGenApi/IChunkPort.h File Reference . . . 1272
15.92 include/SpinGenApi/ICommand.h File Reference . . . 1273
15.93 include/SpinGenApi/IDestroy.h File Reference . . . 1274
15.94 include/SpinGenApi/IDeviceInfo.h File Reference . . . 1275
15.95 include/SpinGenApi/IEnumEntry.h File Reference . . . 1276
15.96 include/SpinGenApi/IEnumeration.h File Reference . . . 1277
15.97 include/SpinGenApi/IEnumerationT.h File Reference . . . 1278
15.98 include/SpinGenApi/IFloat.h File Reference . . . 1279
15.99 include/SpinGenApi/IInteger.h File Reference . . . 1281
15.100 include/SpinGenApi/INode.h File Reference . . . 1282
15.101 include/SpinGenApi/INodeMap.h File Reference . . . 1285
15.102 include/SpinGenApi/INodeMapDyn.h File Reference . . . 1287
15.103 include/SpinGenApi/IntegerNode.h File Reference . . . 1288
15.104 include/SpinGenApi/IntRegNode.h File Reference . . . 1289
15.105 include/SpinGenApi/IPort.h File Reference . . . 1290
15.106 include/SpinGenApi/IPortConstruct.h File Reference . . . 1291
15.107 include/SpinGenApi/IPortRecorder.h File Reference . . . 1292
15.108 include/SpinGenApi/IRegister.h File Reference . . . 1293
15.109 include/SpinGenApi/ISelector.h File Reference . . . 1294
15.110 include/SpinGenApi/ISelectorDigit.h File Reference . . . 1295
15.111 include/SpinGenApi/IString.h File Reference . . . 1296
15.112 include/SpinGenApi/IValue.h File Reference . . . 1297
15.113 include/SpinGenApi/Node.h File Reference . . . 1298
15.114 include/SpinGenApi/NodeCallback.h File Reference . . . 1299

15.115 Include/SpinGenApi/NodeCallbackImpl.h File Reference . . . 1301
15.116 Include/SpinGenApi/NodeMap.h File Reference . . . 1301
15.117 Include/SpinGenApi/NodeMapFactory.h File Reference . . . 1302
15.118 Include/SpinGenApi/NodeMapRef.h File Reference . . . 1303
15.119 Include/SpinGenApi/Persistence.h File Reference . . . 1304
15.120 Include/SpinGenApi/Pointer.h File Reference . . . 1305
15.121 Include/SpinGenApi/PortImpl.h File Reference . . . 1307
15.122 Include/SpinGenApi/PortNode.h File Reference . . . 1308
15.123 Include/SpinGenApi/PortRecorder.h File Reference . . . 1309
15.124 Include/SpinGenApi/PortReplay.h File Reference . . . 1309
15.125 Include/SpinGenApi/PortWriteList.h File Reference . . . 1310
15.126 Include/SpinGenApi/Reference.h File Reference . . . 1311
15.127 Include/SpinGenApi/RegisterNode.h File Reference . . . 1312
15.128 Include/SpinGenApi/RegisterPortImpl.h File Reference . . . 1313
15.129 Include/SpinGenApi/SelectorSet.h File Reference . . . 1313
15.130 Include/SpinGenApi/SpinnakerGenApi.h File Reference . . . 1314
15.131 Include/SpinGenApi/SpinTestCamera.h File Reference . . . 1314
15.132 Include/SpinGenApi/StringNode.h File Reference . . . 1315
15.133 Include/SpinGenApi/StringRegNode.h File Reference . . . 1316
15.134 Include/SpinGenApi/StructPort.h File Reference . . . 1316
15.135 Include/SpinGenApi/Synch.h File Reference . . . 1317
15.136 Include/SpinGenApi/Types.h File Reference . . . 1318
15.136.1 Macro Definition Documentation . . . 1321
15.136.1.1 interface . . . 1321
15.137 Include/SpinGenApi/ValueNode.h File Reference . . . 1321
15.138 Include/Spinnaker.h File Reference . . . 1322
15.139 Include/SpinnakerDefs.h File Reference . . . 1322
15.140 Include/SpinnakerPlatform.h File Reference . . . 1326
15.141 Include/SpinUpdate.h File Reference . . . 1327
15.141.1 Macro Definition Documentation . . . 1327

15.141.1.1 SPINUPDATE_API . . . 1327
15.141.2 Function Documentation . . . 1328
15.141.2.1 GetErrorMessage() . . . 1328
15.141.2.2 SetMessageCallback() . . . 1328
15.141.2.3 SetProgressCallback() . . . 1328
15.141.2.4 UpdateFirmware() . . . 1328
15.141.2.5 UpdateFirmwareConsole() . . . 1328
15.141.2.6 UpdateFirmwareGUI() . . . 1329
15.141.3 Variable Documentation . . . 1329
15.141.3.1 UpdatorMessageCallback . . . 1329
15.141.3.2 UpdatorProgressCallback . . . 1329
15.142 Include/SpinVideo.h File Reference . . . 1329
15.143 Include/SpinVideoDefs.h File Reference . . . 1330
15.144 Include/System.h File Reference . . . 1331
15.144.1 Macro Definition Documentation . . . 1332
15.144.1.1 FLIR_SPINNAKER_VERSION_BUILD . . . 1332
15.144.1.2 FLIR_SPINNAKER_VERSION_MAJOR . . . 1332
15.144.1.3 FLIR_SPINNAKER_VERSION_MINOR . . . 1332
15.144.1.4 FLIR_SPINNAKER_VERSION_TYPE . . . 1332
15.145 Include/SystemEventHandler.h File Reference . . . 1333
15.146 Include/SystemPtr.h File Reference . . . 1334
15.147 Include/TransportLayerDefs.h File Reference . . . 1334
15.148 Include/TransportLayerDevice.h File Reference . . . 1336
15.149 Include/TransportLayerInterface.h File Reference . . . 1337
15.150 Include/TransportLayerStream.h File Reference . . . 1337
15.151 Include/TransportLayerSystem.h File Reference . . . 1338
15.152 Src/Acquisition/Acquisition.cpp File Reference . . . 1338
15.152.1 Function Documentation . . . 1339
15.152.1.1 AcquireImages() . . . 1339
15.152.1.2 main() . . . 1339

15.152.1.3 PrintDeviceInfo() . . . 1339
15.152.1.4 RunSingleCamera() . . . 1339
15.153 src/Acquisition/resource.h File Reference . . . 1340
15.154 src/AcquisitionMultipleCameraRecovery/resource.h File Reference . . . 1340
15.155 src/AcquisitionMultipleThread/resource.h File Reference . . . 1340
15.156 src/ActionCommand/resource.h File Reference . . . 1340
15.157 src/BufferHandling/resource.h File Reference . . . 1340
15.158 src/ChunkData/resource.h File Reference . . . 1340
15.159 src/CounterAndTimer/resource.h File Reference . . . 1340
15.160 src/DeviceEvents/resource.h File Reference . . . 1340
15.161 src/Enumeration/resource.h File Reference . . . 1340
15.162 src/Enumeration_QuickSpin/resource.h File Reference . . . 1340
15.163 src/EnumerationEvents/resource.h File Reference . . . 1340
15.164 src/ExceptionHandling/resource.h File Reference . . . 1340
15.165 src/Exposure/resource.h File Reference . . . 1340
15.166 src/Exposure_QuickSpin/resource.h File Reference . . . 1340
15.167 src/FileAccess_QuickSpin/resource.h File Reference . . . 1340
15.168 src/GigEVisionPerformance/resource.h File Reference . . . 1340
15.169 src/HighDynamicRange/resource.h File Reference . . . 1340
15.170 src/GenTLInfo_QuickSpin/resource.h File Reference . . . 1341
15.171 src/ImageEvents/resource.h File Reference . . . 1341
15.172 src/ImageFormatControl/resource.h File Reference . . . 1341
15.173 src/ImageFormatControl_QuickSpin/resource.h File Reference . . . 1341
15.174 src/Inference/resource.h File Reference . . . 1341
15.175 src/Logging/resource.h File Reference . . . 1341
15.176 src/LogicBlock/resource.h File Reference . . . 1341
15.177 src/LookupTable/resource.h File Reference . . . 1341
15.178 src/NodeMapCallback/resource.h File Reference . . . 1341
15.179 src/NodeMapInfo/resource.h File Reference . . . 1341
15.180 src/Polarization/resource.h File Reference . . . 1341

15.181 src/SaveToAvi/resource.h File Reference . . . 1341
15.182 src/Sequencer/resource.h File Reference . . . 1341
15.183 src/SerialRxTx/resource.h File Reference . . . 1341
15.184 src/Trigger/resource.h File Reference . . . 1341
15.185 src/Trigger_QuickSpin/resource.h File Reference . . . 1341
15.186 src/Acquisition/stdafx.cpp File Reference . . . 1342
15.187 src/ActionCommand/stdafx.cpp File Reference . . . 1342
15.188 src/BufferHandling/stdafx.cpp File Reference . . . 1343
15.189 src/CounterAndTimer/stdafx.cpp File Reference . . . 1344
15.190 src/DeviceEvents/stdafx.cpp File Reference . . . 1344
15.191 src/Enumeration/stdafx.cpp File Reference . . . 1345
15.192 src/Enumeration_QuickSpin/stdafx.cpp File Reference . . . 1345
15.193 src/ExceptionHandling/stdafx.cpp File Reference . . . 1346
15.194 src/Exposure/stdafx.cpp File Reference . . . 1346
15.195 src/Exposure_QuickSpin/stdafx.cpp File Reference . . . 1347
15.196 src/FileAccess_QuickSpin/stdafx.cpp File Reference . . . 1348
15.197 src/GigEVisionPerformance/stdafx.cpp File Reference . . . 1349
15.198 src/NodeMapInfo/stdafx.cpp File Reference . . . 1349
15.199 src/Sequencer/stdafx.cpp File Reference . . . 1350
15.200 src/SerialRxTx/stdafx.cpp File Reference . . . 1350
15.201 src/Acquisition/stdafx.h File Reference . . . 1351
15.202 src/ActionCommand/stdafx.h File Reference . . . 1352
15.203 src/BufferHandling/stdafx.h File Reference . . . 1353
15.204 src/CounterAndTimer/stdafx.h File Reference . . . 1354
15.205 src/DeviceEvents/stdafx.h File Reference . . . 1355
15.206 src/Enumeration/stdafx.h File Reference . . . 1356
15.207 src/Enumeration_QuickSpin/stdafx.h File Reference . . . 1357
15.208 src/ExceptionHandling/stdafx.h File Reference . . . 1358
15.209 src/Exposure/stdafx.h File Reference . . . 1359
15.210 src/Exposure_QuickSpin/stdafx.h File Reference . . . 1360

15.211 src/FileAccess_QuickSpin/stdafx.h File Reference . . . 1361
15.212 src/GigEVisionPerformance/stdafx.h File Reference . . . 1362
15.213 src/ImageFormatControl/stdafx.h File Reference . . . 1363
15.214 src/ImageFormatControl_QuickSpin/stdafx.h File Reference . . . 1363
15.215 src/NodeMapInfo/stdafx.h File Reference . . . 1364
15.216 src/Polarization/stdafx.h File Reference . . . 1365
15.217 src/Sequencer/stdafx.h File Reference . . . 1365
15.218 src/SerialRxTx/stdafx.h File Reference . . . 1366
15.219 src/Acquisition/targetver.h File Reference . . . 1367
15.220 src/ActionCommand/targetver.h File Reference . . . 1368
15.221 src/BufferHandling/targetver.h File Reference . . . 1369
15.222 src/CounterAndTimer/targetver.h File Reference . . . 1370
15.223 src/DeviceEvents/targetver.h File Reference . . . 1371
15.224 src/Enumeration/targetver.h File Reference . . . 1372
15.225 src/Enumeration_QuickSpin/targetver.h File Reference . . . 1373
15.226 src/ExceptionHandling/targetver.h File Reference . . . 1374
15.227 src/Exposure/targetver.h File Reference . . . 1375
15.228 src/Exposure_QuickSpin/targetver.h File Reference . . . 1376
15.229 src/FileAccess_QuickSpin/targetver.h File Reference . . . 1377
15.230 src/GigEVisionPerformance/targetver.h File Reference . . . 1378
15.231 src/GenTLInfo_QuickSpin/targetver.h File Reference . . . 1379
15.232 src/NodeMapInfo/targetver.h File Reference . . . 1379
15.233 src/Sequencer/targetver.h File Reference . . . 1380
15.234 src/SerialRxTx/targetver.h File Reference . . . 1381
15.235 src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp File Reference . 1382
15.235.1 Function Documentation . . . 1382
15.235.1.1 ConfigureCamera() . . . 1382
15.235.1.2 ConfigureUserSet1() . . . 1383
15.235.1.3 GetDeviceSerial() . . . 1383
15.235.1.4 main() . . . 1383

15.235.1.5 PrintExampleStatistics() . . . 1383
15.235.1.6 RefreshCameraList() . . . 1383
15.235.1.7 ResetCameraUserSetToDefault() . . . 1383
15.235.1.8 SleepyWrapper() . . . 1383
15.235.2 Variable Documentation . . . 1384
15.235.2.1 cameraGrabInfoMap . . . 1384
15.235.2.2 globalCamList . . . 1384
15.236 src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference . . . 1384
15.236.1 Function Documentation . . . 1384
15.236.1.1 AcquireImages() . . . 1384
15.236.1.2 main() . . . 1385
15.236.1.3 PrintDeviceInfo() . . . 1385
15.236.1.4 RunMultipleCameras() . . . 1385
15.237 src/ActionCommand/ActionCommand.cpp File Reference . . . 1385
15.237.1 Function Documentation . . . 1386
15.237.1.1 AcquireImages() . . . 1386
15.237.1.2 ConfigureActionControl() . . . 1386
15.237.1.3 ConfigureChunkData() . . . 1386
15.237.1.4 ConfigureIEEE1588() . . . 1386
15.237.1.5 ConfigureInterface() . . . 1386
15.237.1.6 ConfigureOtherNodes() . . . 1386
15.237.1.7 ConfigureTrigger() . . . 1387
15.237.1.8 main() . . . 1387
15.237.1.9 PrintDeviceInfo() . . . 1387
15.237.1.10 RunMultipleCameras() . . . 1387
15.237.1.11 SleepyWrapper() . . . 1387
15.238 src/BufferHandling/BufferHandling.cpp File Reference . . . 1387
15.238.1 Macro Definition Documentation . . . 1388
15.238.1.1 k_numLoops . . . 1388
15.238.1.2 numBuffers . . . 1388

15.238.1.3 z_numTriggers . . . 1388
15.238.2 Function Documentation . . . 1388
15.238.2.1 AcquireImages() . . . 1388
15.238.2.2 ConfigureTrigger() . . . 1389
15.238.2.3 GrabNextImageByTrigger() . . . 1389
15.238.2.4 main() . . . 1389
15.238.2.5 PrintDeviceInfo() . . . 1389
15.238.2.6 ResetTrigger() . . . 1389
15.238.2.7 RunSingleCamera() . . . 1389
15.238.2.8 SleepyWrapper() . . . 1389
15.239 src/ChunkData/ChunkData.cpp File Reference . . . 1390
15.239.1 Enumeration Type Documentation . . . 1390
15.239.1.1 chunkDataType . . . 1390
15.239.2 Function Documentation . . . 1391
15.239.2.1 AcquireImages() . . . 1391
15.239.2.2 ConfigureChunkData() . . . 1391
15.239.2.3 DisableChunkData() . . . 1391
15.239.2.4 DisplayChunkData() [1/2] . . . 1391
15.239.2.5 DisplayChunkData() [2/2] . . . 1391
15.239.2.6 main() . . . 1391
15.239.2.7 PrintDeviceInfo() . . . 1392
15.239.2.8 RunSingleCamera() . . . 1392
15.239.3 Variable Documentation . . . 1392
15.239.3.1 chosenChunkData . . . 1392
15.240 src/CounterAndTimer/CounterAndTimer.cpp File Reference . . . 1392
15.240.1 Function Documentation . . . 1392
15.240.1.1 AcquireImages() . . . 1393
15.240.1.2 ConfigureDigitalIO() . . . 1393
15.240.1.3 ConfigureExposureandTrigger() . . . 1393
15.240.1.4 main() . . . 1393

15.240.1.5 PrintDeviceInfo() . . . 1393
15.240.1.6 ResetTrigger() . . . 1393
15.240.1.7 RunSingleCamera() . . . 1394
15.240.1.8 SetupCounterAndTimer() . . . 1394
15.241 src/DeviceEvents/DeviceEvents.cpp File Reference . . . 1394
15.241.1 Enumeration Type Documentation . . . 1394
15.241.1.1 eventType . . . 1394
15.241.2 Function Documentation . . . 1395
15.241.2.1 AcquireImages() . . . 1395
15.241.2.2 ConfigureDeviceEvents() . . . 1395
15.241.2.3 main() . . . 1395
15.241.2.4 PrintDeviceInfo() . . . 1395
15.241.2.5 ResetDeviceEvents() . . . 1395
15.241.2.6 RunSingleCamera() . . . 1396
15.241.3 Variable Documentation . . . 1396
15.241.3.1 chosenEvent . . . 1396
15.242 src/Enumeration/Enumeration.cpp File Reference . . . 1396
15.242.1 Function Documentation . . . 1396
15.242.1.1 main() . . . 1396
15.242.1.2 QueryInterface() . . . 1396
15.243 src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp File Reference . . . 1397
15.243.1 Function Documentation . . . 1397
15.243.1.1 main() . . . 1397
15.243.1.2 QueryInterface() . . . 1397
15.244 src/EnumerationEvents/EnumerationEvents.cpp File Reference . . . 1397
15.244.1 Function Documentation . . . 1398
15.244.1.1 CheckGevEnabled() . . . 1398
15.244.1.2 main() . . . 1398
15.245 src/ExceptionHandling/ExceptionHandling.cpp File Reference . . . 1398
15.245.1 Enumeration Type Documentation . . . 1399

15.245.1.1 exceptionType . . . 1399
15.245.2 Function Documentation . . . 1399
15.245.2.1 causeSpinnakerException() . . . 1399
15.245.2.2 causeStandardException() . . . 1399
15.245.2.3 main() . . . 1399
15.245.3 Variable Documentation . . . 1399
15.245.3.1 chosenException . . . 1400
15.246 src/Exposure/Exposure.cpp File Reference . . . 1400
15.246.1 Function Documentation . . . 1400
15.246.1.1 AcquireImages() . . . 1400
15.246.1.2 ConfigureExposure() . . . 1400
15.246.1.3 main() . . . 1401
15.246.1.4 PrintDeviceInfo() . . . 1401
15.246.1.5 ResetExposure() . . . 1401
15.246.1.6 RunSingleCamera() . . . 1401
15.247 src/Exposure_QuickSpin/Exposure_QuickSpin.cpp File Reference . . . 1401
15.247.1 Function Documentation . . . 1402
15.247.1.1 AcquireImages() . . . 1402
15.247.1.2 ConfigureExposure() . . . 1402
15.247.1.3 main() . . . 1402
15.247.1.4 PrintDeviceInfo() . . . 1402
15.247.1.5 ResetExposure() . . . 1402
15.247.1.6 RunSingleCamera() . . . 1402
15.248 src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp File Reference . . . 1403
15.248.1 Function Documentation . . . 1403
15.248.1.1 AcquireImages() . . . 1403
15.248.1.2 CloseFile() . . . 1404
15.248.1.3 DownloadImage() . . . 1404
15.248.1.4 ExecuteDeleteCommand() . . . 1404
15.248.1.5 ExecuteReadCommand() . . . 1404

15.248.1.6 ExecuteWriteCommand() . . . 1404
15.248.1.7 InitializeSystem() . . . 1404
15.248.1.8 main() . . . 1404
15.248.1.9 OpenFileToRead() . . . 1405
15.248.1.10 OpenFileToWrite() . . . 1405
15.248.1.11 PrintDebugMessage() . . . 1405
15.248.1.12 PrintDeviceInfo() . . . 1405
15.248.1.13 PrintResultMessage() . . . 1405
15.248.1.14 PrintUsage() . . . 1405
15.248.1.15 UploadImage() . . . 1405
15.248.2 Variable Documentation . . . 1406
15.248.2.1 _enableDebug . . . 1406
15.248.2.2 _fileSelector . . . 1406
15.249 src/GenTLInfo_QuickSpin/GenTLInfo_QuickSpin.cpp File Reference . . . 1406
15.249.1 Function Documentation . . . 1406
15.249.1.1 main() . . . 1406
15.249.1.2 PrintApplicationLayerDeviceInfo() . . . 1407
15.249.1.3 PrintTransportLayerDeviceInfo() . . . 1407
15.249.1.4 PrintTransportLayerInterfaceInfo() . . . 1407
15.249.1.5 PrintTransportLayerStreamInfo() . . . 1407
15.250 src/GigEVisionPerformance/CpuUtil.cpp File Reference . . . 1407
15.251 src/GigEVisionPerformance/CpuUtil.h File Reference . . . 1408
15.252 src/GigEVisionPerformance/GigEVisionPerformance.cpp File Reference . . . 1409
15.252.1 Function Documentation . . . 1410
15.252.1.1 AcquireImages() . . . 1410
15.252.1.2 EnableManualFramerate() . . . 1410
15.252.1.3 getCameraCategory() . . . 1410
15.252.1.4 main() . . . 1411
15.252.1.5 ParseArguments() . . . 1411
15.252.1.6 PrintAllNodes() . . . 1411

15.252.1.7 PrintCPUUsage() . . . 1411
15.252.1.8 PrintDataStreamInfo() . . . 1411
15.252.1.9 PrintDeviceInfo() . . . 1411
15.252.1.10 PrintUsage() . . . 1411
15.252.1.11 RunSingleCamera() . . . 1412
15.252.1.12 SetFrameRate() . . . 1412
15.252.2 Variable Documentation . . . 1412
15.252.2.1 argBayerRG . . . 1412
15.252.2.2 argDuration . . . 1412
15.252.2.3 argMaxFrames . . . 1412
15.252.2.4 argNumImages . . . 1412
15.252.2.5 argPacketDelay . . . 1412
15.252.2.6 argPacketSize . . . 1413
15.252.2.7 argPrintUsage . . . 1413
15.252.2.8 argRelease . . . 1413
15.252.2.9 argUserSetFrames . . . 1413
15.252.2.10 cpuUsageInfo . . . 1413
15.252.2.11 bIsRelease . . . 1413
15.252.2.12 NumImagesToGrab . . . 1413
15.252.2.13 PacketDelayToSet . . . 1413
15.252.2.14 PacketSizeToSet . . . 1414
15.252.2.15 PixelFormatToSet . . . 1414
15.252.2.16 TestDuration . . . 1414
15.252.2.17 UseDuration . . . 1414
15.252.2.18 UseMaxFramerate . . . 1414
15.252.2.19 UserSetFramerate . . . 1414
15.253 src/GigEVisionPerformance/GigEVisionPerformance.h File Reference . . . 1414
15.254 src/HighDynamicRange/HighDynamicRange.cpp File Reference . . . 1415
15.254.1 Function Documentation . . . 1415
15.254.1.1 CheckNodeAccessibility() . . . 1415

15.254.1.2 InitializeHDRImages() . . . 1415
15.254.1.3 main() . . . 1416
15.254.1.4 PrintBuildInfo() . . . 1416
15.254.1.5 PrintDeviceInfo() . . . 1416
15.254.1.6 RunSingleCamera() . . . 1416
15.254.1.7 ToggleHDRMode() . . . 1416
15.254.2 Variable Documentation . . . 1416
15.254.2.1 k_HDRGain1 . . . 1416
15.254.2.2 k_HDRGain2 . . . 1417
15.254.2.3 k_HDRGain3 . . . 1417
15.254.2.4 k_HDRGain4 . . . 1417
15.254.2.5 k_HDRShutter1 . . . 1417
15.254.2.6 k_HDRShutter2 . . . 1417
15.254.2.7 k_HDRShutter3 . . . 1417
15.254.2.8 k_HDRShutter4 . . . 1417
15.255 src/ImageEvents/ImageEvents.cpp File Reference . . . 1418
15.255.1 Function Documentation . . . 1418
15.255.1.1 AcquireImages() . . . 1418
15.255.1.2 ConfigureImageEvents() . . . 1418
15.255.1.3 main() . . . 1419
15.255.1.4 PrintDeviceInfo() . . . 1419
15.255.1.5 ResetImageEvents() . . . 1419
15.255.1.6 RunSingleCamera() . . . 1419
15.255.1.7 SleepyWrapper() . . . 1419
15.255.1.8 WaitForImages() . . . 1419
15.256 src/ImageFormatControl/ImageFormatControl.cpp File Reference . . . 1420
15.256.1 Function Documentation . . . 1420
15.256.1.1 AcquireImages() . . . 1420
15.256.1.2 ConfigureCustomImageSettings() . . . 1420
15.256.1.3 main() . . . 1420

15.256.1.4 PrintDeviceInfo() . . . 1421
15.256.1.5 RunSingleCamera() . . . 1421
15.257 src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp File Reference . . . 1421
15.257.1 Function Documentation . . . 1421
15.257.1.1 AcquireImages() . . . 1421
15.257.1.2 ConfigureCustomImageSettings() . . . 1422
15.257.1.3 main() . . . 1422
15.257.1.4 PrintDeviceInfo() . . . 1422
15.257.1.5 RunSingleCamera() . . . 1422
15.258 src/Inference/Inference.cpp File Reference . . . 1422
15.258.1 Enumeration Type Documentation . . . 1423
15.258.1.1 FileUploadPersistence . . . 1423
15.258.1.2 InferenceNetworkType . . . 1424
15.258.2 Function Documentation . . . 1424
15.258.2.1 AcquireImages() . . . 1424
15.258.2.2 CameraCloseFile() . . . 1424
15.258.2.3 CameraDeleteFile() . . . 1424
15.258.2.4 CameraOpenFile() . . . 1424
15.258.2.5 CameraWriteToFile() . . . 1425
15.258.2.6 ConfigureChunkData() . . . 1425
15.258.2.7 ConfigureInference() . . . 1425
15.258.2.8 ConfigureTestPattern() . . . 1425
15.258.2.9 ConfigureTrigger() . . . 1425
15.258.2.10 DeleteFileOnCamera() . . . 1425
15.258.2.11 DisableChunkData() . . . 1426
15.258.2.12 DisableTrigger() . . . 1426
15.258.2.13 DisplayChunkData() . . . 1426
15.258.2.14 LabelClassification() . . . 1426
15.258.2.15 LabelDetection() . . . 1426
15.258.2.16 LoadFileIntoMemory() . . . 1426

15.258.2.17 main() . . . 1427
15.258.2.18 PrintDeviceInfo() . . . 1427
15.258.2.19 RunSingleCamera() . . . 1427
15.258.2.20 SetChunkEnable() . . . 1427
15.258.2.21 UploadFileToCamera() . . . 1427
15.258.3 Variable Documentation . . . 1427
15.258.3.1 arrayLabelClassification . . . 1427
15.258.3.2 arrayLabelDetection . . . 1428
15.258.3.3 chosenFileUploadPersistence . . . 1428
15.258.3.4 chosenInferenceNetworkType . . . 1428
15.258.3.5 injectedImageFilePath . . . 1428
15.258.3.6 injectedImageHeight . . . 1428
15.258.3.7 injectedImageWidth . . . 1429
15.258.3.8 networkFilePath . . . 1429
15.259 src/Logging/Logging.cpp File Reference . . . 1429
15.259.1 Function Documentation . . . 1429
15.259.1.1 main() . . . 1430
15.259.2 Variable Documentation . . . 1430
15.259.2.1 k_LoggingLevel . . . 1430
15.260 src/LogicBlock/LogicBlock.cpp File Reference . . . 1430
15.260.1 Function Documentation . . . 1430
15.260.1.1 AcquireImages() . . . 1430
15.260.1.2 ConfigureLogicBlock() . . . 1431
15.260.1.3 ConfigureTrigger() . . . 1431
15.260.1.4 GrabTwoImages() . . . 1431
15.260.1.5 main() . . . 1431
15.260.1.6 PrintDeviceInfo() . . . 1431
15.260.1.7 ResetExposure() . . . 1431
15.260.1.8 ResetTrigger() . . . 1431
15.260.1.9 RunSingleCamera() . . . 1432

15.261 src/LookupTable/LookupTable.cpp File Reference . . . 1432
15.261.1 Function Documentation . . . 1432
15.261.1.1 AcquireImages() . . . 1432
15.261.1.2 ConfigureLookupTables() . . . 1432
15.261.1.3 main() . . . 1433
15.261.1.4 PrintDeviceInfo() . . . 1433
15.261.1.5 PrintRetrieveNodeFailure() . . . 1433
15.261.1.6 ResetLookupTables() . . . 1433
15.261.1.7 RunSingleCamera() . . . 1433
15.262 src/NodeMapCallback/NodeMapCallback.cpp File Reference . . . 1433
15.262.1 Function Documentation . . . 1434
15.262.1.1 ChangeHeightAndGain() . . . 1434
15.262.1.2 ConfigureCallbacks() . . . 1434
15.262.1.3 main() . . . 1434
15.262.1.4 OnGainNodeUpdate() . . . 1434
15.262.1.5 OnHeightNodeUpdate() . . . 1435
15.262.1.6 PrintDeviceInfo() . . . 1435
15.262.1.7 ResetCallbacks() . . . 1435
15.262.1.8 RunSingleCamera() . . . 1435
15.263 src/NodeMapInfo/NodeMapInfo.cpp File Reference . . . 1435
15.263.1 Enumeration Type Documentation . . . 1436
15.263.1.1 readType . . . 1436
15.263.2 Function Documentation . . . 1436
15.263.2.1 Indent() . . . 1436
15.263.2.2 main() . . . 1437
15.263.2.3 PrintBooleanNode() . . . 1437
15.263.2.4 PrintCategoryNodeAndAllFeatures() . . . 1437
15.263.2.5 PrintCommandNode() . . . 1437
15.263.2.6 PrintEnumerationNodeAndCurrentEntry() . . . 1437
15.263.2.7 PrintEnumerationSelector() . . . 1437

15.263.2.8 PrintFloatNode() . . . 1438
15.263.2.9 PrintIntegerNode() . . . 1438
15.263.2.10 PrintNode() . . . 1438
15.263.2.11 PrintStringNode() . . . 1438
15.263.2.12 PrintValueNode() . . . 1438
15.263.2.13 RunSingleCamera() . . . 1438
15.263.3 Variable Documentation . . . 1438
15.263.3.1 chosenRead . . . 1439
15.263.3.2 maxChars . . . 1439
15.264 src/Polarization/Polarization.cpp File Reference . . . 1439
15.264.1 Function Documentation . . . 1439
15.264.1.1 AcquireImages() . . . 1440
15.264.1.2 ConfigureStream() . . . 1440
15.264.1.3 CreateAndSaveAolpDolpImages() . . . 1440
15.264.1.4 CreateAndSaveGlareReducedImage() . . . 1440
15.264.1.5 CreateAndSaveStokesImages() . . . 1440
15.264.1.6 CreateHeatmapImages() . . . 1440
15.264.1.7 CreateNormalizedImage() . . . 1441
15.264.1.8 ExtractAndSavePolarQuadImages() . . . 1441
15.264.1.9 GetQuadFileNameAppendage() . . . 1441
15.264.1.10 main() . . . 1441
15.264.1.11 PrintDeviceInfo() . . . 1441
15.264.1.12 RunSingleCamera() . . . 1441
15.264.1.13 SaveImage() . . . 1442
15.264.2 Variable Documentation . . . 1442
15.264.2.1 isPixelFormatColor . . . 1442
15.265 src/SaveToAvi/SaveToAvi.cpp File Reference . . . 1442
15.265.1 Enumeration Type Documentation . . . 1442
15.265.1.1 videoType . . . 1442
15.265.2 Function Documentation . . . 1443

15.265.2.1 AcquireImages() . . . 1443
15.265.2.2 main() . . . 1443
15.265.2.3 PrintDeviceInfo() . . . 1443
15.265.2.4 RunSingleCamera() . . . 1443
15.265.2.5 SaveVectorToVideo() . . . 1443
15.265.3 Variable Documentation . . . 1444
15.265.3.1 chosenVideoType . . . 1444
15.266 src/Sequencer/Sequencer.cpp File Reference . . . 1444
15.266.1 Function Documentation . . . 1444
15.266.1.1 AcquireImages() . . . 1444
15.266.1.2 ConfigureSequencerPartOne() . . . 1445
15.266.1.3 ConfigureSequencerPartTwo() . . . 1445
15.266.1.4 main() . . . 1445
15.266.1.5 PrintDeviceInfo() . . . 1445
15.266.1.6 PrintRetrieveNodeFailure() . . . 1445
15.266.1.7 ResetSequencer() . . . 1445
15.266.1.8 RunSingleCamera() . . . 1445
15.266.1.9 SetSingleState() . . . 1446
15.267 src/SerialRxTx/SerialRxTx.cpp File Reference . . . 1446
15.267.1 Macro Definition Documentation . . . 1446
15.267.1.1 COM_PORT_COUNT_MAX . . . 1447
15.267.1.2 DATA_BITS . . . 1447
15.267.1.3 MILLISECOND . . . 1447
15.267.1.4 SERIAL_PORT_BAUD_RATE . . . 1447
15.267.1.5 SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND . . . 1447
15.267.1.6 SERIAL_PORT_DELAY . . . 1447
15.267.1.7 SERIAL_PORT_PARITY_BITS . . . 1447
15.267.1.8 SERIAL_PORT_STOP_BITS . . . 1447
15.267.1.9 TWO_SECOND_DELAY . . . 1448
15.267.2 Function Documentation . . . 1448

15.267.2.1 CleanUp() . . . 1448
15.267.2.2 ConfigureDevice() . . . 1448
15.267.2.3 main() . . . 1448
15.267.2.4 PrintDeviceInfo() . . . 1448
15.267.2.5 RunSingleCamera() . . . 1448
15.267.2.6 SerialRx() . . . 1449
15.267.2.7 SerialTx() . . . 1449
15.268 src/Trigger/Trigger.cpp File Reference . . . 1449
15.268.1 Enumeration Type Documentation . . . 1450
15.268.1.1 triggerType . . . 1450
15.268.2 Function Documentation . . . 1451
15.268.2.1 AcquireImages() . . . 1451
15.268.2.2 ConfigureTrigger() . . . 1451
15.268.2.3 GrabNextImageByTrigger() . . . 1451
15.268.2.4 main() . . . 1451
15.268.2.5 PrintDeviceInfo() . . . 1451
15.268.2.6 ResetTrigger() . . . 1452
15.268.2.7 RunSingleCamera() . . . 1452
15.268.3 Variable Documentation . . . 1452
15.268.3.1 chosenTrigger . . . 1452
15.269 src/Trigger_QuickSpin/Trigger_QuickSpin.cpp File Reference . . . 1452
15.269.1 Enumeration Type Documentation . . . 1453
15.269.1.1 triggerType . . . 1453
15.269.2 Function Documentation . . . 1453
15.269.2.1 AcquireImages() . . . 1453
15.269.2.2 ConfigureTrigger() . . . 1453
15.269.2.3 GrabNextImageByTrigger() . . . 1453
15.269.2.4 main() . . . 1454
15.269.2.5 PrintDeviceInfo() . . . 1454
15.269.2.6 ResetTrigger() . . . 1454
15.269.2.7 RunSingleCamera() . . . 1454
15.269.3 Variable Documentation . . . 1454
15.269.3.1 chosenTrigger . . . 1454

16 Example Documentation 1455

16.1 Acquisition.cpp . . . 1455

16.2 AcquisitionMultipleCameraRecovery.cpp . . . 1455

16.3 AcquisitionMultipleThread.cpp . . . 1456

16.4 ActionCommand.cpp . . . 1456

16.5 BufferHandling.cpp . . . 1456

16.6 ChunkData.cpp . . . 1456

16.7 CounterAndTimer.cpp . . . 1457

16.8 DeviceEvents.cpp . . . 1457

16.9 Enumeration.cpp . . . 1457

16.10 Enumeration_QuickSpin.cpp . . . 1458

16.11 EnumerationEvents.cpp . . . 1458

16.12 ExceptionHandling.cpp . . . 1458

16.13 Exposure.cpp . . . 1459

16.14 Exposure_QuickSpin.cpp . . . 1459

16.15 FileAccess_Quickspin.cpp . . . 1459

16.16 GenTLInfo_QuickSpin.cpp . . . 1459

16.17 GigEVisionPerformance.cpp . . . 1460

16.18 HighDynamicRange.cpp . . . 1460

16.19 ImageEvents.cpp . . . 1460

16.20 ImageFormatControl.cpp . . . 1460

16.21 ImageFormatControl_QuickSpin.cpp . . . 1461

16.22 Inference.cpp . . . 1461

16.23 Logging.cpp . . . 1461

16.24 LogicBlock.cpp . . . 1462

16.25 LookupTable.cpp . . . 1462

16.26 NodeMapCallback.cpp . . . 1462

16.27 NodeMapInfo.cpp . . . 1463

16.28 Polarization.cpp . . . 1463

16.29 SaveToAvi.cpp . . . 1463

16.30 Sequencer.cpp . . . 1464

16.31 SerialRxTx.cpp . . . 1464

16.32 Trigger.cpp . . . 1464

16.33 Trigger_QuickSpin.cpp . . . 1464

Index 1465

Chapter 1

Getting Started

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

- Benefits of Spinnaker
- Software Licensing Information
- FlyCapture2 Feature Comparison with Spinnaker
- Programmer's Guide
- SpinViewGuide
- Working with GenICam GenTL Devices

Chapter 2

Programmer's Guide

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

Chapter 3

Benefits of Spinnaker

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

Chapter 4

FlyCapture2 Feature Comparison with Spinnaker

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

Chapter 5

Working with GenICam GenTL Devices

5.1 GenTL Overview

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

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

5.2 Installation

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

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

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

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

5.3 Troubleshooting

5.3.1 Enable FLIR GenTL Logging

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

Sample log4cpp.gentl.property configuration file:

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

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

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

5.3.2 USB3 Device Image Tearing

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

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

Chapter 6

Software Licensing Information

Table 6.1 License table

Component	License
Spinnaker	Copyright (c) 2001-2020 FLIR Systems, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("↩ Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARR↩ ANTIES ABOUT THE SUITABILITY OF THE SOFT↩ WARE, EITHER EXPRESSED OR IMPLIED, INCL↩ UDING, BUT NOT LIMITED TO, THE IMPLIED WA↩ RRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEM↩ ENT. FLIR SHALL NOT BE LIABLE FOR ANY DAM↩ AGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SO↩ FTWARE OR ITS DERIVATIVES.
GenICam	GenICam License
AdapterList	The Code Project Open License (CP↩ OL)
Make ListView.ScrollIntoView Scroll the Item into the Center of the ListView	WP:CC_BY-SA License
Work with Bitmaps Faster in C#	The Code Project Open License (CP↩ OL) 1.02
FreeImage	FreeImage public license
Boost	Boost Software License
Libusb	LGPLv2.1 License
Libraw1394	LGPLv2.0 License
FFMPEG	LGPv2.1 License
log4Net	Apache license 2.0
log4Cpp	LGPL License

The licenses mentioned above can also be found in the Spinnaker installed license folder.

Chapter 7

Module Index

7.1 Modules

Here is a list of all modules:

Spinnaker Classes . . . 40
- AVI Recorder Class . . . 44
- BasePtr Class . . . 47
- Camera Class . . . 48
- Camera Base Class . . . 49
- CameraDefs Class . . . 50
- Camera List Class . . . 165
- CameraPtr Class . . . 166
- ChunkData Class . . . 168
- Chunk Data Inference Class . . . 169
- Exception Class . . . 181
- Image Class . . . 182
- ImagePtr Class . . . 184
- ImageStatistics Class . . . 185
- Image Utility Class . . . 186
- Image Utility Heatmap Class . . . 187
- Image Utility Polarization Class . . . 188
- Interface Class . . . 189
- InterfaceList Class . . . 192
- InterfacePtr Class . . . 193
- Spinnaker Video Class . . . 212
- System Class . . . 214
- SystemPtr Class . . . 216
- Camera Base Interface Class . . . 230
- IChunkData Class . . . 231
- IImage Class . . . 232
- IImageStatistics Class . . . 233
- IInterface Class . . . 234
- IInterfaceList Class . . . 235
- ISystem Class . . . 236

Spinnaker EventHandler Classes . . . 175
- DeviceArrivalEventHandler Class . . . 177
- DeviceEventHandler Class . . . 178
- DeviceRemovalEventHandler Class . . . 179
- EventHandler Class . . . 180

ImageEventHandler Class . . . 183
InterfaceArrivalEventHandler Class . . . 190
InterfaceEventHandler Class . . . 191
InterfaceRemovalEventHandler Class . . . 194
Logging EventHandler Class . . . 195
LoggingEventDataPtr Class . . . 196
LoggingEventHandler Class . . . 197
SystemEventHandler Class . . . 215

Spinnaker Headers . . . 198

Spinnaker.h . . . 200
Spinnaker Definitions . . . 201
Spinnaker Platform . . . 211
Spinnaker Video Definitions . . . 213

Spinnaker QuickSpin Classes . . . 217

TransportLayerDefs Class . . . 218
TransportLayerDevice Class . . . 226
TransportLayerInterface Class . . . 227
TransportLayerStream Class . . . 228
TransportLayerSystem Class . . . 229

Spinnaker GenApi Classes . . . 237

AutoVector Class . . . 248
BooleanNode Class . . . 252
CategoryNode Class . . . 253
ChunkAdapter Class . . . 254
ChunkAdapterDcam Class . . . 255
ChunkAdapterGeneric Class . . . 256
ChunkAdapterGEV Class . . . 257
ChunkPort Class . . . 258
CommandNode Class . . . 259
Container Class . . . 260
Counter Class . . . 261
EnumClasses Class . . . 262
EnumEntryNode Class . . . 264
EnumNode Class . . . 265
EnumNodeT Class . . . 266
EventAdapter Class . . . 267
EventAdapter1394 Class . . . 268
EventAdapterGeneric Class . . . 269
EventAdapterGEV Class . . . 270
EventAdapterU3V Class . . . 271
EventPort Class . . . 272
Filestream Class . . . 273
FloatNode Class . . . 274
FloatRegNode Class . . . 275
GCString Class . . . 276
GCSynch Class . . . 277
GCTypes Class . . . 278
IntegerNode Class . . . 329
IntRegNode Class . . . 330
IString Class . . . 342
IValue Class . . . 343
Node Class . . . 345
NodeCallback Class . . . 346
NodeMap Class . . . 349
NodeMapFactory Class . . . 350
NodeMapRef Class . . . 352
Persistence Class . . . 353

Pointer Class . . . 354
PortImpl Class . . . 360
PortNode Class . . . 361
PortRecorder Class . . . 362
PortReplay Class . . . 363
PortWriteList Class . . . 364
RegisterNode Class . . . 366
RegisterPortImpl Class . . . 367
SelectorSet Class . . . 368
SpinTestCamera Class . . . 369
StringNode Class . . . 370
StringRegNode Class . . . 371
StructPort Class . . . 372
Synch Class . . . 373
ValueNode Class . . . 385
ChunkAdapterU3V Class . . . 386
IPortRecorder Interface . . . 334
Spinnaker GenApi Interfaces . . . 249
IBase Interface . . . 251
IBoolean Interface . . . 286
ICategory Interfaces . . . 288
IChunkPort Interface . . . 289
ICommand Interface . . . 291
IDestroy Interface . . . 293
IDeviceInfo Interface . . . 294
IEnumEntry Interface . . . 297
IEnumeration Interface . . . 299
IEnumerationT Interface . . . 302
IFloat Interface . . . 304
IInteger Interface . . . 308
INode Interface . . . 310
INodeMap Interface . . . 321
INodeMapDyn Interface . . . 324
IPort Interface . . . 331
IPortConstruct Interface . . . 333
IPortRecorder Interface . . . 334
IRegister Interfaces . . . 336
ISelector Interface . . . 338
ISelectorDigit Interface . . . 339
Reference Interfaces . . . 365
Spinnaker GenApi Utilities . . . 279
GCUtilities Utility . . . 280
Spinnaker GenApi Enums . . . 374
Types Enums . . . 375

Chapter 8

Namespace Index

8.1 Namespace List

Here is a list of all namespaces with brief descriptions:

AdapterConfig 387
Conversion 391
CpuUtil 392
PerformanceCounter 392
SecondsCounter 393
Spinnaker 395
Spinnaker::GenApi 434
Spinnaker::GenICam 451
Spinnaker::Video 453

Chapter 9

Hierarchical Index

9.1 Class Hierarchy

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

ActionCommandResult 455
AdapterInfo 457
AttachStatistics_t 461
AutoLock 462
AutoLock 462
AVIOption 463
BasePtr< T, B > 464
BasePtr< Camera, ICameraBase > 464
    CameraPtr 645
BasePtr< IImage > 464
    ImagePtr 950
BasePtr< IInterface > 464
    InterfacePtr 1006
BasePtr< ISystem > 464
    SystemPtr 1117
BasePtr< LoggingEventData > 464
    LoggingEventDataPtr 1032
basic_istream
    IDevFileStreamBase< CharType, Traits > 874
basic_ostream
    ODevFileStreamBase< CharType, Traits > 1062
basic_streambuf
    IDevFileStreamBuf< CharType, Traits > 877
    ODevFileStreamBuf< CharType, Traits > 1064
BMPOption 468
CChunkAdapter 649
    CChunkAdapterDcam 652
    CChunkAdapterGeneric 655
    CChunkAdapterGEV 657
    CChunkAdapterU3V 659
CDataStruct
    CTestPortStruct< CDataStruct > 760
CEventAdapter 672

CEventAdapter1394 . . . 674
CEventAdapterGeneric . . . 676
CEventAdapterGEV . . . 679
CEventAdapterU3V . . . 681
CGeneric_XMLLoaderParams . . . 693
CNodeMapRefT< GenApi::CGeneric_XMLLoaderParams > . . . 731
CNodeMapRef . . . 729
CGlobalLock . . . 694
CGlobalLockUnlocker . . . 696
CLock . . . 711
CLockEx . . . 715
CLock . . . 712
CLockEx . . . 716
CNodeCallback . . . 717
Function_NodeCallback< Function > . . . 822
Member_NodeCallback< Client, Member > . . . 1037
CNodeMapFactory . . . 720
Counter . . . 739
CPointer< T, B > . . . 741
CPointer< IFloat, IBase > . . . 741
CFloatPtr . . . 691
CPointer< INode, IBase > . . . 741
CpuUsageInfo . . . 753
DCAM_CHECKSUM . . . 763
DCAM_CHUNK_TRAILER . . . 764
double_autovector_t . . . 774
EAccessModeClass . . . 777
ECachingModeClass . . . 778
EDisplayNotationClass . . . 779
EEndianessClass . . . 780
EGenApiSchemaVersionClass . . . 782
EInputDirectionClass . . . 783
ENameSpaceClass . . . 784
ERepresentationClass . . . 793
ESignClass . . . 794
ESlopeClass . . . 795
EStandardNameSpaceClass . . . 796
EventHandler . . . 798
IDeviceArrivalEventHandler . . . 879
DeviceArrivalEventHandler . . . 765
IInterfaceEventHandler . . . 910
InterfaceEventHandler . . . 995
InterfaceEventHandlerImpl . . . 998
InterfaceEventHandlerImpl . . . 998
IDeviceEventHandler . . . 881
DeviceEventHandler . . . 767
DeviceEventHandlerImpl . . . 770
IDeviceRemovalEventHandler . . . 883
DeviceRemovalEventHandler . . . 772
IInterfaceEventHandler . . . 910
IImageEventHandler . . . 897
ImageEventHandler . . . 944
ImageEventHandlerImpl . . . 947
ImageEventHandlerImpl . . . 947
IInterfaceArrivalEventHandler . . . 908
InterfaceArrivalEventHandler . . . 992

ISystemEventHandler . . . 1020
SystemEventHandler . . . 1110
SystemEventHandlerImpl . . . 1113
IInterfaceRemovalEventHandler . . . 916
InterfaceRemovalEventHandler . . . 1008
ISystemEventHandler . . . 1020
ILoggingEventHandler . . . 918
LoggingEventHandler . . . 1034
LoggingEventHandlerImpl . . . 1036
EVisibilityClass . . . 801
exception
AdapterConfigException . . . 456
Exception . . . 802
EYesNoClass . . . 807
FileProtocolAdapter . . . 808
gcstring . . . 824
GrabInfo . . . 833
GVCP_CHUNK_TRAILER . . . 834
GVCP_EVENT_ITEM . . . 835
GVCP_EVENT_ITEM_BASIC . . . 836
GVCP_EVENT_ITEM_EXTENDED_ID . . . 837
GVCP_EVENT_REQUEST . . . 839
GVCP_EVENT_REQUEST_EXTENDED_ID . . . 840
GVCP_EVENTDATA_REQUEST . . . 841
GVCP_EVENTDATA_REQUEST_EXTENDED_ID . . . 842
GVCP_REQUEST_HEADER . . . 843
H264Option . . . 844
IBoolean
BooleanNode . . . 470
ICameraBase . . . 846
CameraBase . . . 625
Camera . . . 473
ICameraList . . . 855
CameraList . . . 640
ICategory
CategoryNode . . . 647
IChunkData . . . 859
ChunkData . . . 698
IChunkPort
PortNode . . . 1069
PortReplay . . . 1078
PortRecorder . . . 1074
ICommand
CommandNode . . . 737
IDataStream . . . 868
IDeviceInfo
NodeMap . . . 1052
SpinTestCamera . . . 1088
IEnumEntry
EnumEntryNode . . . 785
IEnumeration
EnumNode . . . 788
CEnumerationTRef< EnumT > . . . 667
IEnumerationT
CEnumerationTRef< EnumT > . . . 667
IFloat

FloatNode . . . 812
FloatRegNode . . . 819
IImage . . . 885
Image . . . 920
IImageStatistics . . . 899
ImageStatistics . . . 952
IInteger
IntegerNode . . . 981
IntRegNode . . . 1010
IInterface . . . 903
Interface . . . 987
IInterfaceList . . . 913
InterfaceList . . . 1002
ImageUtility . . . 959
ImageUtilityHeatmap . . . 964
ImageUtilityPolarization . . . 968
InferenceBoundingBox . . . 976
InferenceBoundingBoxResult . . . 976
InferenceBoxCircle . . . 977
InferenceBoxRect . . . 977
InferenceBoxRotatedRect . . . 978
INode
Node . . . 1041
CSelectorSet . . . 757
PortNode . . . 1069
ValueNode . . . 1160
BooleanNode . . . 470
CategoryNode . . . 647
CommandNode . . . 737
EnumEntryNode . . . 785
EnumNode . . . 788
FloatNode . . . 812
IntegerNode . . . 981
RegisterNode . . . 1083
FloatRegNode . . . 819
IntRegNode . . . 1010
StringRegNode . . . 1096
StringNode . . . 1092
StringRegNode . . . 1096
INodeMap
NodeMap . . . 1052
int64_autovector_t . . . 978
IPersistScript
CFeatureBag . . . 688
IpInfo . . . 1012
IPortConstruct
CChunkPort . . . 662
CEventPort . . . 683
CPortImpl . . . 746
CRegisterPortImpl . . . 753
CTestPortStruct< CDataStruct > . . . 760
PortNode . . . 1069
IPortRecorder
PortNode . . . 1069
PortRecorder . . . 1074
IPortReplay

CPortImpl 746
PortReplay 1078
IPortWriteList
CPortWriteList 750
IRegister
RegisterNode 1083
IString
StringNode 1092
ISystem 1014
System 1100
IValue
ValueNode 1160
JPEGOption 1022
JPG2Option 1023
LibraryVersion 1025
LockableObject< Object >::Lock 1026
LockableObject< Object > 1027
LoggingEventData 1028
MJPGOption 1039
CNodeMapFactory::NodeStatistics_t 1061
PGMOption 1066
PNGOption 1067
PPMOption 1081
SingleChunkData_t 1086
SingleChunkDataStr_t 1087
SpinVideo 1089
TIFFOption 1119
TransportLayerDevice 1121
TransportLayerInterface 1131
TransportLayerStream 1142
TransportLayerSystem 1150
U3V_CHUNK_TRAILER 1157
U3V_COMMAND_HEADER 1157
U3V_EVENT_DATA 1158
U3V_EVENT_MESSAGE 1159
Version_t 1163
TCameraParams
CNodeMapRefT< TCameraParams > 731

Chapter 10

Class Index

10.1 Class List

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

ActionCommandResult
Action Command Result 455
AdapterConfigException 456
AdapterInfo 457
AttachStatistics_t
Delivers information about the attached chunks and nodes 461
AutoLock 462
AutoLock 462
AVIOption
Options for saving AVI files 463
BasePtr< T, B >
The base class of the SystemPtr, CameraPtr, InterfacePtr, ImagePtr and LoggingEventDataPtr objects 464
BMPOption
Options for saving Bitmap image 468
BooleanNode
Interface for string properties 470
Camera
The camera object class 473
CameraBase
The base class for the camera object 625
CameraList
Used to hold a list of camera objects 640
CameraPtr
A reference tracked pointer to a camera object 645
CategoryNode
Interface for string properties 647
CChunkAdapter
Connects a chunked buffer to a node map 649
CChunkAdapterDcam
Connects a chunked DCAM buffer to a node map 652
CChunkAdapterGeneric 655
CChunkAdapterGEV
Connects a chunked DCAM buffer to a node map 657
CChunkAdapterU3V
Connects a chunked U3V buffer to a node map 659

CChunkPort
Port attachable to a chunk in a buffer . . . 662
CEnumerationTRef< EnumT >
Interface for string properties . . . 667
CEventAdapter
Delivers Events to ports . . . 672
CEventAdapter1394
Distribute the events to the node map . . . 674
CEventAdapterGeneric
Connects a generic event to a node map . . . 676
CEventAdapterGEV
Connects a GigE Event to a node map . . . 679
CEventAdapterU3V
Connects a U3V Event to a node map . . . 681
CEventPort
Port attachable to an event . . . 683
CFeatureBag
Bag holding streamable features of a nodetree . . . 688
CFloatPtr
SmartPointer for IFloat interface pointer . . . 691
CGeneric_XMLLoaderParams
Empty base class used by class CNodeMapRef as generic template argument . . . 693
CGlobalLock
Named global lock which can be used over process boundaries . . . 694
CGlobalLockUnlocker
Unlocks the global lock object on destruction . . . 696
ChunkData
The chunk data which contains additional information about an image . . . 698
CLock
A lock class . . . 711
CLock
A lock class . . . 712
CLockEx
This class is for testing purposes only . . . 715
CLockEx
This class is for testing purposes only . . . 716
CNodeCallback
Callback body instance for INode pointers . . . 717
CNodeMapFactory
The node map factory is used for creating node maps from camera description files . . . 720
CNodeMapRef
Smartpointer for NodeMaps with create function . . . 729
CNodeMapRefT< TCameraParams >
Smartpointer template for NodeMaps with create function . . . 731
CommandNode
Interface for string properties . . . 737
Counter
Definition of a simple Counter class . . . 739
CPointer< T, B >
Encapsulates a GenApi pointer dealing with the dynamic_cast automatically . . . 741
CPortImpl
Standard implementation for a port . . . 746
CPortWriteList
Container holding a list of port write commands . . . 750
CpuUsageInfo . . . 753
CRegisterPortImpl
Standard implementation for a port using a register based transport layer . . . 753

CSelectorSet
The set of selectors selecting a given node . . . 757
CTestPortStruct< CDataStruct >
Implements a register spaces based on a C++ struct . . . 760
DCAM_CHECKSUM . . . 763
DCAM_CHUNK_TRAILER . . . 764
DeviceArrivalEventHandler
An event handler for capturing the device arrival event . . . 765
DeviceEventHandler
A handler to device events . . . 767
DeviceEventHandlerImpl . . . 770
DeviceRemovalEventHandler
An event handler for capturing the device removal event . . . 772
double_autovector_t
Vector of doubles with reference counting . . . 774
EAccessModeClass
Holds conversion methods for the access mode enumeration . . . 777
ECachingModeClass
Holds conversion methods for the caching mode enumeration . . . 778
EDisplayNotationClass
Holds conversion methods for the notation type of floats . . . 779
EEndianessClass
Holds conversion methods for the endianess enumeration . . . 780
EGenApiSchemaVersionClass
Helper class converting EGenApiSchemaVersion from and to string . . . 782
EInputDirectionClass
Holds conversion methods for the notation type of floats . . . 783
ENameSpaceClass
Holds conversion methods for the namespace enumeration . . . 784
EnumEntryNode
Interface for string properties . . . 785
EnumNode
Interface for string properties . . . 788
ERepresentationClass
Holds conversion methods for the representation enumeration . . . 793
ESignClass
Holds conversion methods for the sign enumeration . . . 794
ESlopeClass
Holds conversion methods for the converter formulas . . . 795
EStandardNameSpaceClass
Holds conversion methods for the standard namespace enumeration . . . 796
EventHandler
The base class for all event handler types . . . 798
EVisibilityClass
Holds conversion methods for the visibility enumeration . . . 801
Exception
The Exception object represents an error that is returned from the library . . . 802
EYesNoClass
Holds conversion methods for the standard namespace enumeration . . . 807
FileProtocolAdapter
Adapter between the std::iostreambuf and the SFNC Features representing the device file system . . . 808
FloatNode
Interface for string properties . . . 812
FloatRegNode
Interface for string properties . . . 819
Function_NodeCallback< Function >
Container for a function pointer . . . 822

gcstring . . . 824
GrabInfo . . . 833
GVCP_CHUNK_TRAILER
Header of a GVCP request packet . . . 834
GVCP_EVENT_ITEM
Layout of a GVCP event item (Extended ID flag not set) . . . 835
GVCP_EVENT_ITEM_BASIC
Layout of a GVCP event item (common to all types) . . . 836
GVCP_EVENT_ITEM_EXTENDED_ID
Layout of a GVCP event item (Extended ID flag set) . . . 837
GVCP_EVENT_REQUEST
Layout of a GVCP event request packet (Extended ID flag not set) . . . 839
GVCP_EVENT_REQUEST_EXTENDED_ID
Layout of a GVCP event request packet (Extended ID flag set) . . . 840
GVCP_EVENTDATA_REQUEST
Layout of a GVCP event data request packet (Extended ID flag not set) . . . 841
GVCP_EVENTDATA_REQUEST_EXTENDED_ID
Layout of a GVCP event data request packet (Extended ID flag set) . . . 842
GVCP_REQUEST_HEADER
Header of a GVCP request packet . . . 843
H264Option
Options for saving H264 files . . . 844
ICameraBase
The interface file for base class for the camera object . . . 846
ICameraList
Used to hold a list of camera objects . . . 855
IChunkData
The Interface file for ChunkData . . . 859
IDataStream . . . 868
IDevFileStreamBase< CharType, Traits > . . . 874
IDevFileStreamBuf< CharType, Traits > . . . 877
IDeviceArrivalEventHandler . . . 879
IDeviceEventHandler . . . 881
IDeviceRemovalEventHandler . . . 883
IImage
The interface file for Image . . . 885
IImageEventHandler . . . 897
IImageStatistics
The interface file for image statistics . . . 899
IInterface
The interface file for Interface . . . 903
IInterfaceArrivalEventHandler . . . 908
IInterfaceEventHandler . . . 910
IInterfaceList
The interface file for InterfaceList class . . . 913
IInterfaceRemovalEventHandler . . . 916
ILoggingEventHandler . . . 918
Image
The image object class . . . 920
ImageEventHandler
A handler for capturing image arrival events . . . 944
ImageEventHandlerImpl . . . 947
ImagePtr
A reference tracked pointer to an image object . . . 950
ImageStatistics
Represents image statistics for an image . . . 952
ImageUtility
Static helper functions for the image object class . . . 959

ImageUtilityHeatmap
Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16 964
ImageUtilityPolarization
Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8 . . . 968
InferenceBoundingBox
Inference Bounding Boxes data structure . . . 976
InferenceBoundingBoxResult
An inference bounding boxes object which holds information about the detected bounding boxes 976
InferenceBoxCircle . . . 977
InferenceBoxRect
Inference Bounding Box Type Data Structures . . . 977
InferenceBoxRotatedRect . . . 978
int64_autovector_t
Vector of integers with reference counting . . . 978
IntegerNode
Interface for string properties . . . 981
Interface
An interface object which holds a list of cameras . . . 987
InterfaceArrivalEventHandler
An event handler for capturing the interface arrival event . . . 992
InterfaceEventHandler
A handler to device arrival and removal events on all interfaces . . . 995
InterfaceEventHandlerImpl . . . 998
InterfaceList
A list of the available interfaces on the system . . . 1002
InterfacePtr
A reference tracked pointer to the interface object . . . 1006
InterfaceRemovalEventHandler
An event handler for capturing the interface removal event . . . 1008
IntRegNode
Interface for string properties . . . 1010
IpInfo . . . 1012
ISystem
The interface file for System . . . 1014
ISystemEventHandler . . . 1020
JPEGOption
Options for saving JPEG image . . . 1022
JPG2Option
Options for saving JPEG2000 image . . . 1023
LibraryVersion
Provides easier access to the current version of Spinnaker . . . 1025
LockableObject< Object >::Lock
A scopelevel Lock class . . . 1026
LockableObject< Object >
Instance-Lock for an object . . . 1027
LoggingEventData
The LoggingEventData object . . . 1028
LoggingEventDataPtr
A reference tracked pointer to the LoggingEvent object . . . 1032
LoggingEventHandler
An event handler for capturing the device logging event . . . 1034
LoggingEventHandlerImpl . . . 1036
Member_NodeCallback< Client, Member >
Container for a member function pointer . . . 1037
MJPGOption
Options for saving MJPG files . . . 1039

Node
Class common to all nodes . . . 1041
NodeMap
Smart pointer template for NodeMaps with create function . . . 1052
CNodeMapFactory::NodeStatistics_t . . . 1061
ODevFileStreamBase< CharType, Traits > . . . 1062
ODevFileStreamBuf< CharType, Traits > . . . 1064
PGMOption
Options for saving PGM images . . . 1066
PNGOption
Options for saving PNG images . . . 1067
PortNode
Interface for value properties . . . 1069
PortRecorder
Interface for recording write commands on a port . . . 1074
PortReplay
Interface for replaying write commands on a port . . . 1078
PPMOption
Options for saving PPM images . . . 1081
RegisterNode
Interface for string properties . . . 1083
SingleChunkData_t . . . 1086
SingleChunkDataStr_t . . . 1087
SpinTestCamera . . . 1088
SpinVideo
Provides the functionality for the user to record images to an AVI/MP4 file . . . 1089
StringNode
Interface for string properties . . . 1092
StringRegNode
Interface for string properties . . . 1096
System
The system object is used to retrieve the list of interfaces and cameras available . . . 1100
SystemEventHandler
A handler to interface arrival and removal events on the system . . . 1110
SystemEventHandlerImpl . . . 1113
SystemPtr
A reference tracked pointer to a system object . . . 1117
TIFFOption
Options for saving TIFF images . . . 1119
TransportLayerDevice
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . 1121
TransportLayerInterface
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . 1131
TransportLayerStream
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . 1142
TransportLayerSystem
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . 1150
U3V_CHUNK_TRAILER
Header of a GVCP request packet . . . 1157
U3V_COMMAND_HEADER
U3V/GenCP command header . . . 1157
U3V_EVENT_DATA
U3V/GenCP EVENT_CMD specific command data . . . 1158

U3V_EVENT_MESSAGE
Entire event data message (without the variable-sized data field) 1159
ValueNode
Interface for value properties 1160
Version_t
Version 1163

Chapter 11

File Index

11.1 File List

Here is a list of all files with brief descriptions:

include/AdapterConfig.h . . . 1165
include/AVIRecorder.h . . . 1167
include/BasePtr.h . . . 1168
include/Camera.h . . . 1169
include/CameraBase.h . . . 1169
include/CameraDefs.h . . . 1170
include/CameraList.h . . . 1202
include/CameraPtr.h . . . 1203
include/ChunkData.h . . . 1203
include/ChunkDataInference.h . . . 1204
include/DeviceArrivalEventHandler.h . . . 1205
include/DeviceEventHandler.h . . . 1206
include/DeviceRemovalEventHandler.h . . . 1207
include/EventHandler.h . . . 1208
include/Exception.h . . . 1208
include/Image.h . . . 1209
include/ImageEventHandler.h . . . 1210
include/ImagePtr.h . . . 1210
include/ImageStatistics.h . . . 1211
include/ImageUtility.h . . . 1212
include/ImageUtilityHeatmap.h . . . 1213
include/ImageUtilityPolarization.h . . . 1214
include/Interface.h . . . 1215
include/InterfaceArrivalEventHandler.h . . . 1230
include/InterfaceEventHandler.h . . . 1231
include/InterfaceList.h . . . 1232
include/InterfacePtr.h . . . 1232
include/InterfaceRemovalEventHandler.h . . . 1233
include/LoggingEventData.h . . . 1234
include/LoggingEventDataPtr.h . . . 1234
include/LoggingEventHandler.h . . . 1235
include/Spinnaker.h . . . 1322
include/SpinnakerDefs.h . . . 1322
include/SpinnakerPlatform.h . . . 1326
include/SpinUpdate.h . . . 1327

include/SpinVideo.h . . . 1329
include/SpinVideoDefs.h . . . 1330
include/System.h . . . 1331
include/SystemEventHandler.h . . . 1333
include/SystemPtr.h . . . 1334
include/TransportLayerDefs.h . . . 1334
include/TransportLayerDevice.h . . . 1336
include/TransportLayerInterface.h . . . 1337
include/TransportLayerStream.h . . . 1337
include/TransportLayerSystem.h . . . 1338
include/Interface/ICameraBase.h . . . 1215
include/Interface/ICameraList.h . . . 1216
include/Interface/IChunkData.h . . . 1217
include/Interface/IDeviceArrivalEventHandler.h . . . 1218
include/Interface/IDeviceEventHandler.h . . . 1219
include/Interface/IDeviceRemovalEventHandler.h . . . 1220
include/Interface/IImage.h . . . 1221
include/Interface/IImageEventHandler.h . . . 1221
include/Interface/IImageStatistics.h . . . 1222
include/Interface/IInterface.h . . . 1223
include/Interface/IInterfaceArrivalEventHandler.h . . . 1224
include/Interface/IInterfaceEventHandler.h . . . 1225
include/Interface/IInterfaceList.h . . . 1225
include/Interface/IInterfaceRemovalEventHandler.h . . . 1226
include/Interface/ILoggingEventHandler.h . . . 1227
include/Interface/IStream.h . . . 1227
include/Interface/ISystem.h . . . 1228
include/Interface/ISystemEventHandler.h . . . 1229
include/SpinGenApi/Autovector.h . . . 1236
include/SpinGenApi/Base.h . . . 1237
include/SpinGenApi/BooleanNode.h . . . 1238
include/SpinGenApi/CategoryNode.h . . . 1239
include/SpinGenApi/ChunkAdapter.h . . . 1240
include/SpinGenApi/ChunkAdapterDcam.h . . . 1241
include/SpinGenApi/ChunkAdapterGeneric.h . . . 1242
include/SpinGenApi/ChunkAdapterGEV.h . . . 1243
include/SpinGenApi/ChunkAdapterU3V.h . . . 1244
include/SpinGenApi/ChunkPort.h . . . 1245
include/SpinGenApi/CommandNode.h . . . 1245
include/SpinGenApi/Compatibility.h . . . 1246
include/SpinGenApi/Container.h . . . 1247
include/SpinGenApi/Counter.h . . . 1247
include/SpinGenApi/EnumClasses.h . . . 1248
include/SpinGenApi/EnumEntryNode.h . . . 1249
include/SpinGenApi/EnumNode.h . . . 1250
include/SpinGenApi/EnumNodeT.h . . . 1251
include/SpinGenApi/EventAdapter.h . . . 1251
include/SpinGenApi/EventAdapter1394.h . . . 1252
include/SpinGenApi/EventAdapterGeneric.h . . . 1253
include/SpinGenApi/EventAdapterGEV.h . . . 1253
include/SpinGenApi/EventAdapterU3V.h . . . 1254
include/SpinGenApi/EventPort.h . . . 1255
include/SpinGenApi/Filestream.h . . . 1256
include/SpinGenApi/FloatNode.h . . . 1257
include/SpinGenApi/FloatRegNode.h . . . 1258
include/SpinGenApi/GCBase.h . . . 1259
include/SpinGenApi/GCString.h . . . 1259
include/SpinGenApi/GCStringVector.h . . . 1261

include/SpinGenApi/GCSynch.h . . . 1262
include/SpinGenApi/GCTypes.h . . . 1263
include/SpinGenApi/GCUtilities.h . . . 1266
include/SpinGenApi/IBoolean.h . . . 1270
include/SpinGenApi/ICategory.h . . . 1271
include/SpinGenApi/IChunkPort.h . . . 1272
include/SpinGenApi/ICommand.h . . . 1273
include/SpinGenApi/IDestroy.h . . . 1274
include/SpinGenApi/IDeviceInfo.h . . . 1275
include/SpinGenApi/IEnumEntry.h . . . 1276
include/SpinGenApi/IEnumeration.h . . . 1277
include/SpinGenApi/IEnumerationT.h . . . 1278
include/SpinGenApi/IFloat.h . . . 1279
include/SpinGenApi/IInteger.h . . . 1281
include/SpinGenApi/INode.h . . . 1282
include/SpinGenApi/INodeMap.h . . . 1285
include/SpinGenApi/INodeMapDyn.h . . . 1287
include/SpinGenApi/IntegerNode.h . . . 1288
include/SpinGenApi/IntRegNode.h . . . 1289
include/SpinGenApi/IPort.h . . . 1290
include/SpinGenApi/IPortConstruct.h . . . 1291
include/SpinGenApi/IPortRecorder.h . . . 1292
include/SpinGenApi/IRegister.h . . . 1293
include/SpinGenApi/ISelector.h . . . 1294
include/SpinGenApi/ISelectorDigit.h . . . 1295
include/SpinGenApi/IString.h . . . 1296
include/SpinGenApi/IValue.h . . . 1297
include/SpinGenApi/Node.h . . . 1298
include/SpinGenApi/NodeCallback.h . . . 1299
include/SpinGenApi/NodeCallbackImpl.h . . . 1301
include/SpinGenApi/NodeMap.h . . . 1301
include/SpinGenApi/NodeMapFactory.h . . . 1302
include/SpinGenApi/NodeMapRef.h . . . 1303
include/SpinGenApi/Persistence.h . . . 1304
include/SpinGenApi/Pointer.h . . . 1305
include/SpinGenApi/PortImpl.h . . . 1307
include/SpinGenApi/PortNode.h . . . 1308
include/SpinGenApi/PortRecorder.h . . . 1309
include/SpinGenApi/PortReplay.h . . . 1309
include/SpinGenApi/PortWriteList.h . . . 1310
include/SpinGenApi/Reference.h . . . 1311
include/SpinGenApi/RegisterNode.h . . . 1312
include/SpinGenApi/RegisterPortImpl.h . . . 1313
include/SpinGenApi/SelectorSet.h . . . 1313
include/SpinGenApi/SpinnakerGenApi.h . . . 1314
include/SpinGenApi/SpinTestCamera.h . . . 1314
include/SpinGenApi/StringNode.h . . . 1315
include/SpinGenApi/StringRegNode.h . . . 1316
include/SpinGenApi/StructPort.h . . . 1316
include/SpinGenApi/Synch.h . . . 1317
include/SpinGenApi/Types.h . . . 1318
include/SpinGenApi/ValueNode.h . . . 1321
src/Acquisition/Acquisition.cpp . . . 1338
src/Acquisition/resource.h . . . 1340
src/Acquisition/stdafx.cpp . . . 1342
src/Acquisition/stdafx.h . . . 1351
src/Acquisition/targetver.h . . . 1367
src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp . . . 1382

src/AcquisitionMultipleCameraRecovery/resource.h . . . 1340
src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp . . . 1384
src/AcquisitionMultipleThread/resource.h . . . 1340
src/ActionCommand/ActionCommand.cpp . . . 1385
src/ActionCommand/resource.h . . . 1340
src/ActionCommand/stdafx.cpp . . . 1342
src/ActionCommand/stdafx.h . . . 1352
src/ActionCommand/targetver.h . . . 1368
src/BufferHandling/BufferHandling.cpp . . . 1387
src/BufferHandling/resource.h . . . 1340
src/BufferHandling/stdafx.cpp . . . 1343
src/BufferHandling/stdafx.h . . . 1353
src/BufferHandling/targetver.h . . . 1369
src/ChunkData/ChunkData.cpp . . . 1390
src/ChunkData/resource.h . . . 1340
src/CounterAndTimer/CounterAndTimer.cpp . . . 1392
src/CounterAndTimer/resource.h . . . 1340
src/CounterAndTimer/stdafx.cpp . . . 1344
src/CounterAndTimer/stdafx.h . . . 1354
src/CounterAndTimer/targetver.h . . . 1370
src/DeviceEvents/DeviceEvents.cpp . . . 1394
src/DeviceEvents/resource.h . . . 1340
src/DeviceEvents/stdafx.cpp . . . 1344
src/DeviceEvents/stdafx.h . . . 1355
src/DeviceEvents/targetver.h . . . 1371
src/Enumeration/Enumeration.cpp . . . 1396
src/Enumeration/resource.h . . . 1340
src/Enumeration/stdafx.cpp . . . 1345
src/Enumeration/stdafx.h . . . 1356
src/Enumeration/targetver.h . . . 1372
src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp . . . 1397
src/Enumeration_QuickSpin/resource.h . . . 1340
src/Enumeration_QuickSpin/stdafx.cpp . . . 1345
src/Enumeration_QuickSpin/stdafx.h . . . 1357
src/Enumeration_QuickSpin/targetver.h . . . 1373
src/EnumerationEvents/EnumerationEvents.cpp . . . 1397
src/EnumerationEvents/resource.h . . . 1340
src/ExceptionHandling/ExceptionHandling.cpp . . . 1398
src/ExceptionHandling/resource.h . . . 1340
src/ExceptionHandling/stdafx.cpp . . . 1346
src/ExceptionHandling/stdafx.h . . . 1358
src/ExceptionHandling/targetver.h . . . 1374
src/Exposure/Exposure.cpp . . . 1400
src/Exposure/resource.h . . . 1340
src/Exposure/stdafx.cpp . . . 1346
src/Exposure/stdafx.h . . . 1359
src/Exposure/targetver.h . . . 1375
src/Exposure_QuickSpin/Exposure_QuickSpin.cpp . . . 1401
src/Exposure_QuickSpin/resource.h . . . 1340
src/Exposure_QuickSpin/stdafx.cpp . . . 1347
src/Exposure_QuickSpin/stdafx.h . . . 1360
src/Exposure_QuickSpin/targetver.h . . . 1376
src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp . . . 1403
src/FileAccess_QuickSpin/resource.h . . . 1340
src/FileAccess_QuickSpin/stdafx.cpp . . . 1348
src/FileAccess_QuickSpin/stdafx.h . . . 1361
src/FileAccess_QuickSpin/targetver.h . . . 1377
src/GenTLInfo_QuickSpin/GenTLInfo_QuickSpin.cpp . . . 1406

src/GenTLInfo_QuickSpin/resource.h . . . 1341
src/GenTLInfo_QuickSpin/targetver.h . . . 1379
src/GigEVisionPerformance/CpuUtil.cpp . . . 1407
src/GigEVisionPerformance/CpuUtil.h . . . 1408
src/GigEVisionPerformance/GigEVisionPerformance.cpp . . . 1409
src/GigEVisionPerformance/GigEVisionPerformance.h . . . 1414
src/GigEVisionPerformance/resource.h . . . 1340
src/GigEVisionPerformance/stdafx.cpp . . . 1349
src/GigEVisionPerformance/stdafx.h . . . 1362
src/GigEVisionPerformance/targetver.h . . . 1378
src/HighDynamicRange/HighDynamicRange.cpp . . . 1415
src/HighDynamicRange/resource.h . . . 1340
src/ImageEvents/ImageEvents.cpp . . . 1418
src/ImageEvents/resource.h . . . 1341
src/ImageFormatControl/ImageFormatControl.cpp . . . 1420
src/ImageFormatControl/resource.h . . . 1341
src/ImageFormatControl/stdafx.h . . . 1363
src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp . . . 1421
src/ImageFormatControl_QuickSpin/resource.h . . . 1341
src/ImageFormatControl_QuickSpin/stdafx.h . . . 1363
src/Inference/Inference.cpp . . . 1422
src/Inference/resource.h . . . 1341
src/Logging/Logging.cpp . . . 1429
src/Logging/resource.h . . . 1341
src/LogicBlock/LogicBlock.cpp . . . 1430
src/LogicBlock/resource.h . . . 1341
src/LookupTable/LookupTable.cpp . . . 1432
src/LookupTable/resource.h . . . 1341
src/NodeMapCallback/NodeMapCallback.cpp . . . 1433
src/NodeMapCallback/resource.h . . . 1341
src/NodeMapInfo/NodeMapInfo.cpp . . . 1435
src/NodeMapInfo/resource.h . . . 1341
src/NodeMapInfo/stdafx.cpp . . . 1349
src/NodeMapInfo/stdafx.h . . . 1364
src/NodeMapInfo/targetver.h . . . 1379
src/Polarization/Polarization.cpp . . . 1439
src/Polarization/resource.h . . . 1341
src/Polarization/stdafx.h . . . 1365
src/SaveToAvi/resource.h . . . 1341
src/SaveToAvi/SaveToAvi.cpp . . . 1442
src/Sequencer/resource.h . . . 1341
src/Sequencer/Sequencer.cpp . . . 1444
src/Sequencer/stdafx.cpp . . . 1350
src/Sequencer/stdafx.h . . . 1365
src/Sequencer/targetver.h . . . 1380
src/SerialRxTx/resource.h . . . 1341
src/SerialRxTx/SerialRxTx.cpp . . . 1446
src/SerialRxTx/stdafx.cpp . . . 1350
src/SerialRxTx/stdafx.h . . . 1366
src/SerialRxTx/targetver.h . . . 1381
src/Trigger/resource.h . . . 1341
src/Trigger/Trigger.cpp . . . 1449
src/Trigger_QuickSpin/resource.h . . . 1341
src/Trigger_QuickSpin/Trigger_QuickSpin.cpp . . . 1452

Chapter 12

Module Documentation

12.1 Spinnaker Classes

Collaboration diagram for Spinnaker Classes:

Modules

- AVI Recorder Class
- BasePtr Class

- Camera Class
- Camera Base Class
- CameraDefs Class
- Camera List Class
- CameraPtr Class
- ChunkData Class
- Chunk Data Inference Class
- Exception Class
- Image Class
- ImagePtr Class
- ImageStatistics Class
- Image Utility Class
- Image Utility 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 {
  INFERENCE_BOX_TYPE_RECTANGLE = 0,
  INFERENCE_BOX_TYPE_CIRCLE = 1,
  INFERENCE_BOX_TYPE_ROTATED_RECTANGLE = 2 }

  Inference Bounding Box Type.

12.1.1 Detailed Description

12.1.2 Enumeration Type Documentation

12.1.2.1 InferenceBoxType

```
enum InferenceBoxType
```

Inference Bounding Box Type.

Enumerator

INFERENCE_BOX_TYPE_RECTANGLE	
INFERENCE_BOX_TYPE_CIRCLE	
INFERENCE_BOX_TYPE_ROTATED_RECTANGLE	

12.2 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.

12.2.1 Detailed Description

12.2.2 Function Documentation

12.2.2.1 DEPRECATED_CLASS()

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

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

NOTE: This class is deprecated and replaced by SpinVideo. Refer to SpinVideo.h instead. Default constructor.

Default destructor.

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

Parameters

pFileName	The filename of the AVI file.
pOption	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

pFileName	The filename of the AVI file.
pOption	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

pFileName	The filename of the MP4 file.
pOption	H264 options to apply to the MP4 file.

See also

AVIClose()
H264Option

Append an image to the AVI/MP4 file.

Parameters

pImage	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

size	The maximum AVI file size in MB.

See also

AVIAppend(ImagePtr pImage)

12.3 BasePtr Class

Collaboration diagram for BasePtr Class:

Classes

- class BasePtr< T, B >

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

Functions

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

  Pointer equal.

12.3.1 Detailed Description

12.3.2 Function Documentation

12.3.2.1 operator==()

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

Pointer equal.

12.4 Camera Class

Collaboration diagram for Camera Class:

Classes

- class Camera

  The camera object class.

12.4.1 Detailed Description

12.5 Camera Base Class

Collaboration diagram for Camera Base Class:

Classes

- class CameraBase

  The base class for the camera object.

12.5.1 Detailed Description

12.6 CameraDefs Class

Collaboration diagram for CameraDefs Class:

Enumerations

- enum LUTSelectorEnums {
  LUTSelector_LUT1,
  NUM_LUTSELECTOR }

  The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.
- enum ExposureModeEnums {
  ExposureMode_Timed,
  ExposureMode_TriggerWidth,
  NUM_EXPOSUREMODE }
- enum AcquisitionModeEnums {
  AcquisitionMode_Continuous,
  AcquisitionMode_SingleFrame,
  AcquisitionMode_MultiFrame,
  NUM_ACQUISITIONMODE }
- enum TriggerSourceEnums {
  TriggerSource_Software,
  TriggerSource_Line0,
  TriggerSource_Line1,
  TriggerSource_Line2,
  TriggerSource_Line3,
  TriggerSource_UserOutput0,
  TriggerSource_UserOutput1,
  TriggerSource_UserOutput2,
  TriggerSource_UserOutput3,
  TriggerSource_Counter0Start,
  TriggerSource_Counter1Start,
  TriggerSource_Counter0End,
  TriggerSource_Counter1End,
  TriggerSource_LogicBlock0,
  TriggerSource_LogicBlock1,
  TriggerSource_Action0,
  NUM_TRIGGERSOURCE }
- enum TriggerActivationEnums {
  TriggerActivation_LevelLow,
  TriggerActivation_LevelHigh,
  TriggerActivation_FallingEdge,
  TriggerActivation_RisingEdge,
  TriggerActivation_AnyEdge,
  NUM_TRIGGERACTIVATION }

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

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

- enum LogicBlockLUTSelectorEnums {
  LogicBlockLUTSelector_Value,
  LogicBlockLUTSelector_Enable,
  NUM_LOGICBLOCKLUTSELECTOR }
- enum ColorTransformationSelectorEnums {
  ColorTransformationSelector_RGBtoRGB,
  ColorTransformationSelector_RGBtoYUV,
  NUM_COLORTRANSFORMATIONSELECTOR }
- enum RgbTransformLightSourceEnums {
  RgbTransformLightSource_General,
  RgbTransformLightSource_Tungsten2800K,
  RgbTransformLightSource_WarmFluorescent3000K,
  RgbTransformLightSource_CoolFluorescent4000K,
  RgbTransformLightSource_Daylight5000K,
  RgbTransformLightSource_Cloudy6500K,
  RgbTransformLightSource_Shade8000K,
  RgbTransformLightSource_Custom,
  NUM_RGBTRANSFORMLIGHTSOURCE }
- enum ColorTransformationValueSelectorEnums {
  ColorTransformationValueSelector_Gain00,
  ColorTransformationValueSelector_Gain01,
  ColorTransformationValueSelector_Gain02,
  ColorTransformationValueSelector_Gain10,
  ColorTransformationValueSelector_Gain11,
  ColorTransformationValueSelector_Gain12,
  ColorTransformationValueSelector_Gain20,
  ColorTransformationValueSelector_Gain21,
  ColorTransformationValueSelector_Gain22,
  ColorTransformationValueSelector_Offset0,
  ColorTransformationValueSelector_Offset1,
  ColorTransformationValueSelector_Offset2,
  NUM_COLORTRANSFORMATIONVALUESELECTOR }
- enum DeviceRegistersEndiannessEnums {
  DeviceRegistersEndianness_Little,
  DeviceRegistersEndianness_Big,
  NUM_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_DEVICETLTYPE }

- enum DevicePowerSupplySelectorEnums {
DevicePowerSupplySelector_External,
NUM_DEVICEPOWERSUPPLYSELECTOR }
- enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICETEMPERATURESELECTOR }
- enum DeviceIndicatorModeEnums {
DeviceIndicatorMode_Inactive,
DeviceIndicatorMode_Active,
DeviceIndicatorMode_ErrorStatus,
NUM_DEVICEINDICATORMODE }
- enum AutoExposureControlPriorityEnums {
AutoExposureControlPriority_Gain,
AutoExposureControlPriority_ExposureTime,
NUM_AUTOEXPOSURECONTROLPRIORITY }
- enum AutoExposureMeteringModeEnums {
AutoExposureMeteringMode_Average,
AutoExposureMeteringMode_Spot,
AutoExposureMeteringMode_Partial,
AutoExposureMeteringMode_CenterWeighted,
AutoExposureMeteringMode_HistgramPeak,
NUM_AUTOEXPOSUREMETERINGMODE }
- enum BalanceWhiteAutoProfileEnums {
BalanceWhiteAutoProfile_Indoor,
BalanceWhiteAutoProfile_Outdoor,
NUM_BALANCEWHITEAUTOPROFILE }
- enum AutoAlgorithmSelectorEnums {
AutoAlgorithmSelector_Awb,
AutoAlgorithmSelector_Ae,
NUM_AUTOALGORITHMSELECTOR }
- enum AutoExposureTargetGreyValueAutoEnums {
AutoExposureTargetGreyValueAuto_Off,
AutoExposureTargetGreyValueAuto_Continuous,
NUM_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum AutoExposureLightingModeEnums {
AutoExposureLightingMode_AutoDetect,
AutoExposureLightingMode_Backlight,
AutoExposureLightingMode_Frontlight,
AutoExposureLightingMode_Normal,
NUM_AUTOEXPOSURELIGHTINGMODE }
- enum GevIEEE1588StatusEnums {
GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,
GevIEEE1588Status_Listening,
GevIEEE1588Status_PreMaster,
GevIEEE1588Status_Master,
GevIEEE1588Status_Passive,
GevIEEE1588Status_Uncalibrated,
GevIEEE1588Status_Slave,
NUM_GEVIEEE1588STATUS }

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

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

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

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

PixelColorFilter_BayerBG,
NUM_PIXELCOLORFILTER }

- enum AdcBitDepthEnums {
AdcBitDepth_Bit8,
AdcBitDepth_Bit10,
AdcBitDepth_Bit12,
AdcBitDepth_Bit14,
NUM_ADCBITDEPTH }
- enum DecimationHorizontalModeEnums {
DecimationHorizontalMode_Discard,
NUM_DECIMATIONHORIZONTALMODE }
- enum BinningVerticalModeEnums {
BinningVerticalMode_Sum,
BinningVerticalMode_Average,
NUM_BINNINGVERTICALMODE }
- enum PixelSizeEnums {
PixelSize_Bpp1,
PixelSize_Bpp2,
PixelSize_Bpp4,
PixelSize_Bpp8,
PixelSize_Bpp10,
PixelSize_Bpp12,
PixelSize_Bpp14,
PixelSize_Bpp16,
PixelSize_Bpp20,
PixelSize_Bpp24,
PixelSize_Bpp30,
PixelSize_Bpp32,
PixelSize_Bpp36,
PixelSize_Bpp48,
PixelSize_Bpp64,
PixelSize_Bpp96,
NUM_PIXELSIZE }
- enum DecimationSelectorEnums {
DecimationSelector_All,
DecimationSelector_Sensor,
NUM_DECIMATIONSELECTOR }
- enum ImageCompressionModeEnums {
ImageCompressionMode_Off,
ImageCompressionMode_Lossless,
NUM_IMAGECOMPRESSIONMODE }
- enum BinningHorizontalModeEnums {
BinningHorizontalMode_Sum,
BinningHorizontalMode_Average,
NUM_BINNINGHORIZONTALMODE }
- enum PixelFormatEnums {
PixelFormat_Mono8,
PixelFormat_Mono16,
PixelFormat_RGB8Packed,
PixelFormat_BayerGR8,
PixelFormat_BayerRG8,
PixelFormat_BayerGB8,
PixelFormat_BayerBG8,
PixelFormat_BayerGR16,
PixelFormat_BayerRG16,
PixelFormat_BayerGB16,
PixelFormat_BayerBG16,
PixelFormat_Mono12Packed,

PixelFormat_BayerGR12Packed,
PixelFormat_BayerRG12Packed,
PixelFormat_BayerGB12Packed,
PixelFormat_BayerBG12Packed,
PixelFormat_YUV411Packed,
PixelFormat_YUV422Packed,
PixelFormat_YUV444Packed,
PixelFormat_Mono12p,
PixelFormat_BayerGR12p,
PixelFormat_BayerRG12p,
PixelFormat_BayerGB12p,
PixelFormat_BayerBG12p,
PixelFormat_YCbCr8,
PixelFormat_YCbCr422_8,
PixelFormat_YCbCr411_8,
PixelFormat_BGR8,
PixelFormat_BGRa8,
PixelFormat_Mono10Packed,
PixelFormat_BayerGR10Packed,
PixelFormat_BayerRG10Packed,
PixelFormat_BayerGB10Packed,
PixelFormat_BayerBG10Packed,
PixelFormat_Mono10p,
PixelFormat_BayerGR10p,
PixelFormat_BayerRG10p,
PixelFormat_BayerGB10p,
PixelFormat_BayerBG10p,
PixelFormat_Mono1p,
PixelFormat_Mono2p,
PixelFormat_Mono4p,
PixelFormat_Mono8s,
PixelFormat_Mono10,
PixelFormat_Mono12,
PixelFormat_Mono14,
PixelFormat_Mono16s,
PixelFormat_Mono32f,
PixelFormat_BayerBG10,
PixelFormat_BayerBG12,
PixelFormat_BayerGB10,
PixelFormat_BayerGB12,
PixelFormat_BayerGR10,
PixelFormat_BayerGR12,
PixelFormat_BayerRG10,
PixelFormat_BayerRG12,
PixelFormat_RGBa8,
PixelFormat_RGBa10,
PixelFormat_RGBa10p,
PixelFormat_RGBa12,
PixelFormat_RGBa12p,
PixelFormat_RGBa14,
PixelFormat_RGBa16,
PixelFormat_RGB8,
PixelFormat_RGB8_Planar,
PixelFormat_RGB10,
PixelFormat_RGB10_Planar,
PixelFormat_RGB10p,
PixelFormat_RGB10p32,
PixelFormat_RGB12,

PixelFormat_RGB12_Planar,
PixelFormat_RGB12p,
PixelFormat_RGB14,
PixelFormat_RGB16,
PixelFormat_RGB16s,
PixelFormat_RGB32f,
PixelFormat_RGB16_Planar,
PixelFormat_RGB565p,
PixelFormat_BGRa10,
PixelFormat_BGRa10p,
PixelFormat_BGRa12,
PixelFormat_BGRa12p,
PixelFormat_BGRa14,
PixelFormat_BGRa16,
PixelFormat_RGBa32f,
PixelFormat_BGR10,
PixelFormat_BGR10p,
PixelFormat_BGR12,
PixelFormat_BGR12p,
PixelFormat_BGR14,
PixelFormat_BGR16,
PixelFormat_BGR565p,
PixelFormat_R8,
PixelFormat_R10,
PixelFormat_R12,
PixelFormat_R16,
PixelFormat_G8,
PixelFormat_G10,
PixelFormat_G12,
PixelFormat_G16,
PixelFormat_B8,
PixelFormat_B10,
PixelFormat_B12,
PixelFormat_B16,
PixelFormat_Coord3D_ABC8,
PixelFormat_Coord3D_ABC8_Planar,
PixelFormat_Coord3D_ABC10p,
PixelFormat_Coord3D_ABC10p_Planar,
PixelFormat_Coord3D_ABC12p,
PixelFormat_Coord3D_ABC12p_Planar,
PixelFormat_Coord3D_ABC16,
PixelFormat_Coord3D_ABC16_Planar,
PixelFormat_Coord3D_ABC32f,
PixelFormat_Coord3D_ABC32f_Planar,
PixelFormat_Coord3D_AC8,
PixelFormat_Coord3D_AC8_Planar,
PixelFormat_Coord3D_AC10p,
PixelFormat_Coord3D_AC10p_Planar,
PixelFormat_Coord3D_AC12p,
PixelFormat_Coord3D_AC12p_Planar,
PixelFormat_Coord3D_AC16,
PixelFormat_Coord3D_AC16_Planar,
PixelFormat_Coord3D_AC32f,
PixelFormat_Coord3D_AC32f_Planar,
PixelFormat_Coord3D_A8,
PixelFormat_Coord3D_A10p,
PixelFormat_Coord3D_A12p,
PixelFormat_Coord3D_A16,

PixelFormat_Coord3D_A32f,
PixelFormat_Coord3D_B8,
PixelFormat_Coord3D_B10p,
PixelFormat_Coord3D_B12p,
PixelFormat_Coord3D_B16,
PixelFormat_Coord3D_B32f,
PixelFormat_Coord3D_C8,
PixelFormat_Coord3D_C10p,
PixelFormat_Coord3D_C12p,
PixelFormat_Coord3D_C16,
PixelFormat_Coord3D_C32f,
PixelFormat_Confidence1,
PixelFormat_Confidence1p,
PixelFormat_Confidence8,
PixelFormat_Confidence16,
PixelFormat_Confidence32f,
PixelFormat_BiColorBGRG8,
PixelFormat_BiColorBGRG10,
PixelFormat_BiColorBGRG10p,
PixelFormat_BiColorBGRG12,
PixelFormat_BiColorBGRG12p,
PixelFormat_BiColorRGBG8,
PixelFormat_BiColorRGBG10,
PixelFormat_BiColorRGBG10p,
PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,
PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,
PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,
PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,
PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,

PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,
PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,
PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,

PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

- enum DecimationVerticalModeEnums {
  DecimationVerticalMode_Discard,
  NUM_DECIMATIONVERTICALMODE }
- enum LineModeEnums {
  LineMode_Input,
  LineMode_Output,
  NUM_LINEMODE }
- enum LineSourceEnums {
  LineSource_Off,
  LineSource_Line0,
  LineSource_Line1,
  LineSource_Line2,
  LineSource_Line3,
  LineSource_UserOutput0,
  LineSource_UserOutput1,
  LineSource_UserOutput2,
  LineSource_UserOutput3,
  LineSource_Counter0Active,
  LineSource_Counter1Active,
  LineSource_LogicBlock0,
  LineSource_LogicBlock1,
  LineSource_ExposureActive,
  LineSource_FrameTriggerWait,
  LineSource_SerialPort0,
  LineSource_PPSSignal,
  LineSource_AllPixel,
  LineSource_AnyPixel,
  NUM_LINESOURCE }
- enum LineInputFilterSelectorEnums {
  LineInputFilterSelector_Deglitch,
  LineInputFilterSelector_Debounce,
  NUM_LINEINPUTFILTERSELECTOR }
- enum UserOutputSelectorEnums {
  UserOutputSelector_UserOutput0,
  UserOutputSelector_UserOutput1,
  UserOutputSelector_UserOutput2,
  UserOutputSelector_UserOutput3,
  NUM_USEROUTPUTSELECTOR }
- enum LineFormatEnums {
  LineFormat_NoConnect,
  LineFormat_TriState,
  LineFormat_TTL,
  LineFormat_LVDS,
  LineFormat_RS422,
  LineFormat_OptoCoupled,
  LineFormat_OpenDrain,
  NUM_LINEFORMAT }
- enum LineSelectorEnums {
  LineSelector_Line0,

LineSelector_Line1,
LineSelector_Line2,
LineSelector_Line3,
NUM_LINESELECTOR }

- enum ExposureActiveModeEnums {
  ExposureActiveMode_Line1,
  ExposureActiveMode_AnyPixels,
  ExposureActiveMode_AllPixels,
  NUM_EXPOSUREACTIVEMODE }
- enum CounterTriggerActivationEnums {
  CounterTriggerActivation_LevelLow,
  CounterTriggerActivation_LevelHigh,
  CounterTriggerActivation_FallingEdge,
  CounterTriggerActivation_RisingEdge,
  CounterTriggerActivation_AnyEdge,
  NUM_COUNTERTRIGGERACTIVATION }
- enum CounterSelectorEnums {
  CounterSelector_Counter0,
  CounterSelector_Counter1,
  NUM_COUNTERSELECTOR }
- enum CounterStatusEnums {
  CounterStatus_CounterIdle,
  CounterStatus_CounterTriggerWait,
  CounterStatus_CounterActive,
  CounterStatus_CounterCompleted,
  CounterStatus_CounterOverflow,
  NUM_COUNTERSTATUS }
- enum CounterTriggerSourceEnums {
  CounterTriggerSource_Off,
  CounterTriggerSource_Line0,
  CounterTriggerSource_Line1,
  CounterTriggerSource_Line2,
  CounterTriggerSource_Line3,
  CounterTriggerSource_UserOutput0,
  CounterTriggerSource_UserOutput1,
  CounterTriggerSource_UserOutput2,
  CounterTriggerSource_UserOutput3,
  CounterTriggerSource_Counter0Start,
  CounterTriggerSource_Counter1Start,
  CounterTriggerSource_Counter0End,
  CounterTriggerSource_Counter1End,
  CounterTriggerSource_LogicBlock0,
  CounterTriggerSource_LogicBlock1,
  CounterTriggerSource_ExposureStart,
  CounterTriggerSource_ExposureEnd,
  CounterTriggerSource_FrameTriggerWait,
  NUM_COUNTERTRIGGERSOURCE }
- enum CounterResetSourceEnums {
  CounterResetSource_Off,
  CounterResetSource_Line0,
  CounterResetSource_Line1,
  CounterResetSource_Line2,
  CounterResetSource_Line3,
  CounterResetSource_UserOutput0,
  CounterResetSource_UserOutput1,
  CounterResetSource_UserOutput2,
  CounterResetSource_UserOutput3,
  CounterResetSource_Counter0Start,

CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }

- enum CounterEventSourceEnums {
  CounterEventSource_Off,
  CounterEventSource_MHzTick,
  CounterEventSource_Line0,
  CounterEventSource_Line1,
  CounterEventSource_Line2,
  CounterEventSource_Line3,
  CounterEventSource_UserOutput0,
  CounterEventSource_UserOutput1,
  CounterEventSource_UserOutput2,
  CounterEventSource_UserOutput3,
  CounterEventSource_Counter0Start,
  CounterEventSource_Counter1Start,
  CounterEventSource_Counter0End,
  CounterEventSource_Counter1End,
  CounterEventSource_LogicBlock0,
  CounterEventSource_LogicBlock1,
  CounterEventSource_ExposureStart,
  CounterEventSource_ExposureEnd,
  CounterEventSource_FrameTriggerWait,
  NUM_COUNTEREVENTSOURCE }
- enum CounterEventActivationEnums {
  CounterEventActivation_LevelLow,
  CounterEventActivation_LevelHigh,
  CounterEventActivation_FallingEdge,
  CounterEventActivation_RisingEdge,
  CounterEventActivation_AnyEdge,
  NUM_COUNTEREVENTACTIVATION }
- enum CounterResetActivationEnums {
  CounterResetActivation_LevelLow,
  CounterResetActivation_LevelHigh,
  CounterResetActivation_FallingEdge,
  CounterResetActivation_RisingEdge,
  CounterResetActivation_AnyEdge,
  NUM_COUNTERRESETACTIVATION }
- enum DeviceTypeEnums {
  DeviceType_Transmitter,
  DeviceType_Receiver,
  DeviceType_Transceiver,
  DeviceType_Peripheral,
  NUM_DEVICETYPE }
- enum DeviceConnectionStatusEnums {
  DeviceConnectionStatus_Active,
  DeviceConnectionStatus_Inactive,
  NUM_DEVICECONNECTIONSTATUS }
- enum DeviceLinkThroughputLimitModeEnums {
  DeviceLinkThroughputLimitMode_On,
  DeviceLinkThroughputLimitMode_Off,
  NUM_DEVICELINKTHROUGHPUTLIMITMODE }

- enum DeviceLinkHeartbeatModeEnums {
  DeviceLinkHeartbeatMode_On,
  DeviceLinkHeartbeatMode_Off,
  NUM_DEVICELINKHEARTBEATMODE }
- enum DeviceStreamChannelTypeEnums {
  DeviceStreamChannelType_Transmitter,
  DeviceStreamChannelType_Receiver,
  NUM_DEVICESTREAMCHANNELTYPE }
- enum DeviceStreamChannelEndiannessEnums {
  DeviceStreamChannelEndianness_Big,
  DeviceStreamChannelEndianness_Little,
  NUM_DEVICESTREAMCHANNELENDIANNESS }
- enum DeviceClockSelectorEnums {
  DeviceClockSelector_Sensor,
  DeviceClockSelector_SensorDigitization,
  DeviceClockSelector_CameraLink,
  NUM_DEVICECLOCKSELECTOR }
- enum DeviceSerialPortSelectorEnums {
  DeviceSerialPortSelector_CameraLink,
  NUM_DEVICESERIALPORTSELECTOR }
- enum DeviceSerialPortBaudRateEnums {
  DeviceSerialPortBaudRate_Baud9600,
  DeviceSerialPortBaudRate_Baud19200,
  DeviceSerialPortBaudRate_Baud38400,
  DeviceSerialPortBaudRate_Baud57600,
  DeviceSerialPortBaudRate_Baud115200,
  DeviceSerialPortBaudRate_Baud230400,
  DeviceSerialPortBaudRate_Baud460800,
  DeviceSerialPortBaudRate_Baud921600,
  NUM_DEVICESERIALPORTBAUDRATE }
- enum SensorTapsEnums {
  SensorTaps_One,
  SensorTaps_Two,
  SensorTaps_Three,
  SensorTaps_Four,
  SensorTaps_Eight,
  SensorTaps_Ten,
  NUM_SENSORTAPS }
- enum SensorDigitizationTapsEnums {
  SensorDigitizationTaps_One,
  SensorDigitizationTaps_Two,
  SensorDigitizationTaps_Three,
  SensorDigitizationTaps_Four,
  SensorDigitizationTaps_Eight,
  SensorDigitizationTaps_Ten,
  NUM_SENSORDIGITIZATIONTAPS }
- enum RegionSelectorEnums {
  RegionSelector_Region0,
  RegionSelector_Region1,
  RegionSelector_Region2,
  RegionSelector_All,
  NUM_REGIONSELECTOR }
- enum RegionModeEnums {
  RegionMode_Off,
  RegionMode_On,
  NUM_REGIONMODE }
- enum RegionDestinationEnums {
  RegionDestination_Stream0,

RegionDestination_Stream1,
RegionDestination_Stream2,
NUM_REGIONDESTINATION }

- enum ImageComponentSelectorEnums {
ImageComponentSelector_Intensity,
ImageComponentSelector_Color,
ImageComponentSelector_Infrared,
ImageComponentSelector_Ultraviolet,
ImageComponentSelector_Range,
ImageComponentSelector_Disparity,
ImageComponentSelector_Confidence,
ImageComponentSelector_Scatter,
NUM_IMAGECOMPONENTSELECTOR }
- enum PixelFormatInfoSelectorEnums {
PixelFormatInfoSelector_Mono1p,
PixelFormatInfoSelector_Mono2p,
PixelFormatInfoSelector_Mono4p,
PixelFormatInfoSelector_Mono8,
PixelFormatInfoSelector_Mono8s,
PixelFormatInfoSelector_Mono10,
PixelFormatInfoSelector_Mono10p,
PixelFormatInfoSelector_Mono12,
PixelFormatInfoSelector_Mono12p,
PixelFormatInfoSelector_Mono14,
PixelFormatInfoSelector_Mono16,
PixelFormatInfoSelector_Mono16s,
PixelFormatInfoSelector_Mono32f,
PixelFormatInfoSelector_BayerBG8,
PixelFormatInfoSelector_BayerBG10,
PixelFormatInfoSelector_BayerBG10p,
PixelFormatInfoSelector_BayerBG12,
PixelFormatInfoSelector_BayerBG12p,
PixelFormatInfoSelector_BayerBG16,
PixelFormatInfoSelector_BayerGB8,
PixelFormatInfoSelector_BayerGB10,
PixelFormatInfoSelector_BayerGB10p,
PixelFormatInfoSelector_BayerGB12,
PixelFormatInfoSelector_BayerGB12p,
PixelFormatInfoSelector_BayerGB16,
PixelFormatInfoSelector_BayerGR8,
PixelFormatInfoSelector_BayerGR10,
PixelFormatInfoSelector_BayerGR10p,
PixelFormatInfoSelector_BayerGR12,
PixelFormatInfoSelector_BayerGR12p,
PixelFormatInfoSelector_BayerGR16,
PixelFormatInfoSelector_BayerRG8,
PixelFormatInfoSelector_BayerRG10,
PixelFormatInfoSelector_BayerRG10p,
PixelFormatInfoSelector_BayerRG12,
PixelFormatInfoSelector_BayerRG12p,
PixelFormatInfoSelector_BayerRG16,
PixelFormatInfoSelector_RGBa8,
PixelFormatInfoSelector_RGBa10,
PixelFormatInfoSelector_RGBa10p,
PixelFormatInfoSelector_RGBa12,
PixelFormatInfoSelector_RGBa12p,
PixelFormatInfoSelector_RGBa14,
PixelFormatInfoSelector_RGBa16,

PixelFormatInfoSelector_RGB8,
PixelFormatInfoSelector_RGB8_Planar,
PixelFormatInfoSelector_RGB10,
PixelFormatInfoSelector_RGB10_Planar,
PixelFormatInfoSelector_RGB10p,
PixelFormatInfoSelector_RGB10p32,
PixelFormatInfoSelector_RGB12,
PixelFormatInfoSelector_RGB12_Planar,
PixelFormatInfoSelector_RGB12p,
PixelFormatInfoSelector_RGB14,
PixelFormatInfoSelector_RGB16,
PixelFormatInfoSelector_RGB16s,
PixelFormatInfoSelector_RGB32f,
PixelFormatInfoSelector_RGB16_Planar,
PixelFormatInfoSelector_RGB565p,
PixelFormatInfoSelector_BGRa8,
PixelFormatInfoSelector_BGRa10,
PixelFormatInfoSelector_BGRa10p,
PixelFormatInfoSelector_BGRa12,
PixelFormatInfoSelector_BGRa12p,
PixelFormatInfoSelector_BGRa14,
PixelFormatInfoSelector_BGRa16,
PixelFormatInfoSelector_RGBa32f,
PixelFormatInfoSelector_BGR8,
PixelFormatInfoSelector_BGR10,
PixelFormatInfoSelector_BGR10p,
PixelFormatInfoSelector_BGR12,
PixelFormatInfoSelector_BGR12p,
PixelFormatInfoSelector_BGR14,
PixelFormatInfoSelector_BGR16,
PixelFormatInfoSelector_BGR565p,
PixelFormatInfoSelector_R8,
PixelFormatInfoSelector_R10,
PixelFormatInfoSelector_R12,
PixelFormatInfoSelector_R16,
PixelFormatInfoSelector_G8,
PixelFormatInfoSelector_G10,
PixelFormatInfoSelector_G12,
PixelFormatInfoSelector_G16,
PixelFormatInfoSelector_B8,
PixelFormatInfoSelector_B10,
PixelFormatInfoSelector_B12,
PixelFormatInfoSelector_B16,
PixelFormatInfoSelector_Coord3D_ABC8,
PixelFormatInfoSelector_Coord3D_ABC8_Planar,
PixelFormatInfoSelector_Coord3D_ABC10p,
PixelFormatInfoSelector_Coord3D_ABC10p_Planar,
PixelFormatInfoSelector_Coord3D_ABC12p,
PixelFormatInfoSelector_Coord3D_ABC12p_Planar,
PixelFormatInfoSelector_Coord3D_ABC16,
PixelFormatInfoSelector_Coord3D_ABC16_Planar,
PixelFormatInfoSelector_Coord3D_ABC32f,
PixelFormatInfoSelector_Coord3D_ABC32f_Planar,
PixelFormatInfoSelector_Coord3D_AC8,
PixelFormatInfoSelector_Coord3D_AC8_Planar,
PixelFormatInfoSelector_Coord3D_AC10p,
PixelFormatInfoSelector_Coord3D_AC10p_Planar,
PixelFormatInfoSelector_Coord3D_AC12p,

PixelFormatInfoSelector_Coord3D_AC12p_Planar,
PixelFormatInfoSelector_Coord3D_AC16,
PixelFormatInfoSelector_Coord3D_AC16_Planar,
PixelFormatInfoSelector_Coord3D_AC32f,
PixelFormatInfoSelector_Coord3D_AC32f_Planar,
PixelFormatInfoSelector_Coord3D_A8,
PixelFormatInfoSelector_Coord3D_A10p,
PixelFormatInfoSelector_Coord3D_A12p,
PixelFormatInfoSelector_Coord3D_A16,
PixelFormatInfoSelector_Coord3D_A32f,
PixelFormatInfoSelector_Coord3D_B8,
PixelFormatInfoSelector_Coord3D_B10p,
PixelFormatInfoSelector_Coord3D_B12p,
PixelFormatInfoSelector_Coord3D_B16,
PixelFormatInfoSelector_Coord3D_B32f,
PixelFormatInfoSelector_Coord3D_C8,
PixelFormatInfoSelector_Coord3D_C10p,
PixelFormatInfoSelector_Coord3D_C12p,
PixelFormatInfoSelector_Coord3D_C16,
PixelFormatInfoSelector_Coord3D_C32f,
PixelFormatInfoSelector_Confidence1,
PixelFormatInfoSelector_Confidence1p,
PixelFormatInfoSelector_Confidence8,
PixelFormatInfoSelector_Confidence16,
PixelFormatInfoSelector_Confidence32f,
PixelFormatInfoSelector_BiColorBGRG8,
PixelFormatInfoSelector_BiColorBGRG10,
PixelFormatInfoSelector_BiColorBGRG10p,
PixelFormatInfoSelector_BiColorBGRG12,
PixelFormatInfoSelector_BiColorBGRG12p,
PixelFormatInfoSelector_BiColorRGBG8,
PixelFormatInfoSelector_BiColorRGBG10,
PixelFormatInfoSelector_BiColorRGBG10p,
PixelFormatInfoSelector_BiColorRGBG12,
PixelFormatInfoSelector_BiColorRGBG12p,
PixelFormatInfoSelector_SCF1WBWG8,
PixelFormatInfoSelector_SCF1WBWG10,
PixelFormatInfoSelector_SCF1WBWG10p,
PixelFormatInfoSelector_SCF1WBWG12,
PixelFormatInfoSelector_SCF1WBWG12p,
PixelFormatInfoSelector_SCF1WBWG14,
PixelFormatInfoSelector_SCF1WBWG16,
PixelFormatInfoSelector_SCF1WGWB8,
PixelFormatInfoSelector_SCF1WGWB10,
PixelFormatInfoSelector_SCF1WGWB10p,
PixelFormatInfoSelector_SCF1WGWB12,
PixelFormatInfoSelector_SCF1WGWB12p,
PixelFormatInfoSelector_SCF1WGWB14,
PixelFormatInfoSelector_SCF1WGWB16,
PixelFormatInfoSelector_SCF1WGWR8,
PixelFormatInfoSelector_SCF1WGWR10,
PixelFormatInfoSelector_SCF1WGWR10p,
PixelFormatInfoSelector_SCF1WGWR12,
PixelFormatInfoSelector_SCF1WGWR12p,
PixelFormatInfoSelector_SCF1WGWR14,
PixelFormatInfoSelector_SCF1WGWR16,
PixelFormatInfoSelector_SCF1WRWG8,
PixelFormatInfoSelector_SCF1WRWG10,

PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,
PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,

PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

- enum DeinterlacingEnums {
  Deinterlacing_Off,
  Deinterlacing_LineDuplication,
  Deinterlacing_Weave,
  NUM_DEINTERLACING }
- enum ImageCompressionRateOptionEnums {
  ImageCompressionRateOption_FixBitrate,
  ImageCompressionRateOption_FixQuality,
  NUM_IMAGECOMPRESSIONRATEOPTION }
- enum ImageCompressionJPEGFormatOptionEnums {
  ImageCompressionJPEGFormatOption_Lossless,
  ImageCompressionJPEGFormatOption_BaselineStandard,
  ImageCompressionJPEGFormatOption_BaselineOptimized,
  ImageCompressionJPEGFormatOption_Progressive,
  NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }
- enum AcquisitionStatusSelectorEnums {
  AcquisitionStatusSelector_AcquisitionTriggerWait,
  AcquisitionStatusSelector_AcquisitionActive,
  AcquisitionStatusSelector_AcquisitionTransfer,
  AcquisitionStatusSelector_FrameTriggerWait,
  AcquisitionStatusSelector_FrameActive,
  AcquisitionStatusSelector_ExposureActive,
  NUM_ACQUISITIONSTATUSSELECTOR }
- enum ExposureTimeModeEnums {
  ExposureTimeMode_Common,
  ExposureTimeMode_Individual,
  NUM_EXPOSURETIMEMODE }
- enum ExposureTimeSelectorEnums {
  ExposureTimeSelector_Common,
  ExposureTimeSelector_Red,
  ExposureTimeSelector_Green,
  ExposureTimeSelector_Blue,
  ExposureTimeSelector_Cyan,
  ExposureTimeSelector_Magenta,
  ExposureTimeSelector_Yellow,
  ExposureTimeSelector_Infrared,
  ExposureTimeSelector_Ultraviolet,
  ExposureTimeSelector_Stage1,
  ExposureTimeSelector_Stage2,
  NUM_EXPOSURETIMESELECTOR }
- enum GainAutoBalanceEnums {
  GainAutoBalance_Off,
  GainAutoBalance_Once,

GainAutoBalance_Continuous,
NUM_GAINAUTOBALANCE }

- enum BlackLevelAutoEnums {
BlackLevelAuto_Off,
BlackLevelAuto_Once,
BlackLevelAuto_Continuous,
NUM_BLACKLEVELAUTO }
- enum BlackLevelAutoBalanceEnums {
BlackLevelAutoBalance_Off,
BlackLevelAutoBalance_Once,
BlackLevelAutoBalance_Continuous,
NUM_BLACKLEVELAUTOBALANCE }
- enum WhiteClipSelectorEnums {
WhiteClipSelector_All,
WhiteClipSelector_Red,
WhiteClipSelector_Green,
WhiteClipSelector_Blue,
WhiteClipSelector_Y,
WhiteClipSelector_U,
WhiteClipSelector_V,
WhiteClipSelector_Tap1,
WhiteClipSelector_Tap2,
NUM_WHITECLIPSELECTOR }
- enum TimerSelectorEnums {
TimerSelector_Timer0,
TimerSelector_Timer1,
TimerSelector_Timer2,
NUM_TIMERSELECTOR }
- enum TimerStatusEnums {
TimerStatus_TimerIdle,
TimerStatus_TimerTriggerWait,
TimerStatus_TimerActive,
TimerStatus_TimerCompleted,
NUM_TIMERSTATUS }
- enum TimerTriggerSourceEnums {
TimerTriggerSource_Off,
TimerTriggerSource_AcquisitionTrigger,
TimerTriggerSource_AcquisitionStart,
TimerTriggerSource_AcquisitionEnd,
TimerTriggerSource_FrameTrigger,
TimerTriggerSource_FrameStart,
TimerTriggerSource_FrameEnd,
TimerTriggerSource_FrameBurstStart,
TimerTriggerSource_FrameBurstEnd,
TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,

TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

- enum TimerTriggerActivationEnums {
  TimerTriggerActivation_RisingEdge,
  TimerTriggerActivation_FallingEdge,
  TimerTriggerActivation_AnyEdge,
  TimerTriggerActivation_LevelHigh,
  TimerTriggerActivation_LevelLow,
  NUM_TIMERTRIGGERACTIVATION }
- enum EncoderSelectorEnums {
  EncoderSelector_Encoder0,
  EncoderSelector_Encoder1,
  EncoderSelector_Encoder2,
  NUM_ENCODERSELECTOR }
- enum EncoderSourceAEnums {
  EncoderSourceA_Off,
  EncoderSourceA_Line0,
  EncoderSourceA_Line1,
  EncoderSourceA_Line2,
  NUM_ENCODERSOURCEA }
- enum EncoderSourceBEnums {
  EncoderSourceB_Off,
  EncoderSourceB_Line0,
  EncoderSourceB_Line1,
  EncoderSourceB_Line2,
  NUM_ENCODERSOURCEB }
- enum EncoderModeEnums {
  EncoderMode_FourPhase,
  EncoderMode_HighResolution,
  NUM_ENCODERMODE }
- enum EncoderOutputModeEnums {
  EncoderOutputMode_Off,
  EncoderOutputMode_PositionUp,
  EncoderOutputMode_PositionDown,
  EncoderOutputMode_DirectionUp,
  EncoderOutputMode_DirectionDown,
  EncoderOutputMode_Motion,
  NUM_ENCODEROUTPUTMODE }

- enum EncoderStatusEnums {
  EncoderStatus_EncoderUp,
  EncoderStatus_EncoderDown,
  EncoderStatus_EncoderIdle,
  EncoderStatus_EncoderStatic,
  NUM_ENCODERSTATUS }
- enum EncoderResetSourceEnums {
  EncoderResetSource_Off,
  EncoderResetSource_AcquisitionTrigger,
  EncoderResetSource_AcquisitionStart,
  EncoderResetSource_AcquisitionEnd,
  EncoderResetSource_FrameTrigger,
  EncoderResetSource_FrameStart,
  EncoderResetSource_FrameEnd,
  EncoderResetSource_ExposureStart,
  EncoderResetSource_ExposureEnd,
  EncoderResetSource_Line0,
  EncoderResetSource_Line1,
  EncoderResetSource_Line2,
  EncoderResetSource_Counter0Start,
  EncoderResetSource_Counter1Start,
  EncoderResetSource_Counter2Start,
  EncoderResetSource_Counter0End,
  EncoderResetSource_Counter1End,
  EncoderResetSource_Counter2End,
  EncoderResetSource_Timer0Start,
  EncoderResetSource_Timer1Start,
  EncoderResetSource_Timer2Start,
  EncoderResetSource_Timer0End,
  EncoderResetSource_Timer1End,
  EncoderResetSource_Timer2End,
  EncoderResetSource_UserOutput0,
  EncoderResetSource_UserOutput1,
  EncoderResetSource_UserOutput2,
  EncoderResetSource_SoftwareSignal0,
  EncoderResetSource_SoftwareSignal1,
  EncoderResetSource_SoftwareSignal2,
  EncoderResetSource_Action0,
  EncoderResetSource_Action1,
  EncoderResetSource_Action2,
  EncoderResetSource_LinkTrigger0,
  EncoderResetSource_LinkTrigger1,
  EncoderResetSource_LinkTrigger2,
  NUM_ENCODERRESETSOURCE }
- enum EncoderResetActivationEnums {
  EncoderResetActivation_RisingEdge,
  EncoderResetActivation_FallingEdge,
  EncoderResetActivation_AnyEdge,
  EncoderResetActivation_LevelHigh,
  EncoderResetActivation_LevelLow,
  NUM_ENCODERRESETACTIVATION }
- enum SoftwareSignalSelectorEnums {
  SoftwareSignalSelector_SoftwareSignal0,
  SoftwareSignalSelector_SoftwareSignal1,
  SoftwareSignalSelector_SoftwareSignal2,
  NUM_SOFTWARESIGNALSELECTOR }
- enum ActionUnconditionalModeEnums {
  ActionUnconditionalMode_Off,

ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }

- enum SourceSelectorEnums {
  SourceSelector_Source0,
  SourceSelector_Source1,
  SourceSelector_Source2,
  SourceSelector_All,
  NUM_SOURCESELECTOR }
- enum TransferSelectorEnums {
  TransferSelector_Stream0,
  TransferSelector_Stream1,
  TransferSelector_Stream2,
  TransferSelector_All,
  NUM_TRANSFERSELECTOR }
- enum TransferTriggerSelectorEnums {
  TransferTriggerSelector_TransferStart,
  TransferTriggerSelector_TransferStop,
  TransferTriggerSelector_TransferAbort,
  TransferTriggerSelector_TransferPause,
  TransferTriggerSelector_TransferResume,
  TransferTriggerSelector_TransferActive,
  TransferTriggerSelector_TransferBurstStart,
  TransferTriggerSelector_TransferBurstStop,
  NUM_TRANSFERTRIGGERSELECTOR }
- enum TransferTriggerModeEnums {
  TransferTriggerMode_Off,
  TransferTriggerMode_On,
  NUM_TRANSFERTRIGGERMODE }
- enum TransferTriggerSourceEnums {
  TransferTriggerSource_Line0,
  TransferTriggerSource_Line1,
  TransferTriggerSource_Line2,
  TransferTriggerSource_Counter0Start,
  TransferTriggerSource_Counter1Start,
  TransferTriggerSource_Counter2Start,
  TransferTriggerSource_Counter0End,
  TransferTriggerSource_Counter1End,
  TransferTriggerSource_Counter2End,
  TransferTriggerSource_Timer0Start,
  TransferTriggerSource_Timer1Start,
  TransferTriggerSource_Timer2Start,
  TransferTriggerSource_Timer0End,
  TransferTriggerSource_Timer1End,
  TransferTriggerSource_Timer2End,
  TransferTriggerSource_SoftwareSignal0,
  TransferTriggerSource_SoftwareSignal1,
  TransferTriggerSource_SoftwareSignal2,
  TransferTriggerSource_Action0,
  TransferTriggerSource_Action1,
  TransferTriggerSource_Action2,
  NUM_TRANSFERTRIGGERSOURCE }
- enum TransferTriggerActivationEnums {
  TransferTriggerActivation_RisingEdge,
  TransferTriggerActivation_FallingEdge,
  TransferTriggerActivation_AnyEdge,
  TransferTriggerActivation_LevelHigh,
  TransferTriggerActivation_LevelLow,
  NUM_TRANSFERTRIGGERACTIVATION }

- enum TransferStatusSelectorEnums {
  TransferStatusSelector_Streaming,
  TransferStatusSelector_Paused,
  TransferStatusSelector_Stopping,
  TransferStatusSelector_Stopped,
  TransferStatusSelector_QueueOverflow,
  NUM_TRANSFERSTATUSSELECTOR }
- enum TransferComponentSelectorEnums {
  TransferComponentSelector_Red,
  TransferComponentSelector_Green,
  TransferComponentSelector_Blue,
  TransferComponentSelector_All,
  NUM_TRANSFERCOMPONENTSELECTOR }
- enum Scan3dDistanceUnitEnums {
  Scan3dDistanceUnit_Millimeter,
  Scan3dDistanceUnit_Inch,
  NUM_SCAN3DDISTANCEUNIT }
- enum Scan3dCoordinateSystemEnums {
  Scan3dCoordinateSystem_Cartesian,
  Scan3dCoordinateSystem_Spherical,
  Scan3dCoordinateSystem_Cylindrical,
  NUM_SCAN3DCOORDINATESYSTEM }
- enum Scan3dOutputModeEnums {
  Scan3dOutputMode_UncalibratedC,
  Scan3dOutputMode_CalibratedABC_Grid,
  Scan3dOutputMode_CalibratedABC_PointCloud,
  Scan3dOutputMode_CalibratedAC,
  Scan3dOutputMode_CalibratedAC_Linescan,
  Scan3dOutputMode_CalibratedC,
  Scan3dOutputMode_CalibratedC_Linescan,
  Scan3dOutputMode_RectifiedC,
  Scan3dOutputMode_RectifiedC_Linescan,
  Scan3dOutputMode_DisparityC,
  Scan3dOutputMode_DisparityC_Linescan,
  NUM_SCAN3DOUTPUTMODE }
- enum Scan3dCoordinateSystemReferenceEnums {
  Scan3dCoordinateSystemReference_Anchor,
  Scan3dCoordinateSystemReference_Transformed,
  NUM_SCAN3DCOORDINATESYSTEMREFERENCE }
- enum Scan3dCoordinateSelectorEnums {
  Scan3dCoordinateSelector_CoordinateA,
  Scan3dCoordinateSelector_CoordinateB,
  Scan3dCoordinateSelector_CoordinateC,
  NUM_SCAN3DCOORDINATESELECTOR }
- enum Scan3dCoordinateTransformSelectorEnums {
  Scan3dCoordinateTransformSelector_RotationX,
  Scan3dCoordinateTransformSelector_RotationY,
  Scan3dCoordinateTransformSelector_RotationZ,
  Scan3dCoordinateTransformSelector_TranslationX,
  Scan3dCoordinateTransformSelector_TranslationY,
  Scan3dCoordinateTransformSelector_TranslationZ,
  NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }
- enum Scan3dCoordinateReferenceSelectorEnums {
  Scan3dCoordinateReferenceSelector_RotationX,
  Scan3dCoordinateReferenceSelector_RotationY,
  Scan3dCoordinateReferenceSelector_RotationZ,
  Scan3dCoordinateReferenceSelector_TranslationX,
  Scan3dCoordinateReferenceSelector_TranslationY,

Scan3dCoordinateReferenceSelector_TranslationZ,
NUM_SCAN3DCOORDINATEREFERENCESELECTOR }

- enum ChunkImageComponentEnums {
  ChunkImageComponent_Intensity,
  ChunkImageComponent_Color,
  ChunkImageComponent_Infrared,
  ChunkImageComponent_Ultraviolet,
  ChunkImageComponent_Range,
  ChunkImageComponent_Disparity,
  ChunkImageComponent_Confidence,
  ChunkImageComponent_Scatter,
  NUM_CHUNKIMAGECOMPONENT }
- enum ChunkCounterSelectorEnums {
  ChunkCounterSelector_Counter0,
  ChunkCounterSelector_Counter1,
  ChunkCounterSelector_Counter2,
  NUM_CHUNKCOUNTERSELECTOR }
- enum ChunkTimerSelectorEnums {
  ChunkTimerSelector_Timer0,
  ChunkTimerSelector_Timer1,
  ChunkTimerSelector_Timer2,
  NUM_CHUNKTIMERSELECTOR }
- enum ChunkEncoderSelectorEnums {
  ChunkEncoderSelector_Encoder0,
  ChunkEncoderSelector_Encoder1,
  ChunkEncoderSelector_Encoder2,
  NUM_CHUNKENCODERSELECTOR }
- enum ChunkEncoderStatusEnums {
  ChunkEncoderStatus_EncoderUp,
  ChunkEncoderStatus_EncoderDown,
  ChunkEncoderStatus_EncoderIdle,
  ChunkEncoderStatus_EncoderStatic,
  NUM_CHUNKENCODERSTATUS }
- enum ChunkExposureTimeSelectorEnums {
  ChunkExposureTimeSelector_Common,
  ChunkExposureTimeSelector_Red,
  ChunkExposureTimeSelector_Green,
  ChunkExposureTimeSelector_Blue,
  ChunkExposureTimeSelector_Cyan,
  ChunkExposureTimeSelector_Magenta,
  ChunkExposureTimeSelector_Yellow,
  ChunkExposureTimeSelector_Infrared,
  ChunkExposureTimeSelector_Ultraviolet,
  ChunkExposureTimeSelector_Stage1,
  ChunkExposureTimeSelector_Stage2,
  NUM_CHUNKEXPOSURETIMESELECTOR }
- enum ChunkSourceIDEnums {
  ChunkSourceID_Source0,
  ChunkSourceID_Source1,
  ChunkSourceID_Source2,
  NUM_CHUNKSOURCEID }
- enum ChunkRegionIDEnums {
  ChunkRegionID_Region0,
  ChunkRegionID_Region1,
  ChunkRegionID_Region2,
  NUM_CHUNKREGIONID }
- enum ChunkTransferStreamIDEnums {
  ChunkTransferStreamID_Stream0,

ChunkTransferStreamID_Stream1,
ChunkTransferStreamID_Stream2,
ChunkTransferStreamID_Stream3,
NUM_CHUNKTRANSFERSTREAMID }

- enum ChunkScan3dDistanceUnitEnums {
ChunkScan3dDistanceUnit_Millimeter,
ChunkScan3dDistanceUnit_Inch,
NUM_CHUNKSCAN3DDISTANCEUNIT }
- enum ChunkScan3dOutputModeEnums {
ChunkScan3dOutputMode_UncalibratedC,
ChunkScan3dOutputMode_CalibratedABC_Grid,
ChunkScan3dOutputMode_CalibratedABC_PointCloud,
ChunkScan3dOutputMode_CalibratedAC,
ChunkScan3dOutputMode_CalibratedAC_Linescan,
ChunkScan3dOutputMode_CalibratedC,
ChunkScan3dOutputMode_CalibratedC_Linescan,
ChunkScan3dOutputMode_RectifiedC,
ChunkScan3dOutputMode_RectifiedC_Linescan,
ChunkScan3dOutputMode_DisparityC,
ChunkScan3dOutputMode_DisparityC_Linescan,
NUM_CHUNKSCAN3DOUTPUTMODE }
- enum ChunkScan3dCoordinateSystemEnums {
ChunkScan3dCoordinateSystem_Cartesian,
ChunkScan3dCoordinateSystem_Spherical,
ChunkScan3dCoordinateSystem_Cylindrical,
NUM_CHUNKSCAN3DCOORDINATESYSTEM }
- enum ChunkScan3dCoordinateSystemReferenceEnums {
ChunkScan3dCoordinateSystemReference_Anchor,
ChunkScan3dCoordinateSystemReference_Transformed,
NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
- enum ChunkScan3dCoordinateSelectorEnums {
ChunkScan3dCoordinateSelector_CoordinateA,
ChunkScan3dCoordinateSelector_CoordinateB,
ChunkScan3dCoordinateSelector_CoordinateC,
NUM_CHUNKSCAN3DCOORDINATESELECTOR }
- enum ChunkScan3dCoordinateTransformSelectorEnums {
ChunkScan3dCoordinateTransformSelector_RotationX,
ChunkScan3dCoordinateTransformSelector_RotationY,
ChunkScan3dCoordinateTransformSelector_RotationZ,
ChunkScan3dCoordinateTransformSelector_TranslationX,
ChunkScan3dCoordinateTransformSelector_TranslationY,
ChunkScan3dCoordinateTransformSelector_TranslationZ,
NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }
- enum ChunkScan3dCoordinateReferenceSelectorEnums {
ChunkScan3dCoordinateReferenceSelector_RotationX,
ChunkScan3dCoordinateReferenceSelector_RotationY,
ChunkScan3dCoordinateReferenceSelector_RotationZ,
ChunkScan3dCoordinateReferenceSelector_TranslationX,
ChunkScan3dCoordinateReferenceSelector_TranslationY,
ChunkScan3dCoordinateReferenceSelector_TranslationZ,
NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }
- enum DeviceTapGeometryEnums {
DeviceTapGeometry_Geometry_1X_1Y,
DeviceTapGeometry_Geometry_1X2_1Y,
DeviceTapGeometry_Geometry_1X2_1Y2,
DeviceTapGeometry_Geometry_2X_1Y,
DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
DeviceTapGeometry_Geometry_2XE_1Y2,

DeviceTapGeometry_Geometry_2XM_1Y,
DeviceTapGeometry_Geometry_2XM_1Y2,
DeviceTapGeometry_Geometry_1X_1Y2,
DeviceTapGeometry_Geometry_1X_2YE,
DeviceTapGeometry_Geometry_1X3_1Y,
DeviceTapGeometry_Geometry_3X_1Y,
DeviceTapGeometry_Geometry_1X,
DeviceTapGeometry_Geometry_1X2,
DeviceTapGeometry_Geometry_2X,
DeviceTapGeometry_Geometry_2XE,
DeviceTapGeometry_Geometry_2XM,
DeviceTapGeometry_Geometry_1X3,
DeviceTapGeometry_Geometry_3X,
DeviceTapGeometry_Geometry_1X4_1Y,
DeviceTapGeometry_Geometry_4X_1Y,
DeviceTapGeometry_Geometry_2X2_1Y,
DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
DeviceTapGeometry_Geometry_1X2_2YE,
DeviceTapGeometry_Geometry_2X_2YE,
DeviceTapGeometry_Geometry_2XE_2YE,
DeviceTapGeometry_Geometry_2XM_2YE,
DeviceTapGeometry_Geometry_1X4,
DeviceTapGeometry_Geometry_4X,
DeviceTapGeometry_Geometry_2X2,
DeviceTapGeometry_Geometry_2X2E,
DeviceTapGeometry_Geometry_2X2M,
DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

- enum GevPhysicalLinkConfigurationEnums {
  GevPhysicalLinkConfiguration_SingleLink,
  GevPhysicalLinkConfiguration_MultiLink,
  GevPhysicalLinkConfiguration_StaticLAG,
  GevPhysicalLinkConfiguration_DynamicLAG,
  NUM_GEVPHYSICALLINKCONFIGURATION }
- enum GevCurrentPhysicalLinkConfigurationEnums {
  GevCurrentPhysicalLinkConfiguration_SingleLink,
  GevCurrentPhysicalLinkConfiguration_MultiLink,
  GevCurrentPhysicalLinkConfiguration_StaticLAG,
  GevCurrentPhysicalLinkConfiguration_DynamicLAG,
  NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }
- enum GevIPConfigurationStatusEnums {
  GevIPConfigurationStatus_None,
  GevIPConfigurationStatus_PersistentIP,
  GevIPConfigurationStatus_DHCP,
  GevIPConfigurationStatus_LLA,
  GevIPConfigurationStatus_ForceIP,

NUM_GEVIPCONFIGURATIONSTATUS }

- enum GevGVCPExtendedStatusCodesSelectorEnums {
  GevGVCPExtendedStatusCodesSelector_Version1_1,
  GevGVCPExtendedStatusCodesSelector_Version2_0,
  NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }
- enum GevGVSPExtendedIDModeEnums {
  GevGVSPExtendedIDMode_Off,
  GevGVSPExtendedIDMode_On,
  NUM_GEVGVSPEXTENDEDIDMODE }
- enum ClConfigurationEnums {
  ClConfiguration_Base,
  ClConfiguration_Medium,
  ClConfiguration_Full,
  ClConfiguration_DualBase,
  ClConfiguration_EightyBit,
  NUM_CLCONFIGURATION }
- enum ClTimeSlotsCountEnums {
  ClTimeSlotsCount_One,
  ClTimeSlotsCount_Two,
  ClTimeSlotsCount_Three,
  NUM_CLTIMESLOTSCOUNT }
- enum CxpLinkConfigurationStatusEnums {
  CxpLinkConfigurationStatus_None,
  CxpLinkConfigurationStatus_Pending,
  CxpLinkConfigurationStatus_CXP1_X1,
  CxpLinkConfigurationStatus_CXP2_X1,
  CxpLinkConfigurationStatus_CXP3_X1,
  CxpLinkConfigurationStatus_CXP5_X1,
  CxpLinkConfigurationStatus_CXP6_X1,
  CxpLinkConfigurationStatus_CXP1_X2,
  CxpLinkConfigurationStatus_CXP2_X2,
  CxpLinkConfigurationStatus_CXP3_X2,
  CxpLinkConfigurationStatus_CXP5_X2,
  CxpLinkConfigurationStatus_CXP6_X2,
  CxpLinkConfigurationStatus_CXP1_X3,
  CxpLinkConfigurationStatus_CXP2_X3,
  CxpLinkConfigurationStatus_CXP3_X3,
  CxpLinkConfigurationStatus_CXP5_X3,
  CxpLinkConfigurationStatus_CXP6_X3,
  CxpLinkConfigurationStatus_CXP1_X4,
  CxpLinkConfigurationStatus_CXP2_X4,
  CxpLinkConfigurationStatus_CXP3_X4,
  CxpLinkConfigurationStatus_CXP5_X4,
  CxpLinkConfigurationStatus_CXP6_X4,
  CxpLinkConfigurationStatus_CXP1_X5,
  CxpLinkConfigurationStatus_CXP2_X5,
  CxpLinkConfigurationStatus_CXP3_X5,
  CxpLinkConfigurationStatus_CXP5_X5,
  CxpLinkConfigurationStatus_CXP6_X5,
  CxpLinkConfigurationStatus_CXP1_X6,
  CxpLinkConfigurationStatus_CXP2_X6,
  CxpLinkConfigurationStatus_CXP3_X6,
  CxpLinkConfigurationStatus_CXP5_X6,
  CxpLinkConfigurationStatus_CXP6_X6,
  NUM_CXPLINKCONFIGURATIONSTATUS }
- enum CxpLinkConfigurationPreferredEnums {
  CxpLinkConfigurationPreferred_CXP1_X1,
  CxpLinkConfigurationPreferred_CXP2_X1,

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

- enum CxpLinkConfigurationEnums {
  CxpLinkConfiguration_Auto,
  CxpLinkConfiguration_CXP1_X1,
  CxpLinkConfiguration_CXP2_X1,
  CxpLinkConfiguration_CXP3_X1,
  CxpLinkConfiguration_CXP5_X1,
  CxpLinkConfiguration_CXP6_X1,
  CxpLinkConfiguration_CXP1_X2,
  CxpLinkConfiguration_CXP2_X2,
  CxpLinkConfiguration_CXP3_X2,
  CxpLinkConfiguration_CXP5_X2,
  CxpLinkConfiguration_CXP6_X2,
  CxpLinkConfiguration_CXP1_X3,
  CxpLinkConfiguration_CXP2_X3,
  CxpLinkConfiguration_CXP3_X3,
  CxpLinkConfiguration_CXP5_X3,
  CxpLinkConfiguration_CXP6_X3,
  CxpLinkConfiguration_CXP1_X4,
  CxpLinkConfiguration_CXP2_X4,
  CxpLinkConfiguration_CXP3_X4,
  CxpLinkConfiguration_CXP5_X4,
  CxpLinkConfiguration_CXP6_X4,
  CxpLinkConfiguration_CXP1_X5,
  CxpLinkConfiguration_CXP2_X5,
  CxpLinkConfiguration_CXP3_X5,
  CxpLinkConfiguration_CXP5_X5,
  CxpLinkConfiguration_CXP6_X5,
  CxpLinkConfiguration_CXP1_X6,
  CxpLinkConfiguration_CXP2_X6,

CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }

- enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
- enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

12.6.1 Detailed Description

12.6.2 Enumeration Type Documentation

12.6.2.1 AcquisitionModeEnums

`enum AcquisitionModeEnums`

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

Enumerator

AcquisitionMode_Continuous	
AcquisitionMode_SingleFrame	
AcquisitionMode_MultiFrame	
NUM_ACQUISITIONMODE	

12.6.2.2 AcquisitionStatusSelectorEnums

`enum AcquisitionStatusSelectorEnums`

< Selects the internal acquisition signal to read using AcquisitionStatus.

Enumerator

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

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	

12.6.2.3 ActionUnconditionalModeEnums

`enum ActionUnconditionalModeEnums`

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

Enumerator

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

12.6.2.4 AdcBitDepthEnums

`enum AdcBitDepthEnums`

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

Enumerator

AdcBitDepth_Bit8	
AdcBitDepth_Bit10	
AdcBitDepth_Bit12	
AdcBitDepth_Bit14	
NUM_ADCBITDEPTH	

12.6.2.5 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	

12.6.2.6 AutoExposureControlPriorityEnums

```
enum AutoExposureControlPriorityEnums
```

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

Enumerator

AutoExposureControlPriority_Gain	
AutoExposureControlPriority_ExposureTime	
NUM_AUTOEXPOSURECONTROLPRIORITY	

12.6.2.7 AutoExposureLightingModeEnums

```
enum AutoExposureLightingModeEnums
```

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

Enumerator

AutoExposureLightingMode_AutoDetect	
AutoExposureLightingMode_Backlight	
AutoExposureLightingMode_Frontlight	
AutoExposureLightingMode_Normal	
NUM_AUTOEXPOSURELIGHTINGMODE	

12.6.2.8 AutoExposureMeteringModeEnums

```
enum AutoExposureMeteringModeEnums
```

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

Enumerator

AutoExposureMeteringMode_Average	
AutoExposureMeteringMode_Spot	
AutoExposureMeteringMode_Partial	
AutoExposureMeteringMode_CenterWeighted	
AutoExposureMeteringMode_HistgramPeak	
NUM_AUTOEXPOSUREMETERINGMODE	

12.6.2.9 AutoExposureTargetGreyValueAutoEnums

```
enum AutoExposureTargetGreyValueAutoEnums
```

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

Enumerator

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_AUTOEXPOSURETARGETGREYVALUEA↩UTO	

12.6.2.10 BalanceRatioSelectorEnums

```
enum BalanceRatioSelectorEnums
```

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

Enumerator

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

12.6.2.11 BalanceWhiteAutoEnums

```
enum BalanceWhiteAutoEnums
```

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

Enumerator

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

12.6.2.12 BalanceWhiteAutoProfileEnums

```
enum BalanceWhiteAutoProfileEnums
```

< Selects the profile used by BalanceWhiteAuto.

Enumerator

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

12.6.2.13 BinningHorizontalModeEnums

```
enum BinningHorizontalModeEnums
```

<

Enumerator

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

12.6.2.14 BinningSelectorEnums

enum BinningSelectorEnums

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

Enumerator

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

12.6.2.15 BinningVerticalModeEnums

enum BinningVerticalModeEnums

<

Enumerator

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

12.6.2.16 BlackLevelAutoBalanceEnums

enum BlackLevelAutoBalanceEnums

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

Enumerator

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

12.6.2.17 BlackLevelAutoEnums

`enum BlackLevelAutoEnums`

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

Enumerator

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

12.6.2.18 BlackLevelSelectorEnums

`enum BlackLevelSelectorEnums`

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

Enumerator

BlackLevelSelector_All	
BlackLevelSelector_Analog	
BlackLevelSelector_Digital	
NUM_BLACKLEVELSELECTOR	

12.6.2.19 ChunkBlackLevelSelectorEnums

`enum ChunkBlackLevelSelectorEnums`

< Selects which black level to retrieve

Enumerator

ChunkBlackLevelSelector_All	
NUM_CHUNKBLACKLEVELSELECTOR	

12.6.2.20 ChunkCounterSelectorEnums

```
enum ChunkCounterSelectorEnums
```

< Selects which counter to retrieve data from.

Enumerator

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

12.6.2.21 ChunkEncoderSelectorEnums

```
enum ChunkEncoderSelectorEnums
```

< Selects which Encoder to retrieve data from.

Enumerator

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

12.6.2.22 ChunkEncoderStatusEnums

```
enum ChunkEncoderStatusEnums
```

< Returns the motion status of the selected encoder.

Enumerator

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

12.6.2.23 ChunkExposureTimeSelectorEnums

```
enum ChunkExposureTimeSelectorEnums
```

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

Enumerator

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

12.6.2.24 ChunkGainSelectorEnums

```
enum ChunkGainSelectorEnums
```

< Selects which gain to retrieve

Enumerator

ChunkGainSelector_All	
ChunkGainSelector_Red	
ChunkGainSelector_Green	
ChunkGainSelector_Blue	
NUM_CHUNKGAINSELECTOR	

12.6.2.25 ChunkImageComponentEnums

```
enum ChunkImageComponentEnums
```

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

Enumerator

ChunkImageComponent_Intensity	The image data is the intensity component.
ChunkImageComponent_Color	The image data is color component.
ChunkImageComponent_Infrared	The image data is infrared component.
ChunkImageComponent_Ultraviolet	The image data is the ultraviolet component.

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	

12.6.2.26 ChunkPixelFormatEnums

```
enum ChunkPixelFormatEnums
```

< Format of the pixel provided by the camera

Enumerator

ChunkPixelFormat_Mono8	
ChunkPixelFormat_Mono12Packed	
ChunkPixelFormat_Mono16	
ChunkPixelFormat_RGB8Packed	
ChunkPixelFormat_YUV422Packed	
ChunkPixelFormat_BayerGR8	
ChunkPixelFormat_BayerRG8	
ChunkPixelFormat_BayerGB8	
ChunkPixelFormat_BayerBG8	
ChunkPixelFormat_YCbCr601_422_8_CbYCrY	
NUM_CHUNKPIXELFORMAT	

12.6.2.27 ChunkRegionIDEnums

```
enum ChunkRegionIDEnums
```

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

Enumerator

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

12.6.2.28 ChunkScan3dCoordinateReferenceSelectorEnums

`enum ChunkScan3dCoordinateReferenceSelectorEnums`

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

Enumerator

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

12.6.2.29 ChunkScan3dCoordinateSelectorEnums

`enum ChunkScan3dCoordinateSelectorEnums`

< Selects which Coordinate to retrieve data from.

Enumerator

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

12.6.2.30 ChunkScan3dCoordinateSystemEnums

`enum ChunkScan3dCoordinateSystemEnums`

< Returns the Coordinate System of the image included in the payload.

Enumerator

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

12.6.2.31 ChunkScan3dCoordinateSystemReferenceEnums

```
enum ChunkScan3dCoordinateSystemReferenceEnums
```

< Returns the Coordinate System Position of the image included in the payload.

Enumerator

ChunkScan3dCoordinateSystemReference_Anchor	Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.
ChunkScan3dCoordinateSystemReference_↩ Transformed	Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices.
NUM_CHUNKSCAN3DCOORDINATESYSTEMRE↩ FERENCE	

12.6.2.32 ChunkScan3dCoordinateTransformSelectorEnums

```
enum ChunkScan3dCoordinateTransformSelectorEnums
```

< Selector for transform values.

Enumerator

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

12.6.2.33 ChunkScan3dDistanceUnitEnums

```
enum ChunkScan3dDistanceUnitEnums
```

< Returns the Distance Unit of the payload image.

Enumerator

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

12.6.2.34 ChunkScan3dOutputModeEnums

```
enum ChunkScan3dOutputModeEnums
```

< Returns the Calibrated Mode of the payload image.

Enumerator

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

12.6.2.35 ChunkSelectorEnums

```
enum ChunkSelectorEnums
```

< Selects which chunk data to enable or disable.

Enumerator

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

12.6.2.36 ChunkSourceIDEnums

```
enum ChunkSourceIDEnums
```

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

Enumerator

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

12.6.2.37 ChunkTimerSelectorEnums

```
enum ChunkTimerSelectorEnums
```

< Selects which Timer to retrieve data from.

Enumerator

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

12.6.2.38 ChunkTransferStreamIDEnums

```
enum ChunkTransferStreamIDEnums
```

< Returns identifier of the stream that generated this block.

Enumerator

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

12.6.2.39 ClConfigurationEnums

```
enum ClConfigurationEnums
```

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

Enumerator

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

12.6.2.40 ClTimeSlotsCountEnums

```
enum ClTimeSlotsCountEnums
```

< 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

ClTimeSlotsCount_One	One
ClTimeSlotsCount_Two	Two
ClTimeSlotsCount_Three	Three
NUM_CLTIMESLOTSCOUNT	

12.6.2.41 ColorTransformationSelectorEnums

```
enum ColorTransformationSelectorEnums
```

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

Enumerator

ColorTransformationSelector_RGBtoRGB	
ColorTransformationSelector_RGBtoYUV	
NUM_COLORTRANSFORMATIONSELECTOR	

12.6.2.42 ColorTransformationValueSelectorEnums

```
enum ColorTransformationValueSelectorEnums
```

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

Enumerator

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

12.6.2.43 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	

12.6.2.44 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	

12.6.2.45 CounterResetActivationEnums

```
enum CounterResetActivationEnums
```

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

Enumerator

CounterResetActivation_LevelLow	
CounterResetActivation_LevelHigh	
CounterResetActivation_FallingEdge	
CounterResetActivation_RisingEdge	
CounterResetActivation_AnyEdge	
NUM_COUNTERRESETACTIVATION	

12.6.2.46 CounterResetSourceEnums

```
enum CounterResetSourceEnums
```

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

Enumerator

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

12.6.2.47 CounterSelectorEnums

```
enum CounterSelectorEnums
```

< Selects which counter to configure

Enumerator

CounterSelector_Counter0	
CounterSelector_Counter1	
NUM_COUNTERSELECTOR	

12.6.2.48 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	

12.6.2.49 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	

12.6.2.50 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	

12.6.2.51 CxpConnectionTestModeEnums

```
enum CxpConnectionTestModeEnums
```

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

Enumerator

CxpConnectionTestMode_Off	Off
CxpConnectionTestMode_Mode1	Mode 1
NUM_CXPCONNECTIONTESTMODE	

12.6.2.52 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	

12.6.2.53 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	

12.6.2.54 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	

12.6.2.55 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_CXPPOCXPSTATUS	

12.6.2.56 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	

12.6.2.57 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	

12.6.2.58 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	

12.6.2.59 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	

12.6.2.60 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	

12.6.2.61 DeviceCharacterSetEnums

```
enum DeviceCharacterSetEnums
```

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

Enumerator

DeviceCharacterSet_UTF8	
DeviceCharacterSet_ASCII	
NUM_DEVICECHARACTERSET	

12.6.2.62 DeviceClockSelectorEnums

```
enum DeviceClockSelectorEnums
```

< Selects the clock frequency to access from the device.

Enumerator

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

12.6.2.63 DeviceConnectionStatusEnums

```
enum DeviceConnectionStatusEnums
```

< Indicates the status of the specified Connection.

Enumerator

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

12.6.2.64 DeviceIndicatorModeEnums

```
enum DeviceIndicatorModeEnums
```

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

Enumerator

DeviceIndicatorMode_Inactive	
DeviceIndicatorMode_Active	
DeviceIndicatorMode_ErrorStatus	
NUM_DEVICEINDICATORMODE	

12.6.2.65 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	

12.6.2.66 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	

12.6.2.67 DevicePowerSupplySelectorEnums

```
enum DevicePowerSupplySelectorEnums
```

< Selects the power supply source to control or read.

Enumerator

DevicePowerSupplySelector_External	
NUM_DEVICEPOWERSUPPLYSELECTOR	

12.6.2.68 DeviceRegistersEndiannessEnums

```
enum DeviceRegistersEndiannessEnums
```

< Endianess of the registers of the device.

Enumerator

DeviceRegistersEndianness_Little	
DeviceRegistersEndianness_Big	
NUM_DEVICEREGISTERSENDIANNESS	

12.6.2.69 DeviceScanTypeEnums

```
enum DeviceScanTypeEnums
```

< Scan type of the sensor of the device.

Enumerator

DeviceScanType_Areascan	
NUM_DEVICESCANTYPE	

12.6.2.70 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	

12.6.2.71 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	

12.6.2.72 DeviceStreamChannelEndiannessEnums

```
enum DeviceStreamChannelEndiannessEnums
```

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

Enumerator

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

12.6.2.73 DeviceStreamChannelTypeEnums

```
enum DeviceStreamChannelTypeEnums
```

< Reports the type of the stream channel.

Enumerator

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

12.6.2.74 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

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

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	

12.6.2.75 DeviceTemperatureSelectorEnums

```
enum DeviceTemperatureSelectorEnums
```

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

Enumerator

DeviceTemperatureSelector_Sensor	
NUM_DEVICETEMPERATURESELECTOR	

12.6.2.76 DeviceTLTypeEnums

```
enum DeviceTLTypeEnums
```

< Transport Layer type of the device.

Enumerator

DeviceTLType_GigEVision	
DeviceTLType_CameraLink	
DeviceTLType_CameraLinkHS	
DeviceTLType_CoaXPress	
DeviceTLType_USB3Vision	
DeviceTLType_Custom	
NUM_DEVICETLTYPE	

12.6.2.77 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	

12.6.2.78 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	

12.6.2.79 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	

12.6.2.80 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	

12.6.2.81 EncoderResetSourceEnums

```
enum EncoderResetSourceEnums
```

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

Enumerator

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

Enumerator

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

12.6.2.82 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	

12.6.2.83 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	

12.6.2.84 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	

12.6.2.85 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	

12.6.2.86 EventNotificationEnums

`enum EventNotificationEnums`

< Enables/Disables the selected event.

Enumerator

EventNotification_On	
EventNotification_Off	
NUM_EVENTNOTIFICATION	

12.6.2.87 EventSelectorEnums

`enum EventSelectorEnums`

< Selects which Event to enable or disable.

Enumerator

EventSelector_Error	
EventSelector_ExposureEnd	
EventSelector_SerialPortReceive	
NUM_EVENTSELECTOR	

12.6.2.88 ExposureActiveModeEnums

`enum ExposureActiveModeEnums`

< Control sensor active exposure mode.

Enumerator

ExposureActiveMode_Line1	
ExposureActiveMode_AnyPixels	
ExposureActiveMode_AllPixels	
NUM_EXPOSUREACTIVEMODE	

12.6.2.89 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	

12.6.2.90 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	

12.6.2.91 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	

12.6.2.92 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	

12.6.2.93 FileOpenModeEnums

`enum FileOpenModeEnums`

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

Enumerator

FileOpenMode_Read	
FileOpenMode_Write	
FileOpenMode_ReadWrite	
NUM_FILEOPENMODE	

12.6.2.94 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	

12.6.2.95 FileOperationStatusEnums

`enum FileOperationStatusEnums`

< Represents the file operation execution status.

Enumerator

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

12.6.2.96 FileSelectorEnums

```
enum FileSelectorEnums
```

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

Enumerator

FileSelector_UserSetDefault	
FileSelector_UserSet0	
FileSelector_UserSet1	
FileSelector_UserFile1	
FileSelector_SerialPort0	
NUM_FILESELECTOR	

12.6.2.97 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

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

12.6.2.98 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

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

12.6.2.99 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	

12.6.2.100 GevCCPEnums

`enum GevCCPEnums`

< Controls the device access privilege of an application.

Enumerator

GevCCP_OpenAccess	
GevCCP_ExclusiveAccess	
GevCCP_ControlAccess	
NUM_GEVCCP	

12.6.2.101 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	

12.6.2.102 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	

12.6.2.103 GevGVSPExtendedIDModeEnums

```
enum GevGVSPExtendedIDModeEnums
```

< Enables the extended IDs mode.

Enumerator

GevGVSPExtendedIDMode_Off	Off
GevGVSPExtendedIDMode_On	On
NUM_GEVGVSPEXTENDEDIDMODE	

12.6.2.104 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	

12.6.2.105 GevIEEE1588ModeEnums

```
enum GevIEEE1588ModeEnums
```

< Provides the mode of the IEEE 1588 clock.

Enumerator

GevIEEE1588Mode_Auto	Automatic
GevIEEE1588Mode_SlaveOnly	Slave Only
NUM_GEVIEEE1588MODE	

12.6.2.106 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	

12.6.2.107 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	

12.6.2.108 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	

12.6.2.109 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	

12.6.2.110 ImageComponentSelectorEnums

enum ImageComponentSelectorEnums

< Selects a component to activate data streaming from.

Enumerator

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

12.6.2.111 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

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

Enumerator

ImageCompressionJPEGFormatOption_Progressive	Indicates this is a progressive (multi-scan) DCT-based JPEG.
NUM_IMAGECOMPRESSIONJPEGFORMATOPT↩ ION	

12.6.2.112 ImageCompressionModeEnums

```
enum ImageCompressionModeEnums
```

<

Enumerator

ImageCompressionMode_Off	
ImageCompressionMode_Lossless	
NUM_IMAGECOMPRESSIONMODE	

12.6.2.113 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	

12.6.2.114 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	

12.6.2.115 LineInputFilterSelectorEnums

`enum LineInputFilterSelectorEnums`

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

Enumerator

LineInputFilterSelector_Deglitch	
LineInputFilterSelector_Debounce	
NUM_LINEINPUTFILTERSELECTOR	

12.6.2.116 LineModeEnums

`enum LineModeEnums`

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

Enumerator

LineMode_Input	
LineMode_Output	
NUM_LINEMODE	

12.6.2.117 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	

12.6.2.118 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	

12.6.2.119 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	

12.6.2.120 LogicBlockLUTInputSelectorEnums

`enum LogicBlockLUTInputSelectorEnums`

< Controls which LogicBlockLUT Input Source & Activation to access.

Enumerator

LogicBlockLUTInputSelector_Input0	
LogicBlockLUTInputSelector_Input1	
LogicBlockLUTInputSelector_Input2	
LogicBlockLUTInputSelector_Input3	
NUM_LOGICBLOCKLUTINPUTSELECTOR	

12.6.2.121 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	

12.6.2.122 LogicBlockLUTSelectorEnums

`enum LogicBlockLUTSelectorEnums`

< Selects which LogicBlock LUT to configure

Enumerator

LogicBlockLUTSelector_Value	
LogicBlockLUTSelector_Enable	
NUM_LOGICBLOCKLUTSELECTOR	

12.6.2.123 LogicBlockSelectorEnums

`enum LogicBlockSelectorEnums`

< Selects which LogicBlock to configure

Enumerator

LogicBlockSelector_LogicBlock0	
LogicBlockSelector_LogicBlock1	
NUM_LOGICBLOCKSELECTOR	

12.6.2.124 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	

12.6.2.125 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	

12.6.2.126 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	YCbCr 4:2:2 10-bit packed
PixelFormat_YCbCr422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed
PixelFormat_YCbCr422_12	YCbCr 4:2:2 12-bit unpacked
PixelFormat_YCbCr422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked
PixelFormat_YCbCr422_12p	YCbCr 4:2:2 12-bit packed
PixelFormat_YCbCr422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed
PixelFormat_YCbCr601_8_CbYCr	YCbCr 4:4:4 8-bit BT.601
PixelFormat_YCbCr601_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.601
PixelFormat_YCbCr601_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.601
PixelFormat_YCbCr601_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.601
PixelFormat_YCbCr601_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.601
PixelFormat_YCbCr601_411_8_CbYYCrYY	YCbCr 4:1:1 8-bit BT.601
PixelFormat_YCbCr601_422_8	YCbCr 4:2:2 8-bit BT.601
PixelFormat_YCbCr601_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.601

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	

12.6.2.127 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	YCbCr 4:2:2 12-bit packed
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed
PixelFormatInfoSelector_YCbCr601_8_CbYCr	YCbCr 4:4:4 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_411_8_CbYY↩CrYY	YCbCr 4:1:1 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_8	YCbCr 4:2:2 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_10	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_10_CbY↩CrY	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_10p	YCbCr 4:2:2 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_10p_Cb↩YCrY	YCbCr 4:2:2 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_12	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_12_CbY↩CrY	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_12p	YCbCr 4:2:2 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_12p_Cb↩YCrY	YCbCr 4:2:2 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr709_8_CbYCr	YCbCr 4:4:4 8-bit BT.709

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	YUV 4:2:2 8-bit
PixelFormatInfoSelector_YUV422_8_UYVY	YUV 4:2:2 8-bit
PixelFormatInfoSelector_Polarized8	Monochrome Polarized 8-bit
PixelFormatInfoSelector_Polarized10p	Monochrome Polarized 10-bit packed
PixelFormatInfoSelector_Polarized12p	Monochrome Polarized 12-bit packed
PixelFormatInfoSelector_Polarized16	Monochrome Polarized 16-bit
PixelFormatInfoSelector_BayerRGPolarized8	Polarized Bayer Red Green filter 8-bit
PixelFormatInfoSelector_BayerRGPolarized10p	Polarized Bayer Red Green filter 10-bit packed
PixelFormatInfoSelector_BayerRGPolarized12p	Polarized Bayer Red Green filter 12-bit packed
PixelFormatInfoSelector_BayerRGPolarized16	Polarized Bayer Red Green filter 16-bit
PixelFormatInfoSelector_LLCMono8	Lossless Compression Monochrome 8-bit
PixelFormatInfoSelector_LLCBayerRG8	Lossless Compression Bayer Red Green filter 8-bit
PixelFormatInfoSelector_JPEGMono8	JPEG Monochrome 8-bit
PixelFormatInfoSelector_JPEGColor8	JPEG Color 8-bit
NUM_PIXELFORMATINFOSELECTOR	

12.6.2.128 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	

12.6.2.129 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	

12.6.2.130 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	

12.6.2.131 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

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

12.6.2.132 RgbTransformLightSourceEnums

enum RgbTransformLightSourceEnums

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

Enumerator

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

12.6.2.133 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	

12.6.2.134 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	

12.6.2.135 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	

12.6.2.136 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	

12.6.2.137 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	

12.6.2.138 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	

12.6.2.139 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	

12.6.2.140 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	

12.6.2.141 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	

12.6.2.142 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	

12.6.2.143 SequencerConfigurationModeEnums

```
enum SequencerConfigurationModeEnums
```

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

Enumerator

SequencerConfigurationMode_Off	
SequencerConfigurationMode_On	
NUM_SEQUENCERCONFIGURATIONMODE	

12.6.2.144 SequencerConfigurationValidEnums

```
enum SequencerConfigurationValidEnums
```

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

Enumerator

SequencerConfigurationValid_No	
SequencerConfigurationValid_Yes	
NUM_SEQUENCERCONFIGURATIONVALID	

12.6.2.145 SequencerModeEnums

```
enum SequencerModeEnums
```

< Controls whether or not a sequencer is active.

Enumerator

SequencerMode_Off	
SequencerMode_On	
NUM_SEQUENCERMODE	

12.6.2.146 SequencerSetValidEnums

```
enum SequencerSetValidEnums
```

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

Enumerator

SequencerSetValid_No	
SequencerSetValid_Yes	
NUM_SEQUENCERSETVALID	

12.6.2.147 SequencerTriggerActivationEnums

```
enum SequencerTriggerActivationEnums
```

< Specifies the activation mode of the sequencer trigger.

Enumerator

SequencerTriggerActivation_RisingEdge	
SequencerTriggerActivation_FallingEdge	
SequencerTriggerActivation_AnyEdge	
SequencerTriggerActivation_LevelHigh	
SequencerTriggerActivation_LevelLow	
NUM_SEQUENCERTRIGGERACTIVATION	

12.6.2.148 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	

12.6.2.149 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	

12.6.2.150 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	

12.6.2.151 SerialPortSelectorEnums

`enum SerialPortSelectorEnums`

< Selects which serial port of the device to control.

Enumerator

SerialPortSelector_SerialPort0	
NUM_SERIALPORTSELECTOR	

12.6.2.152 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	

12.6.2.153 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	

12.6.2.154 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	

12.6.2.155 SourceSelectorEnums

```
enum SourceSelectorEnums
```

< Selects the source to control.

Enumerator

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

12.6.2.156 TestPatternEnums

```
enum TestPatternEnums
```

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

Enumerator

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

12.6.2.157 TestPatternGeneratorSelectorEnums

```
enum TestPatternGeneratorSelectorEnums
```

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

Enumerator

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

12.6.2.158 TimerSelectorEnums

```
enum TimerSelectorEnums
```

< Selects which Timer to configure.

Enumerator

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

12.6.2.159 TimerStatusEnums

```
enum TimerStatusEnums
```

< Returns the current status of the Timer.

Enumerator

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

12.6.2.160 TimerTriggerActivationEnums

```
enum TimerTriggerActivationEnums
```

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

Enumerator

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

12.6.2.161 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 specidfied TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_Line1	Starts when the specidfied TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_Line2	Starts when the specidfied 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	

12.6.2.162 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	

12.6.2.163 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	

12.6.2.164 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	

12.6.2.165 TransferQueueModeEnums

`enum TransferQueueModeEnums`

< Specifies the operation mode of the transfer queue.

Enumerator

TransferQueueMode_FirstInFirstOut	Blocks first In are transferred Out first.
NUM_TRANSFERQUEUEMODE	

12.6.2.166 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 simulateneously.
NUM_TRANSFERSELECTOR	

12.6.2.167 TransferStatusSelectorEnums

`enum TransferStatusSelectorEnums`

< Selects which status of the transfer module to read.

Enumerator

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

12.6.2.168 TransferTriggerActivationEnums

`enum TransferTriggerActivationEnums`

< Specifies the activation mode of the transfer control trigger.

Enumerator

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

12.6.2.169 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	

12.6.2.170 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	

12.6.2.171 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	

12.6.2.172 TriggerActivationEnums

```
enum TriggerActivationEnums
```

< Specifies the activation mode of the trigger.

Enumerator

TriggerActivation_LevelLow	
TriggerActivation_LevelHigh	
TriggerActivation_FallingEdge	
TriggerActivation_RisingEdge	
TriggerActivation_AnyEdge	
NUM_TRIGGERACTIVATION	

12.6.2.173 TriggerModeEnums

```
enum TriggerModeEnums
```

< Controls whether or not trigger is active.

Enumerator

TriggerMode_Off	
TriggerMode_On	
NUM_TRIGGERMODE	

12.6.2.174 TriggerOverlapEnums

```
enum TriggerOverlapEnums
```

< Specifies the overlap mode of the trigger.

Enumerator

TriggerOverlap_Off	
TriggerOverlap_ReadOut	
TriggerOverlap_PreviousFrame	
NUM_TRIGGEROVERLAP	

12.6.2.175 TriggerSelectorEnums

```
enum TriggerSelectorEnums
```

< Selects the type of trigger to configure.

Enumerator

TriggerSelector_AcquisitionStart	
TriggerSelector_FrameStart	
TriggerSelector_FrameBurstStart	
NUM_TRIGGERSELECTOR	

12.6.2.176 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	

12.6.2.177 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	

12.6.2.178 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	

12.6.2.179 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	

12.6.2.180 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	

12.7 Camera List Class

Collaboration diagram for Camera List Class:

Classes

- class CameraList

  Used to hold a list of camera objects.
- class ICameraList

  Used to hold a list of camera objects.

12.7.1 Detailed Description

12.8 CameraPtr Class

Collaboration diagram for CameraPtr Class:

Classes

- class CameraPtr

  A reference tracked pointer to a camera object.

Functions

- CameraPtr () throw ()

  Default constructor.
- CameraPtr (const int) throw ()

  Default constructor.
- CameraPtr (const long) throw ()

  Default constructor with argument.
- CameraPtr (const std::nullptr_t) throw ()

12.8.1 Detailed Description

12.8.2 Function Documentation

12.8.2.1 CameraPtr() [1/4]

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

Default constructor.

12.8.2.2 CameraPtr() [2/4]

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

Default constructor.

12.8.2.3 CameraPtr() [3/4]

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

Default constructor with argument.

12.8.2.4 CameraPtr() [4/4]

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

12.9 ChunkData Class

Collaboration diagram for ChunkData Class:

Classes

- class ChunkData

  The chunk data which contains additional information about an image.

12.9.1 Detailed Description

12.10 Chunk Data Inference Class

Collaboration diagram for Chunk Data Inference Class:

Classes

- class InferenceBoundingBoxResult

  An inference bounding boxes object which holds information about the detected bounding boxes.

Functions

- InferenceBoundingBoxResult ()

  Default Constructor.
- ~InferenceBoundingBoxResult ()

  Destructor.
- InferenceBoundingBoxResult (const uint8_t *data, const int64_t lengthInBytes)

  Default Constructor with arguments.
- InferenceBoundingBoxResult (const InferenceBoundingBoxResult &other)

  Copy Constructor.
- InferenceBoundingBoxResult & operator= (const InferenceBoundingBoxResult &rhs)

  Assignment Operator.
- int8_t GetVersion () const

  Returns the bounding box format version number.
- int16_t GetBoxCount () const

  Returns the number of bounding boxes.
- int8_t GetBoxSize () const

  Returns the number of bytes allocated for one bounding box.
- InferenceBoundingBox GetBoxAt (const uint16_t index) const

  Returns the bounding box at specified index.

Variables

- int16_t topLeftXCoord
- int16_t topLeftYCoord
- int16_t bottomRightXCoord
- int16_t bottomRightYCoord
- int16_t centerXCoord
- int16_t centerYCoord
- int16_t radius
- int16_t topLeftXCoord

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

12.10.1 Detailed Description

12.10.2 Function Documentation

12.10.2.1 GetBoxAt()

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

Returns the bounding box at specified index.

Parameters

index	Index of the bounding box to return.

12.10.2.2 GetBoxCount()

```
int16_t GetBoxCount ( ) const
```

Returns the number of bounding boxes.

12.10.2.3 GetBoxSize()

```
int8_t GetBoxSize ( ) const
```

Returns the number of bytes allocated for one bounding box.

12.10.2.4 GetVersion()

```
int8_t GetVersion ( ) const
```

Returns the bounding box format version number.

12.10.2.5 InferenceBoundingBoxResult() [1/3]

```
InferenceBoundingBoxResult ( )
```

Default Constructor.

12.10.2.6 InferenceBoundingBoxResult() [2/3]

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

Default Constructor with arguments.

Parameters

data	The bounding box binary data from chunk data.
lengthInBytes	The length of bounding box binary data in bytes.

12.10.2.7 InferenceBoundingBoxResult() [3/3]

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

Copy Constructor.

12.10.2.8 operator=()

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

Assignment Operator.

12.10.2.9 ~InferenceBoundingBoxResult()

```
~InferenceBoundingBoxResult ( )
```

Destructor.

12.10.3 Variable Documentation

12.10.3.1 bottomRightXCoord [1/2]

```
int16_t bottomRightXCoord
```

12.10.3.2 bottomRightXCoord [2/2]

```
int16_t bottomRightXCoord
```

12.10.3.3 bottomRightYCoord [1/2]

```
int16_t bottomRightYCoord
```

12.10.3.4 bottomRightYCoord [2/2]

```
int16_t bottomRightYCoord
```

12.10.3.5 boxType

```
InferenceBoxType boxType
```

12.10.3.6 centerXCoord

```
int16_t centerXCoord
```

12.10.3.7 centerYCoord

```
int16_t centerYCoord
```

12.10.3.8 circle

```
InferenceBoxCircle circle
```

12.10.3.9 classId

```
int16_t classId
```

12.10.3.10 confidence

```
float32_t confidence
```

12.10.3.11 radius

```
int16_t radius
```

12.10.3.12 rect

```
InferenceBoxRect rect
```

12.10.3.13 rotatedRect

```
InferenceBoxRotatedRect rotatedRect
```

12.10.3.14 rotationAngle

```
short rotationAngle
```

12.10.3.15 topLeftXCoord [1/2]

```
int16_t topLeftXCoord
```

12.10.3.16 topLeftXCoord [2/2]

```
int16_t topLeftXCoord
```

12.10.3.17 topLeftYCoord [1/2]

```
int16_t topLeftYCoord
```

12.10.3.18 topLeftYCoord [2/2]

```
int16_t topLeftYCoord
```

12.11 Spinnaker EventHandler Classes

Collaboration diagram for Spinnaker EventHandler Classes:

Modules

- DeviceArrivalEventHandler Class
- DeviceEventHandler Class

- DeviceRemovalEventHandler Class
- EventHandler Class
- ImageEventHandler Class
- InterfaceArrivalEventHandler Class
- InterfaceEventHandler Class
- InterfaceRemovalEventHandler Class
- Logging EventHandler Class
- LoggingEventDataPtr Class
- LoggingEventHandler Class
- SystemEventHandler Class

Classes

- class DeviceArrivalEventHandler

  An event handler for capturing the device arrival event.
- class DeviceEventHandler

  A handler to device events.
- class DeviceRemovalEventHandler

  An event handler for capturing the device removal event.
- class EventHandler

  The base class for all event handler types.
- class ImageEventHandler

  A handler for capturing image arrival events.
- class InterfaceArrivalEventHandler

  An event handler for capturing the interface arrival event.
- class InterfaceEventHandler

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

  An event handler for capturing the interface removal event.
- class LoggingEventData

  The LoggingEventData object.
- class LoggingEventDataPtr

  A reference tracked pointer to the LoggingEvent object.
- class LoggingEventHandler

  An event handler for capturing the device logging event.
- class SystemEventHandler

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

12.11.1 Detailed Description

12.12 DeviceArrivalEventHandler Class

Collaboration diagram for DeviceArrivalEventHandler Class:

Classes

- class DeviceArrivalEventHandler

  An event handler for capturing the device arrival event.

12.12.1 Detailed Description

12.13 DeviceEventHandler Class

Collaboration diagram for DeviceEventHandler Class:

Classes

- class DeviceEventHandler

  A handler to device events.

12.13.1 Detailed Description

12.14 DeviceRemovalEventHandler Class

Collaboration diagram for DeviceRemovalEventHandler Class:

Classes

- class DeviceRemovalEventHandler

  An event handler for capturing the device removal event.

12.14.1 Detailed Description

12.15 EventHandler Class

Collaboration diagram for EventHandler Class:

Classes

- class EventHandler

  The base class for all event handler types.

12.15.1 Detailed Description

12.16 Exception Class

Collaboration diagram for Exception Class:

Classes

- class Exception

  The Exception object represents an error that is returned from the library.

12.16.1 Detailed Description

12.17 Image Class

Collaboration diagram for Image Class:

Classes

- class Image

  The image object class.

12.17.1 Detailed Description

12.18 ImageEventHandler Class

Collaboration diagram for ImageEventHandler Class:

Classes

- class ImageEventHandler

  A handler for capturing image arrival events.

12.18.1 Detailed Description

12.19 ImagePtr Class

Collaboration diagram for ImagePtr Class:

Classes

- class ImagePtr

  A reference tracked pointer to an image object.

12.19.1 Detailed Description

12.20 ImageStatistics Class

Collaboration diagram for ImageStatistics Class:

Classes

- class ImageStatistics

  Represents image statistics for an image.

12.20.1 Detailed Description

12.21 Image Utility Class

Collaboration diagram for Image Utility Class:

Classes

- class ImageUtility

  Static helper functions for the image object class.

12.21.1 Detailed Description

12.22 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.

12.22.1 Detailed Description

12.23 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.

12.23.1 Detailed Description

12.24 Interface Class

Collaboration diagram for Interface Class:

Classes

- class Interface

  An interface object which holds a list of cameras.

12.24.1 Detailed Description

12.25 InterfaceArrivalEventHandler Class

Collaboration diagram for InterfaceArrivalEventHandler Class:

Classes

- class InterfaceArrivalEventHandler

  An event handler for capturing the interface arrival event.

12.25.1 Detailed Description

12.26 InterfaceEventHandler Class

Collaboration diagram for InterfaceEventHandler Class:

Classes

- class InterfaceEventHandler

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

12.26.1 Detailed Description

12.27 InterfaceList Class

Collaboration diagram for InterfaceList Class:

Classes

- class InterfaceList

  A list of the available interfaces on the system.

12.27.1 Detailed Description

12.28 InterfacePtr Class

Collaboration diagram for InterfacePtr Class:

Classes

- class InterfacePtr

  A reference tracked pointer to the interface object.

12.28.1 Detailed Description

12.29 InterfaceRemovalEventHandler Class

Collaboration diagram for InterfaceRemovalEventHandler Class:

Classes

- class InterfaceRemovalEventHandler

  An event handler for capturing the interface removal event.

12.29.1 Detailed Description

12.30 Logging EventHandler Class

Collaboration diagram for Logging EventHandler Class:

Classes

- class LoggingEventData

  The LoggingEventData object.

12.30.1 Detailed Description

12.31 LoggingEventDataPtr Class

Collaboration diagram for LoggingEventDataPtr Class:

Classes

- class LoggingEventDataPtr

  A reference tracked pointer to the LoggingEvent object.

12.31.1 Detailed Description

12.32 LoggingEventHandler Class

Collaboration diagram for LoggingEventHandler Class:

Classes

- class LoggingEventHandler

  An event handler for capturing the device logging event.

12.32.1 Detailed Description

12.33 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 = 0xFFFFFFFFFFFFFFFF

12.33.1 Detailed Description

12.33.2 Variable Documentation

12.33.2.1 EVENT_TIMEOUT_INFINITE

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

12.33.2.2 EVENT_TIMEOUT_NONE

```
const uint64_t EVENT_TIMEOUT_NONE = 0
```

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

12.34 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-supp

12.35 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,

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 }

12.35.1 Detailed Description

Definitions file for Spinnaker.

12.35.2 Enumeration Type Documentation

12.35.2.1 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	

12.35.2.2 BufferOwnership

`enum BufferOwnership`

Enumerator

BUFFER_OWNERSHIP_SYSTEM	
BUFFER_OWNERSHIP_USER	

12.35.2.3 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.
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.

12.35.2.4 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	
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	

Enumerator

SPINNAKER_ERR_IM_MIN_MAX	
SPINNAKER_ERR_IM_COLOR_CONVERSION	
SPINNAKER_ERR_IM_DECOMPRESSION	
SPINNAKER_ERR_CUSTOM_ID	

12.35.2.5 EventType

`enum EventType`

Event types in Spinnaker.

See also

EventHandler::GetEventType()

Enumerator

SPINNAKER_EVENT_ARRIVAL_REMOVAL	
SPINNAKER_EVENT_DEVICE	
SPINNAKER_EVENT_DEVICE_SPECIFIC	
SPINNAKER_EVENT_NEW_BUFFER	
SPINNAKER_EVENT_LOGGING_EVENT	
SPINNAKER_EVENT_UNKNOWN	
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL	

12.35.2.6 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	

12.35.2.7 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. Potential fixes include enabling jumbo packets and adjusting packet size/delay. For more information see https://www.flir.↩ com/support-center/iis/machine-vision/application
IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT	Image leader is incomplete. Could be caused by missing packet(s). See link above.
IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT	Image trailer is incomplete. Could be caused by missing packet(s). See link above.
IMAGE_PACKETID_INCONSISTENT	Image has an inconsistent packet id. Could be caused by missing packet(s). See link above.
IMAGE_MISSING_LEADER	Image leader is missing. Could be caused by missing packet(s). See link above.
IMAGE_MISSING_TRAILER	Image trailer is missing. Could be caused by missing packet(s). See link above.
IMAGE_DATA_INCOMPLETE	Image data is incomplete. Could be caused by missing packet(s). See link above.
IMAGE_INFO_INCONSISTENT	Image info is corrupted. Could be caused by missing packet(s). See link above.
IMAGE_CHUNK_DATA_INVALID	Image chunk data is invalid.
IMAGE_NO_SYSTEM_RESOURCES	Image cannot be processed due to lack of system resources.

12.35.2.8 PayloadTypeInfoIDs

```
enum PayloadTypeInfoIDs
```

Enumerator

PAYLOAD_TYPE_UNKNOWN	
PAYLOAD_TYPE_IMAGE	
PAYLOAD_TYPE_RAW_DATA	
PAYLOAD_TYPE_FILE	

Enumerator

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	

12.35.2.9 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	

12.35.2.10 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	

12.35.2.11 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	

12.35.2.12 StatisticsChannel

```
enum StatisticsChannel
```

Channels that allow statistics to be calculated.

Enumerator

GREY	
RED	
GREEN	
BLUE	
HUE	
SATURATION	
LIGHTNESS	
NUM_STATISTICS_CHANNELS	

12.36 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")))

12.36.1 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.

12.36.2 Macro Definition Documentation

12.36.2.1 SPINNAKER_API

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

12.36.2.2 SPINNAKER_API_ABSTRACT

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

12.36.2.3 SPINNAKER_LOCAL

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

12.37 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.

12.37.1 Detailed Description

12.38 Spinnaker Video Definitions

Definitions file for Spinnaker video recorder.

Collaboration diagram for Spinnaker Video Definitions:

Definitions file for Spinnaker video recorder.

12.39 System Class

Collaboration diagram for System Class:

Classes

- class System

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

12.39.1 Detailed Description

12.40 SystemEventHandler Class

Collaboration diagram for SystemEventHandler Class:

Classes

- class SystemEventHandler

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

12.40.1 Detailed Description

12.41 SystemPtr Class

Collaboration diagram for SystemPtr Class:

Classes

- class SystemPtr

  A reference tracked pointer to a system object.

12.41.1 Detailed Description

12.42 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.

12.42.1 Detailed Description

12.43 TransportLayerDefs Class

Collaboration diagram for TransportLayerDefs Class:

Enumerations

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

  The enum definitions for TL Device nodes from the transport layer .xml files.
- enum StreamBufferCountModeEnum {
  StreamBufferCountMode_Manual,
  StreamBufferCountMode_Auto,
  NUMSTREAMBUFFERCOUNTMODE }
- enum StreamBufferHandlingModeEnum {
  StreamBufferHandlingMode_OldestFirst,
  StreamBufferHandlingMode_OldestFirstOverwrite,
  StreamBufferHandlingMode_NewestOnly,
  StreamBufferHandlingMode_NewestFirst,
  NUMSTREAMBUFFERHANDLINGMODE }
- enum DeviceTypeEnum {
  DeviceType_GigEVision,
  DeviceType_CameraLink,
  DeviceType_CameraLinkHS,
  DeviceType_CoaXPress,
  DeviceType_USB3Vision,
  DeviceType_Custom,
  NUMDEVICETYPE }
- enum DeviceAccessStatusEnum {
  DeviceAccessStatus_Unknown,
  DeviceAccessStatus_ReadWrite,
  DeviceAccessStatus_ReadOnly,
  DeviceAccessStatus_NoAccess,
  DeviceAccessStatus_Busy,
  DeviceAccessStatus_OpenReadWrite,
  DeviceAccessStatus_OpenReadOnly,
  NUMDEVICEACCESSSTATUS }

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

12.43.1 Detailed Description

12.43.2 Enumeration Type Documentation

12.43.2.1 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	

12.43.2.2 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	

12.43.2.3 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	

12.43.2.4 DeviceTypeEnum

```
enum DeviceTypeEnum
```

< Transport layer type of the device.

Enumerator

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

12.43.2.5 FilterDriverStatusEnum

```
enum FilterDriverStatusEnum
```

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

Enumerator

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

12.43.2.6 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	

12.43.2.7 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	

12.43.2.8 GUIXMLLocationEnum

```
enum GUIXMLLocationEnum
```

< Sets the location to load GUI XML.

Enumerator

GUIXMLLocation_Device	Load XML from device
GUIXMLLocation_Host	Load XML from host
NUMGUIXMLLOCATION	

12.43.2.9 InterfaceTypeEnum

```
enum InterfaceTypeEnum
```

< Transport layer type of the interface.

Enumerator

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

12.43.2.10 POEStatusEnum

```
enum POEStatusEnum
```

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

Enumerator

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

12.43.2.11 StreamBufferCountModeEnum

```
enum StreamBufferCountModeEnum
```

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

Enumerator

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

12.43.2.12 StreamBufferHandlingModeEnum

```
enum StreamBufferHandlingModeEnum
```

< Available buffer handling modes of this data stream:

Enumerator

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

Enumerator

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

12.43.2.13 StreamTypeEnum

`enum StreamTypeEnum`

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

< Stream type of the device.

Enumerator

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

12.43.2.14 TLTypeEnum

`enum TLTypeEnum`

< Transport layer type of the GenTL Producer implementation.

Enumerator

TLType_GigEVision	GigE Vision
TLType_CameraLink	Camera Link
TLType_CameraLinkHS	Camera Link High Speed
TLType_CoaXPress	CoaXPress

Enumerator

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

12.44 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.

12.44.1 Detailed Description

12.45 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.

12.45.1 Detailed Description

12.46 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.

12.46.1 Detailed Description

12.47 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.

12.47.1 Detailed Description

12.48 Camera Base Interface Class

Collaboration diagram for Camera Base Interface Class:

Classes

- class ICameraBase

  The interface file for base class for the camera object.

12.48.1 Detailed Description

12.49 IChunkData Class

Collaboration diagram for IChunkData Class:

Classes

- class IChunkData

  The Interface file for ChunkData.

12.49.1 Detailed Description

12.50 IImage Class

Collaboration diagram for IImage Class:

Classes

- class IImage

  The interface file for Image.

12.50.1 Detailed Description

12.51 IImageStatistics Class

Collaboration diagram for IImageStatistics Class:

Classes

- class IImageStatistics

  The interface file for image statistics.

12.51.1 Detailed Description

12.52 IInterface Class

Collaboration diagram for IInterface Class:

Classes

- class IInterface

  The interface file for Interface.

12.52.1 Detailed Description

12.53 IInterfaceList Class

Collaboration diagram for IInterfaceList Class:

Classes

- class IInterfaceList

  The interface file for InterfaceList class.

12.53.1 Detailed Description

12.54 ISystem Class

Collaboration diagram for ISystem Class:

Classes

- class ISystem

  The interface file for System.

12.54.1 Detailed Description

12.55 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 >

Container for a function pointer.

- class Member_NodeCallback< Client, Member >

  Container for a member function pointer.
- class NodeMap

  Smart pointer template for NodeMaps with create function.
- class CNodeMapFactory

  The node map factory is used for creating node maps from camera description files.
- class CNodeMapRefT< TCameraParams >

  Smartpointer template for NodeMaps with create function.
- class CGeneric_XMLLoaderParams

  Empty base class used by class CNodeMapRef as generic template argument.
- class CNodeMapRef

  Smartpointer for NodeMaps with create function.
- class CFeatureBag

  Bag holding streamable features of a nodetree.
- class CPointer< T, B >

  Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.
- class CFloatPtr

  SmartPointer for IFloat interface pointer.
- class CPortImpl

  Standard implementation for a port.
- class PortNode

  Interface for value properties.
- class PortRecorder

  Interface for recording write commands on a port.
- class PortReplay

  Interface for replaying write commands on a port.
- class CPortWriteList

  Container holding a list of port write commands.
- class RegisterNode

  Interface for string properties.
- class CRegisterPortImpl

  Standard implementation for a port using a register based transport layer.
- class CSelectorSet

  The set of selectors selecting a given node.
- class SpinTestCamera
- 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.

12.55.1 Detailed Description

12.55.2 Typedef Documentation

12.55.2.1 CNodeMapRef

```
typedef NodeMap CNodeMapRef
```

12.55.2.2 CNodeRef

```
typedef Node CNodeRef
```

12.55.2.3 CSelectorRef

```
typedef Node CSelectorRef
```

12.55.3 Function Documentation

12.55.3.1 _ClearXMLCache()

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

12.55.3.2 _Connect() [1/2]

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

12.55.3.3 _Connect() [2/2]

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

12.55.3.4 _Destroy()

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

Destroys the node map.

12.55.3.5 _GetDeviceName()

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

12.55.3.6 _GetNode()

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

12.55.3.7 _GetNodes()

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

12.55.3.8 _GetSupportedSchemaVersions()

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

12.55.3.9 _InvalidateNodes()

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

12.55.3.10 _LoadXMLFromFile()

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

12.55.3.11 _LoadXMLFromFileInject()

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

12.55.3.12 _LoadXMLFromString()

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

12.55.3.13 _LoadXMLFromStringInject()

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

12.55.3.14 _LoadXMLFromZIPData()

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

12.55.3.15 _LoadXMLFromZIPFile()

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

12.55.3.16 _Poll()

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

12.55.3.17 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)

12.55.3.18 CNodeMapRefT() [1/3]

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

Constructor.

12.55.3.19 CNodeMapRefT() [2/3]

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

Constructor.

12.55.3.20 CNodeMapRefT() [3/3]

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

Copy constructor.

12.55.3.21 EatComments()

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

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

12.55.3.22 operator<<()

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

writes out persistent data to a stream

12.55.3.23 operator=() [1/2]

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

Assignment.

12.55.3.24 operator=() [2/2]

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

Assignment of an INodeMap*.

12.55.3.25 operator>>()

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

Reads in persistent data from a stream.

12.55.3.26 ~CNodeMapRefT()

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

Destructor.

12.56 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.

12.56.1 Detailed Description

12.57 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

12.57.1 Detailed Description

12.57.2 Typedef Documentation

12.57.2.1 CallbackHandleType

```
typedef intptr_t CallbackHandleType
```

the callback handle for nodes

12.57.2.2 NodeList_t

```
typedef node_vector NodeList_t
```

a list of node references

12.58 IBase Interface

Collaboration diagram for IBase Interface:

Variables

- interface SPINNAKER_API_ABSTRACT IBase

  Base interface common to all nodes.

12.58.1 Detailed Description

12.58.2 Variable Documentation

12.58.2.1 IBase

```
interface SPINNAKER_API_ABSTRACT IBase
```

Initial value:

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

Base interface common to all nodes.

12.59 BooleanNode Class

Collaboration diagram for BooleanNode Class:

Classes

- class BooleanNode

  Interface for string properties.

Typedefs

- typedef BooleanNode CBooleanRef

12.59.1 Detailed Description

12.59.2 Typedef Documentation

12.59.2.1 CBooleanRef

```
typedef BooleanNode CBooleanRef
```

12.60 CategoryNode Class

Collaboration diagram for CategoryNode Class:

Classes

- class CategoryNode

  Interface for string properties.

Typedefs

- typedef CategoryNode CCategoryRef

12.60.1 Detailed Description

12.60.2 Typedef Documentation

12.60.2.1 CCategoryRef

```
typedef CategoryNode CCategoryRef
```

12.61 ChunkAdapter Class

Collaboration diagram for ChunkAdapter Class:

Classes

- class CChunkAdapter

  Connects a chunked buffer to a node map.

12.61.1 Detailed Description

12.62 ChunkAdapterDcam Class

Collaboration diagram for ChunkAdapterDcam Class:

Classes

- class CChunkAdapterDcam

  Connects a chunked DCAM buffer to a node map.

12.62.1 Detailed Description

12.63 ChunkAdapterGeneric Class

Collaboration diagram for ChunkAdapterGeneric Class:

Classes

- class CChunkAdapterGeneric

12.63.1 Detailed Description

12.64 ChunkAdapterGEV Class

Collaboration diagram for ChunkAdapterGEV Class:

Classes

- class CChunkAdapterGEV

  Connects a chunked DCAM buffer to a node map.

12.64.1 Detailed Description

12.65 ChunkPort Class

Collaboration diagram for ChunkPort Class:

Classes

- class CChunkPort

  Port attachable to a chunk in a buffer.

12.65.1 Detailed Description

12.66 CommandNode Class

Collaboration diagram for CommandNode Class:

Classes

- class CommandNode

  Interface for string properties.

Typedefs

- typedef CommandNode CCommandRef

12.66.1 Detailed Description

12.66.2 Typedef Documentation

12.66.2.1 CCommandRef

```
typedef CommandNode CCommandRef
```

12.67 Container Class

Collaboration diagram for Container Class:

12.68 Counter Class

Collaboration diagram for Counter Class:

Classes

- class Counter

  Definition of a simple Counter class.

12.68.1 Detailed Description

12.69 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

12.69.1 Detailed Description

12.70 EnumEntryNode Class

Collaboration diagram for EnumEntryNode Class:

Classes

- class EnumEntryNode

  Interface for string properties.

Typedefs

- typedef EnumEntryNode CEnumEntryRef

12.70.1 Detailed Description

12.70.2 Typedef Documentation

12.70.2.1 CEnumEntryRef

```
typedef EnumEntryNode CEnumEntryRef
```

12.71 EnumNode Class

Collaboration diagram for EnumNode Class:

Classes

- class EnumNode

  Interface for string properties.

Typedefs

- typedef EnumNode CEnumerationRef

12.71.1 Detailed Description

12.71.2 Typedef Documentation

12.71.2.1 CEnumerationRef

```
typedef EnumNode CEnumerationRef
```

12.72 EnumNodeT Class

Collaboration diagram for EnumNodeT Class:

Classes

- class CEnumerationTRef< EnumT >

  Interface for string properties.

12.72.1 Detailed Description

12.73 EventAdapter Class

Collaboration diagram for EventAdapter Class:

Classes

- class CEventAdapter

  Delivers Events to ports.

12.73.1 Detailed Description

12.74 EventAdapter1394 Class

Collaboration diagram for EventAdapter1394 Class:

Classes

- class CEventAdapter1394

  Distribute the events to the node map.

12.74.1 Detailed Description

12.75 EventAdapterGeneric Class

Collaboration diagram for EventAdapterGeneric Class:

Classes

- class CEventAdapterGeneric

  Connects a generic event to a node map.

12.75.1 Detailed Description

12.76 EventAdapterGEV Class

Collaboration diagram for EventAdapterGEV Class:

Classes

- class CEventAdapterGEV

  Connects a GigE Event to a node map.

12.76.1 Detailed Description

12.77 EventAdapterU3V Class

Collaboration diagram for EventAdapterU3V Class:

Classes

- class CEventAdapterU3V

  Connects a U3V Event to a node map.

12.77.1 Detailed Description

12.78 EventPort Class

Collaboration diagram for EventPort Class:

Classes

- class CEventPort

  Port attachable to an event.

12.78.1 Detailed Description

12.79 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 >

12.79.1 Detailed Description

12.80 FloatNode Class

Collaboration diagram for FloatNode Class:

Classes

- class FloatNode

  Interface for string properties.

Typedefs

- typedef FloatNode CFloatRef

12.80.1 Detailed Description

12.80.2 Typedef Documentation

12.80.2.1 CFloatRef

```
typedef FloatNode CFloatRef
```

12.81 FloatRegNode Class

Collaboration diagram for FloatRegNode Class:

Classes

- class FloatRegNode

  Interface for string properties.

12.81.1 Detailed Description

12.82 GCString Class

Collaboration diagram for GCString Class:

Classes

- class gcstring

12.82.1 Detailed Description

12.83 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.

12.83.1 Detailed Description

12.84 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

12.84.1 Detailed Description

12.84.2 Typedef Documentation

12.84.2.1 float32_t

```
typedef float float32_t
```

32 bit floating point

12.84.2.2 float64_t

```
typedef double float64_t
```

64 bit floating point

12.85 Spinnaker GenApi Utilities

Collaboration diagram for Spinnaker GenApi Utilities:

Modules

- GCUtilities Utility

12.85.1 Detailed Description

12.86 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 &Variable↩Name)

  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 unsave 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 SetGenICam↩CacheFolder().
- 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 SetGenICam↩CLProtocolFolder().
- SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)

  Stores the path of the GenICam cache folder.
- SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)

  Stores the path of the GenICam logging properties file.
- SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)

  Stores the path of the CLProtocol folder.
- SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")

splits str input string into a list of tokens using the delimiter

- SPINNAKER_API void GetFiles (const gcstring &FileTemplate, gcstring_vector &FileNames, const bool DirectoriesOnly=false)

  Gets a list of files or directories matching a given FileTemplate.

- SPINNAKER_API gcstring GetModulePathFromFunction (void *pFunction)

  Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

12.86.1 Detailed Description

12.86.2 Function Documentation

12.86.2.1 DoesEnvironmentVariableExist()

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

Returns true if an environment variable exists.

12.86.2.2 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

FileNames	The file template. Can contain environment variables.
DirectoriesOnly	A list of files matching the file template

12.86.2.3 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 SetGenICam↩CacheFolder().

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.

12.86.2.4 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 SetGenICam↩CLProtocolFolder().

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.

12.86.2.5 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.

12.86.2.6 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

12.86.2.7 GetValueOfEnvironmentVariable() [1/2]

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetValueOfEnvironmentVariable (
            const gcstring & VariableName )
```

Retrieve the value of an environment variable.

Exceptions

runtime_exception	if not found

12.86.2.8 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

12.86.2.9 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)

12.86.2.10 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)

12.86.2.11 ReplaceEnvironmentVariables()

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

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

12.86.2.12 SetGenICamCacheFolder()

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

Stores the path of the GenICam cache folder.

12.86.2.13 SetGenICamCLProtocolFolder()

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

Stores the path of the CLProtocol folder.

12.86.2.14 SetGenICamLogConfig()

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

Stores the path of the GenICam logging properties file.

12.86.2.15 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

str	string to be split
tokens	result of the splitting operation
delimiters	delimiters for the splitting

12.86.2.16 UrlDecode()

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

Replaces xx escapes by their char equivalent.

12.86.2.17 UrlEncode()

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

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

12.87 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

12.87.1 Detailed Description

12.87.2 Function Documentation

12.87.2.1 GetValue()

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

Get 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

12.87.2.2 operator()()

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

Get node value.

Execute the command.

12.87.2.3 operator=()

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

Set node value.

12.87.3 Variable Documentation

12.87.3.1 IBoolean

```
interface SPINNAKER_API_ABSTRACT IBoolean
```

Interface for Boolean properties.

12.87.3.2 Verify

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

12.88 ICategory Interfaces

Collaboration diagram for ICategory Interfaces:

Variables

- interface SPINNAKER_API_ABSTRACT ICategory

  Gives access to a category node.

12.88.1 Detailed Description

12.88.2 Variable Documentation

12.88.2.1 ICategory

```
interface SPINNAKER_API_ABSTRACT ICategory
```

Gives access to a category node.

12.89 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

  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.

12.89.1 Detailed Description

12.89.2 Macro Definition Documentation

12.89.2.1 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)

12.89.2.2 CHUNK_BASE_ADDRESS_REGISTER_LEN

```
#define CHUNK_BASE_ADDRESS_REGISTER_LEN 8
```

Lenght of the CHUNK_BASE_ADDRESS_REGISTER pseudo register.

12.89.2.3 CHUNK_LENGTH_REGISTER

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

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

12.89.2.4 CHUNK_LENGTH_REGISTER_LEN

```
#define CHUNK_LENGTH_REGISTER_LEN 8
```

Lenght of the CHUNK_LENGTH_REGISTER pseudo register.

12.89.3 Function Documentation

12.89.3.1 CacheChunkData()

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

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

12.89.4 Variable Documentation

12.89.4.1 IChunkPort

```
interface SPINNAKER_API_ABSTRACT IChunkPort
```

Interface for ports attached to a chunk.

12.90 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.

12.90.1 Detailed Description

12.90.2 Function Documentation

12.90.2.1 IsDone()

```
virtual bool Spinnaker::GenApi::IsDone (
            bool Verify = true )  [pure virtual]
```

Query whether the command is executed.

Parameters

Verify	Enables Range verification (default = false). The AccessMode is always checked

Returns

True if the Execute command has finished; false otherwise

12.90.3 Variable Documentation

12.90.3.1 ICommand

```
interface SPINNAKER_API_ABSTRACT ICommand
```

Interface for command like properties.

12.91 IDestroy Interface

Collaboration diagram for IDestroy Interface:

Variables

- interface SPINNAKER_API_ABSTRACT IDestroy

  Interface to destroy an object.

12.91.1 Detailed Description

12.91.2 Variable Documentation

12.91.2.1 IDestroy

```
interface SPINNAKER_API_ABSTRACT IDestroy
```

Initial value:

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

Interface to destroy an object.

12.92 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)

12.92.1 Detailed Description

12.92.2 Function Documentation

12.92.2.1 GetDeviceVersion()

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

Get the version of the device description file.

12.92.2.2 GetGenApiVersion()

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

Get the version of the DLL's GenApi implementation.

12.92.2.3 GetProductGuid()

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

Get the Guid describing the product.

12.92.2.4 GetSchemaVersion()

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

Get the schema version number.

12.92.2.5 GetStandardNameSpace()

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

Get the standard name space.

12.92.2.6 GetToolTip()

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

Get tool tip.

Get a short description of the node.

12.92.2.7 GetVendorName()

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

Get the vendor name.

12.92.2.8 GetVersionGuid()

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

Get the Guid describing the product version.

12.92.3 Variable Documentation

12.92.3.1 IDeviceInfo

```
interface SPINNAKER_API_ABSTRACT IDeviceInfo
```

Initial value:

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

Interface to get information about the device (= nodemap)

12.93 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.

12.93.1 Detailed Description

12.93.2 Function Documentation

12.93.2.1 GetNumericValue()

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

Get double number associated with the entry.

12.93.2.2 GetSymbolic()

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

Get symbolic enum value.

12.93.2.3 IsSelfClearing()

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

Indicates if the corresponding EnumEntry is self clearing.

12.93.3 Variable Documentation

12.93.3.1 IEnumEntry

```
interface SPINNAKER_API_ABSTRACT IEnumEntry
```

Interface of single enum value.

Maps of Enum Values to symbolic values

12.94 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.

12.94.1 Detailed Description

12.94.2 Function Documentation

12.94.2.1 GetCurrentEntry()

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

Get the current entry.

12.94.2.2 GetEntries()

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

Get list of entry nodes.

12.94.2.3 GetEntry()

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

Get an entry node by its IntValue.

12.94.2.4 GetEntryByName()

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

Get an entry node by name.

12.94.2.5 GetIntValue()

```
virtual int64_t Spinnaker::GenApi::GetIntValue (
            bool Verify = false,
            bool IgnoreCache = false )  [pure 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

12.94.2.6 operator∗()

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

Get string node value.

Get node value.

12.94.2.7 SetIntValue()

```
virtual void Spinnaker::GenApi::SetIntValue (
            int64_t Value,
            bool Verify = true )  [pure virtual]
```

Set integer node value.

Parameters

Value	The value to set
Verify	Enables AccessMode and Range verification (default = true)

12.94.3 Variable Documentation

12.94.3.1 IEnumeration

```
interface SPINNAKER_API_ABSTRACT IEnumeration
```

Interface for enumeration properties.

12.95 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 GenICam::gcstring &ValueStr)=0

  Set string node value.

Variables

- template<typename EnumT >
  interface SPINNAKER_API_ABSTRACT IEnumerationT

  Interface for enumeration properties.
- template<typename EnumT >
  interface SPINNAKER_API_ABSTRACT virtual public IEnumReference

  Interface to construct an enum reference.

12.95.1 Detailed Description

12.95.2 Function Documentation

12.95.2.1 GetEntry()

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

returns the EnumEntry object belonging to the Value

12.95.2.2 operator=() [1/2]

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

Set node value.

12.95.2.3 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

12.95.3 Variable Documentation

12.95.3.1 IEnumerationT

```
interface SPINNAKER_API_ABSTRACT IEnumerationT
```

Interface for enumeration properties.

12.95.3.2 IEnumReference

```
interface SPINNAKER_API_ABSTRACT IEnumReference
```

Initial value:

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

Interface to construct an enum reference.

12.96 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.

12.96.1 Detailed Description

12.96.2 Function Documentation

12.96.2.1 GetDisplayNotation()

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

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

12.96.2.2 GetDisplayPrecision()

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

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

12.96.2.3 GetInc()

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

Get the constant increment if there is any.

Get increment.

12.96.2.4 GetIncMode()

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

Get increment mode.

12.96.2.5 GetListOfValidValues()

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

Get list of valid value.

12.96.2.6 GetMax()

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

Get maximum value allowed.

12.96.2.7 GetMin()

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

Get minimum value allowed.

12.96.2.8 GetRepresentation()

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

Get recommended representation.

12.96.2.9 GetUnit()

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

Get the physical unit name.

12.96.2.10 HasInc()

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

True if the float has a constant increment.

12.96.2.11 ImposeMax()

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

Restrict maximum value.

12.96.2.12 ImposeMin()

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

Restrict minimum value.

12.96.2.13 operator=()

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

Set node value.

12.96.3 Variable Documentation

12.96.3.1 IFloat

```
interface SPINNAKER_API_ABSTRACT IFloat
```

Interface for float properties.

12.97 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.

12.97.1 Detailed Description

12.97.2 Function Documentation

12.97.2.1 ImposeMax()

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

Restrict maximum value.

12.97.2.2 ImposeMin()

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

Restrict minimum value.

12.97.2.3 operator=()

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

Set node value.

12.97.3 Variable Documentation

12.97.3.1 IInteger

```
interface SPINNAKER_API_ABSTRACT IInteger
```

Interface for integer properties.

12.98 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 MaxVisiblity)

  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.

12.98.1 Detailed Description

12.98.2 Function Documentation

12.98.2.1 Combine() [1/3]

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

Computes which access mode the two guards allow together.

12.98.2.2 Combine() [2/3]

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

Computes which visibility the two guards allow together.

12.98.2.3 Combine() [3/3]

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

Computes which CachingMode results from a combination.

12.98.2.4 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

12.98.2.5 GetAlias()

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

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

12.98.2.6 GetCachingMode()

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

Get Caching Mode.

12.98.2.7 GetCastAlias()

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

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

12.98.2.8 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

out	*Children*	List of children nodes
	LinkType	The link type

12.98.2.9 GetDescription()

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

Get a long description of the node.

12.98.2.10 GetDisplayName()

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

Get a name string for display.

12.98.2.11 GetDocuURL()

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

Gets a URL pointing to the documentation of that feature.

12.98.2.12 GetEventID()

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

Get the EventId of the node.

12.98.2.13 GetNameSpace()

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

Get name space.

12.98.2.14 GetNodeMap()

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

Retrieves the central node map.

12.98.2.15 GetParents()

```
virtual void Spinnaker::GenApi::GetParents (
            GenApi::NodeList_t & Parents ) const  [pure virtual]
```

Gets all nodes this node is directly depending on.

Parameters

out	*Parents*	List of parent nodes

12.98.2.16 GetPollingTime()

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

recommended polling time (for non-cacheable nodes)

12.98.2.17 GetPrincipalInterfaceType()

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

Get the type of the main interface of a node.

12.98.2.18 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.

12.98.2.19 GetPropertyNames()

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

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

12.98.2.20 GetVisibility()

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

Get the recommended visibility of the node.

12.98.2.21 ImposeAccessMode()

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

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

12.98.2.22 ImposeVisibility()

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

Imposes a visibility to the natural visibility of the node.

12.98.2.23 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

12.98.2.24 IsAccessModeCacheable()

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

True if the AccessMode can be cached.

12.98.2.25 IsAvailable() [1/3]

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

Tests if available.

12.98.2.26 IsAvailable() [2/3]

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

Checks if a node is available.

12.98.2.27 IsAvailable() [3/3]

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

Checks if a node is available.

12.98.2.28 IsCachable()

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

Is the node value cacheable.

12.98.2.29 IsCacheable()

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

Tests Cacheability.

12.98.2.30 IsDeprecated()

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

True if the node should not be used any more.

12.98.2.31 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".

12.98.2.32 IsImplemented() [1/3]

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

Tests if implemented.

12.98.2.33 IsImplemented() [2/3]

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

Checks if a node is implemented.

12.98.2.34 IsImplemented() [3/3]

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

Checks if a node is implemented.

12.98.2.35 IsReadable() [1/3]

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

Tests if readable.

12.98.2.36 IsReadable() [2/3]

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

Checks if a node is readable.

12.98.2.37 IsReadable() [3/3]

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

Checks if a node is readable.

12.98.2.38 IsStreamable()

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

True if the node is streamable.

12.98.2.39 IsVisible()

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

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

12.98.2.40 IsWritable() [1/3]

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

Tests if writable.

12.98.2.41 IsWritable() [2/3]

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

Checks if a node is writable.

12.98.2.42 IsWritable() [3/3]

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

Checks if a node is writable.

12.98.2.43 operator"!=()

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

12.98.2.44 operator==()

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

12.98.2.45 RegisterCallback()

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

Register change callback Takes ownership of the CNodeCallback object.

12.98.3 Variable Documentation

12.98.3.1 INode

```
interface SPINNAKER_API_ABSTRACT INode
```

Interface common to all nodes.

12.98.3.2 IReference

```
interface SPINNAKER_API_ABSTRACT IReference
```

Initial value:

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

Interface to construct a reference.

12.99 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.

12.99.1 Detailed Description

12.99.2 Function Documentation

12.99.2.1 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.

12.99.2.2 Connect() [2/2]

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

Connects a port to the standard port "Device".

12.99.2.3 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".

12.99.2.4 GetLock()

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

Returns the lock which guards the node map.

12.99.2.5 GetNode()

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

Retrieves the node from the central map by Name.

12.99.2.6 GetNumNodes()

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

Get the number of nodes in the map.

12.99.2.7 InvalidateNodes()

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

Invalidates all nodes.

12.99.2.8 Poll()

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

Fires nodes which have a polling time.

12.99.3 Variable Documentation

12.99.3.1 INodeMap

```
interface SPINNAKER_API_ABSTRACT INodeMap
```

Initial value:

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

Interface to access the node map.

12.100 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=xv↩Default)=0

  Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void MergeXMLFiles (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectedFileName, const GenICam::gcstring &OutputFileName)=0

  Injects an XML file into a target file.
- virtual void ExtractIndependentSubtree (const GenICam::gcstring &XMLData, const GenICam::gcstring &InjectXMLData, const GenICam::gcstring &SubTreeRootNodeName, GenICam::gcstring &Extracted↩Subtree)=0

  Extract independent subtree.
- virtual void GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions)=0

  Gets a list of supported schema versions.
- virtual void LoadXMLFromZIPFile (const GenICam::gcstring &ZipFileName)=0

  Loads an XML from a ZIP file.
- virtual void LoadXMLFromZIPData (const void *zipData, size_t zipSize)=0

  Loads an XML from a ZIP data buffer.
- virtual void PreprocessXMLFromZIPFile (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xv↩Default)=0

  Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

Variables

- interface SPINNAKER_API_ABSTRACT INodeMapDyn

  Interface to access the node map.

12.100.1 Detailed Description

12.100.2 Function Documentation

12.100.2.1 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

InjectXMLData	The XML data the subtree is extracted from.
SubTreeRootNodeName	Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed.
ExtractedSubtree	The name of the node that represents the root of the subtree that shall be extracted.> The returned extracted subtree as string.

12.100.2.2 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

12.100.2.3 LoadXMLFromFile()

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

Loads an XML from a file.

12.100.2.4 LoadXMLFromFileInject()

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

Loads an XML from a file with injection.

12.100.2.5 LoadXMLFromString()

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

Loads an XML from a string.

12.100.2.6 LoadXMLFromStringInject()

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

Loads an XML from a string with injection.

12.100.2.7 LoadXMLFromZIPData()

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

Loads an XML from a ZIP data buffer.

12.100.2.8 LoadXMLFromZIPFile()

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

Loads an XML from a ZIP file.

12.100.2.9 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

InjectedFileName	Name of the target XML file to process
OutputFileName	Name of the Injected XML file to process> Name of the oputput file

12.100.2.10 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

StyleSheetFileName	The name of the XML file to process
OutputFileName	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
XMLValidation	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

12.100.2.11 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

StyleSheetFileName	The name of the XML file to process
OutputFileName	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
XMLValidation	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

12.100.3 Variable Documentation

12.100.3.1 INodeMapDyn

```
interface SPINNAKER_API_ABSTRACT INodeMapDyn
```

Interface to access the node map.

12.101 IntegerNode Class

Collaboration diagram for IntegerNode Class:

Classes

- class IntegerNode

  Interface for string properties.

Typedefs

- typedef IntegerNode CIntegerRef

12.101.1 Detailed Description

12.101.2 Typedef Documentation

12.101.2.1 CIntegerRef

```
typedef IntegerNode CIntegerRef
```

12.102 IntRegNode Class

Collaboration diagram for IntRegNode Class:

Classes

- class IntRegNode

  Interface for string properties.

12.102.1 Detailed Description

12.103 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

12.103.1 Detailed Description

12.103.2 Function Documentation

12.103.2.1 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.

12.103.3 Variable Documentation

12.103.3.1 Address

```
interface SPINNAKER_API_ABSTRACT int64_t Address
```

12.103.3.2 IPort

```
interface SPINNAKER_API_ABSTRACT IPort
```

Interface for ports.

12.103.3.3 Length

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

12.104 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.

12.104.1 Detailed Description

12.104.2 Function Documentation

12.104.2.1 GetSwapEndianess()

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

Determines if the port adapter must perform an endianess swap.

12.104.3 Variable Documentation

12.104.3.1 IPortConstruct

```
interface SPINNAKER_API IPortConstruct
```

Interface for ports.

12.105 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.

12.105.1 Detailed Description

12.105.2 Function Documentation

12.105.2.1 GetCookie()

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

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

12.105.2.2 Replay()

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

Replays the write command to the given port interface.

12.105.2.3 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.

12.105.2.4 StopRecording()

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

Stops recording.

12.105.3 Variable Documentation

12.105.3.1 Invalidate

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

12.105.3.2 IPortRecorder

```
interface SPINNAKER_API_ABSTRACT IPortRecorder
```

Interface for recording write commands on a port.

12.105.3.3 IPortReplay

```
interface SPINNAKER_API_ABSTRACT IPortReplay
```

Interface for replaying write commands on a port.

12.105.3.4 IPortWriteList

```
interface SPINNAKER_API_ABSTRACT IPortWriteList
```

Initial value:

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

12.106 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.

12.106.1 Detailed Description

12.106.2 Function Documentation

12.106.2.1 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

pBuffer	The buffer receiving the data to read
Length	The number of bytes to retrieve
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

12.106.2.2 GetAddress()

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

Retrieves the Address of the register.

12.106.2.3 GetLength()

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

Retrieves the Length of the register [Bytes].

12.106.3 Variable Documentation

12.106.3.1 IRegister

```
interface SPINNAKER_API_ABSTRACT IRegister
```

Interface for registers.

12.107 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.

12.107.1 Detailed Description

12.107.2 Function Documentation

12.107.2.1 GetSelectedFeatures()

```
virtual void Spinnaker::GenApi::GetSelectedFeatures (
            FeatureList_t &  ) const  [pure virtual]
```

retrieve the group of selected features

12.107.2.2 GetSelectingFeatures()

```
virtual void Spinnaker::GenApi::GetSelectingFeatures (
            FeatureList_t &  ) const  [pure virtual]
```

retrieve the group of features selecting this node

12.107.3 Variable Documentation

12.107.3.1 ISelector

```
interface SPINNAKER_API_ABSTRACT ISelector
```

Interface for groups of features selected by a single one.

12.108 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.

12.108.1 Detailed Description

12.108.2 Function Documentation

12.108.2.1 GetSelectorList()

```
virtual void Spinnaker::GenApi::GetSelectorList (
            FeatureList_t & SelectorList,
            bool Incremental = false )  [pure virtual]
```

Retrieves an ordered list of selectors.

Parameters

Incremental	List to contain the selector pointer> if true only seletor changed since the last GetNext are contained

12.108.2.2 Restore()

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

Restores the selectors' values found at creation.

12.108.2.3 SetNext()

```
virtual bool Spinnaker::GenApi::SetNext (
            bool Tick = true )  [pure virtual]
```

Sets digit to next value.

Parameters

Tick	if false the counter does not tick (but realize it could have)

Returns

true if the resulting value is valid

12.108.2.4 ToString()

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

Returns a string representation of the digit.

12.108.3 Variable Documentation

12.108.3.1 ISelectorDigit

```
interface SPINNAKER_API_ABSTRACT ISelectorDigit
```

Initial value:

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

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

12.109 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.

12.109.1 Detailed Description

12.109.2 Function Documentation

12.109.2.1 GetMaxLength()

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

Retrieves the maximum length of the string in bytes.

12.109.3 Variable Documentation

12.109.3.1 IString

```
interface SPINNAKER_API_ABSTRACT IString
```

Interface for string properties.

12.110 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.

12.110.1 Detailed Description

12.110.2 Function Documentation

12.110.2.1 FromString()

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

Set content of the node as string.

Parameters

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

12.110.2.2 IsValueCacheValid()

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

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

12.110.2.3 ToString()

```
virtual GenICam::gcstring Spinnaker::GenApi::ToString (
            bool Verify = false,
            bool IgnoreCache = false )  [pure virtual]
```

Get content of the node as string.

Parameters

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

12.110.3 Variable Documentation

12.110.3.1 IValue

```
interface SPINNAKER_API_ABSTRACT IValue
```

Interface for value properties.

12.111 Node Class

Collaboration diagram for Node Class:

Classes

- class Node

  class common to all nodes

12.111.1 Detailed Description

12.112 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.

12.112.1 Detailed Description

12.112.2 Enumeration Type Documentation

12.112.2.1 ECallbackType

```
enum ECallbackType
```

the type of callback

Enumerator

cbPostInsideLock	
cbPostOutsideLock	callback is fired on leaving the tree inside the lock-guarded area

12.112.3 Function Documentation

12.112.3.1 Deregister()

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

Unregistering callback by handle.

12.112.3.2 make_NodeCallback() [1/2]

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

make a new callback object for C functions

12.112.3.3 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

12.112.3.4 Register() [1/2]

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

Register a C-function as a callback.

12.112.3.5 Register() [2/2]

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

Register a C++-member function a callback.

12.113 NodeMap Class

Collaboration diagram for NodeMap Class:

Classes

- class NodeMap

  Smart pointer template for NodeMaps with create function.

12.113.1 Detailed Description

12.114 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.

12.114.1 Detailed Description

12.114.2 Enumeration Type Documentation

12.114.2.1 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.

12.114.2.2 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.

12.115 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.

12.115.1 Detailed Description

12.116 Persistence Class

Collaboration diagram for Persistence Class:

Classes

- class CFeatureBag

  Bag holding streamable features of a nodetree.

12.116.1 Detailed Description

12.117 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< 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.

12.117.1 Detailed Description

12.117.2 Typedef Documentation

12.117.2.1 CBasePtr

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

SmartPointer for IBase interface pointer.

12.117.2.2 CBooleanPtr

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

SmartPointer for IBoolean interface pointer.

12.117.2.3 CCategoryPtr

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

SmartPointer for ICategory interface pointer.

12.117.2.4 CChunkPortPtr

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

SmartPointer for IChunkPort interface pointer.

12.117.2.5 CCommandPtr

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

SmartPointer for ICommand interface pointer.

12.117.2.6 CDeviceInfoPtr

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

SmartPointer for IDeviceInfo interface pointer.

12.117.2.7 CEnumEntryPtr

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

SmartPointer for IEnumEntry interface pointer.

12.117.2.8 CEnumerationPtr

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

SmartPointer for IEnumeration interface pointer.

12.117.2.9 CIntegerPtr

```
typedef CPointer<IInteger> CIntegerPtr
```

SmartPointer for IInteger interface pointer.

12.117.2.10 CNodeMapDynPtr

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

SmartPointer for INodeMapDyn interface pointer.

12.117.2.11 CNodeMapPtr

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

SmartPointer for INodeMap interface pointer.

12.117.2.12 CNodePtr

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

SmartPointer for INode interface pointer.

12.117.2.13 CPortConstructPtr

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

SmartPointer for IPortConstruct interface pointer.

12.117.2.14 CPortPtr

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

SmartPointer for IPort interface pointer.

12.117.2.15 CPortRecorderPtr

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

SmartPointer for IPortRecorder interface pointer.

12.117.2.16 CPortReplayPtr

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

SmartPointer for IPortReplay interface pointer.

12.117.2.17 CPortWriteListPtr

```
typedef CPointer<IPortWriteList, IPortWriteList> CPortWriteListPtr
```

SmartPointer for IPortWriteList interface pointer.

12.117.2.18 CRegisterPtr

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

SmartPointer for IRegister interface pointer.

12.117.2.19 CSelectorPtr

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

SmartPointer for ISelector interface pointer.

12.117.2.20 CStringPtr

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

SmartPointer for IString interface pointer.

12.117.2.21 CValuePtr

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

SmartPointer for IValue interface pointer.

12.117.3 Function Documentation

12.117.3.1 GetInterfaceName()

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

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

12.117.3.2 IsAvailable()

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

Checks if a node is Available.

12.117.3.3 IsImplemented()

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

Checks if a node is Implemented.

12.117.3.4 IsReadable()

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

Checks if a node is readable.

12.117.3.5 IsWritable()

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

Checks if a node is Writable.

12.118 PortImpl Class

Collaboration diagram for PortImpl Class:

Classes

- class CPortImpl

  Standard implementation for a port.

12.118.1 Detailed Description

12.119 PortNode Class

Collaboration diagram for PortNode Class:

Classes

- class PortNode

  Interface for value properties.

Typedefs

- typedef PortNode CPortRef

12.119.1 Detailed Description

12.119.2 Typedef Documentation

12.119.2.1 CPortRef

```
typedef PortNode CPortRef
```

12.120 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.

12.120.1 Detailed Description

12.120.2 Typedef Documentation

12.120.2.1 CPortRecorderRef

```
typedef PortRecorder CPortRecorderRef
```

Reference to an IPortRecorder pointer.

12.121 PortReplay Class

Collaboration diagram for PortReplay Class:

Classes

- class PortReplay

  Interface for replaying write commands on a port.

12.121.1 Detailed Description

12.122 PortWriteList Class

Collaboration diagram for PortWriteList Class:

Classes

- class CPortWriteList

  Container holding a list of port write commands.

12.122.1 Detailed Description

12.123 Reference Interfaces

Collaboration diagram for Reference Interfaces:

Functions

- virtual void SetNumEnums (int NumEnums)=0

  sets the number of enum values

12.123.1 Detailed Description

12.123.2 Function Documentation

12.123.2.1 SetNumEnums()

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

sets the number of enum values

12.124 RegisterNode Class

Collaboration diagram for RegisterNode Class:

Classes

- class RegisterNode

  Interface for string properties.

Typedefs

- typedef RegisterNode CRegisterRef

12.124.1 Detailed Description

12.124.2 Typedef Documentation

12.124.2.1 CRegisterRef

```
typedef RegisterNode CRegisterRef
```

12.125 RegisterPortImpl Class

Collaboration diagram for RegisterPortImpl Class:

Classes

- class CRegisterPortImpl

  Standard implementation for a port using a register based transport layer.

12.125.1 Detailed Description

12.126 SelectorSet Class

Collaboration diagram for SelectorSet Class:

Classes

- class CSelectorSet

  The set of selectors selecting a given node.

12.126.1 Detailed Description

12.127 SpinTestCamera Class

Collaboration diagram for SpinTestCamera Class:

Classes

- class SpinTestCamera

12.127.1 Detailed Description

12.128 StringNode Class

Collaboration diagram for StringNode Class:

Classes

- class StringNode

  Interface for string properties.

Typedefs

- typedef StringNode CStringRef

12.128.1 Detailed Description

12.128.2 Typedef Documentation

12.128.2.1 CStringRef

```
typedef StringNode CStringRef
```

12.129 StringRegNode Class

Collaboration diagram for StringRegNode Class:

Classes

- class StringRegNode

  Interface for string properties.

12.129.1 Detailed Description

12.130 StructPort Class

Collaboration diagram for StructPort Class:

Classes

- class CTestPortStruct< CDataStruct >

  Implements a register spaces based on a C++ struct.

12.130.1 Detailed Description

12.131 Synch Class

Collaboration diagram for Synch Class:

Classes

- class CLock

  A lock class.
- class CLockEx

  This class is for testing purposes only.
- class AutoLock

12.131.1 Detailed Description

12.132 Spinnaker GenApi Enums

Collaboration diagram for Spinnaker GenApi Enums:

Modules

- Types Enums

12.132.1 Detailed Description

12.133 Types Enums

Collaboration diagram for Types Enums:

Macros

- #define _UndefindedRepresentation _UndefinedRepresentation

Typedefs

- typedef GenICam::gcstring_vector StringList_t

  A list of strings.

Enumerations

- enum ESign {
  Signed,
  Unsigned,
  _UndefinedSign }

  signed or unsigned integers
- enum EAccessMode {
  NI,
  NA,
  WO,
  RO,
  RW,
  _UndefinedAccesMode,
  _CycleDetectAccesMode }

  access mode of a node
- enum EVisibility {
  Beginner = 0,
  Expert = 1,
  Guru = 2,
  Invisible = 3,
  _UndefinedVisibility = 99 }

  recommended visibility of a node
- enum ECachingMode {
  NoCache,
  WriteThrough,
  WriteAround,
  _UndefinedCachingMode }

  caching mode of a register

- enum ERepresentation {
  Linear,
  Logarithmic,
  Boolean,
  PureNumber,
  HexNumber,
  IPV4Address,
  MACAddress,
  _UndefinedRepresentation }

  recommended representation of a node value
- enum EEndianess {
  BigEndian,
  LittleEndian,
  _UndefinedEndian }

  Endianess of a value in a register.
- enum ENameSpace {
  Custom,
  Standard,
  _UndefinedNameSpace }

  Defines if a node name is standard or custom.
- enum EStandardNameSpace {
  None,
  GEV,
  IIDC,
  CL,
  USB,
  _UndefinedStandardNameSpace }

  Defines from which standard namespace a node name comes from.
- enum EYesNo {
  Yes = 1,
  No = 0,
  _UndefinedYesNo = 2 }

  Defines the choices of a Yes/No alternative.
- enum ESlope {
  Increasing,
  Decreasing,
  Varying,
  Automatic,
  _UndefinedESlope }

  typedef for formula type
- enum EXMLValidation {
  xvLoad = 0x00000001L,
  xvCycles = 0x00000002L,
  xvSFNC = 0x00000004L,
  xvDefault = 0x00000000L,
  xvAll = 0xffffffffL,
  _UndefinedEXMLValidation = 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,
  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.

12.133.1 Detailed Description

12.133.2 Macro Definition Documentation

12.133.2.1 _UndefindedRepresentation

```
#define _UndefindedRepresentation _UndefinedRepresentation
```

12.133.3 Typedef Documentation

12.133.3.1 StringList_t

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

A list of strings.

12.133.4 Enumeration Type Documentation

12.133.4.1 EAccessMode

```
enum EAccessMode
```

access mode of a node

Enumerator

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

12.133.4.2 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

12.133.4.3 EDisplayNotation

```
enum EDisplayNotation
```

typedef for float notation

Enumerator

fnAutomatic	
fnFixed	the notation if either scientific or fixed depending on what is shorter
fnScientific	the notation is fixed, e.g. 123.4
_UndefinedEDisplayNotation	the notation is scientific, e.g. 1.234e2 Object is not yet initialized

12.133.4.4 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

12.133.4.5 EGenApiSchemaVersion

`enum EGenApiSchemaVersion`

GenApi schema version.

Enumerator

v1_0	
v1_1	
_Undefined	

12.133.4.6 EIncMode

`enum EIncMode`

typedef for increment mode

Enumerator

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

12.133.4.7 EInputDirection

```
enum EInputDirection
```

typedef for link type

Enumerator

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

12.133.4.8 EInterfaceType

```
enum EInterfaceType
```

typedef for interface type

Enumerator

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

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

12.133.4.10 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

12.133.4.11 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.

12.133.4.12 ESign

```
enum ESign
```

signed or unsigned integers

Enumerator

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

12.133.4.13 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

12.133.4.14 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

12.133.4.15 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.

12.133.4.16 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

12.133.4.17 EYesNo

`enum EYesNo`

Defines the choices of a Yes/No alternative.

Enumerator

Yes	
No	yes
_UndefinedYesNo	no

12.134 ValueNode Class

Collaboration diagram for ValueNode Class:

Classes

- class ValueNode

  Interface for value properties.

Typedefs

- typedef ValueNode CValueRef

12.134.1 Detailed Description

12.134.2 Typedef Documentation

12.134.2.1 CValueRef

```
typedef ValueNode CValueRef
```

12.135 ChunkAdapterU3V Class

Collaboration diagram for ChunkAdapterU3V Class:

Classes

- class CChunkAdapterU3V

  Connects a chunked U3V buffer to a node map.

12.135.1 Detailed Description

Chapter 13

Namespace Documentation

13.1 AdapterConfig Namespace Reference

Classes

- struct AdapterInfo
- struct IpInfo

Enumerations

- enum AdapterConfigErr {
  IP_ADDRESS_INVALID,
  IP_ADDRESS_IS_NOT_V4,
  IP_ADDRESS_TOO_LARGE,
  IP_ADDRESS_TOO_SMALL,
  HOST_ADDRESS_ZERO,
  SUBNET_MASK_INVALID,
  VALID_SUBNET_NOT_FOUND }

Functions

- ADAPTERCONFIG_API std::vector< AdapterInfo > RetrieveAllAdapters ()
- ADAPTERCONFIG_API void AutoPopulateAdapterInfo (std::vector< AdapterInfo > &adaptersToConfigure, const std::vector< AdapterInfo > &allAdapters)
- ADAPTERCONFIG_API void AutoPopulateAdvancedProperties (std::vector< AdapterInfo > &adaptersTo↩Configure)
- 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 GetEnumerationLogFileName ()
- ADAPTERCONFIG_API std::string GetConfigLogFileName ()

- ADAPTERCONFIG_API void ConfigureAdapter (AdapterInfo &adapter, bool configureIP, bool configure↩ AdvancedProperties)
- 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 ()

13.1.1 Enumeration Type Documentation

13.1.1.1 AdapterConfigErr

```
enum AdapterConfigErr
```

Enumerator

IP_ADDRESS_INVALID	
IP_ADDRESS_IS_NOT_V4	
IP_ADDRESS_TOO_LARGE	
IP_ADDRESS_TOO_SMALL	
HOST_ADDRESS_ZERO	
SUBNET_MASK_INVALID	
VALID_SUBNET_NOT_FOUND	

13.1.2 Function Documentation

13.1.2.1 AutoPopulateAdapterInfo()

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

13.1.2.2 AutoPopulateAdvancedProperties()

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

13.1.2.3 ConfigureAdapter()

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

13.1.2.4 GetAuto10GDesc()

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

13.1.2.5 GetAutoGigabitDesc()

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

13.1.2.6 GetAutoStartIp()

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

13.1.2.7 GetAutoSubnetMask()

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

13.1.2.8 GetAutoSubnetMaskLength()

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

13.1.2.9 GetConfigLogFileName()

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

13.1.2.10 GetEnumerationLogFileName()

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

13.1.2.11 GetMaxIpAddress()

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

13.1.2.12 GetMinIpAddress()

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

13.1.2.13 GetSubnetMaskLength()

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

13.1.2.14 IsOnSameSubnet()

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

13.1.2.15 IsValidIpAddress()

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

13.1.2.16 IsValidSubnetMask()

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

13.1.2.17 PopulateAdapterIpInfo()

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

13.1.2.18 RetrieveAllAdapters()

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

13.1.2.19 ValidateIpAddress()

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

13.2 Conversion Namespace Reference

Functions

- string NumToCString (int number)
- string NumToCString (double number)
- string NumToCString (float number)

13.2.1 Function Documentation

13.2.1.1 NumToCString() [1/3]

```
string NumToCString (
            int number )
```

13.2.1.2 NumToCString() [2/3]

```
string NumToCString (
            double number )
```

13.2.1.3 NumToCString() [3/3]

```
string Conversion::NumToCString (
            float number )
```

13.3 CpuUtil Namespace Reference

Classes

- struct CpuUsageInfo

Functions

- bool StartCpuTracing (CpuUsageInfo *cpuUsage)
- bool StopCpuTracing (CpuUsageInfo *cpuUsage)
- std::string GetCpuStats (CpuUsageInfo *cpuUsage)

13.3.1 Function Documentation

13.3.1.1 GetCpuStats()

```
std::string GetCpuStats (
            CpuUsageInfo * cpuUsage )
```

13.3.1.2 StartCpuTracing()

```
bool StartCpuTracing (
            CpuUsageInfo * cpuUsage )
```

13.3.1.3 StopCpuTracing()

```
bool StopCpuTracing (
            CpuUsageInfo * cpuUsage )
```

13.4 PerformanceCounter Namespace Reference

Functions

- void StartPerformanceCounter ()
- double GetPerformanceCounter ()

Variables

- double PCFreq
- __int64 CounterStart

13.4.1 Function Documentation

13.4.1.1 GetPerformanceCounter()

```
double GetPerformanceCounter ( )
```

13.4.1.2 StartPerformanceCounter()

```
void StartPerformanceCounter ( )
```

13.4.2 Variable Documentation

13.4.2.1 CounterStart

```
__int64 CounterStart
```

13.4.2.2 PCFreq

```
double PCFreq
```

13.5 SecondsCounter Namespace Reference

Functions

- void StartSecondsCounter ()
- int GetSecondsCounter ()

Variables

- time_t startTime
- time_t endTime
- double timeDiff

13.5.1 Function Documentation

13.5.1.1 GetSecondsCounter()

```
int GetSecondsCounter ( )
```

13.5.1.2 StartSecondsCounter()

```
void StartSecondsCounter ( )
```

13.5.2 Variable Documentation

13.5.2.1 endTime

```
time_t endTime
```

13.5.2.2 startTime

```
time_t startTime
```

13.5.2.3 timeDiff

```
double timeDiff
```

13.6 Spinnaker Namespace Reference

Namespaces

- GenApi
- GenICam
- Video

Classes

- struct ActionCommandResult

  Action Command Result.
- class BasePtr

  The base class of the SystemPtr, CameraPtr, InterfacePtr, ImagePtr and LoggingEventDataPtr objects.
- struct BMPOption

  Options for saving Bitmap image.
- class Camera

  The camera object class.
- class CameraBase

  The base class for the camera object.
- class CameraList

  Used to hold a list of camera objects.
- class CameraPtr

  A reference tracked pointer to a camera object.
- class ChunkData

  The chunk data which contains additional information about an image.
- class DeviceArrivalEventHandler

  An event handler for capturing the device arrival event.
- class DeviceEventHandler

  A handler to device events.
- class DeviceRemovalEventHandler

  An event handler for capturing the device removal event.
- class EventHandler

  The base class for all event handler types.
- class Exception

  The Exception object represents an error that is returned from the library.
- class ICameraBase

  The interface file for base class for the camera object.
- class ICameraList

  Used to hold a list of camera objects.
- class IChunkData

  The Interface file for ChunkData.
- class IDataStream
- class IDeviceArrivalEventHandler
- class IDeviceEventHandler
- class IDeviceRemovalEventHandler
- class IImage

  The interface file for Image.
- class IImageEventHandler
- class IImageStatistics

The interface file for image statistics.

- class IInterface

  The interface file for Interface.
- class IInterfaceArrivalEventHandler
- class IInterfaceEventHandler
- class IInterfaceList

  The interface file for InterfaceList class.
- class IInterfaceRemovalEventHandler
- class ILoggingEventHandler
- class Image

  The image object class.
- class ImageEventHandler

  A handler for capturing image arrival events.
- class ImagePtr

  A reference tracked pointer to an image object.
- class ImageStatistics

  Represents image statistics for an image.
- class ImageUtility

  Static helper functions for the image object class.
- class ImageUtilityHeatmap

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

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

  An inference bounding boxes object which holds information about the detected bounding boxes.
- struct InferenceBoxCircle
- struct InferenceBoxRect

  Inference Bounding Box Type Data Structures.
- struct InferenceBoxRotatedRect
- class Interface

  An interface object which holds a list of cameras.
- class InterfaceArrivalEventHandler

  An event handler for capturing the interface arrival event.
- class InterfaceEventHandler

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

  A list of the available interfaces on the system.
- class InterfacePtr

  A reference tracked pointer to the interface object.
- class InterfaceRemovalEventHandler

  An event handler for capturing the interface removal event.
- class ISystem

  The interface file for System.
- class ISystemEventHandler
- struct JPEGOption

  Options for saving JPEG image.
- struct JPG2Option

  Options for saving JPEG2000 image.
- struct LibraryVersion

  Provides easier access to the current version of Spinnaker.
- class LoggingEventData

The LoggingEventData object.

- class LoggingEventDataPtr

  A reference tracked pointer to the LoggingEvent object.
- class LoggingEventHandler

  An event handler for capturing the device logging event.
- struct PGMOption

  Options for saving PGM images.
- struct PNGOption

  Options for saving PNG images.
- struct PPMOption

  Options for saving PPM images.
- class System

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

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

  A reference tracked pointer to a system object.
- struct TIFFOption

  Options for saving TIFF images.
- class TransportLayerDevice

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

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

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

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

Enumerations

- enum LUTSelectorEnums {
  LUTSelector_LUT1,
  NUM_LUTSELECTOR }

  The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.
- enum ExposureModeEnums {
  ExposureMode_Timed,
  ExposureMode_TriggerWidth,
  NUM_EXPOSUREMODE }
- enum AcquisitionModeEnums {
  AcquisitionMode_Continuous,
  AcquisitionMode_SingleFrame,
  AcquisitionMode_MultiFrame,
  NUM_ACQUISITIONMODE }
- enum TriggerSourceEnums {
  TriggerSource_Software,
  TriggerSource_Line0,
  TriggerSource_Line1,
  TriggerSource_Line2,
  TriggerSource_Line3,
  TriggerSource_UserOutput0,
  TriggerSource_UserOutput1,
  TriggerSource_UserOutput2,

TriggerSource_UserOutput3,
TriggerSource_Counter0Start,
TriggerSource_Counter1Start,
TriggerSource_Counter0End,
TriggerSource_Counter1End,
TriggerSource_LogicBlock0,
TriggerSource_LogicBlock1,
TriggerSource_Action0,
NUM_TRIGGERSOURCE }

- enum TriggerActivationEnums {
  TriggerActivation_LevelLow,
  TriggerActivation_LevelHigh,
  TriggerActivation_FallingEdge,
  TriggerActivation_RisingEdge,
  TriggerActivation_AnyEdge,
  NUM_TRIGGERACTIVATION }
- enum SensorShutterModeEnums {
  SensorShutterMode_Global,
  SensorShutterMode_Rolling,
  SensorShutterMode_GlobalReset,
  NUM_SENSORSHUTTERMODE }
- enum TriggerModeEnums {
  TriggerMode_Off,
  TriggerMode_On,
  NUM_TRIGGERMODE }
- enum TriggerOverlapEnums {
  TriggerOverlap_Off,
  TriggerOverlap_ReadOut,
  TriggerOverlap_PreviousFrame,
  NUM_TRIGGEROVERLAP }
- enum TriggerSelectorEnums {
  TriggerSelector_AcquisitionStart,
  TriggerSelector_FrameStart,
  TriggerSelector_FrameBurstStart,
  NUM_TRIGGERSELECTOR }
- enum ExposureAutoEnums {
  ExposureAuto_Off,
  ExposureAuto_Once,
  ExposureAuto_Continuous,
  NUM_EXPOSUREAUTO }
- enum EventSelectorEnums {
  EventSelector_Error,
  EventSelector_ExposureEnd,
  EventSelector_SerialPortReceive,
  NUM_EVENTSELECTOR }
- enum EventNotificationEnums {
  EventNotification_On,
  EventNotification_Off,
  NUM_EVENTNOTIFICATION }
- enum LogicBlockSelectorEnums {
  LogicBlockSelector_LogicBlock0,
  LogicBlockSelector_LogicBlock1,
  NUM_LOGICBLOCKSELECTOR }
- enum LogicBlockLUTInputActivationEnums {
  LogicBlockLUTInputActivation_LevelLow,
  LogicBlockLUTInputActivation_LevelHigh,
  LogicBlockLUTInputActivation_FallingEdge,
  LogicBlockLUTInputActivation_RisingEdge,

LogicBlockLUTInputActivation_AnyEdge,
NUM_LOGICBLOCKLUTINPUTACTIVATION }

- enum LogicBlockLUTInputSelectorEnums {
LogicBlockLUTInputSelector_Input0,
LogicBlockLUTInputSelector_Input1,
LogicBlockLUTInputSelector_Input2,
LogicBlockLUTInputSelector_Input3,
NUM_LOGICBLOCKLUTINPUTSELECTOR }
- enum LogicBlockLUTInputSourceEnums {
LogicBlockLUTInputSource_Zero,
LogicBlockLUTInputSource_Line0,
LogicBlockLUTInputSource_Line1,
LogicBlockLUTInputSource_Line2,
LogicBlockLUTInputSource_Line3,
LogicBlockLUTInputSource_UserOutput0,
LogicBlockLUTInputSource_UserOutput1,
LogicBlockLUTInputSource_UserOutput2,
LogicBlockLUTInputSource_UserOutput3,
LogicBlockLUTInputSource_Counter0Start,
LogicBlockLUTInputSource_Counter1Start,
LogicBlockLUTInputSource_Counter0End,
LogicBlockLUTInputSource_Counter1End,
LogicBlockLUTInputSource_LogicBlock0,
LogicBlockLUTInputSource_LogicBlock1,
LogicBlockLUTInputSource_ExposureStart,
LogicBlockLUTInputSource_ExposureEnd,
LogicBlockLUTInputSource_FrameTriggerWait,
LogicBlockLUTInputSource_AcquisitionActive,
NUM_LOGICBLOCKLUTINPUTSOURCE }
- enum LogicBlockLUTSelectorEnums {
LogicBlockLUTSelector_Value,
LogicBlockLUTSelector_Enable,
NUM_LOGICBLOCKLUTSELECTOR }
- enum ColorTransformationSelectorEnums {
ColorTransformationSelector_RGBtoRGB,
ColorTransformationSelector_RGBtoYUV,
NUM_COLORTRANSFORMATIONSELECTOR }
- enum RgbTransformLightSourceEnums {
RgbTransformLightSource_General,
RgbTransformLightSource_Tungsten2800K,
RgbTransformLightSource_WarmFluorescent3000K,
RgbTransformLightSource_CoolFluorescent4000K,
RgbTransformLightSource_Daylight5000K,
RgbTransformLightSource_Cloudy6500K,
RgbTransformLightSource_Shade8000K,
RgbTransformLightSource_Custom,
NUM_RGBTRANSFORMLIGHTSOURCE }
- enum ColorTransformationValueSelectorEnums {
ColorTransformationValueSelector_Gain00,
ColorTransformationValueSelector_Gain01,
ColorTransformationValueSelector_Gain02,
ColorTransformationValueSelector_Gain10,
ColorTransformationValueSelector_Gain11,
ColorTransformationValueSelector_Gain12,
ColorTransformationValueSelector_Gain20,
ColorTransformationValueSelector_Gain21,
ColorTransformationValueSelector_Gain22,
ColorTransformationValueSelector_Offset0,

ColorTransformationValueSelector_Offset1,
ColorTransformationValueSelector_Offset2,
NUM_COLORTRANSFORMATIONVALUESELECTOR }

- enum DeviceRegistersEndiannessEnums {
DeviceRegistersEndianness_Little,
DeviceRegistersEndianness_Big,
NUM_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_DEVICETLTYPE }
- enum DevicePowerSupplySelectorEnums {
DevicePowerSupplySelector_External,
NUM_DEVICEPOWERSUPPLYSELECTOR }
- enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICETEMPERATURESELECTOR }
- enum DeviceIndicatorModeEnums {
DeviceIndicatorMode_Inactive,
DeviceIndicatorMode_Active,
DeviceIndicatorMode_ErrorStatus,
NUM_DEVICEINDICATORMODE }
- enum AutoExposureControlPriorityEnums {
AutoExposureControlPriority_Gain,
AutoExposureControlPriority_ExposureTime,
NUM_AUTOEXPOSURECONTROLPRIORITY }
- enum AutoExposureMeteringModeEnums {
AutoExposureMeteringMode_Average,
AutoExposureMeteringMode_Spot,
AutoExposureMeteringMode_Partial,
AutoExposureMeteringMode_CenterWeighted,
AutoExposureMeteringMode_HistgramPeak,
NUM_AUTOEXPOSUREMETERINGMODE }
- enum BalanceWhiteAutoProfileEnums {
BalanceWhiteAutoProfile_Indoor,
BalanceWhiteAutoProfile_Outdoor,
NUM_BALANCEWHITEAUTOPROFILE }
- enum AutoAlgorithmSelectorEnums {
AutoAlgorithmSelector_Awb,
AutoAlgorithmSelector_Ae,
NUM_AUTOALGORITHMSELECTOR }
- enum AutoExposureTargetGreyValueAutoEnums {
AutoExposureTargetGreyValueAuto_Off,
AutoExposureTargetGreyValueAuto_Continuous,
NUM_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum AutoExposureLightingModeEnums {
AutoExposureLightingMode_AutoDetect,

AutoExposureLightingMode_Backlight,
AutoExposureLightingMode_Frontlight,
AutoExposureLightingMode_Normal,
NUM_AUTOEXPOSURELIGHTINGMODE }

- enum GevIEEE1588StatusEnums {
GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,
GevIEEE1588Status_Listening,
GevIEEE1588Status_PreMaster,
GevIEEE1588Status_Master,
GevIEEE1588Status_Passive,
GevIEEE1588Status_Uncalibrated,
GevIEEE1588Status_Slave,
NUM_GEVIEEE1588STATUS }
- enum GevIEEE1588ModeEnums {
GevIEEE1588Mode_Auto,
GevIEEE1588Mode_SlaveOnly,
NUM_GEVIEEE1588MODE }
- enum GevIEEE1588ClockAccuracyEnums {
GevIEEE1588ClockAccuracy_Unknown,
NUM_GEVIEEE1588CLOCKACCURACY }
- enum GevCCPEnums {
GevCCP_OpenAccess,
GevCCP_ExclusiveAccess,
GevCCP_ControlAccess,
NUM_GEVCCP }
- enum GevSupportedOptionSelectorEnums {
GevSupportedOptionSelector_UserDefinedName,
GevSupportedOptionSelector_SerialNumber,
GevSupportedOptionSelector_HeartbeatDisable,
GevSupportedOptionSelector_LinkSpeed,
GevSupportedOptionSelector_CCPApplicationSocket,
GevSupportedOptionSelector_ManifestTable,
GevSupportedOptionSelector_TestData,
GevSupportedOptionSelector_DiscoveryAckDelay,
GevSupportedOptionSelector_DiscoveryAckDelayWritable,
GevSupportedOptionSelector_ExtendedStatusCodes,
GevSupportedOptionSelector_Action,
GevSupportedOptionSelector_PendingAck,
GevSupportedOptionSelector_EventData,
GevSupportedOptionSelector_Event,
GevSupportedOptionSelector_PacketResend,
GevSupportedOptionSelector_WriteMem,
GevSupportedOptionSelector_CommandsConcatenation,
GevSupportedOptionSelector_IPConfigurationLLA,
GevSupportedOptionSelector_IPConfigurationDHCP,
GevSupportedOptionSelector_IPConfigurationPersistentIP,
GevSupportedOptionSelector_StreamChannelSourceSocket,
GevSupportedOptionSelector_MessageChannelSourceSocket,
NUM_GEVSUPPORTEDOPTIONSELECTOR }
- enum BlackLevelSelectorEnums {
BlackLevelSelector_All,
BlackLevelSelector_Analog,
BlackLevelSelector_Digital,
NUM_BLACKLEVELSELECTOR }
- enum BalanceWhiteAutoEnums {
BalanceWhiteAuto_Off,

BalanceWhiteAuto_Once,
BalanceWhiteAuto_Continuous,
NUM_BALANCEWHITEAUTO }

- enum GainAutoEnums {
GainAuto_Off,
GainAuto_Once,
GainAuto_Continuous,
NUM_GAINAUTO }
- enum BalanceRatioSelectorEnums {
BalanceRatioSelector_Red,
BalanceRatioSelector_Blue,
NUM_BALANCERATIOSELECTOR }
- enum GainSelectorEnums {
GainSelector_All,
NUM_GAINSELECTOR }
- enum DefectCorrectionModeEnums {
DefectCorrectionMode_Average,
DefectCorrectionMode_Highlight,
DefectCorrectionMode_Zero,
NUM_DEFECTCORRECTIONMODE }
- enum UserSetSelectorEnums {
UserSetSelector_Default,
UserSetSelector_UserSet0,
UserSetSelector_UserSet1,
NUM_USERSETSELECTOR }
- enum UserSetDefaultEnums {
UserSetDefault_Default,
UserSetDefault_UserSet0,
UserSetDefault_UserSet1,
NUM_USERSETDEFAULT }
- enum SerialPortBaudRateEnums {
SerialPortBaudRate_Baud300,
SerialPortBaudRate_Baud600,
SerialPortBaudRate_Baud1200,
SerialPortBaudRate_Baud2400,
SerialPortBaudRate_Baud4800,
SerialPortBaudRate_Baud9600,
SerialPortBaudRate_Baud14400,
SerialPortBaudRate_Baud19200,
SerialPortBaudRate_Baud38400,
SerialPortBaudRate_Baud57600,
SerialPortBaudRate_Baud115200,
SerialPortBaudRate_Baud230400,
SerialPortBaudRate_Baud460800,
SerialPortBaudRate_Baud921600,
NUM_SERIALPORTBAUDRATE }
- enum SerialPortParityEnums {
SerialPortParity_None,
SerialPortParity_Odd,
SerialPortParity_Even,
SerialPortParity_Mark,
SerialPortParity_Space,
NUM_SERIALPORTPARITY }
- enum SerialPortSelectorEnums {
SerialPortSelector_SerialPort0,
NUM_SERIALPORTSELECTOR }
- enum SerialPortStopBitsEnums {
SerialPortStopBits_Bits1,

SerialPortStopBits_Bits1AndAHalf,
SerialPortStopBits_Bits2,
NUM_SERIALPORTSTOPBITS }

- enum SerialPortSourceEnums {
SerialPortSource_Line0,
SerialPortSource_Line1,
SerialPortSource_Line2,
SerialPortSource_Line3,
SerialPortSource_Off,
NUM_SERIALPORTSOURCE }
- enum SequencerModeEnums {
SequencerMode_Off,
SequencerMode_On,
NUM_SEQUENCERMODE }
- enum SequencerConfigurationValidEnums {
SequencerConfigurationValid_No,
SequencerConfigurationValid_Yes,
NUM_SEQUENCERCONFIGURATIONVALID }
- enum SequencerSetValidEnums {
SequencerSetValid_No,
SequencerSetValid_Yes,
NUM_SEQUENCERSETVALID }
- enum SequencerTriggerActivationEnums {
SequencerTriggerActivation_RisingEdge,
SequencerTriggerActivation_FallingEdge,
SequencerTriggerActivation_AnyEdge,
SequencerTriggerActivation_LevelHigh,
SequencerTriggerActivation_LevelLow,
NUM_SEQUENCERTRIGGERACTIVATION }
- enum SequencerConfigurationModeEnums {
SequencerConfigurationMode_Off,
SequencerConfigurationMode_On,
NUM_SEQUENCERCONFIGURATIONMODE }
- enum SequencerTriggerSourceEnums {
SequencerTriggerSource_Off,
SequencerTriggerSource_FrameStart,
NUM_SEQUENCERTRIGGERSOURCE }
- enum TransferQueueModeEnums {
TransferQueueMode_FirstInFirstOut,
NUM_TRANSFERQUEUEMODE }
- enum TransferOperationModeEnums {
TransferOperationMode_Continuous,
TransferOperationMode_MultiBlock,
NUM_TRANSFEROPERATIONMODE }
- enum TransferControlModeEnums {
TransferControlMode_Basic,
TransferControlMode_Automatic,
TransferControlMode_UserControlled,
NUM_TRANSFERCONTROLMODE }
- enum ChunkGainSelectorEnums {
ChunkGainSelector_All,
ChunkGainSelector_Red,
ChunkGainSelector_Green,
ChunkGainSelector_Blue,
NUM_CHUNKGAINSELECTOR }
- enum ChunkSelectorEnums {
ChunkSelector_Image,
ChunkSelector_CRC,

ChunkSelector_FrameID,
ChunkSelector_OffsetX,
ChunkSelector_OffsetY,
ChunkSelector_Width,
ChunkSelector_Height,
ChunkSelector_ExposureTime,
ChunkSelector_Gain,
ChunkSelector_BlackLevel,
ChunkSelector_PixelFormat,
ChunkSelector_Timestamp,
ChunkSelector_SequencerSetActive,
ChunkSelector_SerialData,
ChunkSelector_ExposureEndLineStatusAll,
NUM_CHUNKSELECTOR }

- enum ChunkBlackLevelSelectorEnums {
  ChunkBlackLevelSelector_All,
  NUM_CHUNKBLACKLEVELSELECTOR }
- enum ChunkPixelFormatEnums {
  ChunkPixelFormat_Mono8,
  ChunkPixelFormat_Mono12Packed,
  ChunkPixelFormat_Mono16,
  ChunkPixelFormat_RGB8Packed,
  ChunkPixelFormat_YUV422Packed,
  ChunkPixelFormat_BayerGR8,
  ChunkPixelFormat_BayerRG8,
  ChunkPixelFormat_BayerGB8,
  ChunkPixelFormat_BayerBG8,
  ChunkPixelFormat_YCbCr601_422_8_CbYCrY,
  NUM_CHUNKPIXELFORMAT }
- enum FileOperationStatusEnums {
  FileOperationStatus_Success,
  FileOperationStatus_Failure,
  FileOperationStatus_Overflow,
  NUM_FILEOPERATIONSTATUS }
- enum FileOpenModeEnums {
  FileOpenMode_Read,
  FileOpenMode_Write,
  FileOpenMode_ReadWrite,
  NUM_FILEOPENMODE }
- enum FileOperationSelectorEnums {
  FileOperationSelector_Open,
  FileOperationSelector_Close,
  FileOperationSelector_Read,
  FileOperationSelector_Write,
  FileOperationSelector_Delete,
  NUM_FILEOPERATIONSELECTOR }
- enum FileSelectorEnums {
  FileSelector_UserSetDefault,
  FileSelector_UserSet0,
  FileSelector_UserSet1,
  FileSelector_UserFile1,
  FileSelector_SerialPort0,
  NUM_FILESELECTOR }
- enum BinningSelectorEnums {
  BinningSelector_All,
  BinningSelector_Sensor,
  BinningSelector_ISP,
  NUM_BINNINGSELECTOR }

- enum TestPatternGeneratorSelectorEnums {
  TestPatternGeneratorSelector_Sensor,
  TestPatternGeneratorSelector_PipelineStart,
  NUM_TESTPATTERNGENERATORSELECTOR }
- enum TestPatternEnums {
  TestPattern_Off,
  TestPattern_Increment,
  TestPattern_SensorTestPattern,
  NUM_TESTPATTERN }
- enum PixelColorFilterEnums {
  PixelColorFilter_None,
  PixelColorFilter_BayerRG,
  PixelColorFilter_BayerGB,
  PixelColorFilter_BayerGR,
  PixelColorFilter_BayerBG,
  NUM_PIXELCOLORFILTER }
- enum AdcBitDepthEnums {
  AdcBitDepth_Bit8,
  AdcBitDepth_Bit10,
  AdcBitDepth_Bit12,
  AdcBitDepth_Bit14,
  NUM_ADCBITDEPTH }
- enum DecimationHorizontalModeEnums {
  DecimationHorizontalMode_Discard,
  NUM_DECIMATIONHORIZONTALMODE }
- enum BinningVerticalModeEnums {
  BinningVerticalMode_Sum,
  BinningVerticalMode_Average,
  NUM_BINNINGVERTICALMODE }
- enum PixelSizeEnums {
  PixelSize_Bpp1,
  PixelSize_Bpp2,
  PixelSize_Bpp4,
  PixelSize_Bpp8,
  PixelSize_Bpp10,
  PixelSize_Bpp12,
  PixelSize_Bpp14,
  PixelSize_Bpp16,
  PixelSize_Bpp20,
  PixelSize_Bpp24,
  PixelSize_Bpp30,
  PixelSize_Bpp32,
  PixelSize_Bpp36,
  PixelSize_Bpp48,
  PixelSize_Bpp64,
  PixelSize_Bpp96,
  NUM_PIXELSIZE }
- enum DecimationSelectorEnums {
  DecimationSelector_All,
  DecimationSelector_Sensor,
  NUM_DECIMATIONSELECTOR }
- enum ImageCompressionModeEnums {
  ImageCompressionMode_Off,
  ImageCompressionMode_Lossless,
  NUM_IMAGECOMPRESSIONMODE }
- enum BinningHorizontalModeEnums {
  BinningHorizontalMode_Sum,

BinningHorizontalMode_Average,
NUM_BINNINGHORIZONTALMODE }

- enum PixelFormatEnums {
PixelFormat_Mono8,
PixelFormat_Mono16,
PixelFormat_RGB8Packed,
PixelFormat_BayerGR8,
PixelFormat_BayerRG8,
PixelFormat_BayerGB8,
PixelFormat_BayerBG8,
PixelFormat_BayerGR16,
PixelFormat_BayerRG16,
PixelFormat_BayerGB16,
PixelFormat_BayerBG16,
PixelFormat_Mono12Packed,
PixelFormat_BayerGR12Packed,
PixelFormat_BayerRG12Packed,
PixelFormat_BayerGB12Packed,
PixelFormat_BayerBG12Packed,
PixelFormat_YUV411Packed,
PixelFormat_YUV422Packed,
PixelFormat_YUV444Packed,
PixelFormat_Mono12p,
PixelFormat_BayerGR12p,
PixelFormat_BayerRG12p,
PixelFormat_BayerGB12p,
PixelFormat_BayerBG12p,
PixelFormat_YCbCr8,
PixelFormat_YCbCr422_8,
PixelFormat_YCbCr411_8,
PixelFormat_BGR8,
PixelFormat_BGRa8,
PixelFormat_Mono10Packed,
PixelFormat_BayerGR10Packed,
PixelFormat_BayerRG10Packed,
PixelFormat_BayerGB10Packed,
PixelFormat_BayerBG10Packed,
PixelFormat_Mono10p,
PixelFormat_BayerGR10p,
PixelFormat_BayerRG10p,
PixelFormat_BayerGB10p,
PixelFormat_BayerBG10p,
PixelFormat_Mono1p,
PixelFormat_Mono2p,
PixelFormat_Mono4p,
PixelFormat_Mono8s,
PixelFormat_Mono10,
PixelFormat_Mono12,
PixelFormat_Mono14,
PixelFormat_Mono16s,
PixelFormat_Mono32f,
PixelFormat_BayerBG10,
PixelFormat_BayerBG12,
PixelFormat_BayerGB10,
PixelFormat_BayerGB12,
PixelFormat_BayerGR10,
PixelFormat_BayerGR12,
PixelFormat_BayerRG10,

PixelFormat_BayerRG12,
PixelFormat_RGBa8,
PixelFormat_RGBa10,
PixelFormat_RGBa10p,
PixelFormat_RGBa12,
PixelFormat_RGBa12p,
PixelFormat_RGBa14,
PixelFormat_RGBa16,
PixelFormat_RGB8,
PixelFormat_RGB8_Planar,
PixelFormat_RGB10,
PixelFormat_RGB10_Planar,
PixelFormat_RGB10p,
PixelFormat_RGB10p32,
PixelFormat_RGB12,
PixelFormat_RGB12_Planar,
PixelFormat_RGB12p,
PixelFormat_RGB14,
PixelFormat_RGB16,
PixelFormat_RGB16s,
PixelFormat_RGB32f,
PixelFormat_RGB16_Planar,
PixelFormat_RGB565p,
PixelFormat_BGRa10,
PixelFormat_BGRa10p,
PixelFormat_BGRa12,
PixelFormat_BGRa12p,
PixelFormat_BGRa14,
PixelFormat_BGRa16,
PixelFormat_RGBa32f,
PixelFormat_BGR10,
PixelFormat_BGR10p,
PixelFormat_BGR12,
PixelFormat_BGR12p,
PixelFormat_BGR14,
PixelFormat_BGR16,
PixelFormat_BGR565p,
PixelFormat_R8,
PixelFormat_R10,
PixelFormat_R12,
PixelFormat_R16,
PixelFormat_G8,
PixelFormat_G10,
PixelFormat_G12,
PixelFormat_G16,
PixelFormat_B8,
PixelFormat_B10,
PixelFormat_B12,
PixelFormat_B16,
PixelFormat_Coord3D_ABC8,
PixelFormat_Coord3D_ABC8_Planar,
PixelFormat_Coord3D_ABC10p,
PixelFormat_Coord3D_ABC10p_Planar,
PixelFormat_Coord3D_ABC12p,
PixelFormat_Coord3D_ABC12p_Planar,
PixelFormat_Coord3D_ABC16,
PixelFormat_Coord3D_ABC16_Planar,
PixelFormat_Coord3D_ABC32f,

PixelFormat_Coord3D_ABC32f_Planar,
PixelFormat_Coord3D_AC8,
PixelFormat_Coord3D_AC8_Planar,
PixelFormat_Coord3D_AC10p,
PixelFormat_Coord3D_AC10p_Planar,
PixelFormat_Coord3D_AC12p,
PixelFormat_Coord3D_AC12p_Planar,
PixelFormat_Coord3D_AC16,
PixelFormat_Coord3D_AC16_Planar,
PixelFormat_Coord3D_AC32f,
PixelFormat_Coord3D_AC32f_Planar,
PixelFormat_Coord3D_A8,
PixelFormat_Coord3D_A10p,
PixelFormat_Coord3D_A12p,
PixelFormat_Coord3D_A16,
PixelFormat_Coord3D_A32f,
PixelFormat_Coord3D_B8,
PixelFormat_Coord3D_B10p,
PixelFormat_Coord3D_B12p,
PixelFormat_Coord3D_B16,
PixelFormat_Coord3D_B32f,
PixelFormat_Coord3D_C8,
PixelFormat_Coord3D_C10p,
PixelFormat_Coord3D_C12p,
PixelFormat_Coord3D_C16,
PixelFormat_Coord3D_C32f,
PixelFormat_Confidence1,
PixelFormat_Confidence1p,
PixelFormat_Confidence8,
PixelFormat_Confidence16,
PixelFormat_Confidence32f,
PixelFormat_BiColorBGRG8,
PixelFormat_BiColorBGRG10,
PixelFormat_BiColorBGRG10p,
PixelFormat_BiColorBGRG12,
PixelFormat_BiColorBGRG12p,
PixelFormat_BiColorRGBG8,
PixelFormat_BiColorRGBG10,
PixelFormat_BiColorRGBG10p,
PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,

PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,
PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,
PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,
PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,
PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,
PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,
PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,

PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

- enum DecimationVerticalModeEnums {
  DecimationVerticalMode_Discard,
  NUM_DECIMATIONVERTICALMODE }
- enum LineModeEnums {
  LineMode_Input,
  LineMode_Output,
  NUM_LINEMODE }
- enum LineSourceEnums {
  LineSource_Off,
  LineSource_Line0,
  LineSource_Line1,
  LineSource_Line2,
  LineSource_Line3,
  LineSource_UserOutput0,
  LineSource_UserOutput1,
  LineSource_UserOutput2,
  LineSource_UserOutput3,
  LineSource_Counter0Active,
  LineSource_Counter1Active,
  LineSource_LogicBlock0,
  LineSource_LogicBlock1,
  LineSource_ExposureActive,
  LineSource_FrameTriggerWait,
  LineSource_SerialPort0,
  LineSource_PPSSignal,
  LineSource_AllPixel,
  LineSource_AnyPixel,
  NUM_LINESOURCE }
- enum LineInputFilterSelectorEnums {
  LineInputFilterSelector_Deglitch,
  LineInputFilterSelector_Debounce,
  NUM_LINEINPUTFILTERSELECTOR }
- enum UserOutputSelectorEnums {
  UserOutputSelector_UserOutput0,

UserOutputSelector_UserOutput1,
UserOutputSelector_UserOutput2,
UserOutputSelector_UserOutput3,
NUM_USEROUTPUTSELECTOR }

- enum LineFormatEnums {
LineFormat_NoConnect,
LineFormat_TriState,
LineFormat_TTL,
LineFormat_LVDS,
LineFormat_RS422,
LineFormat_OptoCoupled,
LineFormat_OpenDrain,
NUM_LINEFORMAT }
- enum LineSelectorEnums {
LineSelector_Line0,
LineSelector_Line1,
LineSelector_Line2,
LineSelector_Line3,
NUM_LINESELECTOR }
- enum ExposureActiveModeEnums {
ExposureActiveMode_Line1,
ExposureActiveMode_AnyPixels,
ExposureActiveMode_AllPixels,
NUM_EXPOSUREACTIVEMODE }
- enum CounterTriggerActivationEnums {
CounterTriggerActivation_LevelLow,
CounterTriggerActivation_LevelHigh,
CounterTriggerActivation_FallingEdge,
CounterTriggerActivation_RisingEdge,
CounterTriggerActivation_AnyEdge,
NUM_COUNTERTRIGGERACTIVATION }
- enum CounterSelectorEnums {
CounterSelector_Counter0,
CounterSelector_Counter1,
NUM_COUNTERSELECTOR }
- enum CounterStatusEnums {
CounterStatus_CounterIdle,
CounterStatus_CounterTriggerWait,
CounterStatus_CounterActive,
CounterStatus_CounterCompleted,
CounterStatus_CounterOverflow,
NUM_COUNTERSTATUS }
- enum CounterTriggerSourceEnums {
CounterTriggerSource_Off,
CounterTriggerSource_Line0,
CounterTriggerSource_Line1,
CounterTriggerSource_Line2,
CounterTriggerSource_Line3,
CounterTriggerSource_UserOutput0,
CounterTriggerSource_UserOutput1,
CounterTriggerSource_UserOutput2,
CounterTriggerSource_UserOutput3,
CounterTriggerSource_Counter0Start,
CounterTriggerSource_Counter1Start,
CounterTriggerSource_Counter0End,
CounterTriggerSource_Counter1End,
CounterTriggerSource_LogicBlock0,
CounterTriggerSource_LogicBlock1,

CounterTriggerSource_ExposureStart,
CounterTriggerSource_ExposureEnd,
CounterTriggerSource_FrameTriggerWait,
NUM_COUNTERTRIGGERSOURCE }

- enum CounterResetSourceEnums {
  CounterResetSource_Off,
  CounterResetSource_Line0,
  CounterResetSource_Line1,
  CounterResetSource_Line2,
  CounterResetSource_Line3,
  CounterResetSource_UserOutput0,
  CounterResetSource_UserOutput1,
  CounterResetSource_UserOutput2,
  CounterResetSource_UserOutput3,
  CounterResetSource_Counter0Start,
  CounterResetSource_Counter1Start,
  CounterResetSource_Counter0End,
  CounterResetSource_Counter1End,
  CounterResetSource_LogicBlock0,
  CounterResetSource_LogicBlock1,
  CounterResetSource_ExposureStart,
  CounterResetSource_ExposureEnd,
  CounterResetSource_FrameTriggerWait,
  NUM_COUNTERRESETSOURCE }
- enum CounterEventSourceEnums {
  CounterEventSource_Off,
  CounterEventSource_MHzTick,
  CounterEventSource_Line0,
  CounterEventSource_Line1,
  CounterEventSource_Line2,
  CounterEventSource_Line3,
  CounterEventSource_UserOutput0,
  CounterEventSource_UserOutput1,
  CounterEventSource_UserOutput2,
  CounterEventSource_UserOutput3,
  CounterEventSource_Counter0Start,
  CounterEventSource_Counter1Start,
  CounterEventSource_Counter0End,
  CounterEventSource_Counter1End,
  CounterEventSource_LogicBlock0,
  CounterEventSource_LogicBlock1,
  CounterEventSource_ExposureStart,
  CounterEventSource_ExposureEnd,
  CounterEventSource_FrameTriggerWait,
  NUM_COUNTEREVENTSOURCE }
- enum CounterEventActivationEnums {
  CounterEventActivation_LevelLow,
  CounterEventActivation_LevelHigh,
  CounterEventActivation_FallingEdge,
  CounterEventActivation_RisingEdge,
  CounterEventActivation_AnyEdge,
  NUM_COUNTEREVENTACTIVATION }
- enum CounterResetActivationEnums {
  CounterResetActivation_LevelLow,
  CounterResetActivation_LevelHigh,
  CounterResetActivation_FallingEdge,
  CounterResetActivation_RisingEdge,
  CounterResetActivation_AnyEdge,

NUM_COUNTERRESETACTIVATION }
- enum DeviceTypeEnums {
  DeviceType_Transmitter,
  DeviceType_Receiver,
  DeviceType_Transceiver,
  DeviceType_Peripheral,
  NUM_DEVICETYPE }
- enum DeviceConnectionStatusEnums {
  DeviceConnectionStatus_Active,
  DeviceConnectionStatus_Inactive,
  NUM_DEVICECONNECTIONSTATUS }
- enum DeviceLinkThroughputLimitModeEnums {
  DeviceLinkThroughputLimitMode_On,
  DeviceLinkThroughputLimitMode_Off,
  NUM_DEVICELINKTHROUGHPUTLIMITMODE }
- enum DeviceLinkHeartbeatModeEnums {
  DeviceLinkHeartbeatMode_On,
  DeviceLinkHeartbeatMode_Off,
  NUM_DEVICELINKHEARTBEATMODE }
- enum DeviceStreamChannelTypeEnums {
  DeviceStreamChannelType_Transmitter,
  DeviceStreamChannelType_Receiver,
  NUM_DEVICESTREAMCHANNELTYPE }
- enum DeviceStreamChannelEndiannessEnums {
  DeviceStreamChannelEndianness_Big,
  DeviceStreamChannelEndianness_Little,
  NUM_DEVICESTREAMCHANNELENDIANNESS }
- enum DeviceClockSelectorEnums {
  DeviceClockSelector_Sensor,
  DeviceClockSelector_SensorDigitization,
  DeviceClockSelector_CameraLink,
  NUM_DEVICECLOCKSELECTOR }
- enum DeviceSerialPortSelectorEnums {
  DeviceSerialPortSelector_CameraLink,
  NUM_DEVICESERIALPORTSELECTOR }
- enum DeviceSerialPortBaudRateEnums {
  DeviceSerialPortBaudRate_Baud9600,
  DeviceSerialPortBaudRate_Baud19200,
  DeviceSerialPortBaudRate_Baud38400,
  DeviceSerialPortBaudRate_Baud57600,
  DeviceSerialPortBaudRate_Baud115200,
  DeviceSerialPortBaudRate_Baud230400,
  DeviceSerialPortBaudRate_Baud460800,
  DeviceSerialPortBaudRate_Baud921600,
  NUM_DEVICESERIALPORTBAUDRATE }
- enum SensorTapsEnums {
  SensorTaps_One,
  SensorTaps_Two,
  SensorTaps_Three,
  SensorTaps_Four,
  SensorTaps_Eight,
  SensorTaps_Ten,
  NUM_SENSORTAPS }
- enum SensorDigitizationTapsEnums {
  SensorDigitizationTaps_One,
  SensorDigitizationTaps_Two,
  SensorDigitizationTaps_Three,
  SensorDigitizationTaps_Four,

SensorDigitizationTaps_Eight,
SensorDigitizationTaps_Ten,
NUM_SENSORDIGITIZATIONTAPS }

- enum RegionSelectorEnums {
RegionSelector_Region0,
RegionSelector_Region1,
RegionSelector_Region2,
RegionSelector_All,
NUM_REGIONSELECTOR }
- enum RegionModeEnums {
RegionMode_Off,
RegionMode_On,
NUM_REGIONMODE }
- enum RegionDestinationEnums {
RegionDestination_Stream0,
RegionDestination_Stream1,
RegionDestination_Stream2,
NUM_REGIONDESTINATION }
- enum ImageComponentSelectorEnums {
ImageComponentSelector_Intensity,
ImageComponentSelector_Color,
ImageComponentSelector_Infrared,
ImageComponentSelector_Ultraviolet,
ImageComponentSelector_Range,
ImageComponentSelector_Disparity,
ImageComponentSelector_Confidence,
ImageComponentSelector_Scatter,
NUM_IMAGECOMPONENTSELECTOR }
- enum PixelFormatInfoSelectorEnums {
PixelFormatInfoSelector_Mono1p,
PixelFormatInfoSelector_Mono2p,
PixelFormatInfoSelector_Mono4p,
PixelFormatInfoSelector_Mono8,
PixelFormatInfoSelector_Mono8s,
PixelFormatInfoSelector_Mono10,
PixelFormatInfoSelector_Mono10p,
PixelFormatInfoSelector_Mono12,
PixelFormatInfoSelector_Mono12p,
PixelFormatInfoSelector_Mono14,
PixelFormatInfoSelector_Mono16,
PixelFormatInfoSelector_Mono16s,
PixelFormatInfoSelector_Mono32f,
PixelFormatInfoSelector_BayerBG8,
PixelFormatInfoSelector_BayerBG10,
PixelFormatInfoSelector_BayerBG10p,
PixelFormatInfoSelector_BayerBG12,
PixelFormatInfoSelector_BayerBG12p,
PixelFormatInfoSelector_BayerBG16,
PixelFormatInfoSelector_BayerGB8,
PixelFormatInfoSelector_BayerGB10,
PixelFormatInfoSelector_BayerGB10p,
PixelFormatInfoSelector_BayerGB12,
PixelFormatInfoSelector_BayerGB12p,
PixelFormatInfoSelector_BayerGB16,
PixelFormatInfoSelector_BayerGR8,
PixelFormatInfoSelector_BayerGR10,
PixelFormatInfoSelector_BayerGR10p,
PixelFormatInfoSelector_BayerGR12,

PixelFormatInfoSelector_BayerGR12p,
PixelFormatInfoSelector_BayerGR16,
PixelFormatInfoSelector_BayerRG8,
PixelFormatInfoSelector_BayerRG10,
PixelFormatInfoSelector_BayerRG10p,
PixelFormatInfoSelector_BayerRG12,
PixelFormatInfoSelector_BayerRG12p,
PixelFormatInfoSelector_BayerRG16,
PixelFormatInfoSelector_RGBa8,
PixelFormatInfoSelector_RGBa10,
PixelFormatInfoSelector_RGBa10p,
PixelFormatInfoSelector_RGBa12,
PixelFormatInfoSelector_RGBa12p,
PixelFormatInfoSelector_RGBa14,
PixelFormatInfoSelector_RGBa16,
PixelFormatInfoSelector_RGB8,
PixelFormatInfoSelector_RGB8_Planar,
PixelFormatInfoSelector_RGB10,
PixelFormatInfoSelector_RGB10_Planar,
PixelFormatInfoSelector_RGB10p,
PixelFormatInfoSelector_RGB10p32,
PixelFormatInfoSelector_RGB12,
PixelFormatInfoSelector_RGB12_Planar,
PixelFormatInfoSelector_RGB12p,
PixelFormatInfoSelector_RGB14,
PixelFormatInfoSelector_RGB16,
PixelFormatInfoSelector_RGB16s,
PixelFormatInfoSelector_RGB32f,
PixelFormatInfoSelector_RGB16_Planar,
PixelFormatInfoSelector_RGB565p,
PixelFormatInfoSelector_BGRa8,
PixelFormatInfoSelector_BGRa10,
PixelFormatInfoSelector_BGRa10p,
PixelFormatInfoSelector_BGRa12,
PixelFormatInfoSelector_BGRa12p,
PixelFormatInfoSelector_BGRa14,
PixelFormatInfoSelector_BGRa16,
PixelFormatInfoSelector_RGBa32f,
PixelFormatInfoSelector_BGR8,
PixelFormatInfoSelector_BGR10,
PixelFormatInfoSelector_BGR10p,
PixelFormatInfoSelector_BGR12,
PixelFormatInfoSelector_BGR12p,
PixelFormatInfoSelector_BGR14,
PixelFormatInfoSelector_BGR16,
PixelFormatInfoSelector_BGR565p,
PixelFormatInfoSelector_R8,
PixelFormatInfoSelector_R10,
PixelFormatInfoSelector_R12,
PixelFormatInfoSelector_R16,
PixelFormatInfoSelector_G8,
PixelFormatInfoSelector_G10,
PixelFormatInfoSelector_G12,
PixelFormatInfoSelector_G16,
PixelFormatInfoSelector_B8,
PixelFormatInfoSelector_B10,
PixelFormatInfoSelector_B12,
PixelFormatInfoSelector_B16,

PixelFormatInfoSelector_Coord3D_ABC8,
PixelFormatInfoSelector_Coord3D_ABC8_Planar,
PixelFormatInfoSelector_Coord3D_ABC10p,
PixelFormatInfoSelector_Coord3D_ABC10p_Planar,
PixelFormatInfoSelector_Coord3D_ABC12p,
PixelFormatInfoSelector_Coord3D_ABC12p_Planar,
PixelFormatInfoSelector_Coord3D_ABC16,
PixelFormatInfoSelector_Coord3D_ABC16_Planar,
PixelFormatInfoSelector_Coord3D_ABC32f,
PixelFormatInfoSelector_Coord3D_ABC32f_Planar,
PixelFormatInfoSelector_Coord3D_AC8,
PixelFormatInfoSelector_Coord3D_AC8_Planar,
PixelFormatInfoSelector_Coord3D_AC10p,
PixelFormatInfoSelector_Coord3D_AC10p_Planar,
PixelFormatInfoSelector_Coord3D_AC12p,
PixelFormatInfoSelector_Coord3D_AC12p_Planar,
PixelFormatInfoSelector_Coord3D_AC16,
PixelFormatInfoSelector_Coord3D_AC16_Planar,
PixelFormatInfoSelector_Coord3D_AC32f,
PixelFormatInfoSelector_Coord3D_AC32f_Planar,
PixelFormatInfoSelector_Coord3D_A8,
PixelFormatInfoSelector_Coord3D_A10p,
PixelFormatInfoSelector_Coord3D_A12p,
PixelFormatInfoSelector_Coord3D_A16,
PixelFormatInfoSelector_Coord3D_A32f,
PixelFormatInfoSelector_Coord3D_B8,
PixelFormatInfoSelector_Coord3D_B10p,
PixelFormatInfoSelector_Coord3D_B12p,
PixelFormatInfoSelector_Coord3D_B16,
PixelFormatInfoSelector_Coord3D_B32f,
PixelFormatInfoSelector_Coord3D_C8,
PixelFormatInfoSelector_Coord3D_C10p,
PixelFormatInfoSelector_Coord3D_C12p,
PixelFormatInfoSelector_Coord3D_C16,
PixelFormatInfoSelector_Coord3D_C32f,
PixelFormatInfoSelector_Confidence1,
PixelFormatInfoSelector_Confidence1p,
PixelFormatInfoSelector_Confidence8,
PixelFormatInfoSelector_Confidence16,
PixelFormatInfoSelector_Confidence32f,
PixelFormatInfoSelector_BiColorBGRG8,
PixelFormatInfoSelector_BiColorBGRG10,
PixelFormatInfoSelector_BiColorBGRG10p,
PixelFormatInfoSelector_BiColorBGRG12,
PixelFormatInfoSelector_BiColorBGRG12p,
PixelFormatInfoSelector_BiColorRGBG8,
PixelFormatInfoSelector_BiColorRGBG10,
PixelFormatInfoSelector_BiColorRGBG10p,
PixelFormatInfoSelector_BiColorRGBG12,
PixelFormatInfoSelector_BiColorRGBG12p,
PixelFormatInfoSelector_SCF1WBWG8,
PixelFormatInfoSelector_SCF1WBWG10,
PixelFormatInfoSelector_SCF1WBWG10p,
PixelFormatInfoSelector_SCF1WBWG12,
PixelFormatInfoSelector_SCF1WBWG12p,
PixelFormatInfoSelector_SCF1WBWG14,
PixelFormatInfoSelector_SCF1WBWG16,
PixelFormatInfoSelector_SCF1WGWB8,

PixelFormatInfoSelector_SCF1WGWB10,
PixelFormatInfoSelector_SCF1WGWB10p,
PixelFormatInfoSelector_SCF1WGWB12,
PixelFormatInfoSelector_SCF1WGWB12p,
PixelFormatInfoSelector_SCF1WGWB14,
PixelFormatInfoSelector_SCF1WGWB16,
PixelFormatInfoSelector_SCF1WGWR8,
PixelFormatInfoSelector_SCF1WGWR10,
PixelFormatInfoSelector_SCF1WGWR10p,
PixelFormatInfoSelector_SCF1WGWR12,
PixelFormatInfoSelector_SCF1WGWR12p,
PixelFormatInfoSelector_SCF1WGWR14,
PixelFormatInfoSelector_SCF1WGWR16,
PixelFormatInfoSelector_SCF1WRWG8,
PixelFormatInfoSelector_SCF1WRWG10,
PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,
PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,

PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

- enum DeinterlacingEnums {
  Deinterlacing_Off,
  Deinterlacing_LineDuplication,
  Deinterlacing_Weave,
  NUM_DEINTERLACING }
- enum ImageCompressionRateOptionEnums {
  ImageCompressionRateOption_FixBitrate,
  ImageCompressionRateOption_FixQuality,
  NUM_IMAGECOMPRESSIONRATEOPTION }
- enum ImageCompressionJPEGFormatOptionEnums {
  ImageCompressionJPEGFormatOption_Lossless,
  ImageCompressionJPEGFormatOption_BaselineStandard,
  ImageCompressionJPEGFormatOption_BaselineOptimized,
  ImageCompressionJPEGFormatOption_Progressive,
  NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }
- enum AcquisitionStatusSelectorEnums {
  AcquisitionStatusSelector_AcquisitionTriggerWait,
  AcquisitionStatusSelector_AcquisitionActive,
  AcquisitionStatusSelector_AcquisitionTransfer,
  AcquisitionStatusSelector_FrameTriggerWait,
  AcquisitionStatusSelector_FrameActive,
  AcquisitionStatusSelector_ExposureActive,
  NUM_ACQUISITIONSTATUSSELECTOR }
- enum ExposureTimeModeEnums {
  ExposureTimeMode_Common,
  ExposureTimeMode_Individual,
  NUM_EXPOSURETIMEMODE }
- enum ExposureTimeSelectorEnums {
  ExposureTimeSelector_Common,

ExposureTimeSelector_Red,
ExposureTimeSelector_Green,
ExposureTimeSelector_Blue,
ExposureTimeSelector_Cyan,
ExposureTimeSelector_Magenta,
ExposureTimeSelector_Yellow,
ExposureTimeSelector_Infrared,
ExposureTimeSelector_Ultraviolet,
ExposureTimeSelector_Stage1,
ExposureTimeSelector_Stage2,
NUM_EXPOSURETIMESELECTOR }

- enum GainAutoBalanceEnums {
  GainAutoBalance_Off,
  GainAutoBalance_Once,
  GainAutoBalance_Continuous,
  NUM_GAINAUTOBALANCE }
- enum BlackLevelAutoEnums {
  BlackLevelAuto_Off,
  BlackLevelAuto_Once,
  BlackLevelAuto_Continuous,
  NUM_BLACKLEVELAUTO }
- enum BlackLevelAutoBalanceEnums {
  BlackLevelAutoBalance_Off,
  BlackLevelAutoBalance_Once,
  BlackLevelAutoBalance_Continuous,
  NUM_BLACKLEVELAUTOBALANCE }
- enum WhiteClipSelectorEnums {
  WhiteClipSelector_All,
  WhiteClipSelector_Red,
  WhiteClipSelector_Green,
  WhiteClipSelector_Blue,
  WhiteClipSelector_Y,
  WhiteClipSelector_U,
  WhiteClipSelector_V,
  WhiteClipSelector_Tap1,
  WhiteClipSelector_Tap2,
  NUM_WHITECLIPSELECTOR }
- enum TimerSelectorEnums {
  TimerSelector_Timer0,
  TimerSelector_Timer1,
  TimerSelector_Timer2,
  NUM_TIMERSELECTOR }
- enum TimerStatusEnums {
  TimerStatus_TimerIdle,
  TimerStatus_TimerTriggerWait,
  TimerStatus_TimerActive,
  TimerStatus_TimerCompleted,
  NUM_TIMERSTATUS }
- enum TimerTriggerSourceEnums {
  TimerTriggerSource_Off,
  TimerTriggerSource_AcquisitionTrigger,
  TimerTriggerSource_AcquisitionStart,
  TimerTriggerSource_AcquisitionEnd,
  TimerTriggerSource_FrameTrigger,
  TimerTriggerSource_FrameStart,
  TimerTriggerSource_FrameEnd,
  TimerTriggerSource_FrameBurstStart,
  TimerTriggerSource_FrameBurstEnd,

TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

- enum TimerTriggerActivationEnums {
  TimerTriggerActivation_RisingEdge,
  TimerTriggerActivation_FallingEdge,
  TimerTriggerActivation_AnyEdge,
  TimerTriggerActivation_LevelHigh,
  TimerTriggerActivation_LevelLow,
  NUM_TIMERTRIGGERACTIVATION }
- enum EncoderSelectorEnums {
  EncoderSelector_Encoder0,
  EncoderSelector_Encoder1,
  EncoderSelector_Encoder2,
  NUM_ENCODERSELECTOR }
- enum EncoderSourceAEnums {
  EncoderSourceA_Off,
  EncoderSourceA_Line0,
  EncoderSourceA_Line1,
  EncoderSourceA_Line2,
  NUM_ENCODERSOURCEA }
- enum EncoderSourceBEnums {
  EncoderSourceB_Off,
  EncoderSourceB_Line0,
  EncoderSourceB_Line1,

EncoderSourceB_Line2,
NUM_ENCODERSOURCEB }

- enum EncoderModeEnums {
  EncoderMode_FourPhase,
  EncoderMode_HighResolution,
  NUM_ENCODERMODE }
- enum EncoderOutputModeEnums {
  EncoderOutputMode_Off,
  EncoderOutputMode_PositionUp,
  EncoderOutputMode_PositionDown,
  EncoderOutputMode_DirectionUp,
  EncoderOutputMode_DirectionDown,
  EncoderOutputMode_Motion,
  NUM_ENCODEROUTPUTMODE }
- enum EncoderStatusEnums {
  EncoderStatus_EncoderUp,
  EncoderStatus_EncoderDown,
  EncoderStatus_EncoderIdle,
  EncoderStatus_EncoderStatic,
  NUM_ENCODERSTATUS }
- enum EncoderResetSourceEnums {
  EncoderResetSource_Off,
  EncoderResetSource_AcquisitionTrigger,
  EncoderResetSource_AcquisitionStart,
  EncoderResetSource_AcquisitionEnd,
  EncoderResetSource_FrameTrigger,
  EncoderResetSource_FrameStart,
  EncoderResetSource_FrameEnd,
  EncoderResetSource_ExposureStart,
  EncoderResetSource_ExposureEnd,
  EncoderResetSource_Line0,
  EncoderResetSource_Line1,
  EncoderResetSource_Line2,
  EncoderResetSource_Counter0Start,
  EncoderResetSource_Counter1Start,
  EncoderResetSource_Counter2Start,
  EncoderResetSource_Counter0End,
  EncoderResetSource_Counter1End,
  EncoderResetSource_Counter2End,
  EncoderResetSource_Timer0Start,
  EncoderResetSource_Timer1Start,
  EncoderResetSource_Timer2Start,
  EncoderResetSource_Timer0End,
  EncoderResetSource_Timer1End,
  EncoderResetSource_Timer2End,
  EncoderResetSource_UserOutput0,
  EncoderResetSource_UserOutput1,
  EncoderResetSource_UserOutput2,
  EncoderResetSource_SoftwareSignal0,
  EncoderResetSource_SoftwareSignal1,
  EncoderResetSource_SoftwareSignal2,
  EncoderResetSource_Action0,
  EncoderResetSource_Action1,
  EncoderResetSource_Action2,
  EncoderResetSource_LinkTrigger0,
  EncoderResetSource_LinkTrigger1,
  EncoderResetSource_LinkTrigger2,
  NUM_ENCODERRESETSOURCE }

- enum EncoderResetActivationEnums {
  EncoderResetActivation_RisingEdge,
  EncoderResetActivation_FallingEdge,
  EncoderResetActivation_AnyEdge,
  EncoderResetActivation_LevelHigh,
  EncoderResetActivation_LevelLow,
  NUM_ENCODERRESETACTIVATION }
- enum SoftwareSignalSelectorEnums {
  SoftwareSignalSelector_SoftwareSignal0,
  SoftwareSignalSelector_SoftwareSignal1,
  SoftwareSignalSelector_SoftwareSignal2,
  NUM_SOFTWARESIGNALSELECTOR }
- enum ActionUnconditionalModeEnums {
  ActionUnconditionalMode_Off,
  ActionUnconditionalMode_On,
  NUM_ACTIONUNCONDITIONALMODE }
- enum SourceSelectorEnums {
  SourceSelector_Source0,
  SourceSelector_Source1,
  SourceSelector_Source2,
  SourceSelector_All,
  NUM_SOURCESELECTOR }
- enum TransferSelectorEnums {
  TransferSelector_Stream0,
  TransferSelector_Stream1,
  TransferSelector_Stream2,
  TransferSelector_All,
  NUM_TRANSFERSELECTOR }
- enum TransferTriggerSelectorEnums {
  TransferTriggerSelector_TransferStart,
  TransferTriggerSelector_TransferStop,
  TransferTriggerSelector_TransferAbort,
  TransferTriggerSelector_TransferPause,
  TransferTriggerSelector_TransferResume,
  TransferTriggerSelector_TransferActive,
  TransferTriggerSelector_TransferBurstStart,
  TransferTriggerSelector_TransferBurstStop,
  NUM_TRANSFERTRIGGERSELECTOR }
- enum TransferTriggerModeEnums {
  TransferTriggerMode_Off,
  TransferTriggerMode_On,
  NUM_TRANSFERTRIGGERMODE }
- enum TransferTriggerSourceEnums {
  TransferTriggerSource_Line0,
  TransferTriggerSource_Line1,
  TransferTriggerSource_Line2,
  TransferTriggerSource_Counter0Start,
  TransferTriggerSource_Counter1Start,
  TransferTriggerSource_Counter2Start,
  TransferTriggerSource_Counter0End,
  TransferTriggerSource_Counter1End,
  TransferTriggerSource_Counter2End,
  TransferTriggerSource_Timer0Start,
  TransferTriggerSource_Timer1Start,
  TransferTriggerSource_Timer2Start,
  TransferTriggerSource_Timer0End,
  TransferTriggerSource_Timer1End,
  TransferTriggerSource_Timer2End,

TransferTriggerSource_SoftwareSignal0,
TransferTriggerSource_SoftwareSignal1,
TransferTriggerSource_SoftwareSignal2,
TransferTriggerSource_Action0,
TransferTriggerSource_Action1,
TransferTriggerSource_Action2,
NUM_TRANSFERTRIGGERSOURCE }

- enum TransferTriggerActivationEnums {
  TransferTriggerActivation_RisingEdge,
  TransferTriggerActivation_FallingEdge,
  TransferTriggerActivation_AnyEdge,
  TransferTriggerActivation_LevelHigh,
  TransferTriggerActivation_LevelLow,
  NUM_TRANSFERTRIGGERACTIVATION }
- enum TransferStatusSelectorEnums {
  TransferStatusSelector_Streaming,
  TransferStatusSelector_Paused,
  TransferStatusSelector_Stopping,
  TransferStatusSelector_Stopped,
  TransferStatusSelector_QueueOverflow,
  NUM_TRANSFERSTATUSSELECTOR }
- enum TransferComponentSelectorEnums {
  TransferComponentSelector_Red,
  TransferComponentSelector_Green,
  TransferComponentSelector_Blue,
  TransferComponentSelector_All,
  NUM_TRANSFERCOMPONENTSELECTOR }
- enum Scan3dDistanceUnitEnums {
  Scan3dDistanceUnit_Millimeter,
  Scan3dDistanceUnit_Inch,
  NUM_SCAN3DDISTANCEUNIT }
- enum Scan3dCoordinateSystemEnums {
  Scan3dCoordinateSystem_Cartesian,
  Scan3dCoordinateSystem_Spherical,
  Scan3dCoordinateSystem_Cylindrical,
  NUM_SCAN3DCOORDINATESYSTEM }
- enum Scan3dOutputModeEnums {
  Scan3dOutputMode_UncalibratedC,
  Scan3dOutputMode_CalibratedABC_Grid,
  Scan3dOutputMode_CalibratedABC_PointCloud,
  Scan3dOutputMode_CalibratedAC,
  Scan3dOutputMode_CalibratedAC_Linescan,
  Scan3dOutputMode_CalibratedC,
  Scan3dOutputMode_CalibratedC_Linescan,
  Scan3dOutputMode_RectifiedC,
  Scan3dOutputMode_RectifiedC_Linescan,
  Scan3dOutputMode_DisparityC,
  Scan3dOutputMode_DisparityC_Linescan,
  NUM_SCAN3DOUTPUTMODE }
- enum Scan3dCoordinateSystemReferenceEnums {
  Scan3dCoordinateSystemReference_Anchor,
  Scan3dCoordinateSystemReference_Transformed,
  NUM_SCAN3DCOORDINATESYSTEMREFERENCE }
- enum Scan3dCoordinateSelectorEnums {
  Scan3dCoordinateSelector_CoordinateA,
  Scan3dCoordinateSelector_CoordinateB,
  Scan3dCoordinateSelector_CoordinateC,
  NUM_SCAN3DCOORDINATESELECTOR }

- enum Scan3dCoordinateTransformSelectorEnums {
  Scan3dCoordinateTransformSelector_RotationX,
  Scan3dCoordinateTransformSelector_RotationY,
  Scan3dCoordinateTransformSelector_RotationZ,
  Scan3dCoordinateTransformSelector_TranslationX,
  Scan3dCoordinateTransformSelector_TranslationY,
  Scan3dCoordinateTransformSelector_TranslationZ,
  NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }
- enum Scan3dCoordinateReferenceSelectorEnums {
  Scan3dCoordinateReferenceSelector_RotationX,
  Scan3dCoordinateReferenceSelector_RotationY,
  Scan3dCoordinateReferenceSelector_RotationZ,
  Scan3dCoordinateReferenceSelector_TranslationX,
  Scan3dCoordinateReferenceSelector_TranslationY,
  Scan3dCoordinateReferenceSelector_TranslationZ,
  NUM_SCAN3DCOORDINATEREFERENCESELECTOR }
- enum ChunkImageComponentEnums {
  ChunkImageComponent_Intensity,
  ChunkImageComponent_Color,
  ChunkImageComponent_Infrared,
  ChunkImageComponent_Ultraviolet,
  ChunkImageComponent_Range,
  ChunkImageComponent_Disparity,
  ChunkImageComponent_Confidence,
  ChunkImageComponent_Scatter,
  NUM_CHUNKIMAGECOMPONENT }
- enum ChunkCounterSelectorEnums {
  ChunkCounterSelector_Counter0,
  ChunkCounterSelector_Counter1,
  ChunkCounterSelector_Counter2,
  NUM_CHUNKCOUNTERSELECTOR }
- enum ChunkTimerSelectorEnums {
  ChunkTimerSelector_Timer0,
  ChunkTimerSelector_Timer1,
  ChunkTimerSelector_Timer2,
  NUM_CHUNKTIMERSELECTOR }
- enum ChunkEncoderSelectorEnums {
  ChunkEncoderSelector_Encoder0,
  ChunkEncoderSelector_Encoder1,
  ChunkEncoderSelector_Encoder2,
  NUM_CHUNKENCODERSELECTOR }
- enum ChunkEncoderStatusEnums {
  ChunkEncoderStatus_EncoderUp,
  ChunkEncoderStatus_EncoderDown,
  ChunkEncoderStatus_EncoderIdle,
  ChunkEncoderStatus_EncoderStatic,
  NUM_CHUNKENCODERSTATUS }
- enum ChunkExposureTimeSelectorEnums {
  ChunkExposureTimeSelector_Common,
  ChunkExposureTimeSelector_Red,
  ChunkExposureTimeSelector_Green,
  ChunkExposureTimeSelector_Blue,
  ChunkExposureTimeSelector_Cyan,
  ChunkExposureTimeSelector_Magenta,
  ChunkExposureTimeSelector_Yellow,
  ChunkExposureTimeSelector_Infrared,
  ChunkExposureTimeSelector_Ultraviolet,
  ChunkExposureTimeSelector_Stage1,

ChunkExposureTimeSelector_Stage2,
NUM_CHUNKEXPOSURETIMESELECTOR }

- enum ChunkSourceIDEnums {
ChunkSourceID_Source0,
ChunkSourceID_Source1,
ChunkSourceID_Source2,
NUM_CHUNKSOURCEID }
- enum ChunkRegionIDEnums {
ChunkRegionID_Region0,
ChunkRegionID_Region1,
ChunkRegionID_Region2,
NUM_CHUNKREGIONID }
- enum ChunkTransferStreamIDEnums {
ChunkTransferStreamID_Stream0,
ChunkTransferStreamID_Stream1,
ChunkTransferStreamID_Stream2,
ChunkTransferStreamID_Stream3,
NUM_CHUNKTRANSFERSTREAMID }
- enum ChunkScan3dDistanceUnitEnums {
ChunkScan3dDistanceUnit_Millimeter,
ChunkScan3dDistanceUnit_Inch,
NUM_CHUNKSCAN3DDISTANCEUNIT }
- enum ChunkScan3dOutputModeEnums {
ChunkScan3dOutputMode_UncalibratedC,
ChunkScan3dOutputMode_CalibratedABC_Grid,
ChunkScan3dOutputMode_CalibratedABC_PointCloud,
ChunkScan3dOutputMode_CalibratedAC,
ChunkScan3dOutputMode_CalibratedAC_Linescan,
ChunkScan3dOutputMode_CalibratedC,
ChunkScan3dOutputMode_CalibratedC_Linescan,
ChunkScan3dOutputMode_RectifiedC,
ChunkScan3dOutputMode_RectifiedC_Linescan,
ChunkScan3dOutputMode_DisparityC,
ChunkScan3dOutputMode_DisparityC_Linescan,
NUM_CHUNKSCAN3DOUTPUTMODE }
- enum ChunkScan3dCoordinateSystemEnums {
ChunkScan3dCoordinateSystem_Cartesian,
ChunkScan3dCoordinateSystem_Spherical,
ChunkScan3dCoordinateSystem_Cylindrical,
NUM_CHUNKSCAN3DCOORDINATESYSTEM }
- enum ChunkScan3dCoordinateSystemReferenceEnums {
ChunkScan3dCoordinateSystemReference_Anchor,
ChunkScan3dCoordinateSystemReference_Transformed,
NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
- enum ChunkScan3dCoordinateSelectorEnums {
ChunkScan3dCoordinateSelector_CoordinateA,
ChunkScan3dCoordinateSelector_CoordinateB,
ChunkScan3dCoordinateSelector_CoordinateC,
NUM_CHUNKSCAN3DCOORDINATESELECTOR }
- enum ChunkScan3dCoordinateTransformSelectorEnums {
ChunkScan3dCoordinateTransformSelector_RotationX,
ChunkScan3dCoordinateTransformSelector_RotationY,
ChunkScan3dCoordinateTransformSelector_RotationZ,
ChunkScan3dCoordinateTransformSelector_TranslationX,
ChunkScan3dCoordinateTransformSelector_TranslationY,
ChunkScan3dCoordinateTransformSelector_TranslationZ,
NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }

- enum ChunkScan3dCoordinateReferenceSelectorEnums {
  ChunkScan3dCoordinateReferenceSelector_RotationX,
  ChunkScan3dCoordinateReferenceSelector_RotationY,
  ChunkScan3dCoordinateReferenceSelector_RotationZ,
  ChunkScan3dCoordinateReferenceSelector_TranslationX,
  ChunkScan3dCoordinateReferenceSelector_TranslationY,
  ChunkScan3dCoordinateReferenceSelector_TranslationZ,
  NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }
- enum DeviceTapGeometryEnums {
  DeviceTapGeometry_Geometry_1X_1Y,
  DeviceTapGeometry_Geometry_1X2_1Y,
  DeviceTapGeometry_Geometry_1X2_1Y2,
  DeviceTapGeometry_Geometry_2X_1Y,
  DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
  DeviceTapGeometry_Geometry_2XE_1Y2,
  DeviceTapGeometry_Geometry_2XM_1Y,
  DeviceTapGeometry_Geometry_2XM_1Y2,
  DeviceTapGeometry_Geometry_1X_1Y2,
  DeviceTapGeometry_Geometry_1X_2YE,
  DeviceTapGeometry_Geometry_1X3_1Y,
  DeviceTapGeometry_Geometry_3X_1Y,
  DeviceTapGeometry_Geometry_1X,
  DeviceTapGeometry_Geometry_1X2,
  DeviceTapGeometry_Geometry_2X,
  DeviceTapGeometry_Geometry_2XE,
  DeviceTapGeometry_Geometry_2XM,
  DeviceTapGeometry_Geometry_1X3,
  DeviceTapGeometry_Geometry_3X,
  DeviceTapGeometry_Geometry_1X4_1Y,
  DeviceTapGeometry_Geometry_4X_1Y,
  DeviceTapGeometry_Geometry_2X2_1Y,
  DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
  DeviceTapGeometry_Geometry_1X2_2YE,
  DeviceTapGeometry_Geometry_2X_2YE,
  DeviceTapGeometry_Geometry_2XE_2YE,
  DeviceTapGeometry_Geometry_2XM_2YE,
  DeviceTapGeometry_Geometry_1X4,
  DeviceTapGeometry_Geometry_4X,
  DeviceTapGeometry_Geometry_2X2,
  DeviceTapGeometry_Geometry_2X2E,
  DeviceTapGeometry_Geometry_2X2M,
  DeviceTapGeometry_Geometry_1X8_1Y,
  DeviceTapGeometry_Geometry_8X_1Y,
  DeviceTapGeometry_Geometry_4X2_1Y,
  DeviceTapGeometry_Geometry_2X2E_2YE,
  DeviceTapGeometry_Geometry_1X8,
  DeviceTapGeometry_Geometry_8X,
  DeviceTapGeometry_Geometry_4X2,
  DeviceTapGeometry_Geometry_4X2E,
  DeviceTapGeometry_Geometry_4X2E_1Y,
  DeviceTapGeometry_Geometry_1X10_1Y,
  DeviceTapGeometry_Geometry_10X_1Y,
  DeviceTapGeometry_Geometry_1X10,
  DeviceTapGeometry_Geometry_10X,
  NUM_DEVICETAPGEOMETRY }
- enum GevPhysicalLinkConfigurationEnums {
  GevPhysicalLinkConfiguration_SingleLink,
  GevPhysicalLinkConfiguration_MultiLink,

GevPhysicalLinkConfiguration_StaticLAG,
GevPhysicalLinkConfiguration_DynamicLAG,
NUM_GEVPHYSICALLINKCONFIGURATION }

- enum GevCurrentPhysicalLinkConfigurationEnums {
GevCurrentPhysicalLinkConfiguration_SingleLink,
GevCurrentPhysicalLinkConfiguration_MultiLink,
GevCurrentPhysicalLinkConfiguration_StaticLAG,
GevCurrentPhysicalLinkConfiguration_DynamicLAG,
NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }
- enum GevIPConfigurationStatusEnums {
GevIPConfigurationStatus_None,
GevIPConfigurationStatus_PersistentIP,
GevIPConfigurationStatus_DHCP,
GevIPConfigurationStatus_LLA,
GevIPConfigurationStatus_ForceIP,
NUM_GEVIPCONFIGURATIONSTATUS }
- enum GevGVCPExtendedStatusCodesSelectorEnums {
GevGVCPExtendedStatusCodesSelector_Version1_1,
GevGVCPExtendedStatusCodesSelector_Version2_0,
NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }
- enum GevGVSPExtendedIDModeEnums {
GevGVSPExtendedIDMode_Off,
GevGVSPExtendedIDMode_On,
NUM_GEVGVSPEXTENDEDIDMODE }
- enum ClConfigurationEnums {
ClConfiguration_Base,
ClConfiguration_Medium,
ClConfiguration_Full,
ClConfiguration_DualBase,
ClConfiguration_EightyBit,
NUM_CLCONFIGURATION }
- enum ClTimeSlotsCountEnums {
ClTimeSlotsCount_One,
ClTimeSlotsCount_Two,
ClTimeSlotsCount_Three,
NUM_CLTIMESLOTSCOUNT }
- enum CxpLinkConfigurationStatusEnums {
CxpLinkConfigurationStatus_None,
CxpLinkConfigurationStatus_Pending,
CxpLinkConfigurationStatus_CXP1_X1,
CxpLinkConfigurationStatus_CXP2_X1,
CxpLinkConfigurationStatus_CXP3_X1,
CxpLinkConfigurationStatus_CXP5_X1,
CxpLinkConfigurationStatus_CXP6_X1,
CxpLinkConfigurationStatus_CXP1_X2,
CxpLinkConfigurationStatus_CXP2_X2,
CxpLinkConfigurationStatus_CXP3_X2,
CxpLinkConfigurationStatus_CXP5_X2,
CxpLinkConfigurationStatus_CXP6_X2,
CxpLinkConfigurationStatus_CXP1_X3,
CxpLinkConfigurationStatus_CXP2_X3,
CxpLinkConfigurationStatus_CXP3_X3,
CxpLinkConfigurationStatus_CXP5_X3,
CxpLinkConfigurationStatus_CXP6_X3,
CxpLinkConfigurationStatus_CXP1_X4,
CxpLinkConfigurationStatus_CXP2_X4,
CxpLinkConfigurationStatus_CXP3_X4,
CxpLinkConfigurationStatus_CXP5_X4,

CxpLinkConfigurationStatus_CXP6_X4,
CxpLinkConfigurationStatus_CXP1_X5,
CxpLinkConfigurationStatus_CXP2_X5,
CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,
CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

- enum CxpLinkConfigurationPreferredEnums {
  CxpLinkConfigurationPreferred_CXP1_X1,
  CxpLinkConfigurationPreferred_CXP2_X1,
  CxpLinkConfigurationPreferred_CXP3_X1,
  CxpLinkConfigurationPreferred_CXP5_X1,
  CxpLinkConfigurationPreferred_CXP6_X1,
  CxpLinkConfigurationPreferred_CXP1_X2,
  CxpLinkConfigurationPreferred_CXP2_X2,
  CxpLinkConfigurationPreferred_CXP3_X2,
  CxpLinkConfigurationPreferred_CXP5_X2,
  CxpLinkConfigurationPreferred_CXP6_X2,
  CxpLinkConfigurationPreferred_CXP1_X3,
  CxpLinkConfigurationPreferred_CXP2_X3,
  CxpLinkConfigurationPreferred_CXP3_X3,
  CxpLinkConfigurationPreferred_CXP5_X3,
  CxpLinkConfigurationPreferred_CXP6_X3,
  CxpLinkConfigurationPreferred_CXP1_X4,
  CxpLinkConfigurationPreferred_CXP2_X4,
  CxpLinkConfigurationPreferred_CXP3_X4,
  CxpLinkConfigurationPreferred_CXP5_X4,
  CxpLinkConfigurationPreferred_CXP6_X4,
  CxpLinkConfigurationPreferred_CXP1_X5,
  CxpLinkConfigurationPreferred_CXP2_X5,
  CxpLinkConfigurationPreferred_CXP3_X5,
  CxpLinkConfigurationPreferred_CXP5_X5,
  CxpLinkConfigurationPreferred_CXP6_X5,
  CxpLinkConfigurationPreferred_CXP1_X6,
  CxpLinkConfigurationPreferred_CXP2_X6,
  CxpLinkConfigurationPreferred_CXP3_X6,
  CxpLinkConfigurationPreferred_CXP5_X6,
  CxpLinkConfigurationPreferred_CXP6_X6,
  NUM_CXPLINKCONFIGURATIONPREFERRED }
- enum CxpLinkConfigurationEnums {
  CxpLinkConfiguration_Auto,
  CxpLinkConfiguration_CXP1_X1,
  CxpLinkConfiguration_CXP2_X1,
  CxpLinkConfiguration_CXP3_X1,
  CxpLinkConfiguration_CXP5_X1,
  CxpLinkConfiguration_CXP6_X1,
  CxpLinkConfiguration_CXP1_X2,
  CxpLinkConfiguration_CXP2_X2,
  CxpLinkConfiguration_CXP3_X2,
  CxpLinkConfiguration_CXP5_X2,
  CxpLinkConfiguration_CXP6_X2,
  CxpLinkConfiguration_CXP1_X3,
  CxpLinkConfiguration_CXP2_X3,

CxpLinkConfiguration_CXP3_X3,
CxpLinkConfiguration_CXP5_X3,
CxpLinkConfiguration_CXP6_X3,
CxpLinkConfiguration_CXP1_X4,
CxpLinkConfiguration_CXP2_X4,
CxpLinkConfiguration_CXP3_X4,
CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,
CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }

- enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
- enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }
- enum InferenceBoxType {
  INFERENCE_BOX_TYPE_RECTANGLE = 0,
  INFERENCE_BOX_TYPE_CIRCLE = 1,
  INFERENCE_BOX_TYPE_ROTATED_RECTANGLE = 2 }

  Inference Bounding Box Type.
- enum Error {
  SPINNAKER_ERR_SUCCESS = 0,
  SPINNAKER_ERR_ERROR = -1001,
  SPINNAKER_ERR_NOT_INITIALIZED = -1002,
  SPINNAKER_ERR_NOT_IMPLEMENTED = -1003,
  SPINNAKER_ERR_RESOURCE_IN_USE = -1004,
  SPINNAKER_ERR_ACCESS_DENIED = -1005,
  SPINNAKER_ERR_INVALID_HANDLE = -1006,
  SPINNAKER_ERR_INVALID_ID = -1007,
  SPINNAKER_ERR_NO_DATA = -1008,
  SPINNAKER_ERR_INVALID_PARAMETER = -1009,
  SPINNAKER_ERR_IO = -1010,
  SPINNAKER_ERR_TIMEOUT = -1011,
  SPINNAKER_ERR_ABORT = -1012,
  SPINNAKER_ERR_INVALID_BUFFER = -1013,
  SPINNAKER_ERR_NOT_AVAILABLE = -1014,
  SPINNAKER_ERR_INVALID_ADDRESS = -1015,
  SPINNAKER_ERR_BUFFER_TOO_SMALL = -1016,
  SPINNAKER_ERR_INVALID_INDEX = -1017,
  SPINNAKER_ERR_PARSING_CHUNK_DATA = -1018,
  SPINNAKER_ERR_INVALID_VALUE = -1019,
  SPINNAKER_ERR_RESOURCE_EXHAUSTED = -1020,
  SPINNAKER_ERR_OUT_OF_MEMORY = -1021,
  SPINNAKER_ERR_BUSY = -1022,
  GENICAM_ERR_INVALID_ARGUMENT = -2001,

GENICAM_ERR_OUT_OF_RANGE = -2002,
GENICAM_ERR_PROPERTY = -2003,
GENICAM_ERR_RUN_TIME = -2004,
GENICAM_ERR_LOGICAL = -2005,
GENICAM_ERR_ACCESS = -2006,
GENICAM_ERR_TIMEOUT = -2007,
GENICAM_ERR_DYNAMIC_CAST = -2008,
GENICAM_ERR_GENERIC = -2009,
GENICAM_ERR_BAD_ALLOCATION = -2010,
SPINNAKER_ERR_IM_CONVERT = -3001,
SPINNAKER_ERR_IM_COPY = -3002,
SPINNAKER_ERR_IM_MALLOC = -3003,
SPINNAKER_ERR_IM_NOT_SUPPORTED = -3004,
SPINNAKER_ERR_IM_HISTOGRAM_RANGE = -3005,
SPINNAKER_ERR_IM_HISTOGRAM_MEAN = -3006,
SPINNAKER_ERR_IM_MIN_MAX = -3007,
SPINNAKER_ERR_IM_COLOR_CONVERSION = -3008,
SPINNAKER_ERR_IM_DECOMPRESSION = -3009,
SPINNAKER_ERR_CUSTOM_ID = -10000 }

Spinnaker enum definitions.

- enum EventType {
  SPINNAKER_EVENT_ARRIVAL_REMOVAL,
  SPINNAKER_EVENT_DEVICE,
  SPINNAKER_EVENT_DEVICE_SPECIFIC,
  SPINNAKER_EVENT_NEW_BUFFER,
  SPINNAKER_EVENT_LOGGING_EVENT,
  SPINNAKER_EVENT_UNKNOWN,
  SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL }

  Event types in Spinnaker.

- enum PixelFormatNamespaceID {
  SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
  SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
  SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
  SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
  SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,
  SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }

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

- enum ColorProcessingAlgorithm {
  DEFAULT,
  NO_COLOR_PROCESSING,
  NEAREST_NEIGHBOR,
  NEAREST_NEIGHBOR_AVG,
  BILINEAR,
  EDGE_SENSING,
  HQ_LINEAR,
  IPP,
  DIRECTIONAL_FILTER,
  RIGOROUS,
  WEIGHTED_DIRECTIONAL_FILTER }

  Color processing algorithms.

- enum ImageFileFormat {
  FROM_FILE_EXT = -1,
  PGM,
  PPM,
  BMP,
  JPEG,
  JPEG2000,

TIFF,
PNG,
RAW,
JPEG12_C,
IMAGE_FILE_FORMAT_FORCE_32BITS = 0x7FFFFFFF }

File formats to be used for saving images to disk.

- enum ImageStatus {
  IMAGE_UNKNOWN_ERROR = -1,
  IMAGE_NO_ERROR = 0,
  IMAGE_CRC_CHECK_FAILED = 1,
  IMAGE_DATA_OVERFLOW = 2,
  IMAGE_MISSING_PACKETS = 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_GigEVision,
  StreamType_CameraLink,
  StreamType_CameraLinkHS,
  StreamType_CoaXPress,
  StreamType_USB3Vision,
  StreamType_Custom,
  NUMSTREAMTYPE }

  The enum definitions for TL Device nodes from the transport layer .xml files.
- enum StreamBufferCountModeEnum {
  StreamBufferCountMode_Manual,
  StreamBufferCountMode_Auto,
  NUMSTREAMBUFFERCOUNTMODE }
- enum StreamBufferHandlingModeEnum {
  StreamBufferHandlingMode_OldestFirst,
  StreamBufferHandlingMode_OldestFirstOverwrite,
  StreamBufferHandlingMode_NewestOnly,
  StreamBufferHandlingMode_NewestFirst,
  NUMSTREAMBUFFERHANDLINGMODE }
- enum DeviceTypeEnum {
  DeviceType_GigEVision,
  DeviceType_CameraLink,
  DeviceType_CameraLinkHS,
  DeviceType_CoaXPress,
  DeviceType_USB3Vision,
  DeviceType_Custom,
  NUMDEVICETYPE }
- enum DeviceAccessStatusEnum {
  DeviceAccessStatus_Unknown,
  DeviceAccessStatus_ReadWrite,
  DeviceAccessStatus_ReadOnly,
  DeviceAccessStatus_NoAccess,

DeviceAccessStatus_Busy,
DeviceAccessStatus_OpenReadWrite,
DeviceAccessStatus_OpenReadOnly,
NUMDEVICEACCESSSTATUS }

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

Functions

- class DEPRECATED_CLASS ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API A↩ VIRecorder

  Provides the functionality for the user to record images to an AVI file.
- template<class T , class B >
  bool operator== (const std::nullptr_t, const BasePtr< T, B > &rhs)

  Pointer equal.

Variables

- const uint64_t EVENT_TIMEOUT_NONE = 0

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

13.7 Spinnaker::GenApi Namespace Reference

Classes

- class AutoLock
- class BooleanNode

  Interface for string properties.
- class CategoryNode

  Interface for string properties.
- class CChunkAdapter

  Connects a chunked buffer to a node map.
- class CChunkAdapterDcam

  Connects a chunked DCAM buffer to a node map.
- class CChunkAdapterGeneric
- class CChunkAdapterGEV

  Connects a chunked DCAM buffer to a node map.
- class CChunkAdapterU3V

  Connects a chunked U3V buffer to a node map.
- class CChunkPort

  Port attachable to a chunk in a buffer.
- class CEnumerationTRef

  Interface for string properties.
- class CEventAdapter

  Delivers Events to ports.
- class CEventAdapter1394

  Distribute the events to the node map.
- class CEventAdapterGeneric

  Connects a generic event to a node map.
- class CEventAdapterGEV

  Connects a GigE Event to a node map.
- class CEventAdapterU3V

  Connects a U3V Event to a node map.
- class CEventPort

  Port attachable to an event.

- class CFeatureBag

  Bag holding streamable features of a nodetree.
- class CFloatPtr

  SmartPointer for IFloat interface pointer.
- class CGeneric_XMLLoaderParams

  Empty base class used by class CNodeMapRef as generic template argument.
- class CLock

  A lock class.
- class CLockEx

  This class is for testing purposes only.
- class CNodeCallback

  callback body instance for INode pointers
- class CNodeMapFactory

  The node map factory is used for creating node maps from camera description files.
- class CNodeMapRef

  Smartpointer for NodeMaps with create function.
- class CNodeMapRefT

  Smartpointer template for NodeMaps with create function.
- class CommandNode

  Interface for string properties.
- class Counter

  Definition of a simple Counter class.
- class CPointer

  Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.
- class CPortImpl

  Standard implementation for a port.
- class CPortWriteList

  Container holding a list of port write commands.
- class CRegisterPortImpl

  Standard implementation for a port using a register based transport layer.
- class CSelectorSet

  The set of selectors selecting a given node.
- class CTestPortStruct

  Implements a register spaces based on a C++ struct.
- class double_autovector_t

  Vector of doubles with reference counting.
- class EAccessModeClass

  Holds conversion methods for the access mode enumeration.
- class ECachingModeClass

  Holds conversion methods for the caching mode enumeration.
- class EDisplayNotationClass

  Holds conversion methods for the notation type of floats.
- class EEndianessClass

  Holds conversion methods for the endianess enumeration.
- class EGenApiSchemaVersionClass

  helper class converting EGenApiSchemaVersion from and to string
- class EInputDirectionClass

  Holds conversion methods for the notation type of floats.
- class ENameSpaceClass

  Holds conversion methods for the namespace enumeration.
- class EnumEntryNode

Interface for string properties.

- class EnumNode

  Interface for string properties.
- class ERepresentationClass

  Holds conversion methods for the representation enumeration.
- class ESignClass

  Holds conversion methods for the sign enumeration.
- class ESlopeClass

  Holds conversion methods for the converter formulas.
- class EStandardNameSpaceClass

  Holds conversion methods for the standard namespace enumeration.
- class EVisibilityClass

  Holds conversion methods for the visibility enumeration.
- class EYesNoClass

  Holds conversion methods for the standard namespace enumeration.
- class FileProtocolAdapter

  Adapter between the std::iostreambuf and the SFNC Features representing the device file system.
- class FloatNode

  Interface for string properties.
- class FloatRegNode

  Interface for string properties.
- class Function_NodeCallback

  Container for a function pointer.
- class IDevFileStreamBase
- class IDevFileStreamBuf
- class int64_autovector_t

  Vector of integers with reference counting.
- class IntegerNode

  Interface for string properties.
- class IntRegNode

  Interface for string properties.
- class Member_NodeCallback

  Container for a member function pointer.
- class Node

  class common to all nodes
- class NodeMap

  Smart pointer template for NodeMaps with create function.
- class ODevFileStreamBase
- class ODevFileStreamBuf
- class PortNode

  Interface for value properties.
- class PortRecorder

  Interface for recording write commands on a port.
- class PortReplay

  Interface for replaying write commands on a port.
- class RegisterNode

  Interface for string properties.
- class SpinTestCamera
- class StringNode

  Interface for string properties.
- class StringRegNode

  Interface for string properties.
- class ValueNode

  Interface for value properties.

Typedefs

- typedef BooleanNode CBooleanRef
- typedef CategoryNode CCategoryRef
- typedef CommandNode CCommandRef
- typedef EnumEntryNode CEnumEntryRef
- typedef EnumNode CEnumerationRef
- typedef ODevFileStreamBase< char, std::char_traits< char > > ODevFileStream
- typedef IDevFileStreamBase< char, std::char_traits< char > > IDevFileStream
- typedef FloatNode CFloatRef
- typedef node_vector NodeList_t

  a list of node references
- typedef intptr_t CallbackHandleType

  the callback handle for nodes
- typedef IntegerNode CIntegerRef
- typedef Node CNodeRef
- typedef Node CSelectorRef
- typedef NodeMap CNodeMapRef
- typedef CPointer< IBase > CBasePtr

  SmartPointer for IBase interface pointer.
- typedef CPointer< INode, IBase > CNodePtr

  SmartPointer for INode interface pointer.
- typedef CPointer< IValue > CValuePtr

  SmartPointer for IValue interface pointer.
- typedef CPointer< ICategory > CCategoryPtr

  SmartPointer for ICategory interface pointer.
- typedef CPointer< IBoolean > CBooleanPtr

  SmartPointer for IBoolean interface pointer.
- typedef CPointer< IInteger > CIntegerPtr

  SmartPointer for IInteger interface pointer.
- typedef CPointer< IString > CStringPtr

  SmartPointer for IString interface pointer.
- typedef CPointer< IRegister > CRegisterPtr

  SmartPointer for IRegister interface pointer.
- typedef CPointer< IEnumeration > CEnumerationPtr

  SmartPointer for IEnumeration interface pointer.
- typedef CPointer< IEnumEntry > CEnumEntryPtr

  SmartPointer for IEnumEntry interface pointer.
- typedef CPointer< IPort > CPortPtr

  SmartPointer for IPort interface pointer.
- typedef CPointer< IPortReplay > CPortReplayPtr

  SmartPointer for IPortReplay interface pointer.
- typedef CPointer< IPortRecorder > CPortRecorderPtr

  SmartPointer for IPortRecorder interface pointer.
- typedef CPointer< IPortWriteList, IPortWriteList > CPortWriteListPtr

  SmartPointer for IPortWriteList interface pointer.
- typedef CPointer< IChunkPort > CChunkPortPtr

  SmartPointer for IChunkPort interface pointer.
- typedef CPointer< INodeMap, INodeMap > CNodeMapPtr

  SmartPointer for INodeMap interface pointer.
- typedef CPointer< INodeMapDyn, INodeMap > CNodeMapDynPtr

  SmartPointer for INodeMapDyn interface pointer.

- typedef CPointer< IDeviceInfo, INodeMap > CDeviceInfoPtr

  SmartPointer for IDeviceInfo interface pointer.
- typedef CPointer< ISelector > CSelectorPtr

  SmartPointer for ISelector interface pointer.
- typedef CPointer< ICommand > CCommandPtr

  SmartPointer for ICommand interface pointer.
- typedef CPointer< IPortConstruct > CPortConstructPtr

  SmartPointer for IPortConstruct interface pointer.
- typedef PortNode CPortRef
- typedef PortRecorder CPortRecorderRef

  Reference to an IPortRecorder pointer.
- typedef RegisterNode CRegisterRef
- typedef StringNode CStringRef
- typedef 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,
ctInvalidatingChildren,
ctDependingNodes,
ctTerminalNodes }

  typedef for link type

- enum EIncMode {
noIncrement,
fixedIncrement,
listIncrement }

  typedef for increment mode

- enum EInputDirection {
idFrom,
idTo,
idNone }

  typedef for link type

- enum EGenApiSchemaVersion {
v1_0 = 1,
v1_1 = 2,
_Undefined = -1 }

  GenApi schema version.

Functions

- void SPINNAKER_API SET_GUID (SPIN_GUID &name, uint32_t l, uint16_t w1, uint16_t w2, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4, uint8_t b5, uint8_t b6, uint8_t b7, uint8_t b8)
- virtual void operator= (bool Value)

  Set node value.

- virtual bool GetValue (bool Verify=false, bool IgnoreCache=false) const =0

  Get node value.
- virtual bool operator() () const

  Get node value.
- virtual EYesNo CacheChunkData () const =0

  Indicates if the chunk a adapter must hold a cached version of the chunk data.
- virtual bool IsDone (bool Verify=true)=0

  Query whether the command is executed.
- virtual GenICam::gcstring GetVendorName ()=0

  Get the vendor name.
- virtual GenICam::gcstring GetToolTip ()=0

  Get tool tip.
- virtual GenICam::gcstring GetStandardNameSpace ()=0

  Get the standard name space.
- virtual void GetGenApiVersion (GenICam::Version_t &Version, uint16_t &Build)=0

  Get the version of the DLL's GenApi implementation.
- virtual void GetSchemaVersion (GenICam::Version_t &Version)=0

  Get the schema version number.
- virtual void GetDeviceVersion (GenICam::Version_t &Version)=0

  Get the version of the device description file.
- virtual GenICam::gcstring GetProductGuid ()=0

  Get the Guid describing the product.
- virtual 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_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.
- 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, GenI↩Cam::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 MaxVisiblity)

  Tests Visibility CAVE : this relies on the EVisibility enum's coding.
- EVisibility Combine (EVisibility Peter, EVisibility Paul)

  Computes which visibility the two guards allow together.
- bool IsCacheable (ECachingMode CachingMode)

  Tests Cacheability.
- ECachingMode Combine (ECachingMode Peter, ECachingMode Paul)

  Computes which CachingMode results from a combination.
- virtual INode * GetNode (const GenICam::gcstring &Name) const =0

  Retrieves the node from the central map by Name.
- virtual void InvalidateNodes () const =0

  Invalidates all nodes.
- virtual bool Connect (IPort *pPort, const GenICam::gcstring &PortName) const =0

  Connects a port to a port node with given name.
- virtual bool Connect (IPort *pPort) const =0

  Connects a port to the standard port "Device".
- virtual void Poll (int64_t ElapsedTime)=0

  Fires nodes which have a polling time.
- virtual CLock & GetLock () const =0

  Returns the lock which guards the node map.
- virtual uint64_t GetNumNodes () const =0

  Get the number of nodes in the map.
- virtual void LoadXMLFromFile (const GenICam::gcstring &FileName)=0

  Loads an XML from a file.
- virtual void LoadXMLFromFileInject (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)=0

  Loads an XML from a file with injection.
- virtual void LoadXMLFromString (const GenICam::gcstring &XMLData)=0

  Loads an XML from a string.
- virtual void LoadXMLFromStringInject (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)=0

  Loads an XML from a string with injection.
- virtual void PreprocessXMLFromFile (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xv↩Default)=0

  Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void MergeXMLFiles (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectedFileName, const GenICam::gcstring &OutputFileName)=0

Injects an XML file into a target file.

- virtual void ExtractIndependentSubtree (const GenICam::gcstring &XMLData, const GenICam::gcstring &InjectXMLData, const GenICam::gcstring &SubTreeRootNodeName, GenICam::gcstring &Extracted↩ Subtree)=0

  Extract independent subtree.
- virtual void GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions)=0

  Gets a list of supported schema versions.
- virtual void LoadXMLFromZIPFile (const GenICam::gcstring &ZipFileName)=0

  Loads an XML from a ZIP file.
- virtual void LoadXMLFromZIPData (const void ∗zipData, size_t zipSize)=0

  Loads an XML from a ZIP data buffer.
- virtual void PreprocessXMLFromZIPFile (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xv↩ Default)=0

  Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void Write (const void ∗pBuffer, int64_t Address, int64_t Length)=0

  Writes a chunk of bytes to the port.
- virtual EYesNo 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

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.
- template<class T , class B >
  bool IsWritable (const Spinnaker::GenApi::CPointer< T, B > &ptr)

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

  Checks if a node is Implemented.
- template<class T , class B >
  bool IsAvailable (const Spinnaker::GenApi::CPointer< T, B > &ptr)

  Checks if a node is Available.
- GenICam::gcstring GetInterfaceName (IBase *pBase)

  Returns the name of the main interface as string DEPRICATED, use IBase::GetPrincipalInterfaceType() instead.
- virtual void SetNumEnums (int NumEnums)=0

  sets the number of enum values

Variables

- interface SPINNAKER_API_ABSTRACT IBase

  Base interface common to all nodes.
- const uint8_t COMMAND_MAGIC = 0x42
- const uint32_t U3V_EVENT_PREFIX = 0x45563355
- const uint16_t GENCP_EVENT_CMD_ID = 0x0C00
- const size_t GENCP_COMMAND_HEADER_SIZE = sizeof(U3V_COMMAND_HEADER)
- const size_t GENCP_EVENT_BASIC_SIZE = sizeof(U3V_EVENT_MESSAGE)
- interface SPINNAKER_API_ABSTRACT IBoolean

  Interface for Boolean properties.
- interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0
- interface SPINNAKER_API_ABSTRACT ICategory

  Gives access to a category node.
- interface SPINNAKER_API_ABSTRACT IChunkPort

  Interface for ports attached to a chunk.
- interface SPINNAKER_API_ABSTRACT ICommand

  Interface for command like properties.
- interface SPINNAKER_API_ABSTRACT IDestroy

  Interface to destroy an object.
- interface SPINNAKER_API_ABSTRACT IDeviceInfo

  Interface to get information about the device (= nodemap)
- interface SPINNAKER_API_ABSTRACT IEnumEntry

  Interface of single enum value.
- interface SPINNAKER_API_ABSTRACT IEnumeration

  Interface for enumeration properties.
- template<typename EnumT >
  interface SPINNAKER_API_ABSTRACT IEnumerationT

  Interface for enumeration properties.

- template<typename EnumT >
  interface SPINNAKER_API_ABSTRACT virtual public IEnumReference

  Interface to construct an enum reference.
- interface SPINNAKER_API_ABSTRACT IFloat

  Interface for float properties.
- interface SPINNAKER_API_ABSTRACT IInteger

  Interface for integer properties.
- interface SPINNAKER_API_ABSTRACT INode

  Interface common to all nodes.
- interface SPINNAKER_API_ABSTRACT virtual public IReference

  Interface to construct a reference.
- interface SPINNAKER_API_ABSTRACT INodeMap

  Interface to access the node map.
- interface SPINNAKER_API_ABSTRACT INodeMapDyn

  Interface to access the node map.
- interface SPINNAKER_API_ABSTRACT IPort

  Interface for ports.
- interface SPINNAKER_API_ABSTRACT int64_t Address
- interface SPINNAKER_API_ABSTRACT int64_t int64_t Length = 0
- interface SPINNAKER_API IPortConstruct

  Interface for ports.
- interface SPINNAKER_API_ABSTRACT IPortWriteList
- interface SPINNAKER_API_ABSTRACT IPortReplay

  Interface for replaying write commands on a port.
- interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0
- interface SPINNAKER_API_ABSTRACT IPortRecorder

  Interface for recording write commands on a port.
- interface SPINNAKER_API_ABSTRACT IRegister

  Interface for registers.
- interface SPINNAKER_API_ABSTRACT ISelector

  Interface for groups of features selected by a single one.
- interface SPINNAKER_API_ABSTRACT ISelectorDigit

  Interface of a "digit" of the "counter" formed by the selector set.
- interface SPINNAKER_API_ABSTRACT IString

  Interface for string properties.
- interface SPINNAKER_API_ABSTRACT IValue

  Interface for value properties.
- interface SPINNAKER_API_ABSTRACT IPersistScript

  Basic interface to persist values to.

13.7.1 Typedef Documentation

13.7.1.1 IDevFileStream

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

13.7.1.2 ODevFileStream

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

13.7.2 Enumeration Type Documentation

13.7.2.1 GVCP_MESSAGE_TAGS

```
enum GVCP_MESSAGE_TAGS
```

Enumerator

TAG_EVENT_CMD	
TAG_EVENTDATA_CMD	

13.7.3 Function Documentation

13.7.3.1 PersistFeature()

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

Stores a feature.

13.7.3.2 SET_GUID()

```
void SPINNAKER_API Spinnaker::GenApi::SET_GUID (
            SPIN_GUID & name,
            uint32_t l,
            uint16_t w1,
            uint16_t w2,
            uint8_t b1,
            uint8_t b2,
            uint8_t b3,
            uint8_t b4,
            uint8_t b5,
            uint8_t b6,
            uint8_t b7,
            uint8_t b8 )
```

13.7.4 Variable Documentation

13.7.4.1 COMMAND_MAGIC

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

13.7.4.2 GENCP_COMMAND_HEADER_SIZE

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

13.7.4.3 GENCP_EVENT_BASIC_SIZE

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

13.7.4.4 GENCP_EVENT_CMD_ID

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

13.7.4.5 IPersistScript

```
interface SPINNAKER_API_ABSTRACT IPersistScript
```

Initial value:

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

Basic interface to persist values to.

13.7.4.6 U3V_EVENT_PREFIX

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

13.8 Spinnaker::GenICam Namespace Reference

Classes

- class AutoLock
- class CGlobalLock

  Named global lock which can be used over process boundaries.
- class CGlobalLockUnlocker

  Unlocks the global lock object on destruction.
- class CLock

  A lock class.
- class CLockEx

  This class is for testing purposes only.
- class gcstring
- class LockableObject

  Instance-Lock for an object.
- struct Version_t

  Version.

Functions

- SPINNAKER_API void ThrowBadAlloc ()
- std::istream & getline (std::istream &is, Spinnaker::GenICam::gcstring &str)

  STL getline.
- std::istream & getline (std::istream &is, Spinnaker::GenICam::gcstring &str, char delim)

  STL getline.
- template<typename Td , typename Ts >
  Td INTEGRAL_CAST2 (Ts s)

  This verifies at runtime if there was no loss of data if an type Ts (e.g.
- template<typename T >
  T INTEGRAL_CAST (int64_t ll)

  This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.
- SPINNAKER_API bool DoesEnvironmentVariableExist (const Spinnaker::GenICam::gcstring &Variable↩Name)

  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 unsave 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 SetGenICam↩CacheFolder().
- 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 SetGenICam↩CLProtocolFolder().

- SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)

  Stores the path of the GenICam cache folder.

- SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)

  Stores the path of the GenICam logging properties file.

- SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)

  Stores the path of the CLProtocol folder.

- SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")

  splits str input string into a list of tokens using the delimiter

- SPINNAKER_API void GetFiles (const gcstring &FileTemplate, gcstring_vector &FileNames, const bool DirectoriesOnly=false)

  Gets a list of files or directories matching a given FileTemplate.

- SPINNAKER_API gcstring GetModulePathFromFunction (void *pFunction)

  Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

13.8.1 Function Documentation

13.8.1.1 getline() [1/2]

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

STL getline.

13.8.1.2 getline() [2/2]

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

STL getline.

13.8.1.3 ThrowBadAlloc()

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

13.9 Spinnaker::Video Namespace Reference

Classes

- struct AVIOption

  Options for saving AVI files.
- struct H264Option

  Options for saving H264 files.
- struct MJPGOption

  Options for saving MJPG files.
- class SpinVideo

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

Chapter 14

Class Documentation

14.1 ActionCommandResult Struct Reference

Action Command Result.

Public Attributes

- unsigned int DeviceAddress
- ActionCommandStatus Status

14.1.1 Detailed Description

Action Command Result.

14.1.2 Member Data Documentation

14.1.2.1 DeviceAddress

```
unsigned int DeviceAddress
```

14.1.2.2 Status

```
ActionCommandStatus Status
```

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

- include/SpinnakerDefs.h

14.2 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

14.2.1 Constructor & Destructor Documentation

14.2.1.1 AdapterConfigException() [1/2]

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

14.2.1.2 AdapterConfigException() [2/2]

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

14.2.2 Member Function Documentation

14.2.2.1 ErrCode()

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

14.2.2.2 GetParamStr()

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

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

- include/AdapterConfig.h

14.3 AdapterInfo Struct Reference

Collaboration diagram for AdapterInfo:

Public Member Functions

- AdapterInfo ()

Public Attributes

- std::string adapterName
- std::string adapterGUID
- std::string adapterMACAddress
- std::string adapterDescription
- bool dhcpEnabled
- IpInfo ipInfo
- std::string receiveBuffersRegKey
- std::string transmitBuffersRegKey
- std::string jumboPacketsRegKey
- unsigned int transmitBuffers
- unsigned int receiveBuffers
- unsigned int jumboPackets
- unsigned int receiveBuffersMin
- unsigned int receiveBuffersMax
- unsigned int receiveBuffersStep
- unsigned int transmitBuffersMin
- unsigned int transmitBuffersMax
- unsigned int transmitBuffersStep
- std::vector< unsigned int > jumboPacketValidValues

14.3.1 Constructor & Destructor Documentation

14.3.1.1 AdapterInfo()

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

14.3.2 Member Data Documentation

14.3.2.1 adapterDescription

```
std::string adapterDescription
```

14.3.2.2 adapterGUID

```
std::string adapterGUID
```

14.3.2.3 adapterMACAddress

```
std::string adapterMACAddress
```

14.3.2.4 adapterName

```
std::string adapterName
```

14.3.2.5 dhcpEnabled

```
bool dhcpEnabled
```

14.3.2.6 ipInfo

```
IpInfo ipInfo
```

14.3.2.7 jumboPackets

```
unsigned int jumboPackets
```

14.3.2.8 jumboPacketsRegKey

```
std::string jumboPacketsRegKey
```

14.3.2.9 jumboPacketValidValues

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

14.3.2.10 receiveBuffers

```
unsigned int receiveBuffers
```

14.3.2.11 receiveBuffersMax

```
unsigned int receiveBuffersMax
```

14.3.2.12 receiveBuffersMin

```
unsigned int receiveBuffersMin
```

14.3.2.13 receiveBuffersRegKey

```
std::string receiveBuffersRegKey
```

14.3.2.14 receiveBuffersStep

```
unsigned int receiveBuffersStep
```

14.3.2.15 transmitBuffers

```
unsigned int transmitBuffers
```

14.3.2.16 transmitBuffersMax

```
unsigned int transmitBuffersMax
```

14.3.2.17 transmitBuffersMin

```
unsigned int transmitBuffersMin
```

14.3.2.18 transmitBuffersRegKey

```
std::string transmitBuffersRegKey
```

14.3.2.19 transmitBuffersStep

```
unsigned int transmitBuffersStep
```

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

- include/AdapterConfig.h

14.4 AttachStatistics_t Struct Reference

Delivers information about the attached chunks and nodes.

Public Attributes

- int NumChunkPorts

  Number of chunk ports found in the node map.
- int NumChunks

  Number of chunks found in the buffer.
- int NumAttachedChunks

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

14.4.1 Detailed Description

Delivers information about the attached chunks and nodes.

14.4.2 Member Data Documentation

14.4.2.1 NumAttachedChunks

```
int NumAttachedChunks
```

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

14.4.2.2 NumChunkPorts

```
int NumChunkPorts
```

Number of chunk ports found in the node map.

14.4.2.3 NumChunks

```
int NumChunks
```

Number of chunks found in the buffer.

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

- include/SpinGenApi/ChunkAdapter.h

14.5 AutoLock Class Reference

Public Member Functions

- AutoLock (CLock &lock)
- ~AutoLock ()

14.5.1 Constructor & Destructor Documentation

14.5.1.1 AutoLock()

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

14.5.1.2 ~AutoLock()

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

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

- include/SpinGenApi/GCSynch.h

14.6 AutoLock Class Reference

Public Member Functions

- AutoLock (CLock &lock)
- ~AutoLock ()

14.6.1 Constructor & Destructor Documentation

14.6.1.1 AutoLock()

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

14.6.1.2 ~AutoLock()

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

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

- include/SpinGenApi/Synch.h

14.7 AVIOption Struct Reference

Options for saving AVI files.

Public Member Functions

- AVIOption ()

Public Attributes

- float frameRate

  Frame rate of the stream.
- unsigned int reserved [256]

  Reserved for future use.

14.7.1 Detailed Description

Options for saving AVI files.

14.7.2 Constructor & Destructor Documentation

14.7.2.1 AVIOption()

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

14.7.3 Member Data Documentation

14.7.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

14.7.3.2 reserved

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

Reserved for future use.

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

- include/SpinVideoDefs.h

14.8 BasePtr< T, B > Class Template Reference

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

Public Member Functions

- BasePtr (void) throw ()

  Default constructor.
- virtual ~BasePtr (void)
- BasePtr (const BasePtr &other) throw ()

  Copy constructor.
- virtual BasePtr & operator= (const BasePtr &rhs)

  Assign INode Pointer.
- virtual BasePtr & operator= (const int nMustBeNull)
- virtual BasePtr & operator= (const long nMustBeNull)
- virtual BasePtr & operator= (const std::nullptr_t nullPtr)
- virtual operator T* (void) const

  Dereferencing.
- virtual T * operator-> (void) const

  Dereferencing.

- virtual bool IsValid () const throw ()

  True if the pointer is valid.
- virtual operator bool (void) const throw ()

  True if the pointer is valid.
- virtual bool operator== (const BasePtr &rT) const

  Pointer equal.
- virtual bool operator== (std::nullptr_t) const

  Pointer equal.
- virtual bool operator== (int nMustBeNull) const

  Pointer equal.
- virtual bool operator== (long nMustBeNull) const

  Pointer equal.
- virtual T * get () const

  get()

Protected Attributes

- PointerData * m_pT

  Underlying raw pointer.

14.8.1 Detailed Description

template<class T, class B = T>
class Spinnaker::BasePtr< T, B >

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

14.8.2 Constructor & Destructor Documentation

14.8.2.1 BasePtr() [1/2]

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

Default constructor.

14.8.2.2 ~BasePtr()

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

14.8.2.3 BasePtr() [2/2]

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

Copy constructor.

14.8.3 Member Function Documentation

14.8.3.1 get()

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

get()

14.8.3.2 IsValid()

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

True if the pointer is valid.

14.8.3.3 operator bool()

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

True if the pointer is valid.

14.8.3.4 operator T*()

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

Dereferencing.

14.8.3.5 operator->()

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

Dereferencing.

14.8.3.6 operator=() [1/4]

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

Assign INode Pointer.

14.8.3.7 operator=() [2/4]

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

14.8.3.8 operator=() [3/4]

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

14.8.3.9 operator=() [4/4]

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

14.8.3.10 operator==() [1/4]

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

Pointer equal.

14.8.3.11 operator==() [2/4]

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

Pointer equal.

14.8.3.12 operator==() [3/4]

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

Pointer equal.

14.8.3.13 operator==() [4/4]

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

Pointer equal.

14.8.4 Member Data Documentation

14.8.4.1 m_pT

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

Underlying raw pointer.

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

- include/BasePtr.h

14.9 BMPOption Struct Reference

Options for saving Bitmap image.

Public Member Functions

- BMPOption ()

Public Attributes

- bool indexedColor_8bit
- unsigned int reserved [16]

  Reserved for future use.

14.9.1 Detailed Description

Options for saving Bitmap image.

14.9.2 Constructor & Destructor Documentation

14.9.2.1 BMPOption()

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

14.9.3 Member Data Documentation

14.9.3.1 indexedColor_8bit

```
bool indexedColor_8bit
```

14.9.3.2 reserved

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

Reserved for future use.

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

- include/SpinnakerDefs.h

14.10 BooleanNode Class Reference

Interface for string properties.

Inheritance diagram for BooleanNode:

Collaboration diagram for BooleanNode:

Public Member Functions

- BooleanNode ()
- BooleanNode (std::shared_ptr< Node::NodeImpl > pBoolean)
- virtual ~BooleanNode ()
- void SetValue (bool Value, bool Verify=true)

  Set node value.
- virtual void operator= (bool Value)

  Set node value.
- bool GetValue (bool Verify=false, bool IgnoreCache=false) const

  Get node value.
- virtual void SetReference (INode *pBase)

  overload SetReference for Value

Additional Inherited Members

14.10.1 Detailed Description

Interface for string properties.

14.10.2 Constructor & Destructor Documentation

14.10.2.1 BooleanNode() [1/2]

```
BooleanNode ( )
```

14.10.2.2 BooleanNode() [2/2]

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

14.10.2.3 ~BooleanNode()

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

14.10.3 Member Function Documentation

14.10.3.1 GetValue()

```
bool GetValue (
            bool Verify = false,
            bool IgnoreCache = false ) const
```

Get 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.

14.10.3.2 operator=()

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

Set node value.

14.10.3.3 SetReference()

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

overload SetReference for Value

Reimplemented from ValueNode.

14.10.3.4 SetValue()

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

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/BooleanNode.h

14.11 Camera Class Reference

The camera object class.

Inheritance diagram for Camera:

Collaboration diagram for Camera:

Public Member Functions

- ~Camera ()
- void Init ()

Public Attributes

- GenApi::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::IEnumerationT< 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.
- 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 & LogicBlockLUTOutputValue

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

- GenApi::IInteger & LogicBlockLUTOutputValueAll

  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 matricies 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

  `Description:`
  Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module
- GenApi::IInteger & TimestampLatchValue

  Description: Returns the latched value of the timestamp counter.
- GenApi::ICommand & TimestampReset

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

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

  Description: Device temperature in degrees Celsius (C).
- GenApi::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

  Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.
- GenApi::IEnumerationT< TransferControlModeEnums > & TransferControlMode

  Description: Selects the control method for the transfers.
- GenApi::IFloat & ChunkBlackLevel

  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< FileOpenModeEnums > & FileOpenMode

  `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

  Description: Activate or deactivate the Link's heartbeat.
- GenApi::IFloat & DeviceLinkHeartbeatTimeout

  Description: Controls the current heartbeat timeout of the specific Link.
- GenApi::IFloat & DeviceLinkCommandTimeout

  Description: Indicates the command timeout of the specified Link.
- GenApi::IInteger & DeviceStreamChannelSelector

  Description: Selects the stream channel to control.
- GenApi::IEnumerationT< DeviceStreamChannelTypeEnums > & DeviceStreamChannelType

  Description: Reports the type of the stream channel.
- GenApi::IInteger & DeviceStreamChannelLink

  Description: Index of device's Link to use for streaming the specifed stream channel.
- GenApi::IEnumerationT< DeviceStreamChannelEndiannessEnums > & DeviceStreamChannelEndianness

  Description: Endianess 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

  Description: Controls if the selected component streaming is active.
- GenApi::IInteger & LinePitch

  Description: Total number of bytes between 2 successive lines.
- GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > & PixelFormatInfoSelector

  Description: Select the pixel format for which the information will be returned.
- GenApi::IInteger & PixelFormatInfoID

  Description: Returns the value used by the streaming channels to identify the selected pixel format.
- GenApi::IEnumerationT< DeinterlacingEnums > & Deinterlacing

  Description: Controls how the device performs de-interlacing.
- GenApi::IEnumerationT< ImageCompressionRateOptionEnums > & ImageCompressionRateOption

  Description: Two rate controlling options are offered: fixed bit rate or fixed quality.
- GenApi::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 > & ImageCompressionJPEG↩ FormatOption

  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 & EncoderDivider

  Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.
- GenApi::IEnumerationT< EncoderOutputModeEnums > & EncoderOutputMode

  Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.
- GenApi::IEnumerationT< EncoderStatusEnums > & EncoderStatus

  Description: Returns the motion status of the encoder.
- GenApi::IFloat & EncoderTimeout

  Description: Sets the maximum time interval between encoder counter increments before the status turns to static.
- GenApi::IEnumerationT< EncoderResetSourceEnums > & EncoderResetSource

  Description: Selects the signals that will be the source to reset the Encoder.
- GenApi::IEnumerationT< EncoderResetActivationEnums > & EncoderResetActivation

  Description: Selects the Activation mode of the Encoder Reset Source signal.
- GenApi::ICommand & EncoderReset

  Description: Does a software reset of the selected Encoder and starts it.
- GenApi::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

  Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.
- 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

  Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.
- GenApi::IInteger & EventStream0TransferBurstEndTimestamp

  Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.
- GenApi::IInteger & EventStream0TransferBurstEndFrameID

  Description: Returns the unique Identifier of the Frame (or image) that generated 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 & EventTimer1EndFrameID

  Description: Returns the unique Identifier of the Frame (or image) that generated 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 Line 0 Falling Edge type of Event.
- GenApi::IInteger & EventLine0FallingEdgeTimestamp

  Description: Returns the Timestamp of the Line 0 Falling Edge Event.
- GenApi::IInteger & EventLine0FallingEdgeFrameID

  Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.
- GenApi::IInteger & EventLine1FallingEdge

  Description: Returns the unique Identifier of the Line 1 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

  Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
- GenApi::IInteger & SourceCount

  Description: Controls or returns the number of sources supported by the device.
- GenApi::IEnumerationT< SourceSelectorEnums > & SourceSelector

  Description: Selects the source to control.
- GenApi::IEnumerationT< TransferSelectorEnums > & TransferSelector

  Description: Selects which stream transfers are currently controlled by the selected Transfer features.
- GenApi::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 > & Scan3dCoordinateSystem↩ Reference

  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 > & Scan3dCoordinateTransform↩ Selector

  Description: Sets the index to read/write a coordinate transform value.
- GenApi::IFloat & Scan3dTransformValue

  Description: Specifies the transform value selected.
- GenApi::IEnumerationT< Scan3dCoordinateReferenceSelectorEnums > & Scan3dCoordinateReference↩ Selector

  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

  Description: Returns the identifier of Region that the image comes from.
- 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 > & ChunkScan3d↩ CoordinateSystemReference

  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 > & ChunkScan3d↩CoordinateTransformSelector

  Description: Selector for transform values.

- GenApi::IFloat & ChunkScan3dTransformValue

  Description: Returns the transform value.

- GenApi::IEnumerationT< ChunkScan3dCoordinateReferenceSelectorEnums > & ChunkScan3d↩CoordinateReferenceSelector

  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 > & GevCurrentPhysicalLink↩Configuration

  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 > & GevGVCPExtendedStatus↩CodesSelector

  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< ClConfigurationEnums > & ClConfiguration

  Description: This Camera Link specific feature describes the configuration used by the camera.
- GenApi::IEnumerationT< ClTimeSlotsCountEnums > & ClTimeSlotsCount

  Description: This Camera Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.
- 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 recieved by the device on the connection selected by CxpConnectionSelector.
- GenApi::IInteger & CxpConnectionTestPacketCount

  Description: Reports the current count for test packets recieved by the device on the connection selected by Cxp↩ ConnectionSelector.
- 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 & ChunkInferenceFrameId

  Description: Returns the frame ID associated with the most recent inference result.
- 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

14.11.1 Detailed Description

The camera object class.

14.11.2 Constructor & Destructor Documentation

14.11.2.1 ~Camera()

```
~Camera ( )
```

14.11.2.2 Camera()

```
Camera ( ) [protected]
```

14.11.3 Member Function Documentation

14.11.3.1 Init()

```
void Init ( ) [virtual]
```

Implements ICameraBase.

14.11.4 Member Data Documentation

14.11.4.1 AasRoiEnable

```
GenApi::IBoolean& AasRoiEnable
```

```
Description:
```

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithm↩ Selector feature.

Visibility:

14.11.4.2 AasRoiHeight

```
GenApi::IInteger& AasRoiHeight
```

```
Description:
```

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

14.11.4.3 AasRoiOffsetX

```
GenApi::IInteger& AasRoiOffsetX
```

```
Description:
```

Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

14.11.4.4 AasRoiOffsetY

```
GenApi::IInteger& AasRoiOffsetY
```

```
Description:
```

Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

14.11.4.5 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:

14.11.4.6 AcquisitionAbort

```
GenApi::ICommand& AcquisitionAbort
```

Description: Aborts the Acquisition immediately.

This will end the capture without completing the current Frame or waiting on a trigger. If no Acquisition is in progress, the command is ignored. Visibility: Expert

14.11.4.7 AcquisitionArm

```
GenApi::ICommand& AcquisitionArm
```

Description: Arms the device before an AcquisitionStart command.

This optional command validates all the current features for consistency and prepares the device for a fast start of the Acquisition. Visibility: Expert

14.11.4.8 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:
```

14.11.4.9 AcquisitionFrameCount

```
GenApi::IInteger& AcquisitionFrameCount
```

```
Description:
```

Number of images to acquire during a multi frame acquisition.

Visibility:

14.11.4.10 AcquisitionFrameRate

```
GenApi::IFloat& AcquisitionFrameRate
```

Description: User controlled acquisition frame rate in Hertz Visibility:

14.11.4.11 AcquisitionFrameRateEnable

```
GenApi::IBoolean& AcquisitionFrameRateEnable
```

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

Visibility:

14.11.4.12 AcquisitionLineRate

```
GenApi::IFloat& AcquisitionLineRate
```

Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.

Visibility:

14.11.4.13 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:

14.11.4.14 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:

14.11.4.15 AcquisitionStart

```
GenApi::ICommand& AcquisitionStart
```

Description: This command starts the acquisition of images.

Visibility:

14.11.4.16 AcquisitionStatus

```
GenApi::IBoolean& AcquisitionStatus
```

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.

Visibility: Expert

14.11.4.17 AcquisitionStatusSelector

```
GenApi::IEnumerationT<AcquisitionStatusSelectorEnums>& AcquisitionStatusSelector
```

Description: Selects the internal acquisition signal to read using AcquisitionStatus.

Visibility: Expert

14.11.4.18 AcquisitionStop

```
GenApi::ICommand& AcquisitionStop
```

Description: This command stops the acquisition of images.

Visibility:

14.11.4.19 ActionDeviceKey

```
GenApi::IInteger& ActionDeviceKey
```

Description: Provides the device key that allows the device to check the validity of action commands.

The device internal assertion of an action signal is only authorized if the ActionDeviceKey and the action device key value in the protocol message are equal. Visibility: Guru

14.11.4.20 ActionGroupKey

```
GenApi::IInteger& ActionGroupKey
```

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

14.11.4.21 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

14.11.4.22 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

14.11.4.23 ActionSelector

```
GenApi::IInteger& ActionSelector
```

Description: Selects to which Action Signal further Action settings apply.

Visibility: Guru

14.11.4.24 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

14.11.4.25 AdaptiveCompressionEnable

```
GenApi::IBoolean& AdaptiveCompressionEnable
```

Description: Controls whether lossless compression adapts to the image content.

If disabled, a fixed encoding table is used. Visibility:

14.11.4.26 AdcBitDepth

```
GenApi::IEnumerationT<AdcBitDepthEnums>& AdcBitDepth
```

```
Description:
```

Selects which ADC bit depth to use.

A higher ADC bit depth results in better image quality but slower maximum frame rate.

```
Visibility:
```

14.11.4.27 aPAUSEMACCtrlFramesReceived

```
GenApi::IInteger& aPAUSEMACCtrlFramesReceived
```

Description: Reports the number of received PAUSE frames.

Visibility: Guru

14.11.4.28 aPAUSEMACCtrlFramesTransmitted

```
GenApi::IInteger& aPAUSEMACCtrlFramesTransmitted
```

Description: Reports the number of transmitted PAUSE frames.

Visibility: Guru

14.11.4.29 AutoAlgorithmSelector

```
GenApi::IEnumerationT<AutoAlgorithmSelectorEnums>& AutoAlgorithmSelector
```

```
Description:
```

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

Visibility:

14.11.4.30 AutoExposureControlLoopDamping

```
GenApi::IFloat& AutoExposureControlLoopDamping
```

```
Description:
```

It controls how fast the exposure and gain get settled.

If the value is too small, it may cause the system to be unstable. Range is from 0.0 to 1.0. Default = 0.2.

Visibility:

14.11.4.31 AutoExposureControlPriority

```
GenApi::IEnumerationT<AutoExposureControlPriorityEnums>& AutoExposureControlPriority
```

```
Description:
```

Selects whether to adjust gain or exposure first.

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

Visibility:

14.11.4.32 AutoExposureEVCompensation

```
GenApi::IFloat& AutoExposureEVCompensation
```

```
Description:
```

The EV compensation value used in the exposure compensation.

This allows you to adjust the resultant image intensity with one control. A positive value makes the image brighter. A negative value makes the image darker. Range from -3 to 3 with a step of 1/3. Default = 0.

```
Visibility:
```

14.11.4.33 AutoExposureExposureTimeLowerLimit

```
GenApi::IFloat& AutoExposureExposureTimeLowerLimit
```

```
Description:
```

The smallest exposure time that auto exposure can set.

Visibility:

14.11.4.34 AutoExposureExposureTimeUpperLimit

```
GenApi::IFloat& AutoExposureExposureTimeUpperLimit
```

```
Description:
```

The largest exposure time that auto exposure can set.

Visibility:

14.11.4.35 AutoExposureGainLowerLimit

```
GenApi::IFloat& AutoExposureGainLowerLimit
```

```
Description:
```

The smallest gain that auto exposure can set.

Visibility:

14.11.4.36 AutoExposureGainUpperLimit

```
GenApi::IFloat& AutoExposureGainUpperLimit
```

Description:

The largest gain that auto exposure can set.

Visibility:

14.11.4.37 AutoExposureGreyValueLowerLimit

```
GenApi::IFloat& AutoExposureGreyValueLowerLimit
```

Description:

The lowest value in percentage that the target mean may reach.

Visibility:

14.11.4.38 AutoExposureGreyValueUpperLimit

```
GenApi::IFloat& AutoExposureGreyValueUpperLimit
```

Description:

The highest value in percentage that the target mean may reach.

Visibility:

14.11.4.39 AutoExposureLightingMode

```
GenApi::IEnumerationT<AutoExposureLightingModeEnums>& AutoExposureLightingMode
```

Description:

Selects a lighting mode: Backlight, Frontlight or Normal (default).

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

Visibility:

14.11.4.40 AutoExposureMeteringMode

```
GenApi::IEnumerationT<AutoExposureMeteringModeEnums>& AutoExposureMeteringMode
```

Description:

Selects a metering mode: average, spot, or partial metering.

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

Visibility:

14.11.4.41 AutoExposureTargetGreyValue

```
GenApi::IFloat& AutoExposureTargetGreyValue
```

Description:

This is the user-specified target grey level (image mean) to apply to the current image.

Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

14.11.4.42 AutoExposureTargetGreyValueAuto

```
GenApi::IEnumerationT<AutoExposureTargetGreyValueAutoEnums>& AutoExposureTargetGreyValueAuto
```

Description:

This indicates whether the target image grey level is automatically set by the camera or manually set by the user.

Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

14.11.4.43 BalanceRatio

```
GenApi::IFloat& BalanceRatio
```

```
Description:
```

Controls the balance ratio of the selected color relative to green.

Used for white balancing.

```
Visibility:
```

14.11.4.44 BalanceRatioSelector

```
GenApi::IEnumerationT<BalanceRatioSelectorEnums>& BalanceRatioSelector
```

```
Description:
```

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

Visibility:

14.11.4.45 BalanceWhiteAuto

```
GenApi::IEnumerationT<BalanceWhiteAutoEnums>& BalanceWhiteAuto
```

```
Description:
```

White Balance compensates for color shifts caused by different lighting conditions.

It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

```
Visibility:
```

14.11.4.46 BalanceWhiteAutoDamping

```
GenApi::IFloat& BalanceWhiteAutoDamping
```

```
Description:
```

Controls how quickly 'BalanceWhiteAuto' adjusts the values for Red and Blue BalanceRatio in response to changing conditions.

Higher damping means the changes are more gradual.

```
Visibility:
```

14.11.4.47 BalanceWhiteAutoLowerLimit

```
GenApi::IFloat& BalanceWhiteAutoLowerLimit
```

```
Description:
```

Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.

Visibility:

14.11.4.48 BalanceWhiteAutoProfile

```
GenApi::IEnumerationT<BalanceWhiteAutoProfileEnums>& BalanceWhiteAutoProfile
```

Description: Selects the profile used by BalanceWhiteAuto.

Visibility:

14.11.4.49 BalanceWhiteAutoUpperLimit

```
GenApi::IFloat& BalanceWhiteAutoUpperLimit
```

```
Description:
```

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

Visibility:

14.11.4.50 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:
```

14.11.4.51 BinningHorizontalMode

```
GenApi::IEnumerationT<BinningHorizontalModeEnums>& BinningHorizontalMode
```

Description: Visibility:

14.11.4.52 BinningSelector

```
GenApi::IEnumerationT<BinningSelectorEnums>& BinningSelector
```

```
Description:
```

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

Visibility:

14.11.4.53 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:
```

14.11.4.54 BinningVerticalMode

```
GenApi::IEnumerationT<BinningVerticalModeEnums>& BinningVerticalMode
```

Description: Visibility:

14.11.4.55 BlackLevel

```
GenApi::IFloat& BlackLevel
```

```
Description:
```

Controls the offset of the video signal in percent.

Visibility:

14.11.4.56 BlackLevelAuto

```
GenApi::IEnumerationT<BlackLevelAutoEnums>& BlackLevelAuto
```

Description: Controls the mode for automatic black level adjustment.

The exact algorithm used to implement this adjustment is device-specific. Visibility: Expert

14.11.4.57 BlackLevelAutoBalance

```
GenApi::IEnumerationT<BlackLevelAutoBalanceEnums>& BlackLevelAutoBalance
```

Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.

The black level coefficients of each channel are adjusted so they are matched. Visibility: Expert

14.11.4.58 BlackLevelClampingEnable

```
GenApi::IBoolean& BlackLevelClampingEnable
```

```
Description:
```

Enable the black level auto clamping feature which performing dark current compensation.

Visibility:

14.11.4.59 BlackLevelRaw

```
GenApi::IInteger& BlackLevelRaw
```

```
Description:
```

Controls the offset of the video signal in camera specific units.

Visibility:

14.11.4.60 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:

14.11.4.61 ChunkBlackLevel

```
GenApi::IFloat& ChunkBlackLevel
```

Description: Returns the black level used to capture the image.

Visibility:

14.11.4.62 ChunkBlackLevelSelector

```
GenApi::IEnumerationT<ChunkBlackLevelSelectorEnums>& ChunkBlackLevelSelector
```

Description: Selects which black level to retrieve Visibility:

14.11.4.63 ChunkCounterSelector

```
GenApi::IEnumerationT<ChunkCounterSelectorEnums>& ChunkCounterSelector
```

Description: Selects which counter to retrieve data from.

Visibility: Expert

14.11.4.64 ChunkCounterValue

```
GenApi::IInteger& ChunkCounterValue
```

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

14.11.4.65 ChunkCRC

```
GenApi::IInteger& ChunkCRC
```

Description: Returns the CRC of the image payload.

Visibility:

14.11.4.66 ChunkEnable

```
GenApi::IBoolean& ChunkEnable
```

Description: Enables the inclusion of the selected Chunk data in the payload of the image.

Visibility:

14.11.4.67 ChunkEncoderSelector

```
GenApi::IEnumerationT<ChunkEncoderSelectorEnums>& ChunkEncoderSelector
```

Description: Selects which Encoder to retrieve data from.

Visibility: Expert

14.11.4.68 ChunkEncoderStatus

```
GenApi::IEnumerationT<ChunkEncoderStatusEnums>& ChunkEncoderStatus
```

Description: Returns the motion status of the selected encoder.

Visibility: Expert

14.11.4.69 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

14.11.4.70 ChunkExposureEndLineStatusAll

```
GenApi::IInteger& ChunkExposureEndLineStatusAll
```

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

14.11.4.71 ChunkExposureTime

```
GenApi::IFloat& ChunkExposureTime
```

Description: Returns the exposure time used to capture the image.

Visibility:

14.11.4.72 ChunkExposureTimeSelector

```
GenApi::IEnumerationT<ChunkExposureTimeSelectorEnums>& ChunkExposureTimeSelector
```

Description: Selects which exposure time is read by the ChunkExposureTime feature.

Visibility: Expert

14.11.4.73 ChunkFrameID

```
GenApi::IInteger& ChunkFrameID
```

Description: Returns the image frame ID.

Visibility:

14.11.4.74 ChunkGain

```
GenApi::IFloat& ChunkGain
```

Description: Returns the gain used to capture the image.

Visibility:

14.11.4.75 ChunkGainSelector

```
GenApi::IEnumerationT<ChunkGainSelectorEnums>& ChunkGainSelector
```

Description: Selects which gain to retrieve Visibility:

14.11.4.76 ChunkHeight

```
GenApi::IInteger& ChunkHeight
```

Description: Returns the height of the image included in the payload.

Visibility:

14.11.4.77 ChunkImage

```
GenApi::IInteger& ChunkImage
```

Description: Returns the image payload.

Visibility:

14.11.4.78 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

14.11.4.79 ChunkInferenceBoundingBoxResult

```
GenApi::IRegister& ChunkInferenceBoundingBoxResult
```

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

14.11.4.80 ChunkInferenceConfidence

```
GenApi::IFloat& ChunkInferenceConfidence
```

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

14.11.4.81 ChunkInferenceFrameId

```
GenApi::IInteger& ChunkInferenceFrameId
```

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

14.11.4.82 ChunkInferenceResult

```
GenApi::IInteger& ChunkInferenceResult
```

Description: Returns the chunk data inference result.

Visibility: Expert

14.11.4.83 ChunkLinePitch

```
GenApi::IInteger& ChunkLinePitch
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

14.11.4.84 ChunkLineStatusAll

```
GenApi::IInteger& ChunkLineStatusAll
```

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

14.11.4.85 ChunkModeActive

```
GenApi::IBoolean& ChunkModeActive
```

Description: Activates the inclusion of Chunk data in the payload of the image.

Visibility:

14.11.4.86 ChunkOffsetX

```
GenApi::IInteger& ChunkOffsetX
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

14.11.4.87 ChunkOffsetY

```
GenApi::IInteger& ChunkOffsetY
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

14.11.4.88 ChunkPartSelector

```
GenApi::IInteger& ChunkPartSelector
```

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

14.11.4.89 ChunkPixelDynamicRangeMax

```
GenApi::IInteger& ChunkPixelDynamicRangeMax
```

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

14.11.4.90 ChunkPixelDynamicRangeMin

```
GenApi::IInteger& ChunkPixelDynamicRangeMin
```

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

14.11.4.91 ChunkPixelFormat

```
GenApi::IEnumerationT<ChunkPixelFormatEnums>& ChunkPixelFormat
```

Description: Format of the pixel provided by the camera Visibility:

14.11.4.92 ChunkRegionID

```
GenApi::IEnumerationT<ChunkRegionIDEnums>& ChunkRegionID
```

Description: Returns the identifier of Region that the image comes from.

Visibility: Expert

14.11.4.93 ChunkScan3dAxisMax

```
GenApi::IFloat& ChunkScan3dAxisMax
```

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

14.11.4.94 ChunkScan3dAxisMin

```
GenApi::IFloat& ChunkScan3dAxisMin
```

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

14.11.4.95 ChunkScan3dCoordinateOffset

```
GenApi::IFloat& ChunkScan3dCoordinateOffset
```

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

14.11.4.96 ChunkScan3dCoordinateReferenceSelector

```
GenApi::IEnumerationT<ChunkScan3dCoordinateReferenceSelectorEnums>& ChunkScan3dCoordinate↩
ReferenceSelector
```

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

Visibility: Expert

14.11.4.97 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

14.11.4.98 ChunkScan3dCoordinateScale

```
GenApi::IFloat& ChunkScan3dCoordinateScale
```

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

14.11.4.99 ChunkScan3dCoordinateSelector

```
GenApi::IEnumerationT<ChunkScan3dCoordinateSelectorEnums>& ChunkScan3dCoordinateSelector
```

Description: Selects which Coordinate to retrieve data from.

Visibility: Expert

14.11.4.100 ChunkScan3dCoordinateSystem

```
GenApi::IEnumerationT<ChunkScan3dCoordinateSystemEnums>& ChunkScan3dCoordinateSystem
```

Description: Returns the Coordinate System of the image included in the payload.

Visibility: Expert

14.11.4.101 ChunkScan3dCoordinateSystemReference

```
GenApi::IEnumerationT<ChunkScan3dCoordinateSystemReferenceEnums>& ChunkScan3dCoordinate↩
SystemReference
```

Description: Returns the Coordinate System Position of the image included in the payload.

Visibility: Expert

14.11.4.102 ChunkScan3dCoordinateTransformSelector

```
GenApi::IEnumerationT<ChunkScan3dCoordinateTransformSelectorEnums>& ChunkScan3dCoordinate↩
TransformSelector
```

Description: Selector for transform values.

Visibility: Expert

14.11.4.103 ChunkScan3dDistanceUnit

```
GenApi::IEnumerationT<ChunkScan3dDistanceUnitEnums>& ChunkScan3dDistanceUnit
```

Description: Returns the Distance Unit of the payload image.

Visibility: Expert

14.11.4.104 ChunkScan3dInvalidDataFlag

```
GenApi::IBoolean& ChunkScan3dInvalidDataFlag
```

Description: Returns if a specific non-valid data flag is used in the data stream.

Visibility: Expert

14.11.4.105 ChunkScan3dInvalidDataValue

```
GenApi::IFloat& ChunkScan3dInvalidDataValue
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

14.11.4.106 ChunkScan3dOutputMode

```
GenApi::IEnumerationT<ChunkScan3dOutputModeEnums>& ChunkScan3dOutputMode
```

Description: Returns the Calibrated Mode of the payload image.

Visibility: Expert

14.11.4.107 ChunkScan3dTransformValue

```
GenApi::IFloat& ChunkScan3dTransformValue
```

Description: Returns the transform value.

Visibility: Expert

14.11.4.108 ChunkScanLineSelector

```
GenApi::IInteger& ChunkScanLineSelector
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

14.11.4.109 ChunkSelector

```
GenApi::IEnumerationT<ChunkSelectorEnums>& ChunkSelector
```

Description: Selects which chunk data to enable or disable.

Visibility:

14.11.4.110 ChunkSequencerSetActive

```
GenApi::IInteger& ChunkSequencerSetActive
```

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

14.11.4.111 ChunkSerialData

```
GenApi::IString& ChunkSerialData
```

Description: Returns the serial data that was received.

Visibility:

14.11.4.112 ChunkSerialDataLength

```
GenApi::IInteger& ChunkSerialDataLength
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

14.11.4.113 ChunkSerialReceiveOverflow

```
GenApi::IBoolean& ChunkSerialReceiveOverflow
```

Description: Returns the status of the chunk serial receive overflow.

Visibility:

14.11.4.114 ChunkSourceID

```
GenApi::IEnumerationT<ChunkSourceIDEnums>& ChunkSourceID
```

Description: Returns the identifier of Source that the image comes from.

Visibility: Expert

14.11.4.115 ChunkStreamChannelID

```
GenApi::IInteger& ChunkStreamChannelID
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

14.11.4.116 ChunkTimerSelector

```
GenApi::IEnumerationT<ChunkTimerSelectorEnums>& ChunkTimerSelector
```

Description: Selects which Timer to retrieve data from.

Visibility: Expert

14.11.4.117 ChunkTimerValue

```
GenApi::IFloat& ChunkTimerValue
```

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

14.11.4.118 ChunkTimestamp

```
GenApi::IInteger& ChunkTimestamp
```

Description: Returns the Timestamp of the image.

Visibility:

14.11.4.119 ChunkTimestampLatchValue

```
GenApi::IInteger& ChunkTimestampLatchValue
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

14.11.4.120 ChunkTransferBlockID

```
GenApi::IInteger& ChunkTransferBlockID
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

14.11.4.121 ChunkTransferQueueCurrentBlockCount

```
GenApi::IInteger& ChunkTransferQueueCurrentBlockCount
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

14.11.4.122 ChunkTransferStreamID

```
GenApi::IEnumerationT<ChunkTransferStreamIDEnums>& ChunkTransferStreamID
```

Description: Returns identifier of the stream that generated this block.

Visibility: Expert

14.11.4.123 ChunkWidth

```
GenApi::IInteger& ChunkWidth
```

Description: Returns the width of the image included in the payload.

Visibility:

14.11.4.124 ClConfiguration

```
GenApi::IEnumerationT<ClConfigurationEnums>& ClConfiguration
```

Description: This Camera Link specific feature describes the configuration used by the camera.

It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitizationTaps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera. Visibility: Beginner

14.11.4.125 ClTimeSlotsCount

```
GenApi::IEnumerationT<ClTimeSlotsCountEnums>& ClTimeSlotsCount
```

Description: This Camera Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Visibility: Expert

14.11.4.126 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:
```

14.11.4.127 ColorTransformationSelector

```
GenApi::IEnumerationT<ColorTransformationSelectorEnums>& ColorTransformationSelector
```

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

Visibility:

14.11.4.128 ColorTransformationValue

```
GenApi::IFloat& ColorTransformationValue
```

```
Description:
```

Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

Visibility:

14.11.4.129 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:

14.11.4.130 CompressionRatio

```
GenApi::IFloat& CompressionRatio
```

Description: Reports the ratio between the uncompressed image size and compressed image size.

Visibility:

14.11.4.131 CounterDelay

```
GenApi::IInteger& CounterDelay
```

Description: Sets the delay (or number of events) before the CounterStart event is generated.

Visibility:

14.11.4.132 CounterDuration

```
GenApi::IInteger& CounterDuration
```

Description: Sets the duration (or number of events) before the CounterEnd event is generated.

Visibility:

14.11.4.133 CounterEventActivation

```
GenApi::IEnumerationT<CounterEventActivationEnums>& CounterEventActivation
```

Description: Selects the activation mode of the event to increment the Counter.

Visibility:

14.11.4.134 CounterEventSource

```
GenApi::IEnumerationT<CounterEventSourceEnums>& CounterEventSource
```

Description: Selects the event that will increment the counter Visibility:

14.11.4.135 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

14.11.4.136 CounterResetActivation

```
GenApi::IEnumerationT<CounterResetActivationEnums>& CounterResetActivation
```

Description: Selects the Activation mode of the Counter Reset Source signal.

Visibility:

14.11.4.137 CounterResetSource

```
GenApi::IEnumerationT<CounterResetSourceEnums>& CounterResetSource
```

Description: Selects the signal that will be the source to reset the Counter.

Visibility:

14.11.4.138 CounterSelector

```
GenApi::IEnumerationT<CounterSelectorEnums>& CounterSelector
```

Description: Selects which counter to configure Visibility:

14.11.4.139 CounterStatus

```
GenApi::IEnumerationT<CounterStatusEnums>& CounterStatus
```

Description: Returns the current status of the Counter.

Visibility:

14.11.4.140 CounterTriggerActivation

```
GenApi::IEnumerationT<CounterTriggerActivationEnums>& CounterTriggerActivation
```

Description: Selects the activation mode of the trigger to start the Counter.

Visibility:

14.11.4.141 CounterTriggerSource

```
GenApi::IEnumerationT<CounterTriggerSourceEnums>& CounterTriggerSource
```

Description: Selects the source of the trigger to start the counter Visibility:

14.11.4.142 CounterValue

```
GenApi::IInteger& CounterValue
```

Description: Current counter value Visibility:

14.11.4.143 CounterValueAtReset

```
GenApi::IInteger& CounterValueAtReset
```

Description: Value of the selected Counter when it was reset by a trigger.

Visibility:

14.11.4.144 CxpConnectionSelector

```
GenApi::IInteger& CxpConnectionSelector
```

Description: Selects the CoaXPress physical connection to control.

Visibility: Expert

14.11.4.145 CxpConnectionTestErrorCount

```
GenApi::IInteger& CxpConnectionTestErrorCount
```

Description: Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

14.11.4.146 CxpConnectionTestMode

```
GenApi::IEnumerationT<CxpConnectionTestModeEnums>& CxpConnectionTestMode
```

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

Visibility: Expert

14.11.4.147 CxpConnectionTestPacketCount

```
GenApi::IInteger& CxpConnectionTestPacketCount
```

Description: Reports the current count for test packets recieved by the device on the connection selected by Cxp↩ ConnectionSelector.

Visibility: Expert

14.11.4.148 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

14.11.4.149 CxpLinkConfigurationPreferred

```
GenApi::IEnumerationT<CxpLinkConfigurationPreferredEnums>& CxpLinkConfigurationPreferred
```

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

Visibility: Expert

14.11.4.150 CxpLinkConfigurationStatus

```
GenApi::IEnumerationT<CxpLinkConfigurationStatusEnums>& CxpLinkConfigurationStatus
```

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

Visibility: Beginner

14.11.4.151 CxpPoCxpAuto

```
GenApi::ICommand& CxpPoCxpAuto
```

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

14.11.4.152 CxpPoCxpStatus

```
GenApi::IEnumerationT<CxpPoCxpStatusEnums>& CxpPoCxpStatus
```

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

Visibility: Expert

14.11.4.153 CxpPoCxpTripReset

```
GenApi::ICommand& CxpPoCxpTripReset
```

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

Visibility: Expert

14.11.4.154 CxpPoCxpTurnOff

```
GenApi::ICommand& CxpPoCxpTurnOff
```

Description: Disable Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

14.11.4.155 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:
```

14.11.4.156 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:
```

14.11.4.157 DecimationSelector

```
GenApi::IEnumerationT<DecimationSelectorEnums>& DecimationSelector
```

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

Visibility:

14.11.4.158 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:
```

14.11.4.159 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:
```

14.11.4.160 DefectCorrectionMode

```
GenApi::IEnumerationT<DefectCorrectionModeEnums>& DefectCorrectionMode
```

Description: Controls the method used for replacing defective pixels.

Visibility:

14.11.4.161 DefectCorrectStaticEnable

```
GenApi::IBoolean& DefectCorrectStaticEnable
```

Description: Enables/Disables table-based defective pixel correction.

Visibility:

14.11.4.162 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:

14.11.4.163 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:
```

14.11.4.164 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:
```

14.11.4.165 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:

14.11.4.166 DefectTableIndex

```
GenApi::IInteger& DefectTableIndex
```

```
Description:
```

Controls the offset of the element to access in the defective pixel location table.

Visibility:

14.11.4.167 DefectTablePixelCount

```
GenApi::IInteger& DefectTablePixelCount
```

```
Description:
```

The number of defective pixel locations in the current table.

Visibility:

14.11.4.168 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:

14.11.4.169 Deinterlacing

```
GenApi::IEnumerationT<DeinterlacingEnums>& Deinterlacing
```

Description: Controls how the device performs de-interlacing.

Visibility: Beginner

14.11.4.170 DeviceCharacterSet

```
GenApi::IEnumerationT<DeviceCharacterSetEnums>& DeviceCharacterSet
```

```
Description:
```

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

Visibility:

14.11.4.171 DeviceClockFrequency

```
GenApi::IFloat& DeviceClockFrequency
```

Description: Returns the frequency of the selected Clock.

Visibility: Expert

14.11.4.172 DeviceClockSelector

```
GenApi::IEnumerationT<DeviceClockSelectorEnums>& DeviceClockSelector
```

Description: Selects the clock frequency to access from the device.

Visibility: Expert

14.11.4.173 DeviceConnectionSelector

```
GenApi::IInteger& DeviceConnectionSelector
```

Description: Selects which Connection of the device to control.

Visibility: Beginner

14.11.4.174 DeviceConnectionSpeed

```
GenApi::IInteger& DeviceConnectionSpeed
```

Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.

14.11.4.175 DeviceConnectionStatus

```
GenApi::IEnumerationT<DeviceConnectionStatusEnums>& DeviceConnectionStatus
```

Description: Indicates the status of the specified Connection.

Visibility: Expert

14.11.4.176 DeviceEventChannelCount

```
GenApi::IInteger& DeviceEventChannelCount
```

```
Description:
```

Indicates the number of event channels supported by the device.

Visibility:

14.11.4.177 DeviceFamilyName

```
GenApi::IString& DeviceFamilyName
```

Description: Identifier of the product family of the device.

Visibility: Beginner

14.11.4.178 DeviceFeaturePersistenceEnd

```
GenApi::ICommand& DeviceFeaturePersistenceEnd
```

Description: Indicate to the device the end of feature persistence.

Visibility: Guru

14.11.4.179 DeviceFeaturePersistenceStart

```
GenApi::ICommand& DeviceFeaturePersistenceStart
```

Description: Indicate to the device and GenICam XML to get ready for persisting of all streamable features.

Visibility: Guru

14.11.4.180 DeviceFirmwareVersion

```
GenApi::IString& DeviceFirmwareVersion
```

Description: Version of the firmware on the device.

Visibility:

14.11.4.181 DeviceGenCPVersionMajor

```
GenApi::IInteger& DeviceGenCPVersionMajor
```

Description: Major version of the GenCP protocol supported by the device.

Visibility: Beginner

14.11.4.182 DeviceGenCPVersionMinor

```
GenApi::IInteger& DeviceGenCPVersionMinor
```

Description: Minor version of the GenCP protocol supported by the device.

Visibility: Beginner

14.11.4.183 DeviceID

```
GenApi::IString& DeviceID
```

Description: Device identifier (serial number).

Visibility:

14.11.4.184 DeviceIndicatorMode

```
GenApi::IEnumerationT<DeviceIndicatorModeEnums>& DeviceIndicatorMode
```

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

Visibility:

14.11.4.185 DeviceLinkBandwidthReserve

```
GenApi::IFloat& DeviceLinkBandwidthReserve
```

```
Description:
```

Percentage of streamed data bandwidth reserved for packet resend.

Visibility:

14.11.4.186 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

14.11.4.187 DeviceLinkConnectionCount

```
GenApi::IInteger& DeviceLinkConnectionCount
```

Description: Returns the number of physical connection of the device used by a particular Link.

Visibility: Beginner

14.11.4.188 DeviceLinkCurrentThroughput

```
GenApi::IInteger& DeviceLinkCurrentThroughput
```

Description: Current bandwidth of streamed data.

Visibility:

14.11.4.189 DeviceLinkHeartbeatMode

```
GenApi::IEnumerationT<DeviceLinkHeartbeatModeEnums>& DeviceLinkHeartbeatMode
```

Description: Activate or deactivate the Link's heartbeat.

Visibility: Expert

14.11.4.190 DeviceLinkHeartbeatTimeout

```
GenApi::IFloat& DeviceLinkHeartbeatTimeout
```

Description: Controls the current heartbeat timeout of the specific Link.

Visibility: Guru

14.11.4.191 DeviceLinkSelector

```
GenApi::IInteger& DeviceLinkSelector
```

Description: Selects which Link of the device to control.

Visibility: Beginner

14.11.4.192 DeviceLinkSpeed

```
GenApi::IInteger& DeviceLinkSpeed
```

```
Description:
```

Indicates the speed of transmission negotiated on the specified Link.

(Bps)

```
Visibility:
```

14.11.4.193 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:
```

14.11.4.194 DeviceLinkThroughputLimitMode

```
GenApi::IEnumerationT<DeviceLinkThroughputLimitModeEnums>& DeviceLinkThroughputLimitMode
```

Description: Controls if the DeviceLinkThroughputLimit is active.

When disabled, lower level TL specific features are expected to control the throughput. When enabled, Device↩LinkThroughputLimit controls the overall throughput. Visibility: Expert

14.11.4.195 DeviceManifestEntrySelector

```
GenApi::IInteger& DeviceManifestEntrySelector
```

Description: Selects the manifest entry to reference.

Visibility: Guru

14.11.4.196 DeviceManifestPrimaryURL

```
GenApi::IString& DeviceManifestPrimaryURL
```

Description: Indicates the first URL to the GenICam XML device description file of the selected manifest entry.

Visibility: Guru

14.11.4.197 DeviceManifestSchemaMajorVersion

```
GenApi::IInteger& DeviceManifestSchemaMajorVersion
```

Description: Indicates the major version number of the schema file of the selected manifest entry.

Visibility: Guru

14.11.4.198 DeviceManifestSchemaMinorVersion

```
GenApi::IInteger& DeviceManifestSchemaMinorVersion
```

Description: Indicates the minor version number of the schema file of the selected manifest entry.

Visibility: Guru

14.11.4.199 DeviceManifestSecondaryURL

```
GenApi::IString& DeviceManifestSecondaryURL
```

Description: Indicates the second URL to the GenICam XML device description file of the selected manifest entry.

Visibility: Guru

14.11.4.200 DeviceManifestXMLMajorVersion

```
GenApi::IInteger& DeviceManifestXMLMajorVersion
```

Description: Indicates the major version number of the GenICam XML file of the selected manifest entry.

Visibility: Guru

14.11.4.201 DeviceManifestXMLMinorVersion

```
GenApi::IInteger& DeviceManifestXMLMinorVersion
```

Description: Indicates the minor version number of the GenICam XML file of the selected manifest entry.

Visibility: Guru

14.11.4.202 DeviceManifestXMLSubMinorVersion

```
GenApi::IInteger& DeviceManifestXMLSubMinorVersion
```

Description: Indicates the subminor version number of the GenICam XML file of the selected manifest entry.

Visibility: Guru

14.11.4.203 DeviceManufacturerInfo

```
GenApi::IString& DeviceManufacturerInfo
```

Description: Manufacturer information about the device.

Visibility:

14.11.4.204 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:
```

14.11.4.205 DeviceModelName

```
GenApi::IString& DeviceModelName
```

Description: Model of the device.

Visibility:

14.11.4.206 DevicePowerSupplySelector

```
GenApi::IEnumerationT<DevicePowerSupplySelectorEnums>& DevicePowerSupplySelector
```

```
Description:
```

Selects the power supply source to control or read.

Visibility:

14.11.4.207 DeviceRegistersCheck

```
GenApi::ICommand& DeviceRegistersCheck
```

Description: Perform the validation of the current register set for consistency.

This will update the DeviceRegistersValid flag. Visibility: Expert

14.11.4.208 DeviceRegistersEndianness

```
GenApi::IEnumerationT<DeviceRegistersEndiannessEnums>& DeviceRegistersEndianness
```

Description: Endianess of the registers of the device.

Visibility:

14.11.4.209 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

14.11.4.210 DeviceRegistersStreamingStart

```
GenApi::ICommand& DeviceRegistersStreamingStart
```

Description: Prepare the device for registers streaming without checking for consistency.

Visibility: Guru

14.11.4.211 DeviceRegistersValid

```
GenApi::IBoolean& DeviceRegistersValid
```

Description: Returns if the current register set is valid and consistent.

Visibility: Expert

14.11.4.212 DeviceReset

```
GenApi::ICommand& DeviceReset
```

Description: This is a command that immediately resets and reboots the device.

Visibility:

14.11.4.213 DeviceScanType

```
GenApi::IEnumerationT<DeviceScanTypeEnums>& DeviceScanType
```

Description: Scan type of the sensor of the device.

Visibility:

14.11.4.214 DeviceSerialNumber

```
GenApi::IString& DeviceSerialNumber
```

```
 Description:
```

Device's serial number.

This string is a unique identifier of the device.

```
Visibility:
```

14.11.4.215 DeviceSerialPortBaudRate

```
GenApi::IEnumerationT<DeviceSerialPortBaudRateEnums>& DeviceSerialPortBaudRate
```

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

Visibility: Expert

14.11.4.216 DeviceSerialPortSelector

```
GenApi::IEnumerationT<DeviceSerialPortSelectorEnums>& DeviceSerialPortSelector
```

Description: Selects which serial port of the device to control.

Visibility: Expert

14.11.4.217 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

14.11.4.218 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

14.11.4.219 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

14.11.4.220 DeviceStreamChannelCount

```
GenApi::IInteger& DeviceStreamChannelCount
```

```
Description:
```

Indicates the number of streaming channels supported by the device.

Visibility:

14.11.4.221 DeviceStreamChannelEndianness

```
GenApi::IEnumerationT<DeviceStreamChannelEndiannessEnums>& DeviceStreamChannelEndianness
```

Description: Endianess of multi-byte pixel data for this stream.

Visibility: Guru

14.11.4.222 DeviceStreamChannelLink

```
GenApi::IInteger& DeviceStreamChannelLink
```

Description: Index of device's Link to use for streaming the specifed stream channel.

Visibility: Guru

14.11.4.223 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

14.11.4.224 DeviceStreamChannelSelector

```
GenApi::IInteger& DeviceStreamChannelSelector
```

Description: Selects the stream channel to control.

Visibility: Expert

14.11.4.225 DeviceStreamChannelType

```
GenApi::IEnumerationT<DeviceStreamChannelTypeEnums>& DeviceStreamChannelType
```

Description: Reports the type of the stream channel.

Visibility: Guru

14.11.4.226 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

14.11.4.227 DeviceTemperature

```
GenApi::IFloat& DeviceTemperature
```

Description: Device temperature in degrees Celsius (C).

Visibility:

14.11.4.228 DeviceTemperatureSelector

```
GenApi::IEnumerationT<DeviceTemperatureSelectorEnums>& DeviceTemperatureSelector
```

```
Description:
```

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

Visibility:

14.11.4.229 DeviceTLType

```
GenApi::IEnumerationT<DeviceTLTypeEnums>& DeviceTLType
```

Description: Transport Layer type of the device.

Visibility:

14.11.4.230 DeviceTLVersionMajor

```
GenApi::IInteger& DeviceTLVersionMajor
```

```
Description:
```

Major version of the Transport Layer of the device.

Visibility:

14.11.4.231 DeviceTLVersionMinor

```
GenApi::IInteger& DeviceTLVersionMinor
```

```
Description:
```

Minor version of the Transport Layer of the device.

Visibility:

14.11.4.232 DeviceTLVersionSubMinor

```
GenApi::IInteger& DeviceTLVersionSubMinor
```

Description: Sub minor version of the Transport Layer of the device.

Visibility: Beginner

14.11.4.233 DeviceType

```
GenApi::IEnumerationT<DeviceTypeEnums>& DeviceType
```

Description: Returns the device type.

Visibility: Guru

14.11.4.234 DeviceUptime

```
GenApi::IInteger& DeviceUptime
```

Description: Total time since the device was powered up in seconds.

Visibility:

14.11.4.235 DeviceUserID

```
GenApi::IString& DeviceUserID
```

Description: User-programmable device identifier.

Visibility:

14.11.4.236 DeviceVendorName

```
GenApi::IString& DeviceVendorName
```

Description: Name of the manufacturer of the device.

Visibility:

14.11.4.237 DeviceVersion

```
GenApi::IString& DeviceVersion
```

Description: Version of the device.

Visibility:

14.11.4.238 EncoderDivider

```
GenApi::IInteger& EncoderDivider
```

Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.

Visibility: Expert

14.11.4.239 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

14.11.4.240 EncoderOutputMode

```
GenApi::IEnumerationT<EncoderOutputModeEnums>& EncoderOutputMode
```

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

Visibility: Expert

14.11.4.241 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

14.11.4.242 EncoderResetActivation

```
GenApi::IEnumerationT<EncoderResetActivationEnums>& EncoderResetActivation
```

Description: Selects the Activation mode of the Encoder Reset Source signal.

Visibility: Expert

14.11.4.243 EncoderResetSource

```
GenApi::IEnumerationT<EncoderResetSourceEnums>& EncoderResetSource
```

Description: Selects the signals that will be the source to reset the Encoder.

Visibility: Expert

14.11.4.244 EncoderSelector

```
GenApi::IEnumerationT<EncoderSelectorEnums>& EncoderSelector
```

Description: Selects which Encoder to configure.

Visibility: Expert

14.11.4.245 EncoderSourceA

```
GenApi::IEnumerationT<EncoderSourceAEnums>& EncoderSourceA
```

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

Visibility: Expert

14.11.4.246 EncoderSourceB

```
GenApi::IEnumerationT<EncoderSourceBEnums>& EncoderSourceB
```

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

Visibility: Expert

14.11.4.247 EncoderStatus

```
GenApi::IEnumerationT<EncoderStatusEnums>& EncoderStatus
```

Description: Returns the motion status of the encoder.

Visibility: Expert

14.11.4.248 EncoderTimeout

```
GenApi::IFloat& EncoderTimeout
```

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

Visibility: Expert

14.11.4.249 EncoderValue

```
GenApi::IInteger& EncoderValue
```

Description: Reads or writes the current value of the position counter of the selected Encoder.

Visibility: Expert

14.11.4.250 EncoderValueAtReset

```
GenApi::IInteger& EncoderValueAtReset
```

Description: Reads the value of the of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

Visibility: Expert

14.11.4.251 EnumerationCount

```
GenApi::IInteger& EnumerationCount
```

Description: Number of enumerations since uptime.

Visibility:

14.11.4.252 EventAcquisitionEnd

```
GenApi::IInteger& EventAcquisitionEnd
```

Description: Returns the unique Identifier of the Acquisition End type of Event.

Visibility: Expert

14.11.4.253 EventAcquisitionEndFrameID

```
GenApi::IInteger& EventAcquisitionEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End Event.

Visibility: Expert

14.11.4.254 EventAcquisitionEndTimestamp

```
GenApi::IInteger& EventAcquisitionEndTimestamp
```

Description: Returns the Timestamp of the Acquisition End Event.

Visibility: Expert

14.11.4.255 EventAcquisitionError

```
GenApi::IInteger& EventAcquisitionError
```

Description: Returns the unique Identifier of the Acquisition Error type of Event.

Visibility: Expert

14.11.4.256 EventAcquisitionErrorFrameID

```
GenApi::IInteger& EventAcquisitionErrorFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.

Visibility: Expert

14.11.4.257 EventAcquisitionErrorTimestamp

```
GenApi::IInteger& EventAcquisitionErrorTimestamp
```

Description: Returns the Timestamp of the Acquisition Error Event.

Visibility: Expert

14.11.4.258 EventAcquisitionStart

```
GenApi::IInteger& EventAcquisitionStart
```

Description: Returns the unique Identifier of the Acquisition Start type of Event.

Visibility: Expert

14.11.4.259 EventAcquisitionStartFrameID

```
GenApi::IInteger& EventAcquisitionStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start Event.

Visibility: Expert

14.11.4.260 EventAcquisitionStartTimestamp

```
GenApi::IInteger& EventAcquisitionStartTimestamp
```

Description: Returns the Timestamp of the Acquisition Start Event.

Visibility: Expert

14.11.4.261 EventAcquisitionTransferEnd

```
GenApi::IInteger& EventAcquisitionTransferEnd
```

Description: Returns the unique Identifier of the Acquisition Transfer End type of Event.

Visibility: Expert

14.11.4.262 EventAcquisitionTransferEndFrameID

```
GenApi::IInteger& EventAcquisitionTransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End Event.

Visibility: Expert

14.11.4.263 EventAcquisitionTransferEndTimestamp

```
GenApi::IInteger& EventAcquisitionTransferEndTimestamp
```

Description: Returns the Timestamp of the Acquisition Transfer End Event.

Visibility: Expert

14.11.4.264 EventAcquisitionTransferStart

```
GenApi::IInteger& EventAcquisitionTransferStart
```

Description: Returns the unique Identifier of the Acquisition Transfer Start type of Event.

Visibility: Expert

14.11.4.265 EventAcquisitionTransferStartFrameID

```
GenApi::IInteger& EventAcquisitionTransferStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start Event.

Visibility: Expert

14.11.4.266 EventAcquisitionTransferStartTimestamp

```
GenApi::IInteger& EventAcquisitionTransferStartTimestamp
```

Description: Returns the Timestamp of the Acquisition Transfer Start Event.

Visibility: Expert

14.11.4.267 EventAcquisitionTrigger

```
GenApi::IInteger& EventAcquisitionTrigger
```

Description: Returns the unique Identifier of the Acquisition Trigger type of Event.

Visibility: Expert

14.11.4.268 EventAcquisitionTriggerFrameID

```
GenApi::IInteger& EventAcquisitionTriggerFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger Event.

Visibility: Expert

14.11.4.269 EventAcquisitionTriggerTimestamp

```
GenApi::IInteger& EventAcquisitionTriggerTimestamp
```

Description: Returns the Timestamp of the Acquisition Trigger Event.

Visibility: Expert

14.11.4.270 EventActionLate

```
GenApi::IInteger& EventActionLate
```

Description: Returns the unique Identifier of the Action Late type of Event.

Visibility: Expert

14.11.4.271 EventActionLateFrameID

```
GenApi::IInteger& EventActionLateFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late Event.

Visibility: Expert

14.11.4.272 EventActionLateTimestamp

```
GenApi::IInteger& EventActionLateTimestamp
```

Description: Returns the Timestamp of the Action Late Event.

Visibility: Expert

14.11.4.273 EventCounter0End

```
GenApi::IInteger& EventCounter0End
```

Description: Returns the unique Identifier of the Counter 0 End type of Event.

Visibility: Expert

14.11.4.274 EventCounter0EndFrameID

```
GenApi::IInteger& EventCounter0EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End Event.

Visibility: Expert

14.11.4.275 EventCounter0EndTimestamp

```
GenApi::IInteger& EventCounter0EndTimestamp
```

Description: Returns the Timestamp of the Counter 0 End Event.

Visibility: Expert

14.11.4.276 EventCounter0Start

```
GenApi::IInteger& EventCounter0Start
```

Description: Returns the unique Identifier of the Counter 0 Start type of Event.

Visibility: Expert

14.11.4.277 EventCounter0StartFrameID

```
GenApi::IInteger& EventCounter0StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.

Visibility: Expert

14.11.4.278 EventCounter0StartTimestamp

```
GenApi::IInteger& EventCounter0StartTimestamp
```

Description: Returns the Timestamp of the Counter 0 Start Event.

Visibility: Expert

14.11.4.279 EventCounter1End

```
GenApi::IInteger& EventCounter1End
```

Description: Returns the unique Identifier of the Counter 1 End type of Event.

Visibility: Expert

14.11.4.280 EventCounter1EndFrameID

```
GenApi::IInteger& EventCounter1EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End Event.

Visibility: Expert

14.11.4.281 EventCounter1EndTimestamp

```
GenApi::IInteger& EventCounter1EndTimestamp
```

Description: Returns the Timestamp of the Counter 1 End Event.

Visibility: Expert

14.11.4.282 EventCounter1Start

```
GenApi::IInteger& EventCounter1Start
```

Description: Returns the unique Identifier of the Counter 1 Start type of Event.

Visibility: Expert

14.11.4.283 EventCounter1StartFrameID

```
GenApi::IInteger& EventCounter1StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start Event.

Visibility: Expert

14.11.4.284 EventCounter1StartTimestamp

```
GenApi::IInteger& EventCounter1StartTimestamp
```

Description: Returns the Timestamp of the Counter 1 Start Event.

Visibility: Expert

14.11.4.285 EventEncoder0Restarted

```
GenApi::IInteger& EventEncoder0Restarted
```

Description: Returns the unique Identifier of the Encoder 0 Restarted type of Event.

Visibility: Expert

14.11.4.286 EventEncoder0RestartedFrameID

```
GenApi::IInteger& EventEncoder0RestartedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted Event.

Visibility: Expert

14.11.4.287 EventEncoder0RestartedTimestamp

```
GenApi::IInteger& EventEncoder0RestartedTimestamp
```

Description: Returns the Timestamp of the Encoder 0 Restarted Event.

Visibility: Expert

14.11.4.288 EventEncoder0Stopped

```
GenApi::IInteger& EventEncoder0Stopped
```

Description: Returns the unique Identifier of the Encoder 0 Stopped type of Event.

Visibility: Expert

14.11.4.289 EventEncoder0StoppedFrameID

```
GenApi::IInteger& EventEncoder0StoppedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped Event.

Visibility: Expert

14.11.4.290 EventEncoder0StoppedTimestamp

```
GenApi::IInteger& EventEncoder0StoppedTimestamp
```

Description: Returns the Timestamp of the Encoder 0 Stopped Event.

Visibility: Expert

14.11.4.291 EventEncoder1Restarted

```
GenApi::IInteger& EventEncoder1Restarted
```

Description: Returns the unique Identifier of the Encoder 1 Restarted type of Event.

Visibility: Expert

14.11.4.292 EventEncoder1RestartedFrameID

```
GenApi::IInteger& EventEncoder1RestartedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted Event.

Visibility: Expert

14.11.4.293 EventEncoder1RestartedTimestamp

```
GenApi::IInteger& EventEncoder1RestartedTimestamp
```

Description: Returns the Timestamp of the Encoder 1 Restarted Event.

Visibility: Expert

14.11.4.294 EventEncoder1Stopped

```
GenApi::IInteger& EventEncoder1Stopped
```

Description: Returns the unique Identifier of the Encoder 1 Stopped type of Event.

Visibility: Expert

14.11.4.295 EventEncoder1StoppedFrameID

```
GenApi::IInteger& EventEncoder1StoppedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped Event.

Visibility: Expert

14.11.4.296 EventEncoder1StoppedTimestamp

```
GenApi::IInteger& EventEncoder1StoppedTimestamp
```

Description: Returns the Timestamp of the Encoder 1 Stopped Event.

Visibility: Expert

14.11.4.297 EventError

```
GenApi::IInteger& EventError
```

Description: Returns the unique identifier of the Error type of Event.

Visibility:

14.11.4.298 EventErrorCode

```
GenApi::IInteger& EventErrorCode
```

Description: Returns the error code for the error that happened Visibility:

14.11.4.299 EventErrorFrameID

```
GenApi::IInteger& EventErrorFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Error Event.

Visibility:

14.11.4.300 EventErrorTimestamp

```
GenApi::IInteger& EventErrorTimestamp
```

Description: Returns the Timestamp of the Error Event.

Visibility:

14.11.4.301 EventExposureEnd

```
GenApi::IInteger& EventExposureEnd
```

Description: Returns the unique identifier of the Exposure End type of Event.

Visibility:

14.11.4.302 EventExposureEndFrameID

```
GenApi::IInteger& EventExposureEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End Event.

Visibility:

14.11.4.303 EventExposureEndTimestamp

```
GenApi::IInteger& EventExposureEndTimestamp
```

Description: Returns the Timestamp of the Exposure End Event.

Visibility:

14.11.4.304 EventExposureStart

```
GenApi::IInteger& EventExposureStart
```

Description: Returns the unique Identifier of the Exposure Start type of Event.

Visibility: Expert

14.11.4.305 EventExposureStartFrameID

```
GenApi::IInteger& EventExposureStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start Event.

Visibility: Expert

14.11.4.306 EventExposureStartTimestamp

```
GenApi::IInteger& EventExposureStartTimestamp
```

Description: Returns the Timestamp of the Exposure Start Event.

Visibility: Expert

14.11.4.307 EventFrameBurstEnd

```
GenApi::IInteger& EventFrameBurstEnd
```

Description: Returns the unique Identifier of the Frame Burst End type of Event.

Visibility: Expert

14.11.4.308 EventFrameBurstEndFrameID

```
GenApi::IInteger& EventFrameBurstEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End Event.

Visibility: Expert

14.11.4.309 EventFrameBurstEndTimestamp

```
GenApi::IInteger& EventFrameBurstEndTimestamp
```

Description: Returns the Timestamp of the Frame Burst End Event.

Visibility: Expert

14.11.4.310 EventFrameBurstStart

```
GenApi::IInteger& EventFrameBurstStart
```

Description: Returns the unique Identifier of the Frame Burst Start type of Event.

Visibility: Expert

14.11.4.311 EventFrameBurstStartFrameID

```
GenApi::IInteger& EventFrameBurstStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start Event.

Visibility: Expert

14.11.4.312 EventFrameBurstStartTimestamp

```
GenApi::IInteger& EventFrameBurstStartTimestamp
```

Description: Returns the Timestamp of the Frame Burst Start Event.

Visibility: Expert

14.11.4.313 EventFrameEnd

```
GenApi::IInteger& EventFrameEnd
```

Description: Returns the unique Identifier of the Frame End type of Event.

Visibility: Expert

14.11.4.314 EventFrameEndFrameID

```
GenApi::IInteger& EventFrameEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End Event.

Visibility: Expert

14.11.4.315 EventFrameEndTimestamp

```
GenApi::IInteger& EventFrameEndTimestamp
```

Description: Returns the Timestamp of the Frame End Event.

Visibility: Expert

14.11.4.316 EventFrameStart

```
GenApi::IInteger& EventFrameStart
```

Description: Returns the unique Identifier of the Frame Start type of Event.

Visibility: Expert

14.11.4.317 EventFrameStartFrameID

```
GenApi::IInteger& EventFrameStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start Event.

Visibility: Expert

14.11.4.318 EventFrameStartTimestamp

```
GenApi::IInteger& EventFrameStartTimestamp
```

Description: Returns the Timestamp of the Frame Start Event.

Visibility: Expert

14.11.4.319 EventFrameTransferEnd

```
GenApi::IInteger& EventFrameTransferEnd
```

Description: Returns the unique Identifier of the Frame Transfer End type of Event.

Visibility: Expert

14.11.4.320 EventFrameTransferEndFrameID

```
GenApi::IInteger& EventFrameTransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End Event.

Visibility: Expert

14.11.4.321 EventFrameTransferEndTimestamp

```
GenApi::IInteger& EventFrameTransferEndTimestamp
```

Description: Returns the Timestamp of the Frame Transfer End Event.

Visibility: Expert

14.11.4.322 EventFrameTransferStart

```
GenApi::IInteger& EventFrameTransferStart
```

Description: Returns the unique Identifier of the Frame Transfer Start type of Event.

Visibility: Expert

14.11.4.323 EventFrameTransferStartFrameID

```
GenApi::IInteger& EventFrameTransferStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start Event.

Visibility: Expert

14.11.4.324 EventFrameTransferStartTimestamp

```
GenApi::IInteger& EventFrameTransferStartTimestamp
```

Description: Returns the Timestamp of the Frame Transfer Start Event.

Visibility: Expert

14.11.4.325 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

14.11.4.326 EventFrameTriggerFrameID

```
GenApi::IInteger& EventFrameTriggerFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

Visibility: Expert

14.11.4.327 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

14.11.4.328 EventLine0AnyEdge

```
GenApi::IInteger& EventLine0AnyEdge
```

Description: Returns the unique Identifier of the Line 0 Any Edge type of Event.

Visibility: Expert

14.11.4.329 EventLine0AnyEdgeFrameID

```
GenApi::IInteger& EventLine0AnyEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge Event.

Visibility: Expert

14.11.4.330 EventLine0AnyEdgeTimestamp

```
GenApi::IInteger& EventLine0AnyEdgeTimestamp
```

Description: Returns the Timestamp of the Line 0 Any Edge Event.

Visibility: Expert

14.11.4.331 EventLine0FallingEdge

```
GenApi::IInteger& EventLine0FallingEdge
```

Description: Returns the unique Identifier of the Line 0 Falling Edge type of Event.

Visibility: Expert

14.11.4.332 EventLine0FallingEdgeFrameID

```
GenApi::IInteger& EventLine0FallingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.

Visibility: Expert

14.11.4.333 EventLine0FallingEdgeTimestamp

```
GenApi::IInteger& EventLine0FallingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 0 Falling Edge Event.

Visibility: Expert

14.11.4.334 EventLine0RisingEdge

```
GenApi::IInteger& EventLine0RisingEdge
```

Description: Returns the unique Identifier of the Line 0 Rising Edge type of Event.

Visibility: Expert

14.11.4.335 EventLine0RisingEdgeFrameID

```
GenApi::IInteger& EventLine0RisingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event.

Visibility: Expert

14.11.4.336 EventLine0RisingEdgeTimestamp

```
GenApi::IInteger& EventLine0RisingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 0 Rising Edge Event.

Visibility: Expert

14.11.4.337 EventLine1AnyEdge

```
GenApi::IInteger& EventLine1AnyEdge
```

Description: Returns the unique Identifier of the Line 1 Any Edge type of Event.

Visibility: Expert

14.11.4.338 EventLine1AnyEdgeFrameID

```
GenApi::IInteger& EventLine1AnyEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge Event.

Visibility: Expert

14.11.4.339 EventLine1AnyEdgeTimestamp

```
GenApi::IInteger& EventLine1AnyEdgeTimestamp
```

Description: Returns the Timestamp of the Line 1 Any Edge Event.

Visibility: Expert

14.11.4.340 EventLine1FallingEdge

```
GenApi::IInteger& EventLine1FallingEdge
```

Description: Returns the unique Identifier of the Line 1 Falling Edge type of Event.

Visibility: Expert

14.11.4.341 EventLine1FallingEdgeFrameID

```
GenApi::IInteger& EventLine1FallingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge Event.

Visibility: Expert

14.11.4.342 EventLine1FallingEdgeTimestamp

```
GenApi::IInteger& EventLine1FallingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 1 Falling Edge Event.

Visibility: Expert

14.11.4.343 EventLine1RisingEdge

```
GenApi::IInteger& EventLine1RisingEdge
```

Description: Returns the unique Identifier of the Line 1 Rising Edge type of Event.

Visibility: Expert

14.11.4.344 EventLine1RisingEdgeFrameID

```
GenApi::IInteger& EventLine1RisingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event.

Visibility: Expert

14.11.4.345 EventLine1RisingEdgeTimestamp

```
GenApi::IInteger& EventLine1RisingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 1 Rising Edge Event.

Visibility: Expert

14.11.4.346 EventLinkSpeedChange

```
GenApi::IInteger& EventLinkSpeedChange
```

Description: Returns the unique Identifier of the Link Speed Change type of Event.

Visibility: Expert

14.11.4.347 EventLinkSpeedChangeFrameID

```
GenApi::IInteger& EventLinkSpeedChangeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change Event.

Visibility: Expert

14.11.4.348 EventLinkSpeedChangeTimestamp

```
GenApi::IInteger& EventLinkSpeedChangeTimestamp
```

Description: Returns the Timestamp of the Link Speed Change Event.

Visibility: Expert

14.11.4.349 EventLinkTrigger0

```
GenApi::IInteger& EventLinkTrigger0
```

Description: Returns the unique Identifier of the Link Trigger 0 type of Event.

Visibility: Expert

14.11.4.350 EventLinkTrigger0FrameID

```
GenApi::IInteger& EventLinkTrigger0FrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 Event.

Visibility: Expert

14.11.4.351 EventLinkTrigger0Timestamp

```
GenApi::IInteger& EventLinkTrigger0Timestamp
```

Description: Returns the Timestamp of the Link Trigger 0 Event.

Visibility: Expert

14.11.4.352 EventLinkTrigger1

```
GenApi::IInteger& EventLinkTrigger1
```

Description: Returns the unique Identifier of the Link Trigger 1 type of Event.

Visibility: Expert

14.11.4.353 EventLinkTrigger1FrameID

```
GenApi::IInteger& EventLinkTrigger1FrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 Event.

Visibility: Expert

14.11.4.354 EventLinkTrigger1Timestamp

```
GenApi::IInteger& EventLinkTrigger1Timestamp
```

Description: Returns the Timestamp of the Link Trigger 1 Event.

Visibility: Expert

14.11.4.355 EventNotification

```
GenApi::IEnumerationT<EventNotificationEnums>& EventNotification
```

Description: Enables/Disables the selected event.

Visibility:

14.11.4.356 EventSelector

```
GenApi::IEnumerationT<EventSelectorEnums>& EventSelector
```

Description: Selects which Event to enable or disable.

Visibility:

14.11.4.357 EventSequencerSetChange

```
GenApi::IInteger& EventSequencerSetChange
```

Description: Returns the unique Identifier of the Sequencer Set Change type of Event.

Visibility: Expert

14.11.4.358 EventSequencerSetChangeFrameID

```
GenApi::IInteger& EventSequencerSetChangeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.

Visibility: Expert

14.11.4.359 EventSequencerSetChangeTimestamp

```
GenApi::IInteger& EventSequencerSetChangeTimestamp
```

Description: Returns the Timestamp of the Sequencer Set Change Event.

Visibility: Expert

14.11.4.360 EventSerialData

```
GenApi::IString& EventSerialData
```

Description: Returns the serial data that was received.

Visibility:

14.11.4.361 EventSerialDataLength

```
GenApi::IInteger& EventSerialDataLength
```

Description: Returns the length of the received serial data that was included in the event payload.

Visibility:

14.11.4.362 EventSerialPortReceive

```
GenApi::IInteger& EventSerialPortReceive
```

Description: Returns the unique identifier of the Serial Port Receive type of Event.

Visibility:

14.11.4.363 EventSerialPortReceiveTimestamp

```
GenApi::IInteger& EventSerialPortReceiveTimestamp
```

Description: Returns the Timestamp of the Serial Port Receive Event.

Visibility:

14.11.4.364 EventSerialReceiveOverflow

```
GenApi::IBoolean& EventSerialReceiveOverflow
```

Description: Returns the status of the event serial receive overflow.

Visibility:

14.11.4.365 EventStream0TransferBlockEnd

```
GenApi::IInteger& EventStream0TransferBlockEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.

Visibility: Expert

14.11.4.366 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

14.11.4.367 EventStream0TransferBlockEndTimestamp

```
GenApi::IInteger& EventStream0TransferBlockEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.

Visibility: Expert

14.11.4.368 EventStream0TransferBlockStart

```
GenApi::IInteger& EventStream0TransferBlockStart
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of Event.

Visibility: Expert

14.11.4.369 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

14.11.4.370 EventStream0TransferBlockStartTimestamp

```
GenApi::IInteger& EventStream0TransferBlockStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block Start Event.

Visibility: Expert

14.11.4.371 EventStream0TransferBlockTrigger

```
GenApi::IInteger& EventStream0TransferBlockTrigger
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.

Visibility: Expert

14.11.4.372 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

14.11.4.373 EventStream0TransferBlockTriggerTimestamp

```
GenApi::IInteger& EventStream0TransferBlockTriggerTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.

Visibility: Expert

14.11.4.374 EventStream0TransferBurstEnd

```
GenApi::IInteger& EventStream0TransferBurstEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.

Visibility: Expert

14.11.4.375 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

14.11.4.376 EventStream0TransferBurstEndTimestamp

```
GenApi::IInteger& EventStream0TransferBurstEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.

Visibility: Expert

14.11.4.377 EventStream0TransferBurstStart

```
GenApi::IInteger& EventStream0TransferBurstStart
```

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.

Visibility: Expert

14.11.4.378 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

14.11.4.379 EventStream0TransferBurstStartTimestamp

```
GenApi::IInteger& EventStream0TransferBurstStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.

Visibility: Expert

14.11.4.380 EventStream0TransferEnd

```
GenApi::IInteger& EventStream0TransferEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer End type of Event.

Visibility: Expert

14.11.4.381 EventStream0TransferEndFrameID

```
GenApi::IInteger& EventStream0TransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End Event.

Visibility: Expert

14.11.4.382 EventStream0TransferEndTimestamp

```
GenApi::IInteger& EventStream0TransferEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer End Event.

Visibility: Expert

14.11.4.383 EventStream0TransferOverflow

```
GenApi::IInteger& EventStream0TransferOverflow
```

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.

Visibility: Expert

14.11.4.384 EventStream0TransferOverflowFrameID

```
GenApi::IInteger& EventStream0TransferOverflowFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.

Visibility: Expert

14.11.4.385 EventStream0TransferOverflowTimestamp

```
GenApi::IInteger& EventStream0TransferOverflowTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.

Visibility: Expert

14.11.4.386 EventStream0TransferPause

```
GenApi::IInteger& EventStream0TransferPause
```

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of Event.

Visibility: Expert

14.11.4.387 EventStream0TransferPauseFrameID

```
GenApi::IInteger& EventStream0TransferPauseFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause Event.

Visibility: Expert

14.11.4.388 EventStream0TransferPauseTimestamp

```
GenApi::IInteger& EventStream0TransferPauseTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Pause Event.

Visibility: Expert

14.11.4.389 EventStream0TransferResume

```
GenApi::IInteger& EventStream0TransferResume
```

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of Event.

Visibility: Expert

14.11.4.390 EventStream0TransferResumeFrameID

```
GenApi::IInteger& EventStream0TransferResumeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume Event.

Visibility: Expert

14.11.4.391 EventStream0TransferResumeTimestamp

```
GenApi::IInteger& EventStream0TransferResumeTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Resume Event.

Visibility: Expert

14.11.4.392 EventStream0TransferStart

```
GenApi::IInteger& EventStream0TransferStart
```

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of Event.

Visibility: Expert

14.11.4.393 EventStream0TransferStartFrameID

```
GenApi::IInteger& EventStream0TransferStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start Event.

Visibility: Expert

14.11.4.394 EventStream0TransferStartTimestamp

```
GenApi::IInteger& EventStream0TransferStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Start Event.

Visibility: Expert

14.11.4.395 EventTest

```
GenApi::IInteger& EventTest
```

Description: Returns the unique identifier of the Test type of Event.

Visibility:

14.11.4.396 EventTestTimestamp

```
GenApi::IInteger& EventTestTimestamp
```

Description: Returns the Timestamp of the Test Event.

Visibility:

14.11.4.397 EventTimer0End

```
GenApi::IInteger& EventTimer0End
```

Description: Returns the unique Identifier of the Timer 0 End type of Event.

Visibility: Expert

14.11.4.398 EventTimer0EndFrameID

```
GenApi::IInteger& EventTimer0EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End Event.

Visibility: Expert

14.11.4.399 EventTimer0EndTimestamp

```
GenApi::IInteger& EventTimer0EndTimestamp
```

Description: Returns the Timestamp of the Timer 0 End Event.

Visibility: Expert

14.11.4.400 EventTimer0Start

```
GenApi::IInteger& EventTimer0Start
```

Description: Returns the unique Identifier of the Timer 0 Start type of Event.

Visibility: Expert

14.11.4.401 EventTimer0StartFrameID

```
GenApi::IInteger& EventTimer0StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start Event.

Visibility: Expert

14.11.4.402 EventTimer0StartTimestamp

```
GenApi::IInteger& EventTimer0StartTimestamp
```

Description: Returns the Timestamp of the Timer 0 Start Event.

Visibility: Expert

14.11.4.403 EventTimer1End

```
GenApi::IInteger& EventTimer1End
```

Description: Returns the unique Identifier of the Timer 1 End type of Event.

Visibility: Expert

14.11.4.404 EventTimer1EndFrameID

```
GenApi::IInteger& EventTimer1EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End Event.

Visibility: Expert

14.11.4.405 EventTimer1EndTimestamp

```
GenApi::IInteger& EventTimer1EndTimestamp
```

Description: Returns the Timestamp of the Timer 1 End Event.

Visibility: Expert

14.11.4.406 EventTimer1Start

```
GenApi::IInteger& EventTimer1Start
```

Description: Returns the unique Identifier of the Timer 1 Start type of Event.

Visibility: Expert

14.11.4.407 EventTimer1StartFrameID

```
GenApi::IInteger& EventTimer1StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start Event.

Visibility: Expert

14.11.4.408 EventTimer1StartTimestamp

```
GenApi::IInteger& EventTimer1StartTimestamp
```

Description: Returns the Timestamp of the Timer 1 Start Event.

Visibility: Expert

14.11.4.409 ExposureActiveMode

```
GenApi::IEnumerationT<ExposureActiveModeEnums>& ExposureActiveMode
```

Description: Control sensor active exposure mode.

Visibility:

14.11.4.410 ExposureAuto

```
GenApi::IEnumerationT<ExposureAutoEnums>& ExposureAuto
```

Description: Sets the automatic exposure mode Visibility:

14.11.4.411 ExposureMode

```
GenApi::IEnumerationT<ExposureModeEnums>& ExposureMode
```

```
Description:
```

Sets the operation mode of the Exposure.

Visibility:

14.11.4.412 ExposureTime

```
GenApi::IFloat& ExposureTime
```

```
Description:
```

Exposure time in microseconds when Exposure Mode is Timed.

Visibility:

14.11.4.413 ExposureTimeMode

```
GenApi::IEnumerationT<ExposureTimeModeEnums>& ExposureTimeMode
```

Description: Sets the configuration mode of the ExposureTime feature.

Visibility: Beginner

14.11.4.414 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

14.11.4.415 FactoryReset

```
GenApi::ICommand& FactoryReset
```

Description: Returns all user tables to factory default Visibility:

14.11.4.416 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

14.11.4.417 FileAccessLength

```
GenApi::IInteger& FileAccessLength
```

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

Visibility:

14.11.4.418 FileAccessOffset

```
GenApi::IInteger& FileAccessOffset
```

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

14.11.4.419 FileOpenMode

```
GenApi::IEnumerationT<FileOpenModeEnums>& FileOpenMode
```

```
Description:
```

The mode of the file when it is opened.

The file can be opened for reading, writting or both. This must be set before opening the file.

```
Visibility:
```

14.11.4.420 FileOperationExecute

```
GenApi::ICommand& FileOperationExecute
```

```
Description:
```

This is a command that executes the selected file operation on the selected file.

Visibility:

14.11.4.421 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:

14.11.4.422 FileOperationSelector

```
GenApi::IEnumerationT<FileOperationSelectorEnums>& FileOperationSelector
```

```
Description:
```

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

Visibility:

14.11.4.423 FileOperationStatus

```
GenApi::IEnumerationT<FileOperationStatusEnums>& FileOperationStatus
```

Description: Represents the file operation execution status.

Visibility:

14.11.4.424 FileSelector

```
GenApi::IEnumerationT<FileSelectorEnums>& FileSelector
```

```
Description:
```

Selects which file is being operated on.

This must be set before performing any file operations.

```
Visibility:
```

14.11.4.425 FileSize

```
GenApi::IInteger& FileSize
```

Description: Represents the size of the selected file in bytes.

Visibility:

14.11.4.426 Gain

```
GenApi::IFloat& Gain
```

```
Description:
```

Controls the amplification of the video signal in dB.

Visibility:

14.11.4.427 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:

14.11.4.428 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

14.11.4.429 GainSelector

```
GenApi::IEnumerationT<GainSelectorEnums>& GainSelector
```

Description: Selects which gain to control.

The All selection is a total amplification across all channels (or taps).

Visibility:

14.11.4.430 Gamma

```
GenApi::IFloat& Gamma
```

Description: Controls the gamma correction of pixel intensity.

Visibility:

14.11.4.431 GammaEnable

```
GenApi::IBoolean& GammaEnable
```

Description: Enables/disables gamma correction.

Visibility:

14.11.4.432 GevActiveLinkCount

```
GenApi::IInteger& GevActiveLinkCount
```

Description: Indicates the current number of active logical links.

Visibility: Expert

14.11.4.433 GevCCP

```
GenApi::IEnumerationT<GevCCPEnums>& GevCCP
```

Description: Controls the device access privilege of an application.

Visibility:

14.11.4.434 GevCurrentDefaultGateway

```
GenApi::IInteger& GevCurrentDefaultGateway
```

Description: Reports the default gateway IP address to be used on the given logical link.

Visibility:

14.11.4.435 GevCurrentIPAddress

```
GenApi::IInteger& GevCurrentIPAddress
```

Description: Reports the IP address for the given logical link.

Visibility:

14.11.4.436 GevCurrentIPConfigurationDHCP

```
GenApi::IBoolean& GevCurrentIPConfigurationDHCP
```

Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.

Visibility:

14.11.4.437 GevCurrentIPConfigurationLLA

```
GenApi::IBoolean& GevCurrentIPConfigurationLLA
```

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Visibility:

14.11.4.438 GevCurrentIPConfigurationPersistentIP

```
GenApi::IBoolean& GevCurrentIPConfigurationPersistentIP
```

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.

Visibility:

14.11.4.439 GevCurrentPhysicalLinkConfiguration

```
GenApi::IEnumerationT<GevCurrentPhysicalLinkConfigurationEnums>& GevCurrentPhysicalLink↩
Configuration
```

Description: Indicates the current physical link configuration of the device.

Visibility: Expert

14.11.4.440 GevCurrentSubnetMask

```
GenApi::IInteger& GevCurrentSubnetMask
```

Description: Reports the subnet mask of the given logical link.

Visibility:

14.11.4.441 GevDiscoveryAckDelay

```
GenApi::IInteger& GevDiscoveryAckDelay
```

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

Visibility: Expert

14.11.4.442 GevFirstURL

```
GenApi::IString& GevFirstURL
```

Description: The first choice of URL for the XML device description file.

Visibility:

14.11.4.443 GevGVCPExtendedStatusCodes

```
GenApi::IBoolean& GevGVCPExtendedStatusCodes
```

Description: Enables the generation of extended status codes.

Visibility: Guru

14.11.4.444 GevGVCPExtendedStatusCodesSelector

```
GenApi::IEnumerationT<GevGVCPExtendedStatusCodesSelectorEnums>& GevGVCPExtendedStatusCodes↩
Selector
```

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

Visibility: Guru

14.11.4.445 GevGVCPHeartbeatDisable

```
GenApi::IBoolean& GevGVCPHeartbeatDisable
```

Description: Disables the GVCP heartbeat.

Visibility:

14.11.4.446 GevGVCPPendingAck

```
GenApi::IBoolean& GevGVCPPendingAck
```

Description: Enables the generation of PENDING_ACK.

Visibility:

14.11.4.447 GevGVCPPendingTimeout

```
GenApi::IInteger& GevGVCPPendingTimeout
```

Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.

Visibility:

14.11.4.448 GevGVSPExtendedIDMode

```
GenApi::IEnumerationT<GevGVSPExtendedIDModeEnums>& GevGVSPExtendedIDMode
```

Description: Enables the extended IDs mode.

Visibility: Expert

14.11.4.449 GevHeartbeatTimeout

```
GenApi::IInteger& GevHeartbeatTimeout
```

Description: Indicates the current heartbeat timeout in milliseconds.

Visibility:

14.11.4.450 GevIEEE1588

```
GenApi::IBoolean& GevIEEE1588
```

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

Visibility:

14.11.4.451 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:

14.11.4.452 GevIEEE1588Mode

```
GenApi::IEnumerationT<GevIEEE1588ModeEnums>& GevIEEE1588Mode
```

Description: Provides the mode of the IEEE 1588 clock.

Visibility:

14.11.4.453 GevIEEE1588Status

```
GenApi::IEnumerationT<GevIEEE1588StatusEnums>& GevIEEE1588Status
```

Description: Provides the status of the IEEE 1588 clock.

Visibility:

14.11.4.454 GevInterfaceSelector

```
GenApi::IInteger& GevInterfaceSelector
```

Description: Selects which logical link to control.

Visibility:

14.11.4.455 GevIPConfigurationStatus

```
GenApi::IEnumerationT<GevIPConfigurationStatusEnums>& GevIPConfigurationStatus
```

Description: Reports the current IP configuration status.

Visibility: Beginner

14.11.4.456 GevMACAddress

```
GenApi::IInteger& GevMACAddress
```

Description: MAC address of the logical link.

Visibility:

14.11.4.457 GevMCDA

```
GenApi::IInteger& GevMCDA
```

Description: Controls the destination IP address of the message channel Visibility:

14.11.4.458 GevMCPHostPort

```
GenApi::IInteger& GevMCPHostPort
```

Description: The port to which the device must send messages Visibility:

14.11.4.459 GevMCRC

```
GenApi::IInteger& GevMCRC
```

Description: Indicates the number of retries of the message channel.

Visibility:

14.11.4.460 GevMCSP

```
GenApi::IInteger& GevMCSP
```

Description: Indicates the source port of the message channel.

Visibility:

14.11.4.461 GevMCTT

```
GenApi::IInteger& GevMCTT
```

Description: Indicates the transmission timeout of the message channel.

Visibility:

14.11.4.462 GevNumberOfInterfaces

```
GenApi::IInteger& GevNumberOfInterfaces
```

Description: Indicates the number of physical network interfaces supported by this device.

Visibility:

14.11.4.463 GevPAUSEFrameReception

```
GenApi::IBoolean& GevPAUSEFrameReception
```

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

Visibility: Expert

14.11.4.464 GevPAUSEFrameTransmission

```
GenApi::IBoolean& GevPAUSEFrameTransmission
```

Description: Controls whether PAUSE Frames can be generated on the given logical link.

Visibility: Expert

14.11.4.465 GevPersistentDefaultGateway

```
GenApi::IInteger& GevPersistentDefaultGateway
```

Description: Controls the persistent default gateway for this logical link.

Visibility:

14.11.4.466 GevPersistentIPAddress

```
GenApi::IInteger& GevPersistentIPAddress
```

Description: Controls the Persistent IP address for this logical link.

Visibility:

14.11.4.467 GevPersistentSubnetMask

```
GenApi::IInteger& GevPersistentSubnetMask
```

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.

Visibility:

14.11.4.468 GevPhysicalLinkConfiguration

```
GenApi::IEnumerationT<GevPhysicalLinkConfigurationEnums>& GevPhysicalLinkConfiguration
```

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

Visibility: Expert

14.11.4.469 GevPrimaryApplicationIPAddress

```
GenApi::IInteger& GevPrimaryApplicationIPAddress
```

Description: Returns the address of the primary application.

Visibility: Guru

14.11.4.470 GevPrimaryApplicationSocket

```
GenApi::IInteger& GevPrimaryApplicationSocket
```

Description: Returns the UDP source port of the primary application.

Visibility: Guru

14.11.4.471 GevPrimaryApplicationSwitchoverKey

```
GenApi::IInteger& GevPrimaryApplicationSwitchoverKey
```

Description: Controls the key to use to authenticate primary application switchover requests.

Visibility: Guru

14.11.4.472 GevSCCFGAllInTransmission

```
GenApi::IBoolean& GevSCCFGAllInTransmission
```

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

Visibility: Guru

14.11.4.473 GevSCCFGExtendedChunkData

```
GenApi::IBoolean& GevSCCFGExtendedChunkData
```

Description: Enables cameras to use the extended chunk data payload type for this stream channel.

Visibility:

14.11.4.474 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

14.11.4.475 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:

14.11.4.476 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:

14.11.4.477 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:

14.11.4.478 GevSCPDirection

```
GenApi::IInteger& GevSCPDirection
```

Description: Transmit or Receive of the channel Visibility:

14.11.4.479 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:

14.11.4.480 GevSCPInterfaceIndex

```
GenApi::IInteger& GevSCPInterfaceIndex
```

Description: Index of the logical link to use.

Visibility:

14.11.4.481 GevSCPSBigEndian

```
GenApi::IBoolean& GevSCPSBigEndian
```

Description: Endianess of multi-byte pixel data for this stream.

Visibility:

14.11.4.482 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:

14.11.4.483 GevSCPSFireTestPacket

```
GenApi::IBoolean& GevSCPSFireTestPacket
```

Description: Sends a test packet.

Visibility:

14.11.4.484 GevSCPSPacketSize

```
GenApi::IInteger& GevSCPSPacketSize
```

Description: Specifies the stream packet size (in bytes) to send on this channel.

Visibility:

14.11.4.485 GevSCSP

```
GenApi::IInteger& GevSCSP
```

Description: Indicates the source port of the stream channel.

Visibility:

14.11.4.486 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

14.11.4.487 GevSCZoneCount

```
GenApi::IInteger& GevSCZoneCount
```

Description: Reports the number of zones per block transmitted on the selected stream channel.

Visibility: Guru

14.11.4.488 GevSCZoneDirectionAll

```
GenApi::IInteger& GevSCZoneDirectionAll
```

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.

Visibility: Guru

14.11.4.489 GevSecondURL

```
GenApi::IString& GevSecondURL
```

Description: The second choice of URL to the XML device description file.

Visibility:

14.11.4.490 GevStreamChannelSelector

```
GenApi::IInteger& GevStreamChannelSelector
```

Description: Selects the stream channel to control.

Visibility:

14.11.4.491 GevSupportedOption

```
GenApi::IBoolean& GevSupportedOption
```

Description: Returns if the selected GEV option is supported.

Visibility:

14.11.4.492 GevSupportedOptionSelector

```
GenApi::IEnumerationT<GevSupportedOptionSelectorEnums>& GevSupportedOptionSelector
```

Description: Selects the GEV option to interrogate for existing support.

Visibility:

14.11.4.493 GevTimestampTickFrequency

```
GenApi::IInteger& GevTimestampTickFrequency
```

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

Visibility:

14.11.4.494 GuiXmlManifestAddress

```
GenApi::IInteger& GuiXmlManifestAddress
```

Description: Location of the GUI XML manifest table.

Visibility:

14.11.4.495 Height

```
GenApi::IInteger& Height
```

```
Description:
```

Height of the image provided by the device (in pixels).

Visibility:

14.11.4.496 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:

14.11.4.497 ImageComponentEnable

```
GenApi::IBoolean& ImageComponentEnable
```

Description: Controls if the selected component streaming is active.

Visibility: Beginner

14.11.4.498 ImageComponentSelector

```
GenApi::IEnumerationT<ImageComponentSelectorEnums>& ImageComponentSelector
```

Description: Selects a component to activate data streaming from.

Visibility: Beginner

14.11.4.499 ImageCompressionBitrate

```
GenApi::IFloat& ImageCompressionBitrate
```

Description: Control the rate of the produced compressed stream.

Visibility: Expert

14.11.4.500 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

14.11.4.501 ImageCompressionMode

```
GenApi::IEnumerationT<ImageCompressionModeEnums>& ImageCompressionMode
```

Description: Visibility:

14.11.4.502 ImageCompressionQuality

```
GenApi::IInteger& ImageCompressionQuality
```

Description: Control the quality of the produced compressed stream.

Visibility: Expert

14.11.4.503 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

14.11.4.504 IspEnable

```
GenApi::IBoolean& IspEnable
```

```
Description:
```

Controls whether the image processing core is used for optional pixel format mode (i.e.

mono).

```
Visibility:
```

14.11.4.505 LineFilterWidth

```
GenApi::IFloat& LineFilterWidth
```

Description: Filter width in microseconds for the selected line and filter combination Visibility:

14.11.4.506 LineFormat

```
GenApi::IEnumerationT<LineFormatEnums>& LineFormat
```

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

Visibility:

14.11.4.507 LineInputFilterSelector

```
GenApi::IEnumerationT<LineInputFilterSelectorEnums>& LineInputFilterSelector
```

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

Visibility:

14.11.4.508 LineInverter

```
GenApi::IBoolean& LineInverter
```

Description: Controls the inversion of the signal of the selected input or output line.

Visibility:

14.11.4.509 LineMode

```
GenApi::IEnumerationT<LineModeEnums>& LineMode
```

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

Visibility:

14.11.4.510 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

14.11.4.511 LineSelector

```
GenApi::IEnumerationT<LineSelectorEnums>& LineSelector
```

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:

14.11.4.512 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:

14.11.4.513 LineStatus

```
GenApi::IBoolean& LineStatus
```

Description: Returns the current status of the selected input or output Line Visibility:

14.11.4.514 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:

14.11.4.515 LinkErrorCount

```
GenApi::IInteger& LinkErrorCount
```

Description: Counts the number of error on the link.

Visibility:

14.11.4.516 LinkUptime

```
GenApi::IInteger& LinkUptime
```

Description: Time since the last phy negotiation (enumeration).

Visibility:

14.11.4.517 LogicBlockLUTInputActivation

```
GenApi::IEnumerationT<LogicBlockLUTInputActivationEnums>& LogicBlockLUTInputActivation
```

Description: Selects the activation mode of the Logic Input Source signal.

Visibility:

14.11.4.518 LogicBlockLUTInputSelector

```
GenApi::IEnumerationT<LogicBlockLUTInputSelectorEnums>& LogicBlockLUTInputSelector
```

Description: Controls which LogicBlockLUT Input Source & Activation to access.

Visibility:

14.11.4.519 LogicBlockLUTInputSource

```
GenApi::IEnumerationT<LogicBlockLUTInputSourceEnums>& LogicBlockLUTInputSource
```

Description: Selects the source for the input into the Logic LUT.

Visibility:

14.11.4.520 LogicBlockLUTOutputValue

```
GenApi::IBoolean& LogicBlockLUTOutputValue
```

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

Visibility:

14.11.4.521 LogicBlockLUTOutputValueAll

```
GenApi::IInteger& LogicBlockLUTOutputValueAll
```

Description: Sets the value of all the output bits in the selected LUT.

Visibility:

14.11.4.522 LogicBlockLUTRowIndex

```
GenApi::IInteger& LogicBlockLUTRowIndex
```

Description: Controls the row of the truth table to access in the selected LUT.

Visibility:

14.11.4.523 LogicBlockLUTSelector

```
GenApi::IEnumerationT<LogicBlockLUTSelectorEnums>& LogicBlockLUTSelector
```

Description: Selects which LogicBlock LUT to configure Visibility:

14.11.4.524 LogicBlockSelector

```
GenApi::IEnumerationT<LogicBlockSelectorEnums>& LogicBlockSelector
```

Description: Selects which LogicBlock to configure Visibility:

14.11.4.525 LUTEnable

```
GenApi::IBoolean& LUTEnable
```

```
Description:
```

Activates the selected LUT.

Visibility:

14.11.4.526 LUTIndex

```
GenApi::IInteger& LUTIndex
```

```
Description:
```

Control the index (offset) of the coefficient to access in the selected LUT.

Visibility:

14.11.4.527 LUTSelector

```
GenApi::IEnumerationT<LUTSelectorEnums>& LUTSelector
```

```
Description:
```

Selects which LUT to control.

Visibility:

14.11.4.528 LUTValue

```
GenApi::IInteger& LUTValue
```

```
Description:
```

Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.

Visibility:

14.11.4.529 LUTValueAll

```
GenApi::IRegister& LUTValueAll
```

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.

Visibility: Guru

14.11.4.530 MaxDeviceResetTime

```
GenApi::IInteger& MaxDeviceResetTime
```

Description: Time to wait until device reset complete (ms).

Visibility:

14.11.4.531 OffsetX

```
GenApi::IInteger& OffsetX
```

```
Description:
```

Horizontal offset from the origin to the ROI (in pixels).

Visibility:

14.11.4.532 OffsetY

```
GenApi::IInteger& OffsetY
```

```
Description:
```

Vertical offset from the origin to the ROI (in pixels).

Visibility:

14.11.4.533 PacketResendRequestCount

```
GenApi::IInteger& PacketResendRequestCount
```

Description: Counts the number of resend requests received from the host.

Visibility:

14.11.4.534 PayloadSize

```
GenApi::IInteger& PayloadSize
```

Description: Provides the number of bytes transferred for each image or chunk on the stream channel.

Visibility:

14.11.4.535 PixelColorFilter

```
GenApi::IEnumerationT<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:

14.11.4.536 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:

14.11.4.537 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:

14.11.4.538 PixelFormat

```
GenApi::IEnumerationT<PixelFormatEnums>& PixelFormat
```

Description: Format of the pixel provided by the camera.

Visibility:

14.11.4.539 PixelFormatInfoID

```
GenApi::IInteger& PixelFormatInfoID
```

Description: Returns the value used by the streaming channels to identify the selected pixel format.

Visibility: Guru

14.11.4.540 PixelFormatInfoSelector

```
GenApi::IEnumerationT<PixelFormatInfoSelectorEnums>& PixelFormatInfoSelector
```

Description: Select the pixel format for which the information will be returned.

Visibility: Guru

14.11.4.541 PixelSize

```
GenApi::IEnumerationT<PixelSizeEnums>& PixelSize
```

Description: Total size in bits of a pixel of the image.

Visibility:

14.11.4.542 PowerSupplyCurrent

```
GenApi::IFloat& PowerSupplyCurrent
```

```
Description:
```

Indicates the output current of the selected power supply (A).

Visibility:

14.11.4.543 PowerSupplyVoltage

```
GenApi::IFloat& PowerSupplyVoltage
```

```
Description:
```

Indicates the current voltage of the selected power supply (V).

Visibility:

14.11.4.544 RegionDestination

```
GenApi::IEnumerationT<RegionDestinationEnums>& RegionDestination
```

Description: Control the destination of the selected region.

Visibility: Expert

14.11.4.545 RegionMode

```
GenApi::IEnumerationT<RegionModeEnums>& RegionMode
```

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

Visibility: Beginner

14.11.4.546 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

14.11.4.547 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:

14.11.4.548 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:

14.11.4.549 RgbTransformLightSource

```
GenApi::IEnumerationT<RgbTransformLightSourceEnums>& RgbTransformLightSource
```

```
Description:
```

Used to select from a set of RGBtoRGB transform matricies 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:
```

14.11.4.550 Saturation

```
GenApi::IFloat& Saturation
```

Description: Controls the color saturation.

Visibility:

14.11.4.551 SaturationEnable

```
GenApi::IBoolean& SaturationEnable
```

Description: Enables/disables Saturation adjustment.

Visibility:

14.11.4.552 Scan3dAxisMax

```
GenApi::IFloat& Scan3dAxisMax
```

Description: Maximum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

14.11.4.553 Scan3dAxisMin

```
GenApi::IFloat& Scan3dAxisMin
```

Description: Minimum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

14.11.4.554 Scan3dCoordinateOffset

```
GenApi::IFloat& Scan3dCoordinateOffset
```

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

14.11.4.555 Scan3dCoordinateReferenceSelector

```
GenApi::IEnumerationT<Scan3dCoordinateReferenceSelectorEnums>& Scan3dCoordinateReference↩
Selector
```

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

14.11.4.556 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

14.11.4.557 Scan3dCoordinateScale

```
GenApi::IFloat& Scan3dCoordinateScale
```

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

14.11.4.558 Scan3dCoordinateSelector

```
GenApi::IEnumerationT<Scan3dCoordinateSelectorEnums>& Scan3dCoordinateSelector
```

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

Visibility: Expert

14.11.4.559 Scan3dCoordinateSystem

```
GenApi::IEnumerationT<Scan3dCoordinateSystemEnums>& Scan3dCoordinateSystem
```

Description: Specifies the Coordinate system to use for the device.

Visibility: Beginner

14.11.4.560 Scan3dCoordinateSystemReference

```
GenApi::IEnumerationT<Scan3dCoordinateSystemReferenceEnums>& Scan3dCoordinateSystemReference
```

Description: Defines coordinate system reference location.

Visibility: Expert

14.11.4.561 Scan3dCoordinateTransformSelector

```
GenApi::IEnumerationT<Scan3dCoordinateTransformSelectorEnums>& Scan3dCoordinateTransform↩
Selector
```

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

Visibility: Expert

14.11.4.562 Scan3dDistanceUnit

```
GenApi::IEnumerationT<Scan3dDistanceUnitEnums>& Scan3dDistanceUnit
```

Description: Specifies the unit used when delivering calibrated distance data.

Visibility: Beginner

14.11.4.563 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 Scan3d↩InvalidDataValue may give processing penalties due to special handling. Visibility: Expert

14.11.4.564 Scan3dInvalidDataValue

```
GenApi::IFloat& Scan3dInvalidDataValue
```

Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.

Visibility: Expert

14.11.4.565 Scan3dOutputMode

```
GenApi::IEnumerationT<Scan3dOutputModeEnums>& Scan3dOutputMode
```

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

Visibility: Expert

14.11.4.566 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

14.11.4.567 SensorDescription

```
GenApi::IString& SensorDescription
```

Description: Returns Sensor Description Visibility:

14.11.4.568 SensorDigitizationTaps

```
GenApi::IEnumerationT<SensorDigitizationTapsEnums>& SensorDigitizationTaps
```

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

Visibility: Expert

14.11.4.569 SensorHeight

```
GenApi::IInteger& SensorHeight
```

Description: Effective height of the sensor in pixels.

Visibility:

14.11.4.570 SensorShutterMode

```
GenApi::IEnumerationT<SensorShutterModeEnums>& SensorShutterMode
```

Description: Sets the shutter mode of the device.

Visibility:

14.11.4.571 SensorTaps

```
GenApi::IEnumerationT<SensorTapsEnums>& SensorTaps
```

Description: Number of taps of the camera sensor.

Visibility: Expert

14.11.4.572 SensorWidth

```
GenApi::IInteger& SensorWidth
```

Description: Effective width of the sensor in pixels.

Visibility:

14.11.4.573 SequencerConfigurationMode

```
GenApi::IEnumerationT<SequencerConfigurationModeEnums>& SequencerConfigurationMode
```

```
Description:
```

Controls whether or not a sequencer is in configuration mode.

Visibility:

14.11.4.574 SequencerConfigurationValid

```
GenApi::IEnumerationT<SequencerConfigurationValidEnums>& SequencerConfigurationValid
```

```
Description:
```

Display whether the current sequencer configuration is valid to run.

Visibility:

14.11.4.575 SequencerFeatureEnable

```
GenApi::IBoolean& SequencerFeatureEnable
```

```
Description:
```

Enables the selected feature and makes it active in all sequencer sets.

Visibility:

14.11.4.576 SequencerMode

```
GenApi::IEnumerationT<SequencerModeEnums>& SequencerMode
```

Description: Controls whether or not a sequencer is active.

Visibility:

14.11.4.577 SequencerPathSelector

```
GenApi::IInteger& SequencerPathSelector
```

```
Description:
```

Selects branching path to be used for subsequent settings.

Visibility:

14.11.4.578 SequencerSetActive

```
GenApi::IInteger& SequencerSetActive
```

Description: Displays the currently active sequencer set.

Visibility:

14.11.4.579 SequencerSetLoad

```
GenApi::ICommand& SequencerSetLoad
```

```
Description:
```

Loads currently selected sequencer to the current device configuration.

Visibility:

14.11.4.580 SequencerSetNext

```
GenApi::IInteger& SequencerSetNext
```

Description: Specifies the next sequencer set.

Visibility:

14.11.4.581 SequencerSetSave

```
GenApi::ICommand& SequencerSetSave
```

```
Description:
```

Saves the current device configuration to the currently selected sequencer set.

Visibility:

14.11.4.582 SequencerSetSelector

```
GenApi::IInteger& SequencerSetSelector
```

```
Description:
```

Selects the sequencer set to which subsequent settings apply.

Visibility:

14.11.4.583 SequencerSetStart

```
GenApi::IInteger& SequencerSetStart
```

Description: Sets the first sequencer set to be used.

Visibility:

14.11.4.584 SequencerSetValid

```
GenApi::IEnumerationT<SequencerSetValidEnums>& SequencerSetValid
```

```
Description:
```

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

Visibility:

14.11.4.585 SequencerTriggerActivation

```
GenApi::IEnumerationT<SequencerTriggerActivationEnums>& SequencerTriggerActivation
```

```
Description:
```

Specifies the activation mode of the sequencer trigger.

Visibility:

14.11.4.586 SequencerTriggerSource

```
GenApi::IEnumerationT<SequencerTriggerSourceEnums>& SequencerTriggerSource
```

```
Description:
```

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

Visibility:

14.11.4.587 SerialPortBaudRate

```
GenApi::IEnumerationT<SerialPortBaudRateEnums>& SerialPortBaudRate
```

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

Visibility:

14.11.4.588 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:

14.11.4.589 SerialPortParity

```
GenApi::IEnumerationT<SerialPortParityEnums>& SerialPortParity
```

Description: This feature controls the parity used by the selected serial port.

Visibility:

14.11.4.590 SerialPortSelector

```
GenApi::IEnumerationT<SerialPortSelectorEnums>& SerialPortSelector
```

Description: Selects which serial port of the device to control.

Visibility:

14.11.4.591 SerialPortSource

```
GenApi::IEnumerationT<SerialPortSourceEnums>& SerialPortSource
```

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

Visibility:

14.11.4.592 SerialPortStopBits

```
GenApi::IEnumerationT<SerialPortStopBitsEnums>& SerialPortStopBits
```

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

Visibility:

14.11.4.593 SerialReceiveFramingErrorCount

```
GenApi::IInteger& SerialReceiveFramingErrorCount
```

Description: Returns the number of framing errors that have occurred on the serial port.

Visibility:

14.11.4.594 SerialReceiveParityErrorCount

```
GenApi::IInteger& SerialReceiveParityErrorCount
```

Description: Returns the number of parity errors that have occurred on the serial port.

Visibility:

14.11.4.595 SerialReceiveQueueClear

```
GenApi::ICommand& SerialReceiveQueueClear
```

Description: This is a command that clears the device serial port receive queue.

Visibility:

14.11.4.596 SerialReceiveQueueCurrentCharacterCount

```
GenApi::IInteger& SerialReceiveQueueCurrentCharacterCount
```

Description: Returns the number of characters currently in the serial port receive queue.

Visibility:

14.11.4.597 SerialReceiveQueueMaxCharacterCount

```
GenApi::IInteger& SerialReceiveQueueMaxCharacterCount
```

Description: >Returns the maximum number of characters in the serial port receive queue.

Visibility:

14.11.4.598 SerialTransmitQueueCurrentCharacterCount

```
GenApi::IInteger& SerialTransmitQueueCurrentCharacterCount
```

Description: Returns the number of characters currently in the serial port transmit queue.

Visibility:

14.11.4.599 SerialTransmitQueueMaxCharacterCount

```
GenApi::IInteger& SerialTransmitQueueMaxCharacterCount
```

Description: >Returns the maximum number of characters in the serial port transmit queue.

Visibility:

14.11.4.600 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:
```

14.11.4.601 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:
```

14.11.4.602 SharpeningEnable

```
GenApi::IBoolean& SharpeningEnable
```

```
Description:
```

Enables/disables the sharpening feature.

Sharpening is disabled by default.

```
Visibility:
```

14.11.4.603 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:

14.11.4.604 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

14.11.4.605 SoftwareSignalSelector

```
GenApi::IEnumerationT<SoftwareSignalSelectorEnums>& SoftwareSignalSelector
```

Description: Selects which Software Signal features to control.

Visibility: Beginner

14.11.4.606 SourceCount

```
GenApi::IInteger& SourceCount
```

Description: Controls or returns the number of sources supported by the device.

Visibility: Beginner

14.11.4.607 SourceSelector

```
GenApi::IEnumerationT<SourceSelectorEnums>& SourceSelector
```

Description: Selects the source to control.

Visibility: Beginner

14.11.4.608 Test0001

```
GenApi::IInteger& Test0001
```

Description: For testing only.

Visibility:

14.11.4.609 TestEventGenerate

```
GenApi::ICommand& TestEventGenerate
```

Description: This command generates a test event and sends it to the host.

Visibility:

14.11.4.610 TestPattern

```
GenApi::IEnumerationT<TestPatternEnums>& TestPattern
```

```
Description:
```

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

Visibility:

14.11.4.611 TestPatternGeneratorSelector

```
GenApi::IEnumerationT<TestPatternGeneratorSelectorEnums>& TestPatternGeneratorSelector
```

```
Description:
```

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

Visibility:

14.11.4.612 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

14.11.4.613 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

14.11.4.614 TimerDuration

```
GenApi::IFloat& TimerDuration
```

Description: Sets the duration (in microseconds) of the Timer pulse.

Visibility: Expert

14.11.4.615 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

14.11.4.616 TimerSelector

```
GenApi::IEnumerationT<TimerSelectorEnums>& TimerSelector
```

Description: Selects which Timer to configure.

Visibility: Expert

14.11.4.617 TimerStatus

```
GenApi::IEnumerationT<TimerStatusEnums>& TimerStatus
```

Description: Returns the current status of the Timer.

Visibility: Expert

14.11.4.618 TimerTriggerActivation

```
GenApi::IEnumerationT<TimerTriggerActivationEnums>& TimerTriggerActivation
```

Description: Selects the activation mode of the trigger to start the Timer.

Visibility: Expert

14.11.4.619 TimerTriggerSource

```
GenApi::IEnumerationT<TimerTriggerSourceEnums>& TimerTriggerSource
```

Description: Selects the source of the trigger to start the Timer.

Visibility: Expert

14.11.4.620 TimerValue

```
GenApi::IFloat& TimerValue
```

Description: Reads or writes the current value (in microseconds) of the selected Timer.

Visibility: Expert

14.11.4.621 Timestamp

```
GenApi::IInteger& Timestamp
```

Description: Reports the current value of the device timestamp counter.

Visibility: Expert

14.11.4.622 TimestampLatch

```
GenApi::ICommand& TimestampLatch
```

Description: Latches the current timestamp counter into TimestampLatchValue.

Visibility:

14.11.4.623 TimestampLatchValue

```
GenApi::IInteger& TimestampLatchValue
```

Description: Returns the latched value of the timestamp counter.

Visibility:

14.11.4.624 TimestampReset

```
GenApi::ICommand& TimestampReset
```

Description: Resets the current value of the device timestamp counter.

Visibility:

14.11.4.625 TLParamsLocked

```
GenApi::IInteger& TLParamsLocked
```

Description: Visibility:

14.11.4.626 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

14.11.4.627 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:

14.11.4.628 TransferBurstCount

```
GenApi::IInteger& TransferBurstCount
```

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.

Visibility: Expert

14.11.4.629 TransferComponentSelector

```
GenApi::IEnumerationT<TransferComponentSelectorEnums>& TransferComponentSelector
```

Description: Selects the color component for the control of the TransferStreamChannel feature.

Visibility: Guru

14.11.4.630 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:

14.11.4.631 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:

14.11.4.632 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 transfered. The device will resume its transmission at the reception of a TransferResume command. Visibility: Guru

14.11.4.633 TransferQueueCurrentBlockCount

```
GenApi::IInteger& TransferQueueCurrentBlockCount
```

Description: Returns number of data blocks (images) currently in the transfer queue.

Visibility:

14.11.4.634 TransferQueueMaxBlockCount

```
GenApi::IInteger& TransferQueueMaxBlockCount
```

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

14.11.4.635 TransferQueueMode

```
GenApi::IEnumerationT<TransferQueueModeEnums>& TransferQueueMode
```

Description: Specifies the operation mode of the transfer queue.

Visibility:

14.11.4.636 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:

14.11.4.637 TransferResume

```
GenApi::ICommand& TransferResume
```

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.

Visibility: Guru

14.11.4.638 TransferSelector

```
GenApi::IEnumerationT<TransferSelectorEnums>& TransferSelector
```

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

Visibility: Expert

14.11.4.639 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:

14.11.4.640 TransferStatus

```
GenApi::IBoolean& TransferStatus
```

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.

Visibility: Guru

14.11.4.641 TransferStatusSelector

```
GenApi::IEnumerationT<TransferStatusSelectorEnums>& TransferStatusSelector
```

Description: Selects which status of the transfer module to read.

Visibility: Guru

14.11.4.642 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:

14.11.4.643 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

14.11.4.644 TransferTriggerActivation

```
GenApi::IEnumerationT<TransferTriggerActivationEnums>& TransferTriggerActivation
```

Description: Specifies the activation mode of the transfer control trigger.

Visibility: Guru

14.11.4.645 TransferTriggerMode

```
GenApi::IEnumerationT<TransferTriggerModeEnums>& TransferTriggerMode
```

Description: Controls if the selected trigger is active.

Visibility: Guru

14.11.4.646 TransferTriggerSelector

```
GenApi::IEnumerationT<TransferTriggerSelectorEnums>& TransferTriggerSelector
```

Description: Selects the type of transfer trigger to configure.

Visibility: Guru

14.11.4.647 TransferTriggerSource

```
GenApi::IEnumerationT<TransferTriggerSourceEnums>& TransferTriggerSource
```

Description: Specifies the signal to use as the trigger source for transfers.

Visibility: Guru

14.11.4.648 TriggerActivation

```
GenApi::IEnumerationT<TriggerActivationEnums>& TriggerActivation
```

Description: Specifies the activation mode of the trigger.

Visibility:

14.11.4.649 TriggerDelay

```
GenApi::IFloat& TriggerDelay
```

```
Description:
```

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

Visibility:

14.11.4.650 TriggerDivider

```
GenApi::IInteger& TriggerDivider
```

Description: Specifies a division factor for the incoming trigger pulses.

Visibility: Expert

14.11.4.651 TriggerEventTest

```
GenApi::ICommand& TriggerEventTest
```

Description: This command generates a test event and sends it to the host.

Visibility:

14.11.4.652 TriggerMode

```
GenApi::IEnumerationT<TriggerModeEnums>& TriggerMode
```

```
Description:
```

Controls whether or not trigger is active.

Visibility:

14.11.4.653 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

14.11.4.654 TriggerOverlap

```
GenApi::IEnumerationT<TriggerOverlapEnums>& TriggerOverlap
```

Description: Specifies the overlap mode of the trigger.

Visibility:

14.11.4.655 TriggerSelector

```
GenApi::IEnumerationT<TriggerSelectorEnums>& TriggerSelector
```

Description: Selects the type of trigger to configure.

Visibility:

14.11.4.656 TriggerSoftware

```
GenApi::ICommand& TriggerSoftware
```

```
Description:
```

Generates an internal trigger if Trigger Source is set to Software.

Visibility:

14.11.4.657 TriggerSource

```
GenApi::IEnumerationT<TriggerSourceEnums>& TriggerSource
```

```
Description:
```

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

Visibility:

14.11.4.658 UserOutputSelector

```
GenApi::IEnumerationT<UserOutputSelectorEnums>& UserOutputSelector
```

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

Visibility:

14.11.4.659 UserOutputValue

```
GenApi::IBoolean& UserOutputValue
```

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).

Visibility:

14.11.4.660 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:

14.11.4.661 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

14.11.4.662 UserSetDefault

```
GenApi::IEnumerationT<UserSetDefaultEnums>& UserSetDefault
```

```
Description:
```

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

Visibility:

14.11.4.663 UserSetFeatureEnable

```
GenApi::IBoolean& UserSetFeatureEnable
```

Description: Whether or not the selected feature is saved to user sets.

Visibility:

14.11.4.664 UserSetLoad

```
GenApi::ICommand& UserSetLoad
```

```
Description:
```

Loads the User Set specified by UserSetSelector to the device and makes it active.

Visibility:

14.11.4.665 UserSetSave

```
GenApi::ICommand& UserSetSave
```

```
Description:
```

Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.

Visibility:

14.11.4.666 UserSetSelector

```
GenApi::IEnumerationT<UserSetSelectorEnums>& UserSetSelector
```

```
Description:
```

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

Visibility:

14.11.4.667 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:

14.11.4.668 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

14.11.4.669 WhiteClipSelector

```
GenApi::IEnumerationT<WhiteClipSelectorEnums>& WhiteClipSelector
```

Description: Selects which White Clip to control.

Visibility: Expert

14.11.4.670 Width

```
GenApi::IInteger& Width
```

```
Description:
```

Width of the image provided by the device (in pixels).

Visibility:

14.11.4.671 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

14.12 CameraBase Class Reference

The base class for the camera object.

Inheritance diagram for CameraBase:

Collaboration diagram for CameraBase:

Public Member Functions

- virtual ~CameraBase (void)

  Virtual Destructor.
- void Init ()

  Init 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 RegisterEventHandler (EventHandler &evtHandlerToRegister)

  RegisterEventHandler(EventHandler &) Registers a specific event handler for the camera.
- void RegisterEventHandler (EventHandler &evtHandlerToRegister, const GenICam::gcstring &eventName)

  RegisterEventHandler(EventHandler &, const GenICam::gcstring&) Registers a specific event handler for the camera.
- void UnregisterEventHandler (EventHandler &evtHandlerToUnregister)

  UnregisterEventHandler Unregisters an event handler for the camera Event handlers should be unregistered first before calling camera 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

14.12.1 Detailed Description

The base class for the camera object.

14.12.2 Constructor & Destructor Documentation

14.12.2.1 ~CameraBase()

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

Virtual Destructor.

14.12.2.2 CameraBase() [1/2]

```
CameraBase (
            void ) [protected]
```

Default constructor.

14.12.2.3 CameraBase() [2/2]

```
CameraBase (
            const CameraBase & ) [protected]
```

Copy constructor.

14.12.3 Member Function Documentation

14.12.3.1 BeginAcquisition()

```
void BeginAcquisition ( ) [virtual]
```

BeginAcquisition Starts the image acquisition engine.

The camera must be initialized via a call to Init() before starting an acquisition.

See also

Init()

Implements ICameraBase.

14.12.3.2 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()
UnregisterEventHandler(EventHandler & evtHandlerToUnregister)

Implements ICameraBase.

14.12.3.3 DiscoverMaxPacketSize()

```
unsigned int DiscoverMaxPacketSize ( ) [virtual]
```

DiscoverMaxPacketSize Returns the largest packet size that can be safely used on the interface that device is connected to.

Returns

The maximum packet size returned.

Implements ICameraBase.

14.12.3.4 EndAcquisition()

```
void EndAcquisition ( ) [virtual]
```

EndAcquisition Stops the image acquisition engine.

If EndAcquisition() is called without a prior call to BeginAcquisition() an error message "Camera is not started" will be thrown. All Images that were acquired using GetNextImage() need to be released first using image->Release() before calling EndAcquisition(). All buffers in the input pool and output queue will be discarded when EndAcquisition() is called.

See also

Init()
BeginAcquisition()
GetNextImage(grabTimeout)
Image::Release()

Implements ICameraBase.

14.12.3.5 ForceIP()

```
void ForceIP ( ) [virtual]
```

ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Implements ICameraBase.

14.12.3.6 GetAccessMode()

```
GenApi::EAccessMode GetAccessMode ( ) const [virtual]
```

GetAccessMode Returns the access mode that the software has on the Camera.

The camera does not need to be initialized before calling this function.

See also

Init()

Returns

An enumeration value indicating the access mode

Implements ICameraBase.

14.12.3.7 GetBufferOwnership()

```
BufferOwnership GetBufferOwnership ( ) const [virtual]
```

GetBufferOwnership Gets data buffer ownership.

The data buffers can be owned by System or User. If the system owns the buffers, the memory required for the buffers are allocated and freed by the library. If user owns the buffers, the user is responsible for allocating and ultimately freeing the memory. By default, data buffers are owned by the library.

See also

SetBufferOwnership()
SetUserBuffers()

Returns

Buffer ownership (system or user)

Implements ICameraBase.

14.12.3.8 GetGuiXml()

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

Returns the GUI XML that can be passed into the Spinnaker GUI framework.

Returns

GenICam::gcstring that represents the uncompressed GUI XML file

Implements ICameraBase.

14.12.3.9 GetNextImage()

```
ImagePtr GetNextImage (
            uint64_t grabTimeout = EVENT_TIMEOUT_INFINITE,
            uint64_t streamID = 0 )  [virtual]
```

GetNextImage Gets the next image that was received by the transport layer.

This function will block indefinitely until an image arrives. Most cameras support one stream so the default streamID is 0 but if a camera supports multiple streams the user can input the streamID to select from which stream to grab images

See also

Init()
BeginAcquisition()
EndAcquisition()

Parameters

grabTimeout	a 64bit value that represents a timeout in milliseconds
streamID	The stream to grab the image.

Returns

pointer to an Image object

Implements ICameraBase.

14.12.3.10 GetNodeMap()

```
GenApi::INodeMap& GetNodeMap ( ) const  [virtual]
```

GetNodeMap Gets a reference to the node map that is generated from a GenICam XML file.

The camera must be initialized by a call to Init() first before a node map reference can be successfully acquired.

See also

Init()

Returns

A reference to the INodeMap.

Implements ICameraBase.

14.12.3.11 GetNumDataStreams()

```
unsigned int GetNumDataStreams ( ) [virtual]
```

GetNumDataStreams Returns the number of streams that a device supports.

Returns

The number of data streams

Implements ICameraBase.

14.12.3.12 GetNumImagesInUse()

```
unsigned int GetNumImagesInUse ( ) [virtual]
```

GetNumImagesInUse Returns the number of images that are currently in use.

Each of the images that are currently in use must be cleaned up with a call to image->Release() before calling system->ReleaseInstance().

Returns

The number of images that needs to be cleaned up.

Implements ICameraBase.

14.12.3.13 GetTLDeviceNodeMap()

```
GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [virtual]
```

GetTLDeviceNodeMap Gets a reference to the node map that is generated from a GenICam XML file for the GenTL Device module.

The camera does not need to be initialized before acquiring this node map.

Returns

A reference to the INodeMap.

Implements ICameraBase.

14.12.3.14 GetTLStreamNodeMap()

```
GenApi::INodeMap& GetTLStreamNodeMap ( ) const [virtual]
```

GetTLStreamNodeMap Gets a reference to the node map that is generated from a GenICam XML file for the GenTL Stream module.

The camera does not need to be initialized before acquiring this node map.

Returns

A reference to the INodeMap.

Implements ICameraBase.

14.12.3.15 GetUniqueID()

```
GenICam::gcstring GetUniqueID ( ) [virtual]
```

GetUniqueID This returns a unique id string that identifies the camera.

This is the camera serial number.

Returns

string that uniquely identifies the camera (serial number)

Implements ICameraBase.

14.12.3.16 GetUserBufferCount()

```
uint64_t GetUserBufferCount ( ) const [virtual]
```

GetUserBufferCount Gets the number of user memory buffers.

This will throw an exception if user memory buffer has not been set. If the user memory is contiguous, this will throw an exception unless BeginAcquisition() has been called.

See also

BeginAcquisition()
SetUserBuffers()

Returns

The number of user memory buffers

Implements ICameraBase.

14.12.3.17 GetUserBufferSize()

```
uint64_t GetUserBufferSize ( ) const  [virtual]
```

GetUserBufferSize Gets the size of one user memory buffer (in bytes).

This will throw an exception if user memory buffer has not been set. If the user memory is contiguous, this will throw an exception unless BeginAcquisition() has been called. To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: ((unsigned int) (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.

14.12.3.18 GetUserBufferTotalSize()

```
uint64_t GetUserBufferTotalSize ( ) const  [virtual]
```

GetUserBufferTotalSize Gets the total size of all the user memory buffers (in bytes).

This will throw an exception if user memory buffer has not been set. The total size should be GetUserBufferSize() multiplied by GetUserBufferCount() or larger.

See also

GetUserBufferCount()
GetUserBufferSize()
SetUserBuffers()

Returns

The total size of all the user memory buffers (in bytes)

Implements ICameraBase.

14.12.3.19 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.

14.12.3.20 IsInitialized()

```
bool IsInitialized ( ) [virtual]
```

IsInitialized Checks if camera is initialized.

This function needs to return true in order to retrieve a valid NodeMap from the GetNodeMap() call.

See also

GetNodeMap()

Returns

If camera is initialized or not

Implements ICameraBase.

14.12.3.21 IsStreaming()

```
bool IsStreaming ( ) const [virtual]
```

IsStreaming Returns true if the camera is currently streaming or false if it is not.

See also

Init()

Returns

returns true if camera is streaming and false otherwise.

Implements ICameraBase.

14.12.3.22 IsValid()

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

IsValid Checks a flag to determine if camera is still valid for use.

Returns

If camera is valid or not

Implements ICameraBase.

14.12.3.23 operator=()

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

Assignment operator.

14.12.3.24 ReadPort()

```
void ReadPort (
            uint64_t iAddress,
            void * pBuffer,
            size_t iSize ) [virtual]
```

Implements ICameraBase.

14.12.3.25 RegisterEventHandler() [1/2]

```
void RegisterEventHandler (
            EventHandler & evtHandlerToRegister ) [virtual]
```

RegisterEventHandler(EventHandler &) Registers a specific event handler for the camera.

The camera has to be initialized first with a call to Init() before registering handlers for events.

See also

Init()

Parameters

evtHandlerToRegister	The event handler to register for the camera

Implements ICameraBase.

14.12.3.26 RegisterEventHandler() [2/2]

```
void RegisterEventHandler (
            EventHandler & evtHandlerToRegister,
            const GenICam::gcstring & eventName )  [virtual]
```

RegisterEventHandler(EventHandler &, const GenICam::gcstring&) Registers a specific event handler for the camera.

See also

Init()

Parameters

evtHandlerToRegister	The event handler to register for the camera
eventName	The event name to register

Implements ICameraBase.

14.12.3.27 SetBufferOwnership()

```
void SetBufferOwnership (
            const BufferOwnership mode )  [virtual]
```

SetBufferOwnership Sets data buffer ownership.

The data buffers can be owned by System or User. If the system owns the buffers, the memory required for the buffers are allocated and freed by the library. If user owns the buffers, the user is responsible for allocating and ultimately freeing the memory. By default, data buffers are owned by the library.

See also

GetBufferOwnership()
SetUserBuffers()

Parameters

mode	System owned or User owned buffers

Implements ICameraBase.

14.12.3.28 SetUserBuffers() [1/2]

```
void SetUserBuffers (
            void *const pMemBuffers,
            uint64_t totalSize )  [virtual]
```

SetUserBuffers Specify contiguous user allocated memory to use as data buffers.

To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: ((unsigned int) (bufferSize + 1024 - 1) / 1024) ∗ 1024 where 1024 is the USB3 packet size.

See also

GetBufferOwnership()
SetBufferOwnership()
GetUserBufferCount()
GetUserBufferSize()
GetUserBufferTotalSize()

Parameters

pMemBuffers	Pointer to memory buffers to be written to
totalSize	The total size of the memory allocated for the user buffers (in bytes)

Implements ICameraBase.

14.12.3.29 SetUserBuffers() [2/2]

```
void SetUserBuffers (
            void **const ppMemBuffers,
            const uint64_t bufferCount,
            const uint64_t bufferSize )  [virtual]
```

SetUserBuffers Specify non-contiguous user allocated memory to use as data buffers.

Each pointer to a buffer must have enough memory to hold one image. To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: ((unsigned int) (bufferSize + 1024 - 1) / 1024) ∗ 1024 where 1024 is the USB3 packet size.

See also

GetBufferOwnership()
SetBufferOwnership()
GetUserBufferCount()
GetUserBufferSize()
GetUserBufferTotalSize()

Parameters

ppMemBuffers	Pointer to pointers that each point to a single user memory buffer to be written to
bufferCount	The number of user memory buffers
bufferSize	The size of the memory allocated for each user buffer (in bytes)

Implements ICameraBase.

14.12.3.30 UnregisterEventHandler()

```
void UnregisterEventHandler (
            EventHandler & evtHandlerToUnregister )  [virtual]
```

UnregisterEventHandler Unregisters an event handler for the camera Event handlers should be unregistered first before calling camera DeInit().

Otherwise an exception will be thrown in the DeInit() call and require the user to unregister event handlers before the camera can be re-initialized again.

See also

DeInit()

Parameters

evtHandlerToUnregister	The event handler to unregister from the camera

Implements ICameraBase.

14.12.3.31 WritePort()

```
void WritePort (
            uint64_t iAddress,
            const void * pBuffer,
            size_t iSize )  [virtual]
```

Implements ICameraBase.

14.12.4 Friends And Related Function Documentation

14.12.4.1 InterfaceImpl

```
friend class InterfaceImpl  [friend]
```

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

- include/CameraBase.h

14.13 CameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for CameraList:

Collaboration diagram for CameraList:

Public Member Functions

- CameraList (void)

  Default constructor.
- virtual ~CameraList (void)

  Virtual destructor.
- CameraList (const CameraList &iface)

  Copy constructor.
- CameraList & operator= (const CameraList &iface)

  Assignment operator.
- CameraPtr operator[] (unsigned int index)

  Array subscription operators.
- unsigned int GetSize () const

  Returns the size of the camera list.
- CameraPtr GetByIndex (unsigned int index) const

Returns a pointer to a camera object at the "index".

- CameraPtr GetBySerial (std::string serialNumber) const

  Returns a pointer to a camera object with the specified serial number.
- CameraPtr GetByDeviceID (std::string deviceID) const

  Returns a pointer to a camera object with the specified device identifier.
- void Clear ()

  Clears the list of cameras and destroys their corresponding reference counted objects.
- void RemoveByIndex (unsigned int index)

  Removes a camera at "index" and destroys its corresponding reference counted object.
- void RemoveBySerial (std::string serialNumber)

  Removes a camera using its serial number and destroys its corresponding reference counted object.
- void RemoveByDeviceID (std::string deviceID)

  Removes a camera using its unique device identifier and destroys its corresponding reference counted object.
- void Append (const CameraList &list)

  Appends a copy of the camera list.

Additional Inherited Members

14.13.1 Detailed Description

Used to hold a list of camera objects.

14.13.2 Constructor & Destructor Documentation

14.13.2.1 CameraList() [1/2]

```
CameraList (
            void  )
```

Default constructor.

14.13.2.2 ~CameraList()

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

Virtual destructor.

14.13.2.3 CameraList() [2/2]

```
CameraList (
            const CameraList & iface )
```

Copy constructor.

14.13.3 Member Function Documentation

14.13.3.1 Append()

```
void Append (
            const CameraList & list )  [virtual]
```

Appends a copy of the camera list.

Parameters

list	Another CameraList object, whose elements are added to this list.

Implements ICameraList.

14.13.3.2 Clear()

```
void Clear ( )  [virtual]
```

Clears the list of cameras and destroys their corresponding reference counted objects.

This is necessary in order to clean up the parent interface. It is important that the camera list is destroyed or is cleared before calling system->ReleaseInstance() or else the call to system->ReleaseInstance() will result in an error message thrown that a reference to the camera is still held.

See also

System:ReleaseInstance()

Implements ICameraList.

14.13.3.3 GetByDeviceID()

```
CameraPtr GetByDeviceID (
            std::string deviceID ) const  [virtual]
```

Returns a pointer to a camera object with the specified device identifier.

This function will return a NULL CameraPtr if no matching device identifier is found.

Parameters

deviceID	The unique device identifier of the camera object to retrieve

Returns

A pointer to a camera object.

Implements ICameraList.

14.13.3.4 GetByIndex()

```
CameraPtr GetByIndex (
            unsigned int index ) const  [virtual]
```

Returns a pointer to a camera object at the "index".

This function will throw a Spinnaker exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

index	The index at which to retrieve the camera object

Returns

A pointer to a camera object.

Implements ICameraList.

14.13.3.5 GetBySerial()

```
CameraPtr GetBySerial (
            std::string serialNumber ) const  [virtual]
```

Returns a pointer to a camera object with the specified serial number.

This function will return a NULL CameraPtr if no matching camera serial is found.

Parameters

serialNumber	The serial number of the camera object to retrieve

Returns

A pointer to a camera object.

Implements ICameraList.

14.13.3.6 GetSize()

```
unsigned int GetSize ( ) const  [virtual]
```

Returns the size of the camera list.

The size is the number of Camera objects stored in the list.

Returns

An integer that represents the list size.

Implements ICameraList.

14.13.3.7 operator=()

```
CameraList& operator= (
            const CameraList & iface )
```

Assignment operator.

14.13.3.8 operator[]()

```
CameraPtr operator[] (
            unsigned int index )  [virtual]
```

Array subscription operators.

Implements ICameraList.

14.13.3.9 RemoveByDeviceID()

```
void RemoveByDeviceID (
            std::string deviceID )  [virtual]
```

Removes a camera using its unique device identifier and destroys its corresponding reference counted object.

This function will throw a Spinnaker exception with SPINNAKER_ERR_NOT_AVAILABLE error if no matching device identifier is found.

Parameters

deviceID	The unique device identifier of the camera object to retrieve

Implements ICameraList.

14.13.3.10 RemoveByIndex()

```
void RemoveByIndex (
            unsigned int index )  [virtual]
```

Removes a camera at "index" and destroys its corresponding reference counted object.

This function will throw a Spinnaker exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

index	The index at which to remove the Camera object

Implements ICameraList.

14.13.3.11 RemoveBySerial()

```
void RemoveBySerial (
            std::string serialNumber )  [virtual]
```

Removes a camera using its serial number and destroys its corresponding reference counted object.

This function will throw a Spinnaker exception with SPINNAKER_ERR_NOT_AVAILABLE error if no matching camera serial is found.

Parameters

serialNumber	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

14.14 CameraPtr Class Reference

A reference tracked pointer to a camera object.

Inheritance diagram for CameraPtr:

Collaboration diagram for CameraPtr:

Public Member Functions

- CameraPtr () throw ()

  Default constructor.
- CameraPtr (const int) throw ()

  Default constructor.
- CameraPtr (const long) throw ()

  Default constructor with argument.
- CameraPtr (const std::nullptr_t) throw ()

Additional Inherited Members

14.14.1 Detailed Description

A reference tracked pointer to a camera object.

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

- include/CameraPtr.h

14.15 CategoryNode Class Reference

Interface for string properties.

Inheritance diagram for CategoryNode:

Collaboration diagram for CategoryNode:

Public Member Functions

- CategoryNode ()
- CategoryNode (std::shared_ptr< Node::NodeImpl > pCategory)
- virtual ~CategoryNode ()
- virtual void GetFeatures (FeatureList_t &Features) const

  Get all features of the category (including sub-categories)
- virtual void SetReference (INode *pBase)

  overload SetReference for Value

Additional Inherited Members

14.15.1 Detailed Description

Interface for string properties.

14.15.2 Constructor & Destructor Documentation

14.15.2.1 CategoryNode() [1/2]

```
CategoryNode ( )
```

14.15.2.2 CategoryNode() [2/2]

```
CategoryNode (
            std::shared_ptr< Node::NodeImpl > pCategory )
```

14.15.2.3 ~CategoryNode()

```
virtual ~CategoryNode ( )  [virtual]
```

14.15.3 Member Function Documentation

14.15.3.1 GetFeatures()

```
virtual void GetFeatures (
            FeatureList_t & Features ) const  [virtual]
```

Get all features of the category (including sub-categories)

14.15.3.2 SetReference()

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

overload SetReference for Value

Reimplemented from ValueNode.

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

- include/SpinGenApi/CategoryNode.h

14.16 CChunkAdapter Class Reference

Connects a chunked buffer to a node map.

Inheritance diagram for CChunkAdapter:

Public Member Functions

- virtual ~CChunkAdapter ()

  Destructor.
- void AttachNodeMap (INodeMap *pNodeMap)

  Attaches to a node map and retrieves the chunk ports.
- void DetachNodeMap ()

  Detaches from the node map.
- virtual bool CheckBufferLayout (uint8_t *pBuffer, int64_t BufferLength)=0

  Checks if a buffer contains chunks in a known format.
- virtual void AttachBuffer (uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)=0

  Attaches a buffer to the matching ChunkPort.
- void DetachBuffer ()

  Detaches a buffer.
- void UpdateBuffer (uint8_t *pBaseAddress)

  Updates the base address of the buffer.
- void ClearCaches ()

  Clears the chunk caches.

Protected Member Functions

- CChunkAdapter (INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)

  Serves as default constructor.

Protected Attributes

- void * m_pChunkAdapter

14.16.1 Detailed Description

Connects a chunked buffer to a node map.

14.16.2 Constructor & Destructor Documentation

14.16.2.1 ~CChunkAdapter()

```
virtual ~CChunkAdapter ( )  [virtual]
```

Destructor.

14.16.2.2 CChunkAdapter()

```
CChunkAdapter (
            INodeMap * pNodeMap = NULL,
            int64_t MaxChunkCacheSize = -1 )  [protected]
```

Serves as default constructor.

14.16.3 Member Function Documentation

14.16.3.1 AttachBuffer()

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            int64_t BufferLength,
            AttachStatistics_t * pAttachStatistics = NULL )  [pure virtual]
```

Attaches a buffer to the matching ChunkPort.

Implemented in CChunkAdapterDcam, CChunkAdapterGeneric, CChunkAdapterGEV, and CChunkAdapterU3V.

14.16.3.2 AttachNodeMap()

```
void AttachNodeMap (
            INodeMap * pNodeMap )
```

Attaches to a node map and retrieves the chunk ports.

14.16.3.3 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
            uint8_t * pBuffer,
            int64_t BufferLength )  [pure virtual]
```

Checks if a buffer contains chunks in a known format.

Implemented in CChunkAdapterDcam, CChunkAdapterGeneric, CChunkAdapterGEV, and CChunkAdapterU3V.

14.16.3.4 ClearCaches()

```
void ClearCaches ( )
```

Clears the chunk caches.

14.16.3.5 DetachBuffer()

```
void DetachBuffer ( )
```

Detaches a buffer.

14.16.3.6 DetachNodeMap()

```
void DetachNodeMap ( )
```

Detaches from the node map.

14.16.3.7 UpdateBuffer()

```
void UpdateBuffer (
            uint8_t * pBaseAddress )
```

Updates the base address of the buffer.

14.16.4 Member Data Documentation

14.16.4.1 m_pChunkAdapter

```
void* m_pChunkAdapter [protected]
```

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

- include/SpinGenApi/ChunkAdapter.h

14.17 CChunkAdapterDcam Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterDcam:

Collaboration diagram for CChunkAdapterDcam:

Public Member Functions

- CChunkAdapterDcam (INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)

  Constructor.
- virtual ~CChunkAdapterDcam ()

  Destructor.
- virtual bool CheckBufferLayout (uint8_t *pBuffer, int64_t BufferLength)

  Checks if a buffer contains chunks in a known format.
- virtual void AttachBuffer (uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)

  Attaches a buffer to the matching ChunkPort.
- bool HasCRC (uint8_t *pBuffer, int64_t BufferLength)

  Checks if buffer has a CRC attached.
- bool CheckCRC (uint8_t *pBuffer, int64_t BufferLength)

  Checks CRC sum of buffer.

Additional Inherited Members

14.17.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

14.17.2 Constructor & Destructor Documentation

14.17.2.1 CChunkAdapterDcam()

```
CChunkAdapterDcam (
            INodeMap * pNodeMap = NULL,
            int64_t MaxChunkCacheSize = -1 )
```

Constructor.

14.17.2.2 ~CChunkAdapterDcam()

```
virtual ~CChunkAdapterDcam ( )  [virtual]
```

Destructor.

14.17.3 Member Function Documentation

14.17.3.1 AttachBuffer()

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            int64_t BufferLength,
            AttachStatistics_t * pAttachStatistics = NULL )  [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements CChunkAdapter.

14.17.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
            uint8_t * pBuffer,
            int64_t BufferLength )  [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements CChunkAdapter.

14.17.3.3 CheckCRC()

```
bool CheckCRC (
            uint8_t * pBuffer,
            int64_t BufferLength )
```

Checks CRC sum of buffer.

14.17.3.4 HasCRC()

```
bool HasCRC (
            uint8_t * pBuffer,
            int64_t BufferLength )
```

Checks if buffer has a CRC attached.

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

- include/SpinGenApi/ChunkAdapterDcam.h

14.18 CChunkAdapterGeneric Class Reference

Inheritance diagram for CChunkAdapterGeneric:

Collaboration diagram for CChunkAdapterGeneric:

Public Member Functions

- CChunkAdapterGeneric (INodeMap ∗pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
- virtual ∼CChunkAdapterGeneric ()
- virtual bool CheckBufferLayout (uint8_t ∗pBuffer, int64_t BufferLength)

  Checks if a buffer contains chunks in a known format.
- virtual void AttachBuffer (uint8_t ∗pBuffer, int64_t BufferLength, AttachStatistics_t ∗pAttachStatistics=NULL)

  Attaches a buffer to the matching ChunkPort.
- virtual void AttachBuffer (uint8_t ∗pBuffer, SingleChunkData_t ∗ChunkData, int64_t NumChunks, Attach↩Statistics_t ∗pAttachStatistics=NULL)
- virtual void AttachBuffer (uint8_t ∗pBuffer, SingleChunkDataStr_t ∗ChunkData, int64_t NumChunks, Attach↩Statistics_t ∗pAttachStatistics=NULL)

Additional Inherited Members

14.18.1 Constructor & Destructor Documentation

14.18.1.1 CChunkAdapterGeneric()

```
CChunkAdapterGeneric (
            INodeMap * pNodeMap = NULL,
            int64_t MaxChunkCacheSize = -1 )
```

14.18.1.2 ~CChunkAdapterGeneric()

```
virtual ~CChunkAdapterGeneric ( )  [virtual]
```

14.18.2 Member Function Documentation

14.18.2.1 AttachBuffer() [1/3]

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            int64_t BufferLength,
            AttachStatistics_t * pAttachStatistics = NULL )  [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements CChunkAdapter.

14.18.2.2 AttachBuffer() [2/3]

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            SingleChunkData_t * ChunkData,
            int64_t NumChunks,
            AttachStatistics_t * pAttachStatistics = NULL )  [virtual]
```

14.18.2.3 AttachBuffer() [3/3]

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            SingleChunkDataStr_t * ChunkData,
            int64_t NumChunks,
            AttachStatistics_t * pAttachStatistics = NULL )  [virtual]
```

14.18.2.4 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
            uint8_t * pBuffer,
            int64_t BufferLength )  [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements CChunkAdapter.

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

- include/SpinGenApi/ChunkAdapterGeneric.h

14.19 CChunkAdapterGEV Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterGEV:

Collaboration diagram for CChunkAdapterGEV:

Public Member Functions

- CChunkAdapterGEV (INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)

  Constructor.
- virtual ~CChunkAdapterGEV ()

  Destructor.
- virtual bool CheckBufferLayout (uint8_t *pBuffer, int64_t BufferLength)

  Checks if a buffer contains chunks in a known format.
- virtual void AttachBuffer (uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)

  Attaches a buffer to the matching ChunkPort.

Additional Inherited Members

14.19.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

14.19.2 Constructor & Destructor Documentation

14.19.2.1 CChunkAdapterGEV()

```
CChunkAdapterGEV (
            INodeMap * pNodeMap = NULL,
            int64_t MaxChunkCacheSize = -1 )
```

Constructor.

14.19.2.2 ~CChunkAdapterGEV()

```
virtual ~CChunkAdapterGEV ( )  [virtual]
```

Destructor.

14.19.3 Member Function Documentation

14.19.3.1 AttachBuffer()

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            int64_t BufferLength,
            AttachStatistics_t * pAttachStatistics = NULL )  [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements CChunkAdapter.

14.19.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
            uint8_t * pBuffer,
            int64_t BufferLength )  [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements CChunkAdapter.

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

- include/SpinGenApi/ChunkAdapterGEV.h

14.20 CChunkAdapterU3V Class Reference

Connects a chunked U3V buffer to a node map.

Inheritance diagram for CChunkAdapterU3V:

Collaboration diagram for CChunkAdapterU3V:

Public Member Functions

- CChunkAdapterU3V (INodeMap ∗pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)

  Constructor.
- virtual ∼CChunkAdapterU3V ()

  Destructor.
- virtual bool CheckBufferLayout (uint8_t ∗pBuffer, int64_t BufferLength)

  Checks if a buffer contains chunks in a known format.
- virtual void AttachBuffer (uint8_t ∗pBuffer, int64_t BufferLength, AttachStatistics_t ∗pAttachStatistics=NULL)

  Attaches a buffer to the matching ChunkPort.

Additional Inherited Members

14.20.1 Detailed Description

Connects a chunked U3V buffer to a node map.

14.20.2 Constructor & Destructor Documentation

14.20.2.1 CChunkAdapterU3V()

```
CChunkAdapterU3V (
            INodeMap * pNodeMap = NULL,
            int64_t MaxChunkCacheSize = -1 )
```

Constructor.

14.20.2.2 ~CChunkAdapterU3V()

```
virtual ~CChunkAdapterU3V ( ) [virtual]
```

Destructor.

14.20.3 Member Function Documentation

14.20.3.1 AttachBuffer()

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            int64_t BufferLength,
            AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements CChunkAdapter.

14.20.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
            uint8_t * pBuffer,
            int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements CChunkAdapter.

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

- include/SpinGenApi/ChunkAdapterU3V.h

14.21 CChunkPort Class Reference

Port attachable to a chunk in a buffer.

Inheritance diagram for CChunkPort:

Collaboration diagram for CChunkPort:

Public Member Functions

- CChunkPort (IPort ∗pPort=NULL)

  Constructor; can attach to a port.
- ∼CChunkPort ()

  Destructor; detaches from the port.
- virtual EAccessMode GetAccessMode () const

  Get the access mode of the node.
- virtual EInterfaceType GetPrincipalInterfaceType () const

  Get the type of the main interface of a node.
- virtual void Read (void ∗pBuffer, int64_t Address, int64_t Length)

  Reads a chunk of bytes from the port.
- virtual void Write (const void ∗pBuffer, int64_t Address, int64_t Length)

  Writes a chunk of bytes to the port.

- virtual void SetPortImpl (IPort *pPort)

  Called from the port node to give the chunk port a pointer to itself.
- virtual EYesNo 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

14.21.1 Detailed Description

Port attachable to a chunk in a buffer.

14.21.2 Constructor & Destructor Documentation

14.21.2.1 CChunkPort()

```
CChunkPort (
            IPort * pPort = NULL )
```

Constructor; can attach to a port.

14.21.2.2 ~CChunkPort()

```
~CChunkPort ( )
```

Destructor; detaches from the port.

14.21.3 Member Function Documentation

14.21.3.1 AttachChunk()

```
void AttachChunk (
            uint8_t * pBaseAddress,
            int64_t ChunkOffset,
            int64_t Length,
            bool Cache )
```

Attaches the Chunk to the ChunkPort.

14.21.3.2 AttachPort()

```
bool AttachPort (
            ::Spinnaker::GenApi::IPort * pPort )
```

Attaches the ChunkPort to the Port.

14.21.3.3 CheckChunkID() [1/2]

```
bool CheckChunkID (
            uint8_t * pChunkIDBuffer,
            int ChunkIDLength )
```

Checks if a ChunkID matches.

14.21.3.4 CheckChunkID() [2/2]

```
bool CheckChunkID (
            uint64_t ChunkID )
```

Checks if a ChunkID matches, version using uint64_t ID representation.

14.21.3.5 ClearCache()

```
void ClearCache ( )
```

Clears the chunk cache.

14.21.3.6 DetachChunk()

```
void DetachChunk ( )
```

Detaches the Chunk from the ChunkPort.

14.21.3.7 DetachPort()

```
void DetachPort ( )
```

Detaches the ChunkPort to the Port.

14.21.3.8 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const  [virtual]
```

Get the access mode of the node.

14.21.3.9 GetChunkIDLength()

```
int GetChunkIDLength ( )
```

Gets the ChunkID length.

14.21.3.10 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const  [virtual]
```

Get the type of the main interface of a node.

14.21.3.11 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

14.21.3.12 InvalidateNode()

```
void InvalidateNode ( )
```

14.21.3.13 Read()

```
virtual void Read (
            void * pBuffer,
            int64_t Address,
            int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

14.21.3.14 SetPortImpl()

```
virtual void SetPortImpl (
            IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

14.21.3.15 UpdateBuffer()

```
void UpdateBuffer (
            uint8_t * pBaseAddress )
```

Updates the base address of the chunk.

14.21.3.16 Write()

```
virtual void Write (
            const void * pBuffer,
            int64_t Address,
            int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

14.21.4 Member Data Documentation

14.21.4.1 m_pChunkPort

```
void* m_pChunkPort [protected]
```

14.21.4.2 m_pPort

```
CNodePtr m_pPort [protected]
```

14.21.4.3 m_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

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

- include/SpinGenApi/ChunkPort.h

14.22 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

14.22.1 Detailed Description

template<class EnumT>
class Spinnaker::GenApi::CEnumerationTRef< EnumT >

Interface for string properties.

14.22.2 Constructor & Destructor Documentation

14.22.2.1 CEnumerationTRef() [1/2]

```
CEnumerationTRef ( )
```

14.22.2.2 CEnumerationTRef() [2/2]

```
CEnumerationTRef (
            std::shared_ptr< Node::NodeImpl > pEnumeration )
```

14.22.2.3 ~CEnumerationTRef()

```
virtual ~CEnumerationTRef ( )  [virtual]
```

14.22.3 Member Function Documentation

14.22.3.1 GetCurrentEntry()

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

Get the current entry.

Reimplemented from EnumNode.

14.22.3.2 GetEntry() [1/2]

```
virtual IEnumEntry* GetEntry (
            const EnumT Value )  [virtual]
```

returns the EnumEntry object belonging to the Value

14.22.3.3 GetEntry() [2/2]

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

Get an entry node by its IntValue.

Reimplemented from EnumNode.

14.22.3.4 GetValue()

```
virtual EnumT GetValue (
            bool Verify = false,
            bool IgnoreCache = false )  [virtual]
```

Get 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

14.22.3.5 operator()()

```
virtual EnumT operator() ( ) [virtual]
```

Get node value.

14.22.3.6 operator=() [1/2]

```
virtual IEnumeration& operator= (
            EnumT Value ) [virtual]
```

Set node value.

14.22.3.7 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.

14.22.3.8 SetEnumReference()

```
virtual void SetEnumReference (
            int Index,
            GenICam::gcstring Name ) [virtual]
```

sets the Enum value corresponding to a value

14.22.3.9 SetNumEnums()

```
virtual void SetNumEnums (
            int NumEnums ) [virtual]
```

sets the number of enum values

14.22.3.10 SetReference()

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

overload SetReference for EnumerationT

Reimplemented from EnumNode.

14.22.3.11 SetValue()

```
virtual void SetValue (
            EnumT 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/EnumNodeT.h

14.23 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 emap.
- virtual void DeliverMessage (const uint8_t msg[ ], uint32_t numBytes)=0

  Deliver message.

Protected Attributes

- void * m_pEventAdapter

14.23.1 Detailed Description

Delivers Events to ports.

14.23.2 Constructor & Destructor Documentation

14.23.2.1 CEventAdapter()

```
CEventAdapter (
            INodeMap * pNodeMap = NULL )
```

Constructor.

14.23.2.2 ~CEventAdapter()

```
virtual ~CEventAdapter ( )  [virtual]
```

Destructor.

14.23.3 Member Function Documentation

14.23.3.1 AttachNodeMap()

```
virtual void AttachNodeMap (
            INodeMap * pNodeMap )  [virtual]
```

Attaches to a node map and retrieves the chunk ports.

14.23.3.2 DeliverMessage()

```
virtual void DeliverMessage (
            const uint8_t msg[],
            uint32_t numBytes )  [pure virtual]
```

Deliver message.

Implemented in CEventAdapterGEV, CEventAdapterU3V, CEventAdapter1394, and CEventAdapterGeneric.

14.23.3.3 DetachNodeMap()

```
virtual void DetachNodeMap ( )  [virtual]
```

Detaches from the node emap.

14.23.4 Member Data Documentation

14.23.4.1 m_pEventAdapter

```
void* m_pEventAdapter  [protected]
```

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

- include/SpinGenApi/EventAdapter.h

14.24 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

14.24.1 Detailed Description

Distribute the events to the node map.

14.24.2 Constructor & Destructor Documentation

14.24.2.1 CEventAdapter1394()

```
CEventAdapter1394 (
            INodeMap * pNodeMap = NULL )  [explicit]
```

constructor

14.24.2.2 ~CEventAdapter1394()

```
virtual ~CEventAdapter1394 ( )  [virtual]
```

14.24.3 Member Function Documentation

14.24.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
            EventData1394 & Event,
            uint32_t numBytes )
```

distributes events to node map

14.24.3.2 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

14.25 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

14.25.1 Detailed Description

Connects a generic event to a node map.

14.25.2 Constructor & Destructor Documentation

14.25.2.1 CEventAdapterGeneric()

```
CEventAdapterGeneric (
            INodeMap * pNodeMap = NULL )
```

Constructor.

14.25.2.2 ~CEventAdapterGeneric()

```
virtual ~CEventAdapterGeneric ( ) [virtual]
```

Destructor.

14.25.3 Member Function Documentation

14.25.3.1 DeliverMessage() [1/3]

```
virtual void DeliverMessage (
            const uint8_t msg[],
            uint32_t numBytes ) [virtual]
```

Deliver message.

Implements CEventAdapter.

14.25.3.2 DeliverMessage() [2/3]

```
virtual void DeliverMessage (
            const uint8_t msg[],
            uint32_t numBytes,
            const GenICam::gcstring & EventID ) [virtual]
```

14.25.3.3 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

14.26 CEventAdapterGEV Class Reference

Connects a GigE Event to a node map.

Inheritance diagram for CEventAdapterGEV:

Collaboration diagram for CEventAdapterGEV:

Public Member Functions

- CEventAdapterGEV (INodeMap *pNodeMap=NULL)

  Constructor.
- virtual ~CEventAdapterGEV ()

  Destructor.
- virtual void DeliverMessage (const uint8_t msg[ ], uint32_t numBytes)

  Deliver message.
- void DeliverEventMessage (const GVCP_EVENT_REQUEST *pEvent)

  Delivers the Events listed in the Event packet.
- void DeliverEventMessage (const GVCP_EVENTDATA_REQUEST *pEventData)

  Delivers the Event + Data listed in the EventData packet.

Additional Inherited Members

14.26.1 Detailed Description

Connects a GigE Event to a node map.

14.26.2 Constructor & Destructor Documentation

14.26.2.1 CEventAdapterGEV()

```
CEventAdapterGEV (
            INodeMap * pNodeMap = NULL )
```

Constructor.

14.26.2.2 ~CEventAdapterGEV()

```
virtual ~CEventAdapterGEV ( )  [virtual]
```

Destructor.

14.26.3 Member Function Documentation

14.26.3.1 DeliverEventMessage() [1/2]

```
void DeliverEventMessage (
            const GVCP_EVENT_REQUEST * pEvent )
```

Delivers the Events listed in the Event packet.

14.26.3.2 DeliverEventMessage() [2/2]

```
void DeliverEventMessage (
            const GVCP_EVENTDATA_REQUEST * pEventData )
```

Delivers the Event + Data listed in the EventData packet.

14.26.3.3 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

14.27 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

14.27.1 Detailed Description

Connects a U3V Event to a node map.

14.27.2 Constructor & Destructor Documentation

14.27.2.1 CEventAdapterU3V()

```
CEventAdapterU3V (
            INodeMap * pNodeMap = NULL )
```

Constructor.

14.27.2.2 ~CEventAdapterU3V()

```
virtual ~CEventAdapterU3V ( )  [virtual]
```

Destructor.

14.27.3 Member Function Documentation

14.27.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
            const U3V_EVENT_MESSAGE * pEventMessage )
```

Delivers the Event + Data listed in the packet.

14.27.3.2 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

14.28 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

14.28.1 Detailed Description

Port attachable to an event.

14.28.2 Constructor & Destructor Documentation

14.28.2.1 CEventPort()

```
CEventPort (
            INode * pNode = NULL )
```

Constructor; can attach to a node.

14.28.2.2 ~CEventPort()

```
~CEventPort ( )
```

Destructor; detaches from the port.

14.28.3 Member Function Documentation

14.28.3.1 AttachEvent()

```
void AttachEvent (
            uint8_t * pBaseAddress,
            int64_t Length )
```

Attaches the an Event to the EventPort.

14.28.3.2 AttachNode()

```
bool AttachNode (
            ::Spinnaker::GenApi::INode * pNode )
```

Attaches to the Node.

14.28.3.3 CheckEventID() [1/2]

```
bool CheckEventID (
            uint8_t * pEventIDBuffer,
            int EventIDLength )
```

Checks if a EventID matches.

14.28.3.4 CheckEventID() [2/2]

```
bool CheckEventID (
            uint64_t EventID )
```

Checks if a EventID matches, version using uint64_t ID representation.

14.28.3.5 DetachEvent()

```
void DetachEvent ( )
```

Detaches the Event from the EventPort.

14.28.3.6 DetachNode()

```
void DetachNode ( )
```

Detaches from the Node.

14.28.3.7 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const  [virtual]
```

Get the access mode of the node.

14.28.3.8 GetEventIDLength()

```
int GetEventIDLength ( )
```

Gets the EventID length.

14.28.3.9 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const  [virtual]
```

Get the type of the main interface of a node.

14.28.3.10 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

14.28.3.11 InvalidateNode()

```
void InvalidateNode ( )
```

14.28.3.12 Read()

```
virtual void Read (
            void * pBuffer,
            int64_t Address,
            int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

14.28.3.13 SetPortImpl()

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

Called from the port node to give the chunk port a pointer to itself.

14.28.3.14 Write()

```
virtual void Write (
            const void * pBuffer,
            int64_t Address,
            int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

14.28.4 Member Data Documentation

14.28.4.1 m_pEventPort

```
void* m_pEventPort [protected]
```

14.28.4.2 m_pNode

```
CNodePtr m_pNode [protected]
```

14.28.4.3 m_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

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

- include/SpinGenApi/EventPort.h

14.29 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 ()

14.29.1 Detailed Description

Bag holding streamable features of a nodetree.

14.29.2 Constructor & Destructor Documentation

14.29.2.1 CFeatureBag()

```
CFeatureBag ( )
```

14.29.2.2 ~CFeatureBag()

```
virtual ~CFeatureBag ( )  [virtual]
```

14.29.3 Member Function Documentation

14.29.3.1 GetFeatureBagHandle()

```
void* GetFeatureBagHandle ( )
```

14.29.3.2 LoadFromBag()

```
bool LoadFromBag (
            INodeMap * pNodeMap,
            bool Verify = true,
            GenICam::gcstring_vector * pErrorList = NULL )
```

Loads the features from the bag to the node tree.

Parameters

pNodeMap	The node map
Verify	If true, all streamable features are read back
pErrorList	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.

14.29.3.3 operator==()

```
bool operator== (
            const CFeatureBag & FeatureBag ) const
```

compares the content of two feature bags

14.29.3.4 PersistFeature()

```
virtual void PersistFeature (
            IValue & item )  [virtual]
```

Stores a feature.

14.29.3.5 SetInfo()

```
virtual void SetInfo (
            GenICam::gcstring & Info )  [virtual]
```

sets information about the node map

14.29.3.6 StoreToBag()

```
int64_t StoreToBag (
            INodeMap * pNodeMap,
            const int MaxNumPersistSkriptEntries = -1 )
```

Stores the streamable nodes to this feature bag.

Parameters

pNodeMap	The node map to persist
MaxNumPersistSkriptEntries	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

14.30 CFloatPtr Class Reference

SmartPointer for IFloat interface pointer.

Inheritance diagram for CFloatPtr:

Collaboration diagram for CFloatPtr:

Public Member Functions

- CFloatPtr () throw ()

  Default constructor.
- CFloatPtr (IBase *pB)

  Constructor from IBase pointer type.
- void operator= (IBase *pB)

  Assign IBase Pointer.
- IInteger * GetIntAlias ()

  gets the interface of an integer alias node.
- IEnumeration * GetEnumAlias ()

  gets the interface of an enum alias node.

Additional Inherited Members

14.30.1 Detailed Description

SmartPointer for IFloat interface pointer.

14.30.2 Constructor & Destructor Documentation

14.30.2.1 CFloatPtr() [1/2]

```
CFloatPtr ( ) throw )   [inline]
```

Default constructor.

14.30.2.2 CFloatPtr() [2/2]

```
CFloatPtr (
            IBase * pB )  [inline]
```

Constructor from IBase pointer type.

14.30.3 Member Function Documentation

14.30.3.1 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( )  [inline]
```

gets the interface of an enum alias node.

14.30.3.2 GetIntAlias()

```
IInteger* GetIntAlias ( )  [inline]
```

gets the interface of an integer alias node.

14.30.3.3 operator=()

```
void operator= (
            IBase * pB )  [inline]
```

Assign IBase Pointer.

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

- include/SpinGenApi/Pointer.h

14.31 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 *)

14.31.1 Detailed Description

Empty base class used by class CNodeMapRef as generic template argument.

14.31.2 Member Function Documentation

14.31.2.1 _Initialize()

```
virtual void _Initialize (
            GenApi::INodeMap *  )  [inline], [protected], [virtual]
```

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

- include/SpinGenApi/NodeMapRef.h

14.32 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

14.32.1 Detailed Description

Named global lock which can be used over process boundaries.

14.32.2 Constructor & Destructor Documentation

14.32.2.1 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.

14.32.2.2 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.

14.32.2.3 ~CGlobalLock()

```
~CGlobalLock ( )
```

14.32.3 Member Function Documentation

14.32.3.1 IsValid()

```
bool IsValid (
            void  ) const
```

tests whether the lock is valid

14.32.3.2 Lock()

```
bool Lock (
            unsigned int timeout_ms )
```

enters the lock (may block)

14.32.3.3 TryLock()

```
bool TryLock (
            void  )
```

tries to enter the lock and returns immediately when not possible

14.32.3.4 Unlock()

```
void Unlock (
            void  )
```

leaves the lock

14.32.4 Member Data Documentation

14.32.4.1 m_DebugCount

```
long m_DebugCount  [mutable], [protected]
```

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

- include/SpinGenApi/GCSynch.h

14.33 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

14.33.1 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

14.33.2 Constructor & Destructor Documentation

14.33.2.1 CGlobalLockUnlocker()

```
CGlobalLockUnlocker (
            CGlobalLock & lock )  [inline]
```

14.33.2.2 ~CGlobalLockUnlocker()

```
~CGlobalLockUnlocker ( )  [inline]
```

14.33.3 Member Function Documentation

14.33.3.1 UnlockEarly()

```
void UnlockEarly (
            void  )  [inline]
```

This function allows to unlock the object early before the object is destroyed.

14.33.4 Member Data Documentation

14.33.4.1 m_enabled

```
bool m_enabled [protected]
```

14.33.4.2 m_Lock

```
CGlobalLock& m_Lock [protected]
```

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

- include/SpinGenApi/GCSynch.h

14.34 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 GetInferenceFrameId () const

  Description: Returns the frame ID associated with the most recent inference result.
- int64_t GetInferenceResult () const

  Description: Returns the chunk data inference result.
- float64_t GetInferenceConfidence () const

  Description: Returns the chunk data inference confidence percentage.
- InferenceBoundingBoxResult GetInferenceBoundingBoxResult () const

  Description: Returns the chunk inference bounding box result data.

Additional Inherited Members

14.34.1 Detailed Description

The chunk data which contains additional information about an image.

14.34.2 Constructor & Destructor Documentation

14.34.2.1 ChunkData() [1/2]

```
ChunkData ( )
```

14.34.2.2 ChunkData() [2/2]

```
ChunkData (
            const ChunkData & src )
```

14.34.2.3 ~ChunkData()

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

14.34.3 Member Function Documentation

14.34.3.1 GetBlackLevel()

```
float64_t GetBlackLevel ( ) const [virtual]
```

Description: Returns the black level used to capture the image.

Visibility:

Implements IChunkData.

14.34.3.2 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.

14.34.3.3 GetCRC()

```
int64_t GetCRC ( ) const [virtual]
```

Description: Returns the CRC of the image payload.

Visibility:

Implements IChunkData.

14.34.3.4 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.

14.34.3.5 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.

14.34.3.6 GetExposureTime()

```
float64_t GetExposureTime ( ) const  [virtual]
```

Description: Returns the exposure time used to capture the image.

Visibility:

Implements IChunkData.

14.34.3.7 GetFrameID()

```
int64_t GetFrameID ( ) const  [virtual]
```

Description: Returns the image frame ID.

Visibility:

Implements IChunkData.

14.34.3.8 GetGain()

```
float64_t GetGain ( ) const  [virtual]
```

Description: Returns the gain used to capture the image.

Visibility:

Implements IChunkData.

14.34.3.9 GetHeight()

```
int64_t GetHeight ( ) const  [virtual]
```

Description: Returns the height of the image included in the payload.

Visibility:

Implements IChunkData.

14.34.3.10 GetImage()

```
int64_t GetImage ( ) const  [virtual]
```

Description: Returns the image payload.

Visibility:

Implements IChunkData.

14.34.3.11 GetInferenceBoundingBoxResult()

```
InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const  [virtual]
```

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

Implements IChunkData.

14.34.3.12 GetInferenceConfidence()

```
float64_t GetInferenceConfidence ( ) const  [virtual]
```

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

Implements IChunkData.

14.34.3.13 GetInferenceFrameId()

```
int64_t GetInferenceFrameId ( ) const  [virtual]
```

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

Implements IChunkData.

14.34.3.14 GetInferenceResult()

```
int64_t GetInferenceResult ( ) const  [virtual]
```

Description: Returns the chunk data inference result.

Visibility: Expert

Implements IChunkData.

14.34.3.15 GetLinePitch()

```
int64_t GetLinePitch ( ) const  [virtual]
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

Implements IChunkData.

14.34.3.16 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.

14.34.3.17 GetOffsetX()

```
int64_t GetOffsetX ( ) const  [virtual]
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

Implements IChunkData.

14.34.3.18 GetOffsetY()

```
int64_t GetOffsetY ( ) const  [virtual]
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

Implements IChunkData.

14.34.3.19 GetPartSelector()

```
int64_t GetPartSelector ( ) const  [virtual]
```

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

Implements IChunkData.

14.34.3.20 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.

14.34.3.21 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.

14.34.3.22 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.

14.34.3.23 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.

14.34.3.24 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.

14.34.3.25 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.

14.34.3.26 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.

14.34.3.27 GetScan3dInvalidDataValue()

```
float64_t GetScan3dInvalidDataValue ( ) const  [virtual]
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

Implements IChunkData.

14.34.3.28 GetScan3dTransformValue()

```
float64_t GetScan3dTransformValue ( ) const  [virtual]
```

Description: Returns the transform value.

Visibility: Expert

Implements IChunkData.

14.34.3.29 GetScanLineSelector()

```
int64_t GetScanLineSelector ( ) const  [virtual]
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

Implements IChunkData.

14.34.3.30 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.

14.34.3.31 GetSerialDataLength()

```
int64_t GetSerialDataLength ( ) const  [virtual]
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

Implements IChunkData.

14.34.3.32 GetStreamChannelID()

```
int64_t GetStreamChannelID ( ) const  [virtual]
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

Implements IChunkData.

14.34.3.33 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.

14.34.3.34 GetTimestamp()

```
int64_t GetTimestamp ( ) const  [virtual]
```

Description: Returns the Timestamp of the image.

Visibility:

Implements IChunkData.

14.34.3.35 GetTimestampLatchValue()

```
int64_t GetTimestampLatchValue ( ) const  [virtual]
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

Implements IChunkData.

14.34.3.36 GetTransferBlockID()

```
int64_t GetTransferBlockID ( ) const  [virtual]
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

Implements IChunkData.

14.34.3.37 GetTransferQueueCurrentBlockCount()

```
int64_t GetTransferQueueCurrentBlockCount ( ) const  [virtual]
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

Implements IChunkData.

14.34.3.38 GetWidth()

```
int64_t GetWidth ( ) const  [virtual]
```

Description: Returns the width of the image included in the payload.

Visibility:

Implements IChunkData.

14.34.3.39 SetChunks()

```
void SetChunks (
            GenApi::INodeMap & pNodeMap )  [virtual]
```

Implements IChunkData.

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

- include/ChunkData.h

14.35 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

14.35.1 Detailed Description

A lock class.

14.35.2 Constructor & Destructor Documentation

14.35.2.1 CLock()

```
CLock ( )
```

Constructor.

14.35.2.2 ~CLock()

`~CLock ( )`

Destructor.

14.35.3 Member Function Documentation

14.35.3.1 Lock()

`void Lock ( )`

enters the critical section (may block)

14.35.3.2 TryLock()

`bool TryLock ( )`

tries to enter the critical section; returns true if successful

14.35.3.3 Unlock()

`void Unlock ( )`

leaves the critical section

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

- include/SpinGenApi/GCSynch.h

14.36 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

14.36.1 Detailed Description

A lock class.

14.36.2 Constructor & Destructor Documentation

14.36.2.1 CLock() [1/2]

```
CLock ( )
```

Constructor.

14.36.2.2 CLock() [2/2]

```
CLock (
            void * pLock )
```

Constructor.

14.36.2.3 ~CLock()

```
~CLock ( )
```

Destructor.

14.36.3 Member Function Documentation

14.36.3.1 Lock()

```
void Lock ( )
```

enters the critical section (may block)

14.36.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

14.36.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

14.36.4 Friends And Related Function Documentation

14.36.4.1 NodeMap

```
friend class NodeMap [friend]
```

14.36.5 Member Data Documentation

14.36.5.1 m_bOwnLock

```
bool m_bOwnLock [protected]
```

14.36.5.2 m_lock

```
void* m_lock [protected]
```

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

- include/SpinGenApi/Synch.h

14.37 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:

Collaboration diagram for CLockEx:

Additional Inherited Members

14.37.1 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

14.38 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

14.38.1 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

14.38.2 Member Data Documentation

14.38.2.1 m_lockEx

```
void* m_lockEx  [protected]
```

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

- include/SpinGenApi/Synch.h

14.39 CNodeCallback Class Reference

callback body instance for INode pointers

Inheritance diagram for CNodeCallback:

Public Member Functions

- CNodeCallback (INode *pNode, ECallbackType CallbackType)
- virtual ~CNodeCallback ()

  virtual destructor
- virtual void operator() (ECallbackType CallbackType) const =0

  fires the callback if th type is right
- virtual void Destroy ()=0

  destroys the object
- INode * GetNode ()

  returns the node the callback is registered to
- ECallbackType GetCallbackType ()

Protected Attributes

- INode * m_pNode

  the node were the callback is installed
- ECallbackType m_CallbackType

  the type of the callback

14.39.1 Detailed Description

callback body instance for INode pointers

14.39.2 Constructor & Destructor Documentation

14.39.2.1 CNodeCallback()

```
CNodeCallback (
            INode * pNode,
            ECallbackType CallbackType )  [inline]
```

14.39.2.2 ~CNodeCallback()

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

virtual destructor

14.39.3 Member Function Documentation

14.39.3.1 Destroy()

```
virtual void Destroy ( ) [pure virtual]
```

destroys the object

Implemented in Member_NodeCallback< Client, Member >, and Function_NodeCallback< Function >.

14.39.3.2 GetCallbackType()

```
ECallbackType GetCallbackType ( ) [inline]
```

14.39.3.3 GetNode()

```
INode* GetNode ( ) [inline]
```

returns the node the callback is registered to

14.39.3.4 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 >.

14.39.4 Member Data Documentation

14.39.4.1 m_CallbackType

```
ECallbackType m_CallbackType [protected]
```

the type of the callback

14.39.4.2 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

14.40 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 CNode↩ DataMap (not part of the public interface).
- bool IsPreprocessed () const

  Can be used to check whether the Preprocess() processing step has been performed.
- void ReleaseCameraDescriptionFileData ()

  Advanced: Releases any in constructors provided camera description file data buffers or files.
- bool IsCameraDescriptionFileDataReleased () const

  Can be used to check whether the ReleaseCameraDescriptionFileData() processing step has been performed.
- INodeMap ∗ CreateNodeMap (const GenICam::gcstring &DeviceName="Device", bool DoReleaseCamera↩ DescriptionFileData=true)

  Creates a node map from the preprocessed memory internal representation of the camera description file(s).
- INodeMap ∗ CreateNodeMap (CLock &UserProvidedLock, const GenICam::gcstring &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true)

  Creates a node map from the preprocessed memory internal representation of the camera description file(s).
- void GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions) const
- GenICam::gcstring ToString () const

  Outputs the pre-processed node map in string form (for debug purpose)
- GenICam::gcstring ToXml () const

  Outputs the pre-processed node map in XML form (mainly for debug purpose)
- void GetNodeStatistics (NodeStatistics_t &NodeStatistics)
- const GenICam::gcstring ApplyStyleSheet (const GenICam::gcstring &StyleSheetFileName)

  Applies a style sheet to the pre-processed node map.

Static Public Member Functions

- static INodeMap ∗ CreateEmptyNodeMap ()

  Creates an empty node map usable as placeholder, e.g.
- static bool ClearCache ()

  Deletes all preprocessed camera description files from the cache.
- static CNodeDataMap ∗ CreateNodeDataFromNodeMap (INodeMap ∗pNodeMap)

14.40.1 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.

14.40.2 Constructor & Destructor Documentation

14.40.2.1 CNodeMapFactory() [1/5]

```
CNodeMapFactory ( )
```

Creates an empty node map factory for assigning a non-empty node map factory later.

14.40.2.2 ~CNodeMapFactory()

```
virtual ~CNodeMapFactory ( )  [virtual]
```

Destroys the node map factory data if all references to the data have been released.

14.40.2.3 CNodeMapFactory() [2/5]

```
CNodeMapFactory (
            const CNodeMapFactory &  )
```

Creates another reference to the node map factory data.

No data is copied.

14.40.2.4 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	*FileType*	Defines how the camera description file is stored, e.g. as zipped XML text.
in	*FileName*	The full path of the camera description file to process.
in	*CacheUsage*	Defines if and how to use the cache for preprocessed camera description files.
in	*SuppressStringsOnLoad*	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 IsCamera↩DescriptionFileDataReleased() method can be used to check if releasing has been done.

14.40.2.5 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	*ContentType*	Defines how the camera description file is stored, e.g. as zipped XML text.
in	*pData*	The pointer to the camera description file data.
in	*DataSize*	The size of the camera description file data.
in	*CacheUsage*	Defines if and how to use the cache for preprocessed camera description files.
in	*SuppressStringsOnLoad*	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.

14.40.2.6 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	*XmlData*	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 cdfData; //... fill cdfData ... CNodeMapFactory factory(ContentType_Xml, cfdData.c_str( ), cfdData.size()); // Create the node map. The node map can be destroyed using the IDestroy interface later. INodeMap* pNodeMap = factory.CreateNodeMap(); // The next step is attaching the device port (not shown).`
in	*CacheUsage*	Defines if and how to use the cache for preprocessed camera description files.
in	*SuppressStringsOnLoad*	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.

14.40.3 Member Function Documentation

14.40.3.1 AddInjectionData()

```
void AddInjectionData (
            CNodeMapFactory & injectionData )
```

Adds a node map factory representing a camera description file to inject.

Parameters

in	*injectionData*	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.

14.40.3.2 ApplyStyleSheet()

```
const GenICam::gcstring ApplyStyleSheet (
            const GenICam::gcstring & StyleSheetFileName )
```

Applies a style sheet to the pre-processed node map.

14.40.3.3 ClearCache()

```
static bool ClearCache ( )  [static]
```

Deletes all preprocessed camera description files from the cache.

14.40.3.4 CreateEmptyNodeMap()

```
static INodeMap* CreateEmptyNodeMap ( )  [static]
```

Creates an empty node map usable as placeholder, e.g.

if certain features are not supported by a module.

14.40.3.5 CreateNodeDataFromNodeMap()

```
static CNodeDataMap* CreateNodeDataFromNodeMap (
            INodeMap * pNodeMap )  [static]
```

14.40.3.6 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.

14.40.3.7 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.

14.40.3.8 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	*SubTreeRootNodeName*	The root of the branch to extract, e.g. "ChunkData".
in	*doRenameToRoot*	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.

14.40.3.9 GetNodeStatistics()

```
void GetNodeStatistics (
            NodeStatistics_t & NodeStatistics )
```

14.40.3.10 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

14.40.3.11 IsCameraDescriptionFileDataReleased()

```
bool IsCameraDescriptionFileDataReleased ( ) const
```

Can be used to check whether the ReleaseCameraDescriptionFileData() processing step has been performed.

14.40.3.12 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.

14.40.3.13 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).

14.40.3.14 IsPreprocessed()

```
bool IsPreprocessed ( ) const
```

Can be used to check whether the Preprocess() processing step has been performed.

14.40.3.15 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.

14.40.3.16 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.

14.40.3.17 Preprocess()

```
void Preprocess ( )
```

Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the C↩ NodeDataMap (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.

14.40.3.18 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.

14.40.3.19 ToString()

```
GenICam::gcstring ToString ( ) const
```

Outputs the pre-processed node map in string form (for debug purpose)

14.40.3.20 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

14.41 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

14.41.1 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.

14.41.2 Constructor & Destructor Documentation

14.41.2.1 CNodeMapRef() [1/3]

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

Constructor.

14.41.2.2 CNodeMapRef() [2/3]

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

Constructor.

14.41.2.3 CNodeMapRef() [3/3]

```
CNodeMapRef (
            const CNodeMapRef & Them )  [inline]
```

Copy constructor.

14.41.3 Member Function Documentation

14.41.3.1 operator=() [1/2]

```
CNodeMapRef& operator= (
            const CNodeMapRef & Them )  [inline]
```

Assignment.

14.41.3.2 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

14.42 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.

14.42.1 Detailed Description

template<class TCameraParams>
class Spinnaker::GenApi::CNodeMapRefT< TCameraParams >

Smartpointer template for NodeMaps with create function.

Parameters

TCameraParams	The camera specific parameter class (auto generated from camera xml file)

14.42.2 Member Function Documentation

14.42.2.1 _ClearXMLCache()

```
static bool _ClearXMLCache ( )  [static]
```

Clears the cache of the camera description files.

14.42.2.2 _Connect() [1/2]

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

Connects a port to a port node with given name.

14.42.2.3 _Connect() [2/2]

```
virtual bool _Connect (
            IPort * pPort )  [virtual]
```

Connects a port to the standard port "Device".

14.42.2.4 _GetDeviceName()

```
virtual GenICam::gcstring _GetDeviceName ( )  [virtual]
```

Get device name.

14.42.2.5 _GetNode()

```
virtual INode* _GetNode (
            const GenICam::gcstring & key )  [virtual]
```

Retrieves the node from the central map by name.

14.42.2.6 _GetNodes()

```
virtual void _GetNodes (
            NodeList_t & Nodes )  [virtual]
```

Retrieves all nodes in the node map.

14.42.2.7 _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

14.42.2.8 _InvalidateNodes()

```
virtual void _InvalidateNodes ( )  [virtual]
```

Invalidates all nodes.

14.42.2.9 _LoadXMLFromFile()

```
void _LoadXMLFromFile (
            const GenICam::gcstring & FileName )
```

Creates the object from a XML file with given file name.

14.42.2.10 _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.

14.42.2.11 _LoadXMLFromString()

```
void _LoadXMLFromString (
            const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

14.42.2.12 _LoadXMLFromStringInject()

```
void _LoadXMLFromStringInject (
            const GenICam::gcstring & TargetXMLDataconst,
            const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

14.42.2.13 _LoadXMLFromZIPData()

```
void _LoadXMLFromZIPData (
            const void * zipData,
            size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

14.42.2.14 _LoadXMLFromZIPFile()

```
void _LoadXMLFromZIPFile (
            const GenICam::gcstring & ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

14.42.2.15 _Poll()

```
virtual void _Poll (
            int64_t ElapsedTime )  [virtual]
```

Fires nodes which have a polling time.

14.42.3 Member Data Documentation

14.42.3.1 _Ptr

```
INodeMap* _Ptr
```

Pointer to the NodeMap.

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

- include/SpinGenApi/NodeMapRef.h

14.43 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

14.43.1 Detailed Description

Interface for string properties.

14.43.2 Constructor & Destructor Documentation

14.43.2.1 CommandNode() [1/2]

```
CommandNode ( )
```

14.43.2.2 CommandNode() [2/2]

```
CommandNode (
            std::shared_ptr< Node::NodeImpl > pCommand )
```

14.43.2.3 ~CommandNode()

```
virtual ~CommandNode ( ) [virtual]
```

14.43.3 Member Function Documentation

14.43.3.1 Execute()

```
virtual void Execute (
            bool Verify = true ) [virtual]
```

Execute the command.

Parameters

Verify	Enables AccessMode and Range verification (default = true)

14.43.3.2 IsDone()

```
virtual bool IsDone (
            bool Verify = true )  [virtual]
```

Query whether the command is executed.

Parameters

Verify	Enables Range verification (default = false). The AccessMode is always checked

Returns

True if the Execute command has finished; false otherwise

14.43.3.3 operator()()

```
virtual void operator() ( )  [virtual]
```

Execute the command.

14.43.3.4 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

14.44 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 ()

14.44.1 Detailed Description

Definition of a simple Counter class.

14.44.2 Constructor & Destructor Documentation

14.44.2.1 Counter()

```
Counter ( ) [inline]
```

14.44.3 Member Function Documentation

14.44.3.1 GetValue()

```
unsigned int GetValue ( ) const [inline]
```

14.44.3.2 IsZero()

```
bool IsZero ( ) [inline]
```

14.44.3.3 operator unsigned int()

```
operator unsigned int ( ) [inline]
```

14.44.3.4 operator++() [1/2]

```
unsigned int operator++ ( ) [inline]
```

14.44.3.5 operator++() [2/2]

```
unsigned int operator++ (
            int ) [inline]
```

14.44.3.6 operator--() [1/2]

```
unsigned int operator-- (
            int ) [inline]
```

14.44.3.7 operator--() [2/2]

```
unsigned int operator-- ( ) [inline]
```

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

- include/SpinGenApi/Counter.h

14.45 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 inequal
- bool operator!= (T *pT) const

  pointer inequal
- bool operator!= (const long int nMustBeNull) const

  pointer inequal
- bool operator!= (const int nMustBeNull) const

  pointer inequal
- bool operator!= (const std::nullptr_t nullPtr) const

  pointer inequal

Protected Attributes

- T * m_pT

  Underlying raw pointer.

14.45.1 Detailed Description

template<class T, class B = IBase>
class Spinnaker::GenApi::CPointer< T, B >

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

14.45.2 Constructor & Destructor Documentation

14.45.2.1 CPointer() [1/2]

```
CPointer (
            void  )  [inline]
```

Default constructor.

14.45.2.2 CPointer() [2/2]

```
CPointer (
            B * pB )  [inline]
```

Constructor from INode pointer type.

14.45.2.3 ~CPointer()

```
virtual ~CPointer (
            void  )  [inline], [virtual]
```

14.45.3 Member Function Documentation

14.45.3.1 IsValid()

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

true if the pointer is valid

14.45.3.2 operator bool()

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

true if the pointer is valid

14.45.3.3 operator T*()

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

Dereferencing.

14.45.3.4 operator"!=() [1/5]

```
bool operator!= (
            const CPointer< T, B > & rT ) const  [inline]
```

pointer inequal

14.45.3.5 operator"!=() [2/5]

```
bool operator!= (
            T * pT ) const  [inline]
```

pointer inequal

14.45.3.6 operator"!=() [3/5]

```
bool operator!= (
            const long int nMustBeNull ) const  [inline]
```

pointer inequal

14.45.3.7 operator"!=() [4/5]

```
bool operator!= (
            const int nMustBeNull ) const  [inline]
```

pointer inequal

14.45.3.8 operator"!=() [5/5]

```
bool operator!= (
            const std::nullptr_t nullPtr ) const  [inline]
```

pointer inequal

14.45.3.9 operator()()

```
T& operator() (
            void ) const [inline]
```

Dereferencing.

14.45.3.10 operator*()

```
T& operator* (
            void ) const [inline]
```

Dereferencing.

14.45.3.11 operator->()

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

Dereferencing.

14.45.3.12 operator=()

```
void operator= (
            B * pB ) [inline]
```

Assign INode Pointer.

14.45.3.13 operator==() [1/3]

```
bool operator== (
            T * pT ) const [inline]
```

pointer equal

14.45.3.14 operator==() [2/3]

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

pointer equal

14.45.3.15 operator==() [3/3]

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

pointer equal

14.45.4 Member Data Documentation

14.45.4.1 m_pT

```
T* m_pT [protected]
```

Underlying raw pointer.

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

- include/SpinGenApi/Pointer.h

14.46 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.

14.46.1 Detailed Description

Standard implementation for a port.

14.46.2 Constructor & Destructor Documentation

14.46.2.1 CPortImpl()

```
CPortImpl ( ) [inline]
```

Constructor.

14.46.2.2 ~CPortImpl()

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

Destructor.

14.46.3 Member Function Documentation

14.46.3.1 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 >.

14.46.3.2 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

14.46.3.3 InvalidateNode()

```
void InvalidateNode ( ) [inline]
```

14.46.3.4 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 >.

14.46.3.5 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.

14.46.3.6 SetPortImpl()

```
virtual void SetPortImpl (
            IPort * pPort )  [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented in CRegisterPortImpl.

14.46.3.7 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 >.

14.46.4 Member Data Documentation

14.46.4.1 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

14.47 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

14.47.1 Detailed Description

Container holding a list of port write commands.

14.47.2 Constructor & Destructor Documentation

14.47.2.1 CPortWriteList()

```
CPortWriteList ( )
```

Constructor.

14.47.2.2 ~CPortWriteList()

```
~CPortWriteList ( )
```

Destructor.

14.47.3 Member Function Documentation

14.47.3.1 GetCookie()

```
virtual int64_t GetCookie ( ) [virtual]
```

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

14.47.3.2 GetPortWriteListHandle()

```
void* GetPortWriteListHandle ( )
```

14.47.3.3 Replay()

```
virtual void Replay (
            IPort * pPort ) [virtual]
```

Replays the write command to the given port interface.

14.47.3.4 SetCookie()

```
virtual void SetCookie (
            const int64_t Value ) [virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

14.47.3.5 Write()

```
virtual void Write (
            const void * pBuffer,
            int64_t Address,
            int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

14.47.4 Member Data Documentation

14.47.4.1 m_pWriteList

```
void* m_pWriteList  [protected]
```

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

- include/SpinGenApi/PortWriteList.h

14.48 CpuUsageInfo Struct Reference

Public Attributes

- bool dummy

14.48.1 Member Data Documentation

14.48.1.1 dummy

```
bool dummy
```

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

- src/GigEVisionPerformance/CpuUtil.h

14.49 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

14.49.1 Detailed Description

Standard implementation for a port using a register based transport layer.

14.49.2 Constructor & Destructor Documentation

14.49.2.1 CRegisterPortImpl()

```
CRegisterPortImpl (
            int MaxNumQuadlets = 1,
            bool TransportLayerSwapsEndianess = false )  [inline]
```

Constructor.

14.49.2.2 ~CRegisterPortImpl()

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

Destructor.

14.49.3 Member Function Documentation

14.49.3.1 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.

14.49.3.2 Read()

```
virtual void Read (
            void * pBuffer,
            int64_t Address,
            int64_t Length )  [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements CPortImpl.

14.49.3.3 ReadRegister()

```
virtual void ReadRegister (
            uint32_t * pRegisters,
            int64_t Address,
            int64_t Length )  [pure virtual]
```

Reads an array of quadlets from the port.

14.49.3.4 SetPortImpl()

```
virtual void SetPortImpl (
            IPort * pPort )  [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented from CPortImpl.

14.49.3.5 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.

14.49.3.6 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

14.50 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 GenICam::gcstring ToString ()
- virtual void GetSelectorList (FeatureList_t &SelectorList, bool Incremental=false)

Additional Inherited Members

14.50.1 Detailed Description

The set of selectors selecting a given node.

14.50.2 Constructor & Destructor Documentation

14.50.2.1 CSelectorSet()

```
CSelectorSet (
            IBase * pBase )
```

Constructor.

Parameters

pBase	Feature selected by the selector set

14.50.2.2 ~CSelectorSet()

```
~CSelectorSet ( )
```

Destructor.

14.50.3 Member Function Documentation

14.50.3.1 GetSelectorList()

```
virtual void GetSelectorList (
            FeatureList_t & SelectorList,
            bool Incremental = false )  [virtual]
```

14.50.3.2 IsEmpty()

```
bool IsEmpty ( )
```

returns true if no selectors are present

14.50.3.3 Restore()

```
virtual void Restore ( )  [virtual]
```

14.50.3.4 SetFirst()

```
virtual bool SetFirst ( )  [virtual]
```

14.50.3.5 SetNext()

```
virtual bool SetNext (
            bool Tick = true )  [virtual]
```

14.50.3.6 ToString()

```
virtual GenICam::gcstring ToString ( )  [virtual]
```

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

- include/SpinGenApi/SelectorSet.h

14.51 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 GetPrincipalInterfaceType () const

  Get the type of the main interface of a node.
- virtual void Read (void *pBuffer, int64_t Address, int64_t Length)

  Reads a chunk of bytes from the port.

- virtual void Write (const void *pBuffer, int64_t Address, int64_t Length)

  Writes a chunk of bytes to the port.
- void MemSet (const char FillValue)
- void ResetStatistics ()

  Resets the read/write statistics.
- int64_t GetNumReads ()

  Returns the number of reads since lastReset Statistics.
- int64_t GetNumWrites ()

  Returns the number of writes since lastReset Statistics.

Protected Attributes

- int64_t m_NumReads

  Number of reads since last reset.
- int64_t m_NumWrites

  Number of writes since last reset.
- int64_t m_BaseAddress

  the base address used for the struct

14.51.1 Detailed Description

template<class CDataStruct>
class Spinnaker::GenApi::CTestPortStruct< CDataStruct >

Implements a register spaces based on a C++ struct.

14.51.2 Constructor & Destructor Documentation

14.51.2.1 CTestPortStruct()

```
CTestPortStruct (
            int64_t BaseAddress = 0 )  [inline]
```

14.51.3 Member Function Documentation

14.51.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const  [inline], [virtual]
```

Get the access mode of the node.

Implements CPortImpl.

14.51.3.2 GetNumReads()

```
int64_t GetNumReads ( ) [inline]
```

Returns the number of reads since lastReset Statistics.

14.51.3.3 GetNumWrites()

```
int64_t GetNumWrites ( ) [inline]
```

Returns the number of writes since lastReset Statistics.

14.51.3.4 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [inline], [virtual]
```

Get the type of the main interface of a node.

14.51.3.5 MemSet()

```
void MemSet (
            const char FillValue ) [inline]
```

14.51.3.6 Read()

```
virtual void Read (
            void * pBuffer,
            int64_t Address,
            int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements CPortImpl.

14.51.3.7 ResetStatistics()

```
void ResetStatistics ( ) [inline]
```

Resets the read/write statistics.

14.51.3.8 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.

14.51.4 Member Data Documentation

14.51.4.1 m_BaseAddress

```
int64_t m_BaseAddress  [protected]
```

the base address used for the struct

14.51.4.2 m_NumReads

```
int64_t m_NumReads  [protected]
```

Number of reads since last reset.

14.51.4.3 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

14.52 DCAM_CHECKSUM Struct Reference

Public Attributes

- uint32_t CRCChecksum

14.52.1 Member Data Documentation

14.52.1.1 CRCChecksum

```
uint32_t CRCChecksum
```

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

- include/SpinGenApi/ChunkAdapterDcam.h

14.53 DCAM_CHUNK_TRAILER Struct Reference

Public Attributes

- SPIN_GUID ChunkID
- uint32_t ChunkLength
- uint32_t InverseChunkLength

14.53.1 Member Data Documentation

14.53.1.1 ChunkID

```
SPIN_GUID ChunkID
```

14.53.1.2 ChunkLength

```
uint32_t ChunkLength
```

14.53.1.3 InverseChunkLength

```
uint32_t InverseChunkLength
```

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

- include/SpinGenApi/ChunkAdapterDcam.h

14.54 DeviceArrivalEventHandler Class Reference

An event handler for capturing the device arrival event.

Inheritance diagram for DeviceArrivalEventHandler:

Collaboration diagram for DeviceArrivalEventHandler:

Public Member Functions

- DeviceArrivalEventHandler ()

  Default constructor.
- virtual ~DeviceArrivalEventHandler ()

  Virtual destructor.
- virtual void OnDeviceArrival (uint64_t serialNumber)=0

  Callback to the device arrival event.

Protected Member Functions

- DeviceArrivalEventHandler & operator= (const DeviceArrivalEventHandler &)

  Assignment operator.

Additional Inherited Members

14.54.1 Detailed Description

An event handler for capturing the device arrival event.

14.54.2 Constructor & Destructor Documentation

14.54.2.1 DeviceArrivalEventHandler()

```
DeviceArrivalEventHandler ( )
```

Default constructor.

14.54.2.2 ~DeviceArrivalEventHandler()

```
virtual ~DeviceArrivalEventHandler ( )  [virtual]
```

Virtual destructor.

14.54.3 Member Function Documentation

14.54.3.1 OnDeviceArrival()

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

Callback to the device arrival event.

Implements IDeviceArrivalEventHandler.

14.54.3.2 operator=()

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

Assignment operator.

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

- include/DeviceArrivalEventHandler.h

14.55 DeviceEventHandler Class Reference

A handler to device events.

Inheritance diagram for DeviceEventHandler:

Collaboration diagram for DeviceEventHandler:

Public Member Functions

- DeviceEventHandler ()

  Default constructor.
- virtual ~DeviceEventHandler ()

  Virtual destructor.
- virtual void OnDeviceEvent (Spinnaker::GenICam::gcstring eventName)=0

  Device event callback.
- uint64_t GetDeviceEventId () const

  Get the ID of the device event.
- GenICam::gcstring GetDeviceEventName () const

  Get the name of the device event.

Protected Member Functions

- DeviceEventHandler & operator= (const DeviceEventHandler &)

  Assignment operator.

Additional Inherited Members

14.55.1 Detailed Description

A handler to device events.

14.55.2 Constructor & Destructor Documentation

14.55.2.1 DeviceEventHandler()

```
DeviceEventHandler ( )
```

Default constructor.

14.55.2.2 ~DeviceEventHandler()

```
virtual ~DeviceEventHandler ( )  [virtual]
```

Virtual destructor.

14.55.3 Member Function Documentation

14.55.3.1 GetDeviceEventId()

```
uint64_t GetDeviceEventId ( ) const  [virtual]
```

Get the ID of the device event.

Returns

The device event ID

Implements IDeviceEventHandler.

14.55.3.2 GetDeviceEventName()

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

Get the name of the device event.

Returns

The device event name

Implements IDeviceEventHandler.

14.55.3.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (
            Spinnaker::GenICam::gcstring eventName )  [pure virtual]
```

Device event callback.

Parameters

eventName	The name of the event

Implements IDeviceEventHandler.

Implemented in DeviceEventHandlerImpl.

14.55.3.4 operator=()

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

Assignment operator.

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

- include/DeviceEventHandler.h

14.56 DeviceEventHandlerImpl Class Reference

Inheritance diagram for DeviceEventHandlerImpl:

Collaboration diagram for DeviceEventHandlerImpl:

Public Member Functions

- DeviceEventHandlerImpl (gcstring eventName)
- ~DeviceEventHandlerImpl ()
- void OnDeviceEvent (gcstring eventName)

  Device event callback.

Additional Inherited Members

14.56.1 Constructor & Destructor Documentation

14.56.1.1 DeviceEventHandlerImpl()

```
DeviceEventHandlerImpl (
            gcstring eventName ) [inline]
```

14.56.1.2 ~DeviceEventHandlerImpl()

```
~DeviceEventHandlerImpl ( ) [inline]
```

14.56.2 Member Function Documentation

14.56.2.1 OnDeviceEvent()

```
void OnDeviceEvent (
            gcstring eventName )  [inline], [virtual]
```

Device event callback.

Parameters

eventName	The name of the event

Implements DeviceEventHandler.

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

- src/DeviceEvents/DeviceEvents.cpp

14.57 DeviceRemovalEventHandler Class Reference

An event handler for capturing the device removal event.

Inheritance diagram for DeviceRemovalEventHandler:

Collaboration diagram for DeviceRemovalEventHandler:

Public Member Functions

- DeviceRemovalEventHandler ()

  Default Constructor.
- virtual ~DeviceRemovalEventHandler ()

  Virtual Destructor.
- virtual void OnDeviceRemoval (uint64_t serialNumber)=0

  Device removal event callback.

Protected Member Functions

- DeviceRemovalEventHandler & operator= (const DeviceRemovalEventHandler &)

  Assignment operator.

Additional Inherited Members

14.57.1 Detailed Description

An event handler for capturing the device removal event.

14.57.2 Constructor & Destructor Documentation

14.57.2.1 DeviceRemovalEventHandler()

```
DeviceRemovalEventHandler ( )
```

Default Constructor.

14.57.2.2 ~DeviceRemovalEventHandler()

```
virtual ~DeviceRemovalEventHandler ( )  [virtual]
```

Virtual Destructor.

14.57.3 Member Function Documentation

14.57.3.1 OnDeviceRemoval()

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

Device removal event callback.

Parameters

serialNumber	The serial number of the device removed

Implements IDeviceRemovalEventHandler.

14.57.3.2 operator=()

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

Assignment operator.

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

- include/DeviceRemovalEventHandler.h

14.58 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

14.58.1 Detailed Description

Vector of doubles with reference counting.

14.58.2 Constructor & Destructor Documentation

14.58.2.1 double_autovector_t() [1/3]

```
double_autovector_t ( )
```

14.58.2.2 double_autovector_t() [2/3]

```
double_autovector_t (
            const double_autovector_t & obj )
```

14.58.2.3 double_autovector_t() [3/3]

```
double_autovector_t (
            size_t n )  [explicit]
```

14.58.2.4 ~double_autovector_t()

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

14.58.3 Member Function Documentation

14.58.3.1 operator delete()

```
void operator delete (
            void * pWhere )
```

14.58.3.2 operator new()

```
void* operator new (
            size_t uiSize )
```

14.58.3.3 operator=()

```
double_autovector_t& operator= (
            const double_autovector_t & obj )
```

14.58.3.4 operator[]() [1/2]

```
double& operator[] (
            size_t uiIndex )
```

14.58.3.5 operator[]() [2/2]

```
const double& operator[] (
            size_t uiIndex ) const
```

14.58.3.6 size()

```
size_t size ( ) const
```

14.58.4 Member Data Documentation

14.58.4.1 _pCount

```
ATOMIC_VARIABLE* _pCount  [protected]
```

14.58.4.2 _pv

```
std::vector<double>* _pv  [protected]
```

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

- include/SpinGenApi/Autovector.h

14.59 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.

14.59.1 Detailed Description

Holds conversion methods for the access mode enumeration.

14.59.2 Member Function Documentation

14.59.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EAccessMode * pValue )  [static]
```

Converts a string to enum value.

14.59.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EAccessMode * pValue )  [static]
```

Converts a string to an int32_t property.

14.59.2.3 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

14.60 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.

14.60.1 Detailed Description

Holds conversion methods for the caching mode enumeration.

14.60.2 Member Function Documentation

14.60.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            ECachingMode * pValue )  [static]
```

Converts a string to enum value.

14.60.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            ECachingMode * pValue )  [static]
```

14.60.2.3 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

14.61 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.

14.61.1 Detailed Description

Holds conversion methods for the notation type of floats.

14.61.2 Member Function Documentation

14.61.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EDisplayNotation * pValue )  [static]
```

Converts a string to enum value.

14.61.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EDisplayNotation * pValue )  [static]
```

Converts a string to an int32_t property.

14.61.2.3 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

14.62 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.

14.62.1 Detailed Description

Holds conversion methods for the endianess enumeration.

14.62.2 Member Function Documentation

14.62.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EEndianess * pValue )  [static]
```

Converts a string to enum value.

14.62.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EEndianess * pValue )  [static]
```

Converts a string to an int32_t property.

14.62.2.3 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

14.63 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.

14.63.1 Detailed Description

helper class converting EGenApiSchemaVersion from and to string

14.63.2 Member Function Documentation

14.63.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EGenApiSchemaVersion * pValue )  [static]
```

Converts a string to enum value.

14.63.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EGenApiSchemaVersion * pValue )  [static]
```

Converts a string to an int32_t property.

14.63.2.3 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

14.64 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.

14.64.1 Detailed Description

Holds conversion methods for the notation type of floats.

14.64.2 Member Function Documentation

14.64.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EInputDirection * pValue )  [static]
```

Converts a string to enum value.

14.64.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EInputDirection * pValue )  [static]
```

Converts a string to an int32_t property.

14.64.2.3 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

14.65 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.

14.65.1 Detailed Description

Holds conversion methods for the namespace enumeration.

14.65.2 Member Function Documentation

14.65.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            ENameSpace * pValue )  [static]
```

Converts a string to enum value.

14.65.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            ENameSpace * pValue )  [static]
```

Converts a string to an int32_t property.

14.65.2.3 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

14.66 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

14.66.1 Detailed Description

Interface for string properties.

14.66.2 Constructor & Destructor Documentation

14.66.2.1 EnumEntryNode() [1/2]

```
EnumEntryNode ( )
```

14.66.2.2 EnumEntryNode() [2/2]

```
EnumEntryNode (
            std::shared_ptr< Node::NodeImpl > pEnumEntry )
```

14.66.2.3 ~EnumEntryNode()

```
virtual ~EnumEntryNode ( ) [virtual]
```

14.66.3 Member Function Documentation

14.66.3.1 GetNumericValue()

```
virtual double GetNumericValue ( )  [virtual]
```

Get double number associated with the entry.

14.66.3.2 GetSymbolic()

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

Get symbolic enum value.

14.66.3.3 GetValue()

```
virtual int64_t GetValue ( )  [virtual]
```

Get numeric enum value.

14.66.3.4 IsSelfClearing()

```
virtual bool IsSelfClearing ( )  [virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

14.66.3.5 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

14.67 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

14.67.1 Detailed Description

Interface for string properties.

14.67.2 Constructor & Destructor Documentation

14.67.2.1 EnumNode() [1/2]

```
EnumNode ( )
```

14.67.2.2 EnumNode() [2/2]

```
EnumNode (
            std::shared_ptr< Node::NodeImpl > pEnumeration )
```

14.67.2.3 ~EnumNode()

```
virtual ~EnumNode ( )  [virtual]
```

14.67.3 Member Function Documentation

14.67.3.1 GetCurrentEntry()

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

Get the current entry.

Reimplemented in CEnumerationTRef< EnumT >.

14.67.3.2 GetEntries()

```
virtual void GetEntries (
            NodeList_t & Entries )  [virtual]
```

Get list of entry nodes.

14.67.3.3 GetEntry()

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

Get an entry node by its IntValue.

Reimplemented in CEnumerationTRef< EnumT >.

14.67.3.4 GetEntryByName()

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

Get an entry node by name.

14.67.3.5 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

14.67.3.6 GetSymbolics()

```
virtual void GetSymbolics (
            StringList_t & Symbolics )  [virtual]
```

Get list of symbolic Values.

14.67.3.7 operator∗()

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

Get string node value.

14.67.3.8 operator=()

```
virtual IEnumeration& operator= (
            const GenICam::gcstring & ValueStr )  [virtual]
```

Set string node value.

Reimplemented in CEnumerationTRef< EnumT >.

14.67.3.9 SetIntValue()

```
virtual void SetIntValue (
            int64_t Value,
            bool Verify = true )  [virtual]
```

Set integer node value.

Parameters

Value	The value to set
Verify	Enables AccessMode and Range verification (default = true)

14.67.3.10 SetReference()

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

overload SetReference for Enumeration

Reimplemented from ValueNode.

Reimplemented in CEnumerationTRef< EnumT >.

14.67.4 Member Data Documentation

14.67.4.1 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

14.68 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.

14.68.1 Detailed Description

Holds conversion methods for the representation enumeration.

14.68.2 Member Function Documentation

14.68.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            ERepresentation * pValue )  [static]
```

Converts a string to enum value.

14.68.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            ERepresentation * pValue )  [static]
```

Converts a string to an int32_t property.

14.68.2.3 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

14.69 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.

14.69.1 Detailed Description

Holds conversion methods for the sign enumeration.

14.69.2 Member Function Documentation

14.69.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            ESign * pValue )  [static]
```

Converts a string to enum value.

14.69.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            ESign * pValue )  [static]
```

Converts a string to an int32_t property.

14.69.2.3 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

14.70 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.

14.70.1 Detailed Description

Holds conversion methods for the converter formulas.

14.70.2 Member Function Documentation

14.70.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            ESlope * pValue )  [static]
```

Converts a string to enum value.

14.70.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            ESlope * pValue )  [static]
```

Converts a string to an int32_t property.

14.70.2.3 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

14.71 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.

14.71.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

14.71.2 Member Function Documentation

14.71.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EStandardNameSpace * pValue )  [static]
```

Converts a string to enum value.

14.71.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EStandardNameSpace * pValue )  [static]
```

Converts a string to an int32_t property.

14.71.2.3 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

14.72 EventHandler Class Reference

The base class for all event handler types.

Inheritance diagram for EventHandler:

Public Member Functions

- virtual ~EventHandler ()

  Virtual Destructor.
- void SetEventType (EventType eventType)

  Sets the event type.
- EventType GetEventType ()

  Gets the event type.
- const uint8_t * GetEventPayloadData ()

  Gets the event payload data.
- const size_t GetEventPayloadDataSize ()

  Gets the event payload data size.

Protected Member Functions

- EventHandler ()
- EventHandler & operator= (const EventHandler &)
- void SetEventPayload (uint8_t *offset, size_t length)

Protected Attributes

- EventData * m_pEventData

Friends

- class EventProcessor
- class IDataStream
- class Stream

14.72.1 Detailed Description

The base class for all event handler types.

14.72.2 Constructor & Destructor Documentation

14.72.2.1 ~EventHandler()

```
virtual ~EventHandler ( ) [virtual]
```

Virtual Destructor.

14.72.2.2 EventHandler()

```
EventHandler ( ) [protected]
```

14.72.3 Member Function Documentation

14.72.3.1 GetEventPayloadData()

```
const uint8_t* GetEventPayloadData ( )
```

Gets the event payload data.

Returns

The event payload data

14.72.3.2 GetEventPayloadDataSize()

```
const size_t GetEventPayloadDataSize ( )
```

Gets the event payload data size.

Returns

The event payload data size

14.72.3.3 GetEventType()

```
EventType GetEventType ( )
```

Gets the event type.

Returns

The event type

14.72.3.4 operator=()

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

14.72.3.5 SetEventPayload()

```
void SetEventPayload (
            uint8_t * offset,
            size_t length )  [protected]
```

14.72.3.6 SetEventType()

```
void SetEventType (
            EventType eventType )
```

Sets the event type.

Parameters

eventType	The event type

14.72.4 Friends And Related Function Documentation

14.72.4.1 EventProcessor

```
friend class EventProcessor  [friend]
```

14.72.4.2 IDataStream

```
friend class IDataStream [friend]
```

14.72.4.3 Stream

```
friend class Stream [friend]
```

14.72.5 Member Data Documentation

14.72.5.1 m_pEventData

```
EventData* m_pEventData [protected]
```

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

- include/EventHandler.h

14.73 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.

14.73.1 Detailed Description

Holds conversion methods for the visibility enumeration.

14.73.2 Member Function Documentation

14.73.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EVisibility * pValue )  [static]
```

Converts a string to enum value.

14.73.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EVisibility * pValue )  [static]
```

Converts a string to an int32_t property.

14.73.2.3 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

14.74 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 ∗build↩ Time, 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

14.74.1 Detailed Description

The Exception object represents an error that is returned from the library.

Overloaded operators allow comparisons against other Exception objects.

14.74.2 Constructor & Destructor Documentation

14.74.2.1 Exception() [1/4]

```
Exception ( )
```

Default constructor.

14.74.2.2 Exception() [2/4]

```
Exception (
            int line,
            const char * fileName,
            const char * funcName,
            const char * errMsg,
            Error err )
```

Message constructor.

Parameters

line	Line number where the exception is thrown
fileName	Name of the file called
funcName	Name of the function called
errMsg	A pointer to the exception message string
err	Error code

14.74.2.3 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

line	Line number where the exception is thrown
fileName	Name of the file called

Parameters

funcName	Name of the function called
buildDate	Build date
buildTime	Build time
errMsg	A pointer to the exception message string
err	Error code

14.74.2.4 Exception() [4/4]

```
Exception (
            const Exception & except )
```

Copy constructor.

14.74.2.5 ~Exception()

```
virtual ~Exception ( ) throw )   [virtual]
```

Default destructor.

14.74.3 Member Function Documentation

14.74.3.1 GetBuildDate()

```
const char* GetBuildDate ( ) const
```

14.74.3.2 GetBuildTime()

```
const char* GetBuildTime ( ) const
```

14.74.3.3 GetError()

```
Error GetError ( ) const
```

14.74.3.4 GetErrorMessage()

```
const char* GetErrorMessage ( ) const
```

Accessor Functions.

14.74.3.5 GetFileName()

```
const char* GetFileName ( ) const
```

14.74.3.6 GetFullErrorMessage()

```
const char* GetFullErrorMessage ( ) const
```

Gets the error code and full error message including the line, file, function, build date, and time.

14.74.3.7 GetFunctionName()

```
const char* GetFunctionName ( ) const
```

14.74.3.8 GetLineNumber()

```
int GetLineNumber ( ) const
```

14.74.3.9 operator"!=()

```
bool operator!= (
            const Error err ) const
```

Inequality operator.

14.74.3.10 operator=()

```
Exception& operator= (
            const Exception & except )
```

Assignment operator.

14.74.3.11 operator==()

```
bool operator== (
            const Error err ) const
```

Equality operator.

14.74.3.12 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

14.75 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.

14.75.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

14.75.2 Member Function Documentation

14.75.2.1 FromString()

```
static bool FromString (
            const GenICam::gcstring & ValueStr,
            EYesNo * pValue )  [static]
```

Converts a string to enum value.

14.75.2.2 ToString() [1/2]

```
static void ToString (
            GenICam::gcstring & ValueStr,
            EYesNo * pValue )  [static]
```

Converts a string to an int32_t property.

14.75.2.3 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

14.76 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.

14.76.1 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

14.76.2 Constructor & Destructor Documentation

14.76.2.1 FileProtocolAdapter()

```
FileProtocolAdapter ( )
```

Constructor.

14.76.2.2 ~FileProtocolAdapter()

```
virtual ~FileProtocolAdapter ( )  [virtual]
```

14.76.3 Member Function Documentation

14.76.3.1 attach()

```
bool attach (
            ::Spinnaker::GenApi::INodeMap * pInterface )
```

attach file protocol adapter to NodeMap

Parameters

pInterface	NodeMap of the device to which the FileProtocolAdapter is attached

Returns

true if attach was successful, false if not

14.76.3.2 closeFile()

```
bool closeFile (
            const char * pFileName )
```

close a file on the device

Parameters

pFileName	filename of the file to open. The filename must exist in the Enumeration FileSelector

Returns

true on success, false on error

14.76.3.3 deleteFile()

```
bool deleteFile (
            const char * pFileName )
```

Delete the content of the file.

Parameters

pFileName	filename of the file to open. The filename must exist in the Enumeration FileSelector

Returns

true on success, false on error

14.76.3.4 getBufSize()

```
int64_t getBufSize (
            const char * pFileName,
            std::ios_base::openmode mode )
```

fetch max FileAccessBuffer length for a file

Parameters

pFileName	filename of the file to open. The filename must exist in the Enumeration FileSelector
mode	mode to open the file. The mode must exist in the Enumeration FileOpenMode

Returns

max length of FileAccessBuffer in the given mode on the given file

14.76.3.5 openFile()

```
bool openFile (
            const char * pFileName,
            std::ios_base::openmode mode )
```

open a file on the device

Parameters

pFileName	filename of the file to open. The filename must exist in the Enumeration FileSelector
mode	mode to open the file. The mode must exist in the Enumeration FileOpenMode

Returns

true on success, false on error

14.76.3.6 read()

```
std::streamsize read (
            char * buf,
            int64_t offs,
            std::streamsize len,
            const char * pFileName )
```

read data from the device into a buffer

Parameters

buf	target buffer
offs	offset in the device file to read from
len	count of bytes to read
pFileName	filename of the file to write into The filename must exist in the Enumeration FileSelector

Returns

count of bytes successfully read

14.76.3.7 write()

```
std::streamsize write (
            const char * buf,
            int64_t offs,
            int64_t len,
            const char * pFileName )
```

writes data into a file.

Parameters

buf	source buffer
offs	offset into the device file
len	count of bytes to write
pFileName	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

14.77 FloatNode Class Reference

Interface for string properties.

Inheritance diagram for FloatNode:

Collaboration diagram for FloatNode:

Public Member Functions

- FloatNode ()
- FloatNode (std::shared_ptr< Node::NodeImpl > pFloat)
- virtual ~FloatNode ()
- virtual void SetValue (double Value, bool Verify=true)

  Set node value.
- virtual IFloat & operator= (double Value)

  Set node value.
- virtual double GetValue (bool Verify=false, bool IgnoreCache=false)

  Get node value.
- virtual double operator() ()

  Get node value.
- virtual double operator* ()

  Get node value.
- virtual double GetMin ()

  Get minimum value allowed.
- virtual double GetMax ()

  Get maximum value allowed.
- virtual bool HasInc ()

  True if the float has a constant increment.
- virtual EIncMode GetIncMode ()

  Get increment mode.
- virtual double GetInc ()

  Get the constant increment if there is any.
- virtual double_autovector_t GetListOfValidValues (bool bounded=true)

  Get list of valid value.
- virtual ERepresentation GetRepresentation ()

  Get recommended representation.
- virtual 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

14.77.1 Detailed Description

Interface for string properties.

14.77.2 Constructor & Destructor Documentation

14.77.2.1 FloatNode() [1/2]

```
FloatNode ( )
```

14.77.2.2 FloatNode() [2/2]

```
FloatNode (
            std::shared_ptr< Node::NodeImpl > pFloat )
```

14.77.2.3 ~FloatNode()

```
virtual ~FloatNode ( ) [virtual]
```

14.77.3 Member Function Documentation

14.77.3.1 GetDisplayNotation()

```
virtual EDisplayNotation GetDisplayNotation ( ) const [virtual]
```

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

14.77.3.2 GetDisplayPrecision()

```
virtual int64_t GetDisplayPrecision ( ) const [virtual]
```

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

14.77.3.3 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( )
```

gets the interface of an alias node.

14.77.3.4 GetInc()

```
virtual double GetInc ( )  [virtual]
```

Get the constant increment if there is any.

14.77.3.5 GetIncMode()

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

Get increment mode.

14.77.3.6 GetIntAlias()

```
IInteger* GetIntAlias ( )
```

gets the interface of an alias node.

14.77.3.7 GetListOfValidValues()

```
virtual double_autovector_t GetListOfValidValues (
            bool bounded = true )  [virtual]
```

Get list of valid value.

14.77.3.8 GetMax()

```
virtual double GetMax ( )  [virtual]
```

Get maximum value allowed.

14.77.3.9 GetMin()

```
virtual double GetMin ( )  [virtual]
```

Get minimum value allowed.

14.77.3.10 GetRepresentation()

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

Get recommended representation.

14.77.3.11 GetUnit()

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

Get the physical unit name.

14.77.3.12 GetValue()

```
virtual double GetValue (
            bool Verify = false,
            bool IgnoreCache = false ) [virtual]
```

Get 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

14.77.3.13 HasInc()

```
virtual bool HasInc ( ) [virtual]
```

True if the float has a constant increment.

14.77.3.14 ImposeMax()

```
virtual void ImposeMax (
            double Value ) [virtual]
```

Restrict maximum value.

14.77.3.15 ImposeMin()

```
virtual void ImposeMin (
            double Value )  [virtual]
```

Restrict minimum value.

14.77.3.16 operator()()

```
virtual double operator() ( )  [virtual]
```

Get node value.

14.77.3.17 operator*()

```
virtual double operator* ( )  [virtual]
```

Get node value.

14.77.3.18 operator=()

```
virtual IFloat& operator= (
            double Value )  [virtual]
```

Set node value.

14.77.3.19 SetReference()

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

overload SetReference for Float

Reimplemented from ValueNode.

Reimplemented in FloatRegNode.

14.77.3.20 SetValue()

```
virtual void SetValue (
            double 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/FloatNode.h

14.78 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

14.78.1 Detailed Description

Interface for string properties.

14.78.2 Constructor & Destructor Documentation

14.78.2.1 FloatRegNode() [1/2]

```
FloatRegNode ( )
```

14.78.2.2 FloatRegNode() [2/2]

```
FloatRegNode (
            std::shared_ptr< Node::NodeImpl > pFloat )
```

14.78.2.3 ~FloatRegNode()

```
virtual ~FloatRegNode ( ) [virtual]
```

14.78.3 Member Function Documentation

14.78.3.1 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

14.79 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

14.79.1 Detailed Description

template<class Function>
class Spinnaker::GenApi::Function_NodeCallback< Function >

Container for a function pointer.

14.79.2 Constructor & Destructor Documentation

14.79.2.1 Function_NodeCallback()

```
Function_NodeCallback (
            INode * pNode,
            const Function & function,
            ECallbackType CallbackType )  [inline]
```

Constructor.

14.79.3 Member Function Documentation

14.79.3.1 Destroy()

```
virtual void Destroy ( )  [inline], [virtual]
```

destroys teh object

Implements CNodeCallback.

14.79.3.2 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

14.80 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)

14.80.1 Constructor & Destructor Documentation

14.80.1.1 gcstring() [1/5]

```
gcstring ( )
```

14.80.1.2 gcstring() [2/5]

```
gcstring (
            const char * pc )
```

14.80.1.3 gcstring() [3/5]

```
gcstring (
            const char * pc,
            size_t n )
```

14.80.1.4 gcstring() [4/5]

```
gcstring (
            size_t count,
            char ch )
```

14.80.1.5 gcstring() [5/5]

```
gcstring (
            const gcstring & str )
```

14.80.1.6 ~gcstring()

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

14.80.2 Member Function Documentation

14.80.2.1 _npos()

```
static size_t _npos (
            void ) [static]
```

14.80.2.2 append() [1/2]

```
virtual gcstring& append (
            const gcstring & str ) [virtual]
```

14.80.2.3 append() [2/2]

```
virtual gcstring& append (
            size_t count,
            char ch ) [virtual]
```

14.80.2.4 assign() [1/4]

```
virtual gcstring& assign (
            const gcstring & str ) [virtual]
```

14.80.2.5 assign() [2/4]

```
virtual gcstring& assign (
            size_t count,
            char ch )  [virtual]
```

14.80.2.6 assign() [3/4]

```
virtual gcstring& assign (
            const char * pc )  [virtual]
```

14.80.2.7 assign() [4/4]

```
virtual gcstring& assign (
            const char * pc,
            size_t n )  [virtual]
```

14.80.2.8 c_str()

```
virtual const char* c_str (
            void  ) const  [virtual]
```

14.80.2.9 compare()

```
virtual int compare (
            const gcstring & str ) const  [virtual]
```

14.80.2.10 empty()

```
virtual bool empty (
            void  ) const  [virtual]
```

14.80.2.11 find() [1/5]

```
virtual size_t find (
            char ch,
            size_t offset = 0 ) const  [virtual]
```

14.80.2.12 find() [2/5]

```
virtual size_t find (
            const gcstring & str,
            size_t offset = 0 ) const  [virtual]
```

14.80.2.13 find() [3/5]

```
virtual size_t find (
            const gcstring & str,
            size_t offset,
            size_t count ) const  [virtual]
```

14.80.2.14 find() [4/5]

```
virtual size_t find (
            const char * pc,
            size_t offset = 0 ) const  [virtual]
```

14.80.2.15 find() [5/5]

```
virtual size_t find (
            const char * pc,
            size_t offset,
            size_t count ) const  [virtual]
```

14.80.2.16 find_first_not_of()

```
virtual size_t find_first_not_of (
            const gcstring & str,
            size_t offset = 0 ) const  [virtual]
```

14.80.2.17 find_first_of()

```
virtual size_t find_first_of (
            const gcstring & str,
            size_t offset = 0 ) const  [virtual]
```

14.80.2.18 length()

```
virtual size_t length (
            void  ) const  [virtual]
```

14.80.2.19 max_size()

```
virtual size_t max_size ( ) const  [virtual]
```

14.80.2.20 operator const char *()

```
operator const char * (
            void  ) const
```

14.80.2.21 operator delete() [1/2]

```
void operator delete (
            void * pWhere )
```

14.80.2.22 operator delete() [2/2]

```
void operator delete (
            void * pWhere,
            void * pNewWhere )
```

14.80.2.23 operator new() [1/2]

```
void* operator new (
            size_t uiSize )
```

14.80.2.24 operator new() [2/2]

```
void* operator new (
            size_t uiSize,
            void * pWhere )
```

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

```
bool operator!= (
            const gcstring & str ) const
```

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

```
bool operator!= (
            const char * pc ) const
```

14.80.2.27 operator+=() [1/5]

```
gcstring& operator+= (
            const gcstring & str )
```

14.80.2.28 operator+=() [2/5]

```
gcstring operator+= (
            const gcstring & str ) const
```

14.80.2.29 operator+=() [3/5]

```
gcstring& operator+= (
            const char * pc )
```

14.80.2.30 operator+=() [4/5]

```
gcstring& operator+= (
            char ch )
```

14.80.2.31 operator+=() [5/5]

```
gcstring operator+= (
            char ch ) const
```

14.80.2.32 operator<()

```
bool operator< (
            const gcstring & str ) const
```

14.80.2.33 operator=()

```
gcstring& operator= (
            const gcstring & str )
```

14.80.2.34 operator==() [1/2]

```
bool operator== (
            const gcstring & str ) const
```

14.80.2.35 operator==() [2/2]

```
bool operator== (
            const char * pc ) const
```

14.80.2.36 operator>()

```
bool operator> (
            const gcstring & str ) const
```

14.80.2.37 resize()

```
virtual void resize (
            size_t n )  [virtual]
```

14.80.2.38 size()

```
virtual size_t size (
            void  ) const  [virtual]
```

14.80.2.39 substr()

```
virtual gcstring substr (
            size_t offset = 0,
            size_t count = GCSTRING_NPOS ) const  [virtual]
```

14.80.2.40 swap()

```
virtual void swap (
            gcstring & Right )  [virtual]
```

14.80.3 Friends And Related Function Documentation

14.80.3.1 operator+ [1/3]

```
SPINNAKER_API friend gcstring operator+ (
            const gcstring & left,
            const gcstring & right )  [friend]
```

14.80.3.2 operator+ [2/3]

```
SPINNAKER_API friend gcstring operator+ (
            const gcstring & left,
            const char * right )  [friend]
```

14.80.3.3 operator+ [3/3]

```
SPINNAKER_API friend gcstring operator+ (
            const char * left,
            const gcstring & right )  [friend]
```

14.80.4 Member Data Documentation

14.80.4.1 npos

```
const size_t npos  [static]
```

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

- include/SpinGenApi/GCString.h

14.81 GrabInfo Struct Reference

Public Member Functions

- GrabInfo (const string &deviceSerial)

Public Attributes

- unsigned int numImagesGrabbed
- unsigned int numIncompleteImages
- unsigned int numRemovals
- std::shared_ptr< ImageEventHandlerImpl > imageEventHandler

14.81.1 Constructor & Destructor Documentation

14.81.1.1 GrabInfo()

```
GrabInfo (
            const string & deviceSerial )  [inline]
```

14.81.2 Member Data Documentation

14.81.2.1 imageEventHandler

```
std::shared_ptr<ImageEventHandlerImpl> imageEventHandler
```

14.81.2.2 numImagesGrabbed

```
unsigned int numImagesGrabbed
```

14.81.2.3 numIncompleteImages

```
unsigned int numIncompleteImages
```

14.81.2.4 numRemovals

```
unsigned int numRemovals
```

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

- src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp

14.82 GVCP_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t ChunkID
- uint32_t ChunkLength

14.82.1 Detailed Description

header of a GVCP request packet

14.82.2 Member Data Documentation

14.82.2.1 ChunkID

```
uint32_t ChunkID
```

14.82.2.2 ChunkLength

```
uint32_t ChunkLength
```

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

- include/SpinGenApi/ChunkAdapterGEV.h

14.83 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

14.83.1 Detailed Description

layout of a GVCP event item (Extended ID flag not set)

14.83.2 Member Data Documentation

14.83.2.1 BlockId

```
uint16_t BlockId
```

14.83.2.2 EventId

```
uint16_t EventId
```

14.83.2.3 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

14.83.2.4 StreamChannelId

```
uint16_t StreamChannelId
```

14.83.2.5 TimestampHigh

```
uint32_t TimestampHigh
```

14.83.2.6 TimestampLow

```
uint32_t TimestampLow
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.84 GVCP_EVENT_ITEM_BASIC Struct Reference

layout of a GVCP event item (common to all types)

Public Attributes

- uint16_t ReservedOrEventSize
- uint16_t EventId

14.84.1 Detailed Description

layout of a GVCP event item (common to all types)

14.84.2 Member Data Documentation

14.84.2.1 EventId

```
uint16_t EventId
```

14.84.2.2 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.85 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

14.85.1 Detailed Description

layout of a GVCP event item (Extended ID flag set)

14.85.2 Member Data Documentation

14.85.2.1 BlockId

```
uint16_t BlockId
```

14.85.2.2 BlockId64High

```
uint32_t BlockId64High
```

14.85.2.3 BlockId64Low

```
uint32_t BlockId64Low
```

14.85.2.4 EventId

```
uint16_t EventId
```

14.85.2.5 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

14.85.2.6 StreamChannelId

```
uint16_t StreamChannelId
```

14.85.2.7 TimestampHigh

```
uint32_t TimestampHigh
```

14.85.2.8 TimestampLow

```
uint32_t TimestampLow
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.86 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]

14.86.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag not set)

14.86.2 Member Data Documentation

14.86.2.1 Header

```
GVCP_REQUEST_HEADER Header
```

14.86.2.2 Items

```
GVCP_EVENT_ITEM Items[1]
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.87 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]

14.87.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag set)

14.87.2 Member Data Documentation

14.87.2.1 Header

```
GVCP_REQUEST_HEADER Header
```

14.87.2.2 Items

```
GVCP_EVENT_ITEM_EXTENDED_ID Items[1]
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.88 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]

14.88.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag not set)

14.88.2 Member Data Documentation

14.88.2.1 Data

```
uint32_t Data[1]
```

14.88.2.2 Event

```
GVCP_EVENT_ITEM Event
```

14.88.2.3 Header

```
GVCP_REQUEST_HEADER Header
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.89 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]

14.89.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag set)

14.89.2 Member Data Documentation

14.89.2.1 Data

```
uint32_t Data[1]
```

14.89.2.2 Event

```
GVCP_EVENT_ITEM_EXTENDED_ID Event
```

14.89.2.3 Header

```
GVCP_REQUEST_HEADER Header
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.90 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

14.90.1 Detailed Description

header of a GVCP request packet

14.90.2 Member Data Documentation

14.90.2.1 Command

```
uint16_t Command
```

14.90.2.2 Flags

```
uint8_t Flags
```

14.90.2.3 Length

```
uint16_t Length
```

14.90.2.4 Magic

```
uint8_t Magic
```

14.90.2.5 ReqId

```
uint16_t ReqId
```

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

- include/SpinGenApi/EventAdapterGEV.h

14.91 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.

14.91.1 Detailed Description

Options for saving H264 files.

14.91.2 Constructor & Destructor Documentation

14.91.2.1 H264Option()

```
H264Option ( ) [inline]
```

14.91.3 Member Data Documentation

14.91.3.1 bitrate

```
unsigned int bitrate
```

Bit-rate to encode at.

14.91.3.2 frameRate

```
float frameRate
```

Frame rate of the stream.

14.91.3.3 height

```
unsigned int height
```

Height of source image.

14.91.3.4 reserved

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

Reserved for future use.

14.91.3.5 width

```
unsigned int width
```

Width of source image.

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

- include/SpinVideoDefs.h

14.92 ICameraBase Class Reference

The interface file for base class for the camera object.

Inheritance diagram for ICameraBase:

Collaboration diagram for ICameraBase:

Public Member Functions

- virtual ~ICameraBase (void)
- virtual void Init ()=0
- virtual void 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 buffer↩Size)=0
- virtual ImagePtr GetNextImage (uint64_t grabTimeout=EVENT_TIMEOUT_INFINITE, uint64_t stream↩ID=0)=0
- virtual GenICam::gcstring GetUniqueID ()=0
- virtual bool IsStreaming () const =0
- virtual GenICam::gcstring GetGuiXml () const =0
- virtual void RegisterEventHandler (EventHandler &evtHandlerToRegister)=0
- virtual void RegisterEventHandler (EventHandler &evtHandlerToRegister, const GenICam::gcstring &event↩Name)=0
- virtual void UnregisterEventHandler (EventHandler &evtHandlerToUnregister)=0
- virtual unsigned int GetNumImagesInUse ()=0
- virtual unsigned int GetNumDataStreams ()=0
- virtual unsigned int DiscoverMaxPacketSize ()=0
- virtual void ForceIP ()=0

Public Attributes

- TransportLayerDevice TLDevice

  Gets vital camera information by connecting to the camera's bootstrap registers.
- TransportLayerStream TLStream

  Gets information about the stream data by connecting to the camera's bootstrap registers.

Protected Member Functions

- ICameraBase ()
- ICameraBase (const ICameraBase &)
- ICameraBase & operator= (const ICameraBase &)

Protected Attributes

- CameraBaseData * m_pCameraBaseData

Friends

- class CameraInternal
- class InterfaceImpl

14.92.1 Detailed Description

The interface file for base class for the camera object.

14.92.2 Constructor & Destructor Documentation

14.92.2.1 ~ICameraBase()

```
virtual ~ICameraBase (
            void  )  [inline], [virtual]
```

14.92.2.2 ICameraBase() [1/2]

```
ICameraBase ( )  [inline], [protected]
```

14.92.2.3 ICameraBase() [2/2]

```
ICameraBase (
            const ICameraBase &  )  [inline], [protected]
```

14.92.3 Member Function Documentation

14.92.3.1 BeginAcquisition()

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

Implemented in CameraBase.

14.92.3.2 DeInit()

```
virtual void DeInit ( )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.3 DiscoverMaxPacketSize()

```
virtual unsigned int DiscoverMaxPacketSize ( )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.4 EndAcquisition()

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

Implemented in CameraBase.

14.92.3.5 ForceIP()

```
virtual void ForceIP ( )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.6 GetAccessMode()

```
virtual GenApi::EAccessMode GetAccessMode ( ) const  [pure virtual]
```

Implemented in CameraBase.

14.92.3.7 GetBufferOwnership()

```
virtual BufferOwnership GetBufferOwnership ( ) const  [pure virtual]
```

Implemented in CameraBase.

14.92.3.8 GetGuiXml()

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

Implemented in CameraBase.

14.92.3.9 GetNextImage()

```
virtual ImagePtr GetNextImage (
            uint64_t grabTimeout = EVENT_TIMEOUT_INFINITE,
            uint64_t streamID = 0 )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.10 GetNodeMap()

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

Implemented in CameraBase.

14.92.3.11 GetNumDataStreams()

```
virtual unsigned int GetNumDataStreams ( )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.12 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.13 GetTLDeviceNodeMap()

```
virtual GenApi::INodeMap& GetTLDeviceNodeMap ( ) const  [pure virtual]
```

Implemented in CameraBase.

14.92.3.14 GetTLStreamNodeMap()

```
virtual GenApi::INodeMap& GetTLStreamNodeMap ( ) const  [pure virtual]
```

Implemented in CameraBase.

14.92.3.15 GetUniqueID()

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

Implemented in CameraBase.

14.92.3.16 GetUserBufferCount()

```
virtual uint64_t GetUserBufferCount ( ) const  [pure virtual]
```

Implemented in CameraBase.

14.92.3.17 GetUserBufferSize()

```
virtual uint64_t GetUserBufferSize ( ) const  [pure virtual]
```

Implemented in CameraBase.

14.92.3.18 GetUserBufferTotalSize()

```
virtual uint64_t GetUserBufferTotalSize ( ) const  [pure virtual]
```

Implemented in CameraBase.

14.92.3.19 Init()

```
virtual void Init ( )  [pure virtual]
```

Implemented in CameraBase, and Camera.

14.92.3.20 IsInitialized()

```
virtual bool IsInitialized ( ) [pure virtual]
```

Implemented in CameraBase.

14.92.3.21 IsStreaming()

```
virtual bool IsStreaming ( ) const [pure virtual]
```

Implemented in CameraBase.

14.92.3.22 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in CameraBase.

14.92.3.23 operator=()

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

14.92.3.24 ReadPort()

```
virtual void ReadPort (
            uint64_t iAddress,
            void * pBuffer,
            size_t iSize ) [pure virtual]
```

Implemented in CameraBase.

14.92.3.25 RegisterEventHandler() [1/2]

```
virtual void RegisterEventHandler (
            EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in CameraBase.

14.92.3.26 RegisterEventHandler() [2/2]

```
virtual void RegisterEventHandler (
            EventHandler & evtHandlerToRegister,
            const GenICam::gcstring & eventName )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.27 SetBufferOwnership()

```
virtual void SetBufferOwnership (
            const BufferOwnership mode )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.28 SetUserBuffers() [1/2]

```
virtual void SetUserBuffers (
            void *const pMemBuffers,
            uint64_t totalSize )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.29 SetUserBuffers() [2/2]

```
virtual void SetUserBuffers (
            void **const ppMemBuffers,
            const uint64_t bufferCount,
            const uint64_t bufferSize )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.30 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
            EventHandler & evtHandlerToUnregister )  [pure virtual]
```

Implemented in CameraBase.

14.92.3.31 WritePort()

```
virtual void WritePort (
            uint64_t iAddress,
            const void * pBuffer,
            size_t iSize )  [pure virtual]
```

Implemented in CameraBase.

14.92.4 Friends And Related Function Documentation

14.92.4.1 CameraInternal

```
friend class CameraInternal  [friend]
```

14.92.4.2 InterfaceImpl

```
friend class InterfaceImpl  [friend]
```

14.92.5 Member Data Documentation

14.92.5.1 m_pCameraBaseData

```
CameraBaseData* m_pCameraBaseData  [protected]
```

14.92.5.2 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.

14.92.5.3 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

14.93 ICameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for ICameraList:

Public Member Functions

- virtual ~ICameraList ()
- virtual CameraPtr operator[ ] (unsigned int index)=0
- virtual unsigned int GetSize () const =0
- virtual CameraPtr GetByIndex (unsigned int index) const =0
- virtual CameraPtr GetBySerial (std::string serialNumber) const =0
- virtual CameraPtr GetByDeviceID (std::string deviceID) const =0
- virtual void Clear ()=0
- virtual void RemoveBySerial (std::string serialNumber)=0
- virtual void RemoveByIndex (unsigned int index)=0
- virtual void RemoveByDeviceID (std::string deviceID)=0
- virtual void Append (const CameraList &list)=0

Protected Member Functions

- ICameraList ()
- ICameraList (const ICameraList &)
- ICameraList & operator= (const ICameraList &)

Protected Attributes

- CameraListData * m_pCameraListData

Friends

- class InterfaceImpl
- class CameraListImpl

14.93.1 Detailed Description

Used to hold a list of camera objects.

14.93.2 Constructor & Destructor Documentation

14.93.2.1 ~ICameraList()

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

14.93.2.2 ICameraList() [1/2]

```
ICameraList ( ) [inline], [protected]
```

14.93.2.3 ICameraList() [2/2]

```
ICameraList (
            const ICameraList & ) [inline], [protected]
```

14.93.3 Member Function Documentation

14.93.3.1 Append()

```
virtual void Append (
            const CameraList & list ) [pure virtual]
```

Implemented in CameraList.

14.93.3.2 Clear()

```
virtual void Clear ( )  [pure virtual]
```

Implemented in CameraList.

14.93.3.3 GetByDeviceID()

```
virtual CameraPtr GetByDeviceID (
            std::string deviceID ) const  [pure virtual]
```

Implemented in CameraList.

14.93.3.4 GetByIndex()

```
virtual CameraPtr GetByIndex (
            unsigned int index ) const  [pure virtual]
```

Implemented in CameraList.

14.93.3.5 GetBySerial()

```
virtual CameraPtr GetBySerial (
            std::string serialNumber ) const  [pure virtual]
```

Implemented in CameraList.

14.93.3.6 GetSize()

```
virtual unsigned int GetSize ( ) const  [pure virtual]
```

Implemented in CameraList.

14.93.3.7 operator=()

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

14.93.3.8 operator[]()

```
virtual CameraPtr operator[] (
            unsigned int index )  [pure virtual]
```

Implemented in CameraList.

14.93.3.9 RemoveByDeviceID()

```
virtual void RemoveByDeviceID (
            std::string deviceID )  [pure virtual]
```

Implemented in CameraList.

14.93.3.10 RemoveByIndex()

```
virtual void RemoveByIndex (
            unsigned int index )  [pure virtual]
```

Implemented in CameraList.

14.93.3.11 RemoveBySerial()

```
virtual void RemoveBySerial (
            std::string serialNumber )  [pure virtual]
```

Implemented in CameraList.

14.93.4 Friends And Related Function Documentation

14.93.4.1 CameraListImpl

```
friend class CameraListImpl  [friend]
```

14.93.4.2 InterfaceImpl

```
friend class InterfaceImpl  [friend]
```

14.93.5 Member Data Documentation

14.93.5.1 m_pCameraListData

```
CameraListData* m_pCameraListData  [protected]
```

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

- include/Interface/ICameraList.h

14.94 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 GetInferenceFrameId () const =0
- virtual int64_t GetInferenceResult () const =0
- virtual float64_t GetInferenceConfidence () const =0
- virtual InferenceBoundingBoxResult GetInferenceBoundingBoxResult () const =0

Protected Member Functions

- IChunkData ()

14.94.1 Detailed Description

The Interface file for ChunkData.

14.94.2 Constructor & Destructor Documentation

14.94.2.1 ~IChunkData()

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

14.94.2.2 IChunkData()

```
IChunkData ( )  [inline], [protected]
```

14.94.3 Member Function Documentation

14.94.3.1 GetBlackLevel()

```
virtual float64_t GetBlackLevel ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.2 GetCounterValue()

```
virtual int64_t GetCounterValue ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.3 GetCRC()

```
virtual int64_t GetCRC ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.4 GetEncoderValue()

```
virtual int64_t GetEncoderValue ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.5 GetExposureEndLineStatusAll()

```
virtual int64_t GetExposureEndLineStatusAll ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.6 GetExposureTime()

```
virtual float64_t GetExposureTime ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.7 GetFrameID()

```
virtual int64_t GetFrameID ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.8 GetGain()

```
virtual float64_t GetGain ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.9 GetHeight()

```
virtual int64_t GetHeight ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.10 GetImage()

```
virtual int64_t GetImage ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.11 GetInferenceBoundingBoxResult()

```
virtual InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.12 GetInferenceConfidence()

```
virtual float64_t GetInferenceConfidence ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.13 GetInferenceFrameId()

```
virtual int64_t GetInferenceFrameId ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.14 GetInferenceResult()

```
virtual int64_t GetInferenceResult ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.15 GetLinePitch()

```
virtual int64_t GetLinePitch ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.16 GetLineStatusAll()

```
virtual int64_t GetLineStatusAll ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.17 GetOffsetX()

```
virtual int64_t GetOffsetX ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.18 GetOffsetY()

```
virtual int64_t GetOffsetY ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.19 GetPartSelector()

```
virtual int64_t GetPartSelector ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.20 GetPixelDynamicRangeMax()

```
virtual int64_t GetPixelDynamicRangeMax ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.21 GetPixelDynamicRangeMin()

```
virtual int64_t GetPixelDynamicRangeMin ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.22 GetScan3dAxisMax()

```
virtual float64_t GetScan3dAxisMax ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.23 GetScan3dAxisMin()

```
virtual float64_t GetScan3dAxisMin ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.24 GetScan3dCoordinateOffset()

```
virtual float64_t GetScan3dCoordinateOffset ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.25 GetScan3dCoordinateReferenceValue()

```
virtual float64_t GetScan3dCoordinateReferenceValue ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.26 GetScan3dCoordinateScale()

```
virtual float64_t GetScan3dCoordinateScale ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.27 GetScan3dInvalidDataValue()

```
virtual float64_t GetScan3dInvalidDataValue ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.28 GetScan3dTransformValue()

```
virtual float64_t GetScan3dTransformValue ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.29 GetScanLineSelector()

```
virtual int64_t GetScanLineSelector ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.30 GetSequencerSetActive()

```
virtual int64_t GetSequencerSetActive ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.31 GetSerialDataLength()

```
virtual int64_t GetSerialDataLength ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.32 GetStreamChannelID()

```
virtual int64_t GetStreamChannelID ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.33 GetTimerValue()

```
virtual float64_t GetTimerValue ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.34 GetTimestamp()

```
virtual int64_t GetTimestamp ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.35 GetTimestampLatchValue()

```
virtual int64_t GetTimestampLatchValue ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.36 GetTransferBlockID()

```
virtual int64_t GetTransferBlockID ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.37 GetTransferQueueCurrentBlockCount()

```
virtual int64_t GetTransferQueueCurrentBlockCount ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.38 GetWidth()

```
virtual int64_t GetWidth ( ) const  [pure virtual]
```

Implemented in ChunkData.

14.94.3.39 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

14.95 IDataStream Class Reference

Public Member Functions

- virtual ~IDataStream ()
- virtual void ResetStreamHandle (std::shared_ptr< Spinnaker::IGenTLDevice > pDevice)=0
- virtual StreamTypeEnum GetStreamType () const =0
- virtual void 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 size_t GetStreamInfoSizeType (GenTL::STREAM_INFO_CMD iInfoCmd)=0
- virtual bool GetStreamInfoBool8Type (GenTL::STREAM_INFO_CMD iInfoCmd)=0
- virtual void * GetBufferInfoPtrType (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual size_t GetBufferInfoSizeType (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual uint64_t GetBufferInfoUInt64Type (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual bool GetBufferInfoBool8Type (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual void RegisterImageEventHandler (IImageEventHandler &imageEventHandler, EventPollingOptions pollingOption)=0
- virtual void UnregisterImageEventHandler (IImageEventHandler &imageEventHandler)=0
- virtual void WaitOnImageEvent (uint64_t timeout)=0
- virtual void InitChunkAdapter (GenApi::INodeMap &nodemap)=0
- virtual void CleanupChunkAdapter ()=0
- virtual GenTL::GC_ERROR GetBufferChunkData (GenTL::BUFFER_HANDLE hBuffer, GenTL::SINGLE_CHUNK_DATA *pChunkData, size_t *piNumChunks)=0
- virtual void AttachBuffer (uint8_t *pBuffer, GenApi::SingleChunkData_t *ChunkData, int64_t NumChunks)=0
- virtual bool IsCRCCheckEnabled () const =0
- virtual GenApi::INodeMap & GetNodeMap () const =0
- virtual GenApi::INodeMap * GetDeviceNodeMap () const =0
- virtual Port & GetPort () const =0
- virtual const TransportLayerStream & TransportLayerStreamInfo () const =0

Protected Member Functions

- IDataStream ()

14.95.1 Constructor & Destructor Documentation

14.95.1.1 ~IDataStream()

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

14.95.1.2 IDataStream()

```
IDataStream ( ) [inline], [protected]
```

14.95.2 Member Function Documentation

14.95.2.1 AnnounceImage() [1/3]

```
virtual void AnnounceImage (
            size_t size ) [pure virtual]
```

14.95.2.2 AnnounceImage() [2/3]

```
virtual void AnnounceImage (
            size_t size,
            void * pPrivate ) [pure virtual]
```

14.95.2.3 AnnounceImage() [3/3]

```
virtual void AnnounceImage (
            size_t size,
            void * pData,
            void * pPrivate ) [pure virtual]
```

14.95.2.4 AttachBuffer()

```
virtual void AttachBuffer (
            uint8_t * pBuffer,
            GenApi::SingleChunkData_t * ChunkData,
            int64_t NumChunks ) [pure virtual]
```

14.95.2.5 CleanupChunkAdapter()

```
virtual void CleanupChunkAdapter ( )  [pure virtual]
```

14.95.2.6 FlushQueueAllDiscard()

```
virtual void FlushQueueAllDiscard ( )  [pure virtual]
```

14.95.2.7 GetBufferChunkData()

```
virtual GenTL::GC_ERROR GetBufferChunkData (
            GenTL::BUFFER_HANDLE hBuffer,
            GenTL::SINGLE_CHUNK_DATA * pChunkData,
            size_t * piNumChunks )  [pure virtual]
```

14.95.2.8 GetBufferInfoBool8Type()

```
virtual bool GetBufferInfoBool8Type (
            GenTL::BUFFER_HANDLE hBuffer,
            GenTL::BUFFER_INFO_CMD iInfoCmd )  [pure virtual]
```

14.95.2.9 GetBufferInfoPtrType()

```
virtual void* GetBufferInfoPtrType (
            GenTL::BUFFER_HANDLE hBuffer,
            GenTL::BUFFER_INFO_CMD iInfoCmd )  [pure virtual]
```

14.95.2.10 GetBufferInfoSizeType()

```
virtual size_t GetBufferInfoSizeType (
            GenTL::BUFFER_HANDLE hBuffer,
            GenTL::BUFFER_INFO_CMD iInfoCmd )  [pure virtual]
```

14.95.2.11 GetBufferInfoUInt64Type()

```
virtual uint64_t GetBufferInfoUInt64Type (
            GenTL::BUFFER_HANDLE hBuffer,
            GenTL::BUFFER_INFO_CMD iInfoCmd )  [pure virtual]
```

14.95.2.12 GetDeviceNodeMap()

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

14.95.2.13 GetNextImage()

```
virtual ImagePtr GetNextImage (
            uint64_t grabTimeout )  [pure virtual]
```

14.95.2.14 GetNextImageInternal()

```
virtual ImagePtr GetNextImageInternal (
            void ** ppPrivate,
            uint64_t grabTimeout )  [pure virtual]
```

14.95.2.15 GetNodeMap()

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

14.95.2.16 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) const  [pure virtual]
```

14.95.2.17 GetPort()

```
virtual Port& GetPort ( ) const  [pure virtual]
```

14.95.2.18 GetStreamInfoBool8Type()

```
virtual bool GetStreamInfoBool8Type (
            GenTL::STREAM_INFO_CMD iInfoCmd )  [pure virtual]
```

14.95.2.19 GetStreamInfoSizeType()

```
virtual size_t GetStreamInfoSizeType (
            GenTL::STREAM_INFO_CMD iInfoCmd )  [pure virtual]
```

14.95.2.20 GetStreamType()

```
virtual StreamTypeEnum GetStreamType ( ) const  [pure virtual]
```

14.95.2.21 InitChunkAdapter()

```
virtual void InitChunkAdapter (
            GenApi::INodeMap & nodemap )  [pure virtual]
```

14.95.2.22 IsCRCCheckEnabled()

```
virtual bool IsCRCCheckEnabled ( ) const  [pure virtual]
```

14.95.2.23 IsImageInUse()

```
virtual bool IsImageInUse (
            const uint64_t imageID )  [pure virtual]
```

14.95.2.24 IsStreaming()

```
virtual bool IsStreaming ( )  [pure virtual]
```

14.95.2.25 KillBufferEvent()

```
virtual void KillBufferEvent ( ) [pure virtual]
```

14.95.2.26 RegisterImageEventHandler()

```
virtual void RegisterImageEventHandler (
            IImageEventHandler & imageEventHandler,
            EventPollingOptions pollingOption ) [pure virtual]
```

14.95.2.27 ReleaseImage()

```
virtual void ReleaseImage (
            const uint64_t imageID ) [pure virtual]
```

14.95.2.28 ResetStreamHandle()

```
virtual void ResetStreamHandle (
            std::shared_ptr< Spinnaker::IGenTLDevice > pDevice ) [pure virtual]
```

14.95.2.29 RevokeImages()

```
virtual void RevokeImages ( ) [pure virtual]
```

14.95.2.30 StartStream()

```
virtual void StartStream (
            const unsigned int stream_index = 0 ) [pure virtual]
```

14.95.2.31 StopStream()

```
virtual void StopStream ( ) [pure virtual]
```

14.95.2.32 TransportLayerStreamInfo()

```
virtual const TransportLayerStream& TransportLayerStreamInfo ( ) const  [pure virtual]
```

14.95.2.33 UnregisterImageEventHandler()

```
virtual void UnregisterImageEventHandler (
            IImageEventHandler & imageEventHandler )  [pure virtual]
```

14.95.2.34 WaitOnImageEvent()

```
virtual void WaitOnImageEvent (
            uint64_t timeout )  [pure virtual]
```

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

- include/Interface/IStream.h

14.96 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.

14.96.1 Member Typedef Documentation

14.96.1.1 filebuf_type

```
typedef IDevFileStreamBuf<CharType, Traits> filebuf_type
```

14.96.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

14.96.1.3 istream_type

```
typedef std::basic_istream<CharType, Traits> istream_type
```

14.96.2 Member Function Documentation

14.96.2.1 close()

```
void close ( ) [inline]
```

Close the file on the device.

14.96.2.2 is_open()

```
bool is_open ( ) const [inline]
```

14.96.2.3 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

pInterface	NodeMap of the device to which the FileProtocolAdapter is attached
pFileName	Name of the file to open
mode	open mode

14.96.2.4 rdbuf()

```
filebuf_type* rdbuf ( ) const [inline]
```

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

- include/SpinGenApi/Filestream.h

14.97 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)

14.97.1 Constructor & Destructor Documentation

14.97.1.1 IDevFileStreamBuf()

```
IDevFileStreamBuf ( ) [inline]
```

14.97.1.2 ~IDevFileStreamBuf()

```
~IDevFileStreamBuf ( ) [inline]
```

14.97.2 Member Function Documentation

14.97.2.1 close()

```
filebuf_type* close ( ) [inline]
```

14.97.2.2 is_open()

```
bool is_open ( ) const [inline]
```

14.97.2.3 open()

```
filebuf_type* open (
            Spinnaker::GenApi::INodeMap * pInterface,
            const char * pFileName,
            std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

14.97.2.4 pbackfail()

```
int_type pbackfail (
            int_type c ) [inline], [protected]
```

14.97.2.5 underflow()

```
int_type underflow ( )  [inline], [protected]
```

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

- include/SpinGenApi/Filestream.h

14.98 IDeviceArrivalEventHandler Class Reference

Inheritance diagram for IDeviceArrivalEventHandler:

Collaboration diagram for IDeviceArrivalEventHandler:

Public Member Functions

- virtual ~IDeviceArrivalEventHandler ()
- virtual void OnDeviceArrival (uint64_t serialNumber)=0

Protected Member Functions

- IDeviceArrivalEventHandler ()
- IDeviceArrivalEventHandler (const IDeviceArrivalEventHandler &)
- IDeviceArrivalEventHandler & operator= (const IDeviceArrivalEventHandler &)

Additional Inherited Members

14.98.1 Constructor & Destructor Documentation

14.98.1.1 ~IDeviceArrivalEventHandler()

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

14.98.1.2 IDeviceArrivalEventHandler() [1/2]

```
IDeviceArrivalEventHandler ( ) [inline], [protected]
```

14.98.1.3 IDeviceArrivalEventHandler() [2/2]

```
IDeviceArrivalEventHandler (
            const IDeviceArrivalEventHandler & ) [inline], [protected]
```

14.98.2 Member Function Documentation

14.98.2.1 OnDeviceArrival()

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

Implemented in InterfaceEventHandlerImpl, InterfaceEventHandlerImpl, DeviceArrivalEventHandler, Interface↩EventHandler, and IInterfaceEventHandler.

14.98.2.2 operator=()

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

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

- include/Interface/IDeviceArrivalEventHandler.h

14.99 IDeviceEventHandler Class Reference

Inheritance diagram for IDeviceEventHandler:

Collaboration diagram for IDeviceEventHandler:

Public Member Functions

- virtual ~IDeviceEventHandler ()
- virtual void OnDeviceEvent (Spinnaker::GenICam::gcstring eventName)=0
- virtual uint64_t GetDeviceEventId () const =0
- virtual GenICam::gcstring GetDeviceEventName () const =0

Protected Member Functions

- IDeviceEventHandler ()
- IDeviceEventHandler (const IDeviceEventHandler &)
- IDeviceEventHandler & operator= (const IDeviceEventHandler &)

Additional Inherited Members

14.99.1 Constructor & Destructor Documentation

14.99.1.1 ~IDeviceEventHandler()

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

14.99.1.2 IDeviceEventHandler() [1/2]

```
IDeviceEventHandler ( ) [inline], [protected]
```

14.99.1.3 IDeviceEventHandler() [2/2]

```
IDeviceEventHandler (
            const IDeviceEventHandler & ) [inline], [protected]
```

14.99.2 Member Function Documentation

14.99.2.1 GetDeviceEventId()

```
virtual uint64_t GetDeviceEventId ( ) const [pure virtual]
```

Implemented in DeviceEventHandler.

14.99.2.2 GetDeviceEventName()

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

Implemented in DeviceEventHandler.

14.99.2.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (
            Spinnaker::GenICam::gcstring eventName )  [pure virtual]
```

Implemented in DeviceEventHandlerImpl, and DeviceEventHandler.

14.99.2.4 operator=()

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

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

- include/Interface/IDeviceEventHandler.h

14.100 IDeviceRemovalEventHandler Class Reference

Inheritance diagram for IDeviceRemovalEventHandler:

Collaboration diagram for IDeviceRemovalEventHandler:

Public Member Functions

- virtual ~IDeviceRemovalEventHandler ()
- virtual void OnDeviceRemoval (uint64_t serialNumber)=0

Protected Member Functions

- IDeviceRemovalEventHandler ()
- IDeviceRemovalEventHandler (const IDeviceRemovalEventHandler &)
- IDeviceRemovalEventHandler & operator= (const IDeviceRemovalEventHandler &)

Additional Inherited Members

14.100.1 Constructor & Destructor Documentation

14.100.1.1 ~IDeviceRemovalEventHandler()

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

14.100.1.2 IDeviceRemovalEventHandler() [1/2]

```
IDeviceRemovalEventHandler ( ) [inline], [protected]
```

14.100.1.3 IDeviceRemovalEventHandler() [2/2]

```
IDeviceRemovalEventHandler (
            const IDeviceRemovalEventHandler &  )  [inline], [protected]
```

14.100.2 Member Function Documentation

14.100.2.1 OnDeviceRemoval()

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

Implemented in InterfaceEventHandlerImpl, InterfaceEventHandlerImpl, InterfaceEventHandler, DeviceRemoval↩EventHandler, and IInterfaceEventHandler.

14.100.2.2 operator=()

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

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

- include/Interface/IDeviceRemovalEventHandler.h

14.101 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
- virtual void ResetImage (size_t width, size_t height, size_t offsetX, size_t offsetY, PixelFormatEnums pixel↩Format)=0
- virtual void ResetImage (size_t width, size_t height, size_t offsetX, size_t offsetY, PixelFormatEnums pixel↩Format, void *pData)=0
- virtual void Release ()=0
- virtual uint64_t GetID () const =0
- virtual void * GetData () const =0
- virtual void * GetPrivateData () const =0
- virtual float GetDataAbsoluteMax () const =0
- virtual float GetDataAbsoluteMin () const =0
- virtual size_t GetBufferSize () const =0
- virtual void DeepCopy (const ImagePtr pSrcImage)=0
- virtual size_t GetWidth () const =0
- virtual size_t GetHeight () const =0
- virtual size_t GetStride () const =0
- virtual size_t GetBitsPerPixel () const =0
- virtual size_t GetNumChannels () const =0
- virtual size_t GetXOffset () const =0
- virtual size_t GetYOffset () const =0
- virtual size_t GetXPadding () const =0
- virtual size_t GetYPadding () const =0
- virtual uint64_t GetFrameID () const =0
- virtual size_t GetPayloadType () const =0
- virtual PayloadTypeInfoIDs GetTLPayloadType () const =0
- virtual uint64_t GetTLPixelFormat () const =0
- virtual PixelFormatNamespaceID GetTLPixelFormatNamespace () const =0
- virtual 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) const =0
- virtual void Save (const char *pFilename, PNGOption &pOption) const =0
- virtual void Save (const char *pFilename, PPMOption &pOption) const =0
- virtual void Save (const char *pFilename, PGMOption &pOption) const =0
- virtual void Save (const char *pFilename, TIFFOption &pOption) const =0
- virtual void Save (const char *pFilename, JPEGOption &pOption) const =0
- virtual void Save (const char *pFilename, JPG2Option &pOption) const =0
- virtual void Save (const char *pFilename, BMPOption &pOption) const =0
- virtual const ChunkData & GetChunkData () const =0
- virtual void CalculateStatistics (ImageStatistics &pStatistics)=0
- virtual bool HasCRC () const =0
- virtual bool CheckCRC () const =0
- virtual size_t GetImageSize () const =0
- virtual bool IsInUse ()=0
- virtual ImageStatus GetImageStatus () const =0

Protected Member Functions

- IImage ()
- virtual ImageData * GetImageData () const =0

Friends

- class Stream

14.101.1 Detailed Description

The interface file for Image.

14.101.2 Constructor & Destructor Documentation

14.101.2.1 ~IImage()

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

14.101.2.2 IImage()

```
IImage ( ) [inline], [protected]
```

14.101.3 Member Function Documentation

14.101.3.1 CalculateStatistics()

```
virtual void CalculateStatistics (
            ImageStatistics & pStatistics ) [pure virtual]
```

Implemented in Image.

14.101.3.2 CheckCRC()

```
virtual bool CheckCRC ( ) const [pure virtual]
```

Implemented in Image.

14.101.3.3 Convert() [1/2]

```
virtual ImagePtr Convert (
            PixelFormatEnums format,
            ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const  [pure virtual]
```

Implemented in Image.

14.101.3.4 Convert() [2/2]

```
virtual void Convert (
            ImagePtr destinationImage,
            PixelFormatEnums format,
            ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const  [pure virtual]
```

Implemented in Image.

14.101.3.5 DeepCopy()

```
virtual void DeepCopy (
            const ImagePtr pSrcImage )  [pure virtual]
```

Implemented in Image.

14.101.3.6 GetBitsPerPixel()

```
virtual size_t GetBitsPerPixel ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.7 GetBufferSize()

```
virtual size_t GetBufferSize ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.8 GetChunkData()

```
virtual const ChunkData& GetChunkData ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.9 GetChunkLayoutId()

```
virtual uint64_t GetChunkLayoutId ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.10 GetColorProcessing()

```
virtual ColorProcessingAlgorithm GetColorProcessing ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.11 GetData()

```
virtual void* GetData ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.12 GetDataAbsoluteMax()

```
virtual float GetDataAbsoluteMax ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.13 GetDataAbsoluteMin()

```
virtual float GetDataAbsoluteMin ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.14 GetFrameID()

```
virtual uint64_t GetFrameID ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.15 GetHeight()

```
virtual size_t GetHeight ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.16 GetID()

```
virtual uint64_t GetID ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.17 GetImageData()

```
virtual ImageData* GetImageData ( ) const  [protected], [pure virtual]
```

Implemented in Image.

14.101.3.18 GetImageSize()

```
virtual size_t GetImageSize ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.19 GetImageStatus()

```
virtual ImageStatus GetImageStatus ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.20 GetNumChannels()

```
virtual size_t GetNumChannels ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.21 GetPayloadType()

```
virtual size_t GetPayloadType ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.22 GetPixelFormat()

```
virtual PixelFormatEnums GetPixelFormat ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.23 GetPixelFormatIntType()

```
virtual PixelFormatIntType GetPixelFormatIntType ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.24 GetPixelFormatName()

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

Implemented in Image.

14.101.3.25 GetPrivateData()

```
virtual void* GetPrivateData ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.26 GetStride()

```
virtual size_t GetStride ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.27 GetTimeStamp()

```
virtual uint64_t GetTimeStamp ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.28 GetTLPayloadType()

```
virtual PayloadTypeInfoIDs GetTLPayloadType ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.29 GetTLPixelFormat()

```
virtual uint64_t GetTLPixelFormat ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.30 GetTLPixelFormatNamespace()

```
virtual PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.31 GetValidPayloadSize()

```
virtual size_t GetValidPayloadSize ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.32 GetWidth()

```
virtual size_t GetWidth ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.33 GetXOffset()

```
virtual size_t GetXOffset ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.34 GetXPadding()

```
virtual size_t GetXPadding ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.35 GetYOffset()

```
virtual size_t GetYOffset ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.36 GetYPadding()

```
virtual size_t GetYPadding ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.37 HasCRC()

```
virtual bool HasCRC ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.38 IsIncomplete()

```
virtual bool IsIncomplete ( ) const  [pure virtual]
```

Implemented in Image.

14.101.3.39 IsInUse()

```
virtual bool IsInUse ( )  [pure virtual]
```

Implemented in Image.

14.101.3.40 Release()

```
virtual void Release ( )  [pure virtual]
```

Implemented in Image.

14.101.3.41 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.

14.101.3.42 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.

14.101.3.43 Save() [1/8]

```
virtual void Save (
            const char * pFilename,
            ImageFileFormat format = FROM_FILE_EXT ) const  [pure virtual]
```

Implemented in Image.

14.101.3.44 Save() [2/8]

```
virtual void Save (
            const char * pFilename,
            PNGOption & pOption ) const  [pure virtual]
```

Implemented in Image.

14.101.3.45 Save() [3/8]

```
virtual void Save (
            const char * pFilename,
            PPMOption & pOption ) const  [pure virtual]
```

Implemented in Image.

14.101.3.46 Save() [4/8]

```
virtual void Save (
            const char * pFilename,
            PGMOption & pOption ) const  [pure virtual]
```

Implemented in Image.

14.101.3.47 Save() [5/8]

```
virtual void Save (
            const char * pFilename,
            TIFFOption & pOption ) const  [pure virtual]
```

Implemented in Image.

14.101.3.48 Save() [6/8]

```
virtual void Save (
            const char * pFilename,
            JPEGOption & pOption ) const  [pure virtual]
```

Implemented in Image.

14.101.3.49 Save() [7/8]

```
virtual void Save (
            const char * pFilename,
            JPG2Option & pOption ) const  [pure virtual]
```

Implemented in Image.

14.101.3.50 Save() [8/8]

```
virtual void Save (
            const char * pFilename,
            BMPOption & pOption ) const  [pure virtual]
```

Implemented in Image.

14.101.4 Friends And Related Function Documentation

14.101.4.1 Stream

```
friend class Stream  [friend]
```

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

- include/Interface/IImage.h

14.102 IImageEventHandler Class Reference

Inheritance diagram for IImageEventHandler:

Collaboration diagram for IImageEventHandler:

Public Member Functions

- virtual ~IImageEventHandler ()
- virtual void OnImageEvent (ImagePtr image)=0

Protected Member Functions

- IImageEventHandler ()
- IImageEventHandler (const IImageEventHandler &)
- IImageEventHandler & operator= (const IImageEventHandler &)

Additional Inherited Members

14.102.1 Constructor & Destructor Documentation

14.102.1.1 ~IImageEventHandler()

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

14.102.1.2 IImageEventHandler() [1/2]

```
IImageEventHandler ( ) [inline], [protected]
```

14.102.1.3 IImageEventHandler() [2/2]

```
IImageEventHandler (
            const IImageEventHandler & ) [inline], [protected]
```

14.102.2 Member Function Documentation

14.102.2.1 OnImageEvent()

```
virtual void OnImageEvent (
            ImagePtr image ) [pure virtual]
```

Implemented in ImageEventHandlerImpl, ImageEventHandlerImpl, and ImageEventHandler.

14.102.2.2 operator=()

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

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

- include/Interface/IImageEventHandler.h

14.103 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 *p↩ PixelValueMax) 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 *p↩ RangeMax=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 &)

14.103.1 Detailed Description

The interface file for image statistics.

14.103.2 Constructor & Destructor Documentation

14.103.2.1 ~IImageStatistics()

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

14.103.2.2 IImageStatistics() [1/2]

```
IImageStatistics ( ) [inline], [protected]
```

14.103.2.3 IImageStatistics() [2/2]

```
IImageStatistics (
            const IImageStatistics & ) [inline], [protected]
```

14.103.3 Member Function Documentation

14.103.3.1 DisableAll()

```
virtual void DisableAll ( ) [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.2 EnableAll()

```
virtual void EnableAll ( ) [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( )  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( )  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
            StatisticsChannel channel,
            bool * pEnabled ) const  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.7 GetHistogram()

```
virtual void GetHistogram (
            StatisticsChannel channel,
            int ** ppHistogram ) const  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.8 GetMean()

```
virtual void GetMean (
            StatisticsChannel channel,
            float * pPixelValueMean ) const  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
            StatisticsChannel channel,
            unsigned int * pNumPixelValues ) const  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.10 GetPixelValueRange()

```
virtual void GetPixelValueRange (
            StatisticsChannel channel,
            unsigned int * pPixelValueMin,
            unsigned int * pPixelValueMax ) const  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.11 GetRange()

```
virtual void GetRange (
            StatisticsChannel channel,
            unsigned int * pMin,
            unsigned int * pMax ) const  [pure virtual]
```

Implemented in ImageStatistics.

14.103.3.12 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.

14.103.3.13 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

14.104 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 RegisterEventHandler (EventHandler &evtHandlerToRegister)=0
- virtual void UnregisterEventHandler (EventHandler &evtHandlerToUnregister)=0
- virtual bool IsInUse () const =0
- virtual void SendActionCommand (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[ ]=NULL) const =0
- virtual bool IsValid ()=0

Public Attributes

- TransportLayerInterface TLInterface

Protected Member Functions

- IInterface ()
- IInterface (const IInterface &)
- IInterface & operator= (const IInterface &)

Protected Attributes

- InterfaceData * m_pInterfaceData

Friends

- class InterfaceInternal
- class ProducerImpl

14.104.1 Detailed Description

The interface file for Interface.

14.104.2 Constructor & Destructor Documentation

14.104.2.1 ~IInterface()

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

14.104.2.2 IInterface() [1/2]

```
IInterface ( ) [inline], [protected]
```

14.104.2.3 IInterface() [2/2]

```
IInterface (
            const IInterface & ) [inline], [protected]
```

14.104.3 Member Function Documentation

14.104.3.1 GetCameras()

```
virtual CameraList GetCameras (
            bool updateCameras = true ) const [pure virtual]
```

Implemented in Interface.

14.104.3.2 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in Interface.

14.104.3.3 IsInUse()

```
virtual bool IsInUse ( ) const [pure virtual]
```

Implemented in Interface.

14.104.3.4 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in Interface.

14.104.3.5 operator=()

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

14.104.3.6 RegisterEventHandler()

```
virtual void RegisterEventHandler (
            EventHandler & evtHandlerToRegister )  [pure virtual]
```

Implemented in Interface.

14.104.3.7 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.

14.104.3.8 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
            EventHandler & evtHandlerToUnregister )  [pure virtual]
```

Implemented in Interface.

14.104.3.9 UpdateCameras()

```
virtual bool UpdateCameras ( )  [pure virtual]
```

Implemented in Interface.

14.104.4 Friends And Related Function Documentation

14.104.4.1 InterfaceInternal

```
friend class InterfaceInternal  [friend]
```

14.104.4.2 ProducerImpl

```
friend class ProducerImpl  [friend]
```

14.104.5 Member Data Documentation

14.104.5.1 m_pInterfaceData

```
InterfaceData* m_pInterfaceData  [protected]
```

14.104.5.2 TLInterface

```
TransportLayerInterface TLInterface
```

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

- include/Interface/IInterface.h

14.105 IInterfaceArrivalEventHandler Class Reference

Inheritance diagram for IInterfaceArrivalEventHandler:

Collaboration diagram for IInterfaceArrivalEventHandler:

Public Member Functions

- virtual ~IInterfaceArrivalEventHandler ()
- virtual void OnInterfaceArrival (std::string interfaceID)=0

Protected Member Functions

- IInterfaceArrivalEventHandler ()
- IInterfaceArrivalEventHandler (const IInterfaceArrivalEventHandler &)
- IInterfaceArrivalEventHandler & operator= (const IInterfaceArrivalEventHandler &)

Additional Inherited Members

14.105.1 Constructor & Destructor Documentation

14.105.1.1 ~IInterfaceArrivalEventHandler()

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

14.105.1.2 IInterfaceArrivalEventHandler() [1/2]

```
IInterfaceArrivalEventHandler ( ) [inline], [protected]
```

14.105.1.3 IInterfaceArrivalEventHandler() [2/2]

```
IInterfaceArrivalEventHandler (
            const IInterfaceArrivalEventHandler & ) [inline], [protected]
```

14.105.2 Member Function Documentation

14.105.2.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
            std::string interfaceID ) [pure virtual]
```

Implemented in SystemEventHandlerImpl, InterfaceArrivalEventHandler, SystemEventHandler, and ISystem↩EventHandler.

14.105.2.2 operator=()

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

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

- include/Interface/IInterfaceArrivalEventHandler.h

14.106 IInterfaceEventHandler Class Reference

Inheritance diagram for IInterfaceEventHandler:

Collaboration diagram for IInterfaceEventHandler:

Public Member Functions

- virtual ~IInterfaceEventHandler ()
- virtual void OnDeviceArrival (uint64_t serialNumber)=0
- virtual void OnDeviceRemoval (uint64_t serialNumber)=0

Protected Member Functions

- IInterfaceEventHandler ()
- IInterfaceEventHandler (const IInterfaceEventHandler &)
- IInterfaceEventHandler & operator= (const IInterfaceEventHandler &)

Additional Inherited Members

14.106.1 Constructor & Destructor Documentation

14.106.1.1 ~IInterfaceEventHandler()

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

14.106.1.2 IInterfaceEventHandler() [1/2]

```
IInterfaceEventHandler ( ) [inline], [protected]
```

14.106.1.3 IInterfaceEventHandler() [2/2]

```
IInterfaceEventHandler (
            const IInterfaceEventHandler &  )  [inline], [protected]
```

14.106.2 Member Function Documentation

14.106.2.1 OnDeviceArrival()

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

Implements IDeviceArrivalEventHandler.

Implemented in InterfaceEventHandlerImpl, InterfaceEventHandlerImpl, and InterfaceEventHandler.

14.106.2.2 OnDeviceRemoval()

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

Implements IDeviceRemovalEventHandler.

Implemented in InterfaceEventHandlerImpl, InterfaceEventHandlerImpl, and InterfaceEventHandler.

14.106.2.3 operator=()

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

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

- include/Interface/IInterfaceEventHandler.h

14.107 IInterfaceList Class Reference

The interface file for InterfaceList class.

Inheritance diagram for IInterfaceList:

Public Member Functions

- virtual ~IInterfaceList (void)
- virtual InterfacePtr operator[ ] (unsigned int index)=0
- virtual unsigned int GetSize () const =0
- virtual InterfacePtr GetByIndex (unsigned int index) const =0
- virtual void Clear ()=0
- virtual void Append (const InterfaceList *list)=0

Protected Member Functions

- IInterfaceList (void)
- IInterfaceList (const IInterfaceList &)
- IInterfaceList & operator= (const IInterfaceList &)

Protected Attributes

- InterfaceListData * m_pInterfaceListData

Friends

- class InterfaceListImpl

14.107.1 Detailed Description

The interface file for InterfaceList class.

14.107.2 Constructor & Destructor Documentation

14.107.2.1 ~IInterfaceList()

```
virtual ~IInterfaceList (
            void ) [inline], [virtual]
```

14.107.2.2 IInterfaceList() [1/2]

```
IInterfaceList (
            void ) [inline], [protected]
```

14.107.2.3 IInterfaceList() [2/2]

```
IInterfaceList (
            const IInterfaceList & ) [inline], [protected]
```

14.107.3 Member Function Documentation

14.107.3.1 Append()

```
virtual void Append (
            const InterfaceList * list ) [pure virtual]
```

Implemented in InterfaceList.

14.107.3.2 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in InterfaceList.

14.107.3.3 GetByIndex()

```
virtual InterfacePtr GetByIndex (
            unsigned int index ) const  [pure virtual]
```

Implemented in InterfaceList.

14.107.3.4 GetSize()

```
virtual unsigned int GetSize ( ) const  [pure virtual]
```

Implemented in InterfaceList.

14.107.3.5 operator=()

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

14.107.3.6 operator[]()

```
virtual InterfacePtr operator[] (
            unsigned int index )  [pure virtual]
```

Implemented in InterfaceList.

14.107.4 Friends And Related Function Documentation

14.107.4.1 InterfaceListImpl

```
friend class InterfaceListImpl  [friend]
```

14.107.5 Member Data Documentation

14.107.5.1 m_pInterfaceListData

```
InterfaceListData* m_pInterfaceListData [protected]
```

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

- include/Interface/IInterfaceList.h

14.108 IInterfaceRemovalEventHandler Class Reference

Inheritance diagram for IInterfaceRemovalEventHandler:

Collaboration diagram for IInterfaceRemovalEventHandler:

Public Member Functions

- virtual ~IInterfaceRemovalEventHandler ()
- virtual void OnInterfaceRemoval (std::string interfaceID)=0

Protected Member Functions

- IInterfaceRemovalEventHandler ()
- IInterfaceRemovalEventHandler (const IInterfaceRemovalEventHandler &)
- IInterfaceRemovalEventHandler & operator= (const IInterfaceRemovalEventHandler &)

Additional Inherited Members

14.108.1 Constructor & Destructor Documentation

14.108.1.1 ~IInterfaceRemovalEventHandler()

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

14.108.1.2 IInterfaceRemovalEventHandler() [1/2]

```
IInterfaceRemovalEventHandler ( )  [inline], [protected]
```

14.108.1.3 IInterfaceRemovalEventHandler() [2/2]

```
IInterfaceRemovalEventHandler (
            const IInterfaceRemovalEventHandler &  )  [inline], [protected]
```

14.108.2 Member Function Documentation

14.108.2.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
            std::string interfaceID )  [pure virtual]
```

Implemented in SystemEventHandlerImpl, SystemEventHandler, InterfaceRemovalEventHandler, and ISystem↩EventHandler.

14.108.2.2 operator=()

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

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

- include/Interface/IInterfaceRemovalEventHandler.h

14.109 ILoggingEventHandler Class Reference

Inheritance diagram for ILoggingEventHandler:

Collaboration diagram for ILoggingEventHandler:

Public Member Functions

- virtual ~ILoggingEventHandler ()
- virtual void OnLogEvent (LoggingEventDataPtr eventPtr)=0

Protected Member Functions

- ILoggingEventHandler ()
- ILoggingEventHandler (const ILoggingEventHandler &)
- ILoggingEventHandler & operator= (const ILoggingEventHandler &)

Additional Inherited Members

14.109.1 Constructor & Destructor Documentation

14.109.1.1 ~ILoggingEventHandler()

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

14.109.1.2 ILoggingEventHandler() [1/2]

```
ILoggingEventHandler ( ) [inline], [protected]
```

14.109.1.3 ILoggingEventHandler() [2/2]

```
ILoggingEventHandler (
            const ILoggingEventHandler & ) [inline], [protected]
```

14.109.2 Member Function Documentation

14.109.2.1 OnLogEvent()

```
virtual void OnLogEvent (
            LoggingEventDataPtr eventPtr ) [pure virtual]
```

Implemented in LoggingEventHandler.

14.109.2.2 operator=()

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

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

- include/Interface/ILoggingEventHandler.h

14.110 Image Class Reference

The image object class.

Inheritance diagram for Image:

Collaboration diagram for Image:

Public Member Functions

- virtual ~Image ()

  Virtual destructor.
- ColorProcessingAlgorithm GetColorProcessing () const

  Gets the color algorithm used to produce the image.
- ImagePtr Convert (Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const

  Converts the current image buffer to the specified output pixel format and stores the result in the specified image.
- void Convert (ImagePtr destinationImage, Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const

  Converts the current image buffer to the specified output pixel format and stores the result in the specified destination image.
- void ResetImage (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat)

  Sets new dimensions of the image object and allocates memory.
- void ResetImage (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void *pData)

  Sets new dimensions of the image object.
- void Release ()
- uint64_t GetID () const

  Gets a unique ID for this image.
- void * GetData () const

  Gets a pointer to the data associated with the image.
- float GetDataAbsoluteMax () const

  Get the value for which no image data will exceed.
- float GetDataAbsoluteMin () const

  Get the value for which no image data will be less than.
- void * GetPrivateData () const

  Gets a pointer to the user passed data associated with the image.
- size_t GetBufferSize () const

  Gets the size of the buffer associated with the image in bytes.
- void DeepCopy (const ImagePtr pSrcImage)

  Performs a deep copy of the Image.
- size_t GetWidth () const

  Gets the width of the image in pixels.
- size_t GetHeight () const

  Gets the height of the image in pixels.
- size_t GetStride () const

  Gets the stride of the image in bytes.
- size_t GetBitsPerPixel () const

  Gets the number of bits used per pixel in the image.
- size_t GetNumChannels () const

  Gets the number of channels (depth) used in the image.
- size_t GetXOffset () const

  Gets the ROI x offset in pixels for this image.
- size_t GetYOffset () const

  Gets the ROI y offset in pixels for this image.
- size_t GetXPadding () const

  Gets the x padding in bytes for this image.
- size_t GetYPadding () const

  Gets the y padding in bytes for this image.

- uint64_t GetFrameID () const

  Gets the frame ID for this image.
- size_t GetPayloadType () const

  Gets the payload type that was transmitted.
- PayloadTypeInfoIDs GetTLPayloadType () const

  Gets the GenTL specific payload type that was transmitted.
- uint64_t GetTLPixelFormat () const

  Gets the pixel format of the image.
- PixelFormatNamespaceID GetTLPixelFormatNamespace () const

  Returns an enum value that represents the namespace in which this image's TL specific pixel format resides.
- GenICam::gcstring GetPixelFormatName () const

  Returns a string value that represents this image's pixel format.
- Spinnaker::PixelFormatEnums GetPixelFormat () const

  Returns an enum value that represents the pixel format of this image.
- Spinnaker::PixelFormatIntType GetPixelFormatIntType () const

  Returns an enum value that represents the integer type used in the pixel format of this image.
- bool IsIncomplete () const

  Returns a boolean value indicating if this image was incomplete.
- size_t GetValidPayloadSize () const

  Returns the size of valid data in the image payload.
- uint64_t GetChunkLayoutId () const

  Returns the id of the chunk data layout.
- uint64_t GetTimeStamp () const

  Gets the time stamp for the image in nanoseconds.
- void Save (const char *pFilename, ImageFileFormat format=FROM_FILE_EXT) const

  Saves the image to the specified file name with the file format specified.
- void Save (const char *pFilename, PNGOption &pOption) const

  Saves the image to the specified file name with the options specified.
- void Save (const char *pFilename, PPMOption &pOption) const

  Saves the image to the specified file name with the options specified.
- void Save (const char *pFilename, PGMOption &pOption) const

  Saves the image to the specified file name with the options specified.
- void Save (const char *pFilename, TIFFOption &pOption) const

  Saves the image to the specified file name with the options specified.
- void Save (const char *pFilename, JPEGOption &pOption) const

  Saves the image to the specified file name with the options specified.
- void Save (const char *pFilename, JPG2Option &pOption) const

  Saves the image to the specified file name with the options specified.
- void Save (const char *pFilename, BMPOption &pOption) const

  Saves the image to the specified file name with the options specified.
- const ChunkData & GetChunkData () const

  Returns a pointer to a chunk data interface.
- void CalculateStatistics (ImageStatistics &pStatistics)

  Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.
- bool HasCRC () const

  Checks if the image contains ImageCRC checksum from chunk data.
- bool CheckCRC () const

  Checks if the computed checksum matches with chunk data's ImageCRC.
- size_t GetImageSize () const

  Returns the size of the image.
- bool IsInUse ()

Returns true if the image is still in use by the stream.

- ImageStatus GetImageStatus () const

  Returns data integrity status of the image returned from GetNextImage()

- bool IsCompressed () const

  Returns a boolean value indicating whether this image is compressed.

Static Public Member Functions

- static ImagePtr Create ()

  Create an image object.

- static ImagePtr Create (const ImagePtr image)

  Create an image object that is a deep copy of the input image.

- static ImagePtr Create (size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormat↩ Enums pixelFormat, void ∗pData)

  Create an image object with the specified parameters.

- static void SetDefaultColorProcessing (ColorProcessingAlgorithm colorAlgorithm)

  Sets the default color processing algorithm.

- static ColorProcessingAlgorithm GetDefaultColorProcessing ()

  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 color↩ Algorithm=DEFAULT) const

Friends

- class IDataStream
- class Stream
- class ImageConverter
- class ImageFiler
- class ImageStatsCalculator
- class ImageUtilityImpl
- class ImageUtilityPolarizationImpl

14.110.1 Detailed Description

The image object class.

14.110.2 Constructor & Destructor Documentation

14.110.2.1 ~Image()

```
virtual ~Image ( ) [virtual]
```

Virtual destructor.

14.110.2.2 Image() [1/3]

```
Image ( ) [protected]
```

14.110.2.3 Image() [2/3]

```
Image (
            const ImagePtr image ) [protected]
```

14.110.2.4 Image() [3/3]

```
Image (
            size_t width,
            size_t height,
            size_t offsetX,
            size_t offsetY,
            PixelFormatEnums pixelFormat,
            void * pData ) [protected]
```

14.110.3 Member Function Documentation

14.110.3.1 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

pStatistics	The statistics of an image.

Implements IImage.

14.110.3.2 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.

14.110.3.3 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

format	Output format of the converted image.
colorAlgorithm	Optional color processing algorithm for producing the converted image

Returns

The converted image.

Implements IImage.

14.110.3.4 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

destinationImage	Destination image where the converted output result will be stored.
format	Output format of the converted image.
colorAlgorithm	Optional color processing algorithm for producing the converted image.

Implements IImage.

14.110.3.5 Convert() [3/3]

```
void Convert (
            PixelFormatEnums format,
            Image & pDestImage,
            ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const  [protected]
```

14.110.3.6 Create() [1/3]

```
static ImagePtr Create ( )  [static]
```

Create an image object.

14.110.3.7 Create() [2/3]

```
static ImagePtr Create (
            const ImagePtr image )  [static]
```

Create an image object that is a deep copy of the input image.

Parameters

image	The input image to copy

14.110.3.8 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

width	The image width in pixels
height	The image height in pixels
offsetX	The image X offset
offsetY	The image Y offset
pixelFormat	The image pixel format
pData	The image data

14.110.3.9 CreateShared()

```
ImagePtr CreateShared ( ) const  [protected]
```

14.110.3.10 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

pSrcImage	The Image to copy the data from.

Implements IImage.

14.110.3.11 DeepCopy() [2/2]

```
void DeepCopy (
            const Image & pSrcImage )  [protected]
```

14.110.3.12 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.

14.110.3.13 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.

14.110.3.14 GetChunkData()

```
const ChunkData& GetChunkData ( ) const  [virtual]
```

Returns a pointer to a chunk data interface.

No ownership is transfered, 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.

14.110.3.15 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.

14.110.3.16 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.

14.110.3.17 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.

14.110.3.18 GetDataAbsoluteMax()

```
float GetDataAbsoluteMax ( ) const  [virtual]
```

Get the value for which no image data will exceed.

Returns

the maximim theoretical image data value

Implements IImage.

14.110.3.19 GetDataAbsoluteMin()

```
float GetDataAbsoluteMin ( ) const  [virtual]
```

Get the value for which no image data will be less than.

Returns

the minimum theoretical image data value

Implements IImage.

14.110.3.20 GetDefaultColorProcessing()

```
static ColorProcessingAlgorithm GetDefaultColorProcessing ( )  [static]
```

Gets the default color processing algorithm.

See also

SetDefaultColorProcessing()

Returns

The default color processing algorithm.

14.110.3.21 GetFrameID()

```
uint64_t GetFrameID ( ) const  [virtual]
```

Gets the frame ID for this image.

Returns

The frame ID.

Implements IImage.

14.110.3.22 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.

14.110.3.23 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.

14.110.3.24 GetImageData()

```
ImageData* GetImageData ( ) const  [protected], [virtual]
```

Implements IImage.

14.110.3.25 GetImageSize()

```
size_t GetImageSize ( ) const  [virtual]
```

Returns the size of the image.

Returns

The image size in bytes.

Implements IImage.

14.110.3.26 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.

14.110.3.27 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.

14.110.3.28 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.

14.110.3.29 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.

14.110.3.30 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.

14.110.3.31 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.

14.110.3.32 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.

14.110.3.33 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.

14.110.3.34 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.

14.110.3.35 GetTimeStamp()

```
uint64_t GetTimeStamp ( ) const  [virtual]
```

Gets the time stamp for the image in nanoseconds.

Returns

The time stamp of the image.

Implements IImage.

14.110.3.36 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.

14.110.3.37 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.

14.110.3.38 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.

14.110.3.39 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.

14.110.3.40 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.

14.110.3.41 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.

14.110.3.42 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.

14.110.3.43 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.

14.110.3.44 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.

14.110.3.45 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.

14.110.3.46 IsCompressed()

```
bool IsCompressed ( ) const
```

Returns a boolean value indicating whether this image is compressed.

Returns

Returns true if image is compressed, false otherwise.

14.110.3.47 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.

14.110.3.48 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.

14.110.3.49 Release()

```
void Release ( ) [virtual]
```

Implements IImage.

14.110.3.50 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

width	The width of image in pixels to set.
height	The height of image in pixels to set.
offsetX	The x offset in pixels to set.
offsetY	The y offset in pixels to set.
pixelFormat	Pixel format to set.

Implements IImage.

14.110.3.51 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

width	The width of image in pixels to set.
height	The height of image in pixels to set.
offsetX	The x offset in pixels to set.
offsetY	The y offset in pixels to set.
pixelFormat	Pixel format to set.
pData	Pointer to the image buffer.

Implements IImage.

14.110.3.52 Save() [1/8]

```
void Save (
            const char * pFilename,
            ImageFileFormat format = FROM_FILE_EXT ) const  [virtual]
```

Saves the image to the specified file name with the file format specified.

Parameters

pFilename	Filename to save image with.
format	File format to save in.

Implements IImage.

14.110.3.53 Save() [2/8]

```
void Save (
            const char * pFilename,
            PNGOption & pOption ) const  [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

pFilename	Filename to save image with.
pOption	Options to use while saving image.

Implements IImage.

14.110.3.54 Save() [3/8]

```
void Save (
            const char * pFilename,
            PPMOption & pOption ) const  [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

pFilename	Filename to save image with.
pOption	Options to use while saving image.

Implements IImage.

14.110.3.55 Save() [4/8]

```
void Save (
            const char * pFilename,
            PGMOption & pOption ) const  [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

pFilename	Filename to save image with.
pOption	Options to use while saving image.

Implements IImage.

14.110.3.56 Save() [5/8]

```
void Save (
            const char * pFilename,
            TIFFOption & pOption ) const  [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

pFilename	Filename to save image with.
pOption	Options to use while saving image.

Implements IImage.

14.110.3.57 Save() [6/8]

```
void Save (
            const char * pFilename,
            JPEGOption & pOption ) const  [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

pFilename	Filename to save image with.
pOption	Options to use while saving image.

Implements IImage.

14.110.3.58 Save() [7/8]

```
void Save (
            const char * pFilename,
            JPG2Option & pOption ) const  [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

pFilename	Filename to save image with.
pOption	Options to use while saving image.

Implements IImage.

14.110.3.59 Save() [8/8]

```
void Save (
            const char * pFilename,
            BMPOption & pOption ) const  [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

pFilename	Filename to save image with.
pOption	Options to use while saving image.

Implements IImage.

14.110.3.60 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

colorAlgorithm	The color processing algorithm to set.

See also

GetDefaultColorProcessing()

14.110.4 Friends And Related Function Documentation

14.110.4.1 IDataStream

```
friend class IDataStream  [friend]
```

14.110.4.2 ImageConverter

```
friend class ImageConverter  [friend]
```

14.110.4.3 ImageFiler

```
friend class ImageFiler  [friend]
```

14.110.4.4 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

14.110.4.5 ImageUtilityImpl

```
friend class ImageUtilityImpl [friend]
```

14.110.4.6 ImageUtilityPolarizationImpl

```
friend class ImageUtilityPolarizationImpl [friend]
```

14.110.4.7 Stream

```
friend class Stream [friend]
```

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

- include/Image.h

14.111 ImageEventHandler Class Reference

A handler for capturing image arrival events.

Inheritance diagram for ImageEventHandler:

Collaboration diagram for ImageEventHandler:

Public Member Functions

- ImageEventHandler ()

  Default Constructor.
- virtual ~ImageEventHandler ()

  Virtual Destructor.
- virtual void OnImageEvent (ImagePtr image)=0

  Image event callback.

Protected Member Functions

- ImageEventHandler & operator= (const ImageEventHandler &)

  Assignment operator.

Additional Inherited Members

14.111.1 Detailed Description

A handler for capturing image arrival events.

14.111.2 Constructor & Destructor Documentation

14.111.2.1 ImageEventHandler()

```
ImageEventHandler ( )
```

Default Constructor.

14.111.2.2 ~ImageEventHandler()

```
virtual ~ImageEventHandler ( )  [virtual]
```

Virtual Destructor.

14.111.3 Member Function Documentation

14.111.3.1 OnImageEvent()

```
virtual void OnImageEvent (
            ImagePtr image )  [pure virtual]
```

Image event callback.

Parameters

image	The ImagePtr object

Implements IImageEventHandler.

Implemented in ImageEventHandlerImpl, and ImageEventHandlerImpl.

14.111.3.2 operator=()

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

Assignment operator.

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

- include/ImageEventHandler.h

14.112 ImageEventHandlerImpl Class Reference

Inheritance diagram for ImageEventHandlerImpl:

Collaboration diagram for ImageEventHandlerImpl:

Public Member Functions

- ImageEventHandlerImpl (string deviceSerial)
- ~ImageEventHandlerImpl ()
- void OnImageEvent (ImagePtr image)

  Image event callback.
- ImageEventHandlerImpl (CameraPtr pCam)
- ~ImageEventHandlerImpl ()
- void OnImageEvent (ImagePtr image)

  Image event callback.
- int getImageCount ()
- int getMaxImages ()

Additional Inherited Members

14.112.1 Constructor & Destructor Documentation

14.112.1.1 ImageEventHandlerImpl() [1/2]

```
ImageEventHandlerImpl (
            string deviceSerial ) [inline]
```

14.112.1.2 ~ImageEventHandlerImpl() [1/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

14.112.1.3 ImageEventHandlerImpl() [2/2]

```
ImageEventHandlerImpl (
            CameraPtr pCam ) [inline]
```

14.112.1.4 ~ImageEventHandlerImpl() [2/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

14.112.2 Member Function Documentation

14.112.2.1 getImageCount()

```
int getImageCount ( ) [inline]
```

14.112.2.2 getMaxImages()

```
int getMaxImages ( ) [inline]
```

14.112.2.3 OnImageEvent() [1/2]

```
void OnImageEvent (
            ImagePtr image ) [virtual]
```

Image event callback.

Parameters

image	The ImagePtr object

Implements ImageEventHandler.

14.112.2.4 OnImageEvent() [2/2]

```
void OnImageEvent (
            ImagePtr image ) [inline], [virtual]
```

Image event callback.

Parameters

image	The ImagePtr object

Implements ImageEventHandler.

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

- src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp
- src/ImageEvents/ImageEvents.cpp

14.113 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

14.113.1 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.

14.113.2 Constructor & Destructor Documentation

14.113.2.1 ImagePtr() [1/4]

```
ImagePtr ( )
```

Default constructor.

14.113.2.2 ImagePtr() [2/4]

```
ImagePtr (
            const int  )
```

Default constructor with argument.

14.113.2.3 ImagePtr() [3/4]

```
ImagePtr (
            const long  )
```

Default constructor with argument.

14.113.2.4 ImagePtr() [4/4]

```
ImagePtr (
            const std::nullptr_t  )
```

Default constructor with argument.

14.113.2.5 ~ImagePtr()

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

Virtual destructor.

14.113.3 Member Function Documentation

14.113.3.1 operator=()

```
virtual ImagePtr& operator= (
            const ImagePtr & ) [virtual]
```

Assignment operator.

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

- include/ImagePtr.h

14.114 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 *p↩PixelValueMax) 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 *p↩RangeMax=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

14.114.1 Detailed Description

Represents image statistics for an image.

14.114.2 Constructor & Destructor Documentation

14.114.2.1 ImageStatistics() [1/2]

```
ImageStatistics ( )
```

Default constructor.

14.114.2.2 ~ImageStatistics()

```
virtual ~ImageStatistics ( ) [virtual]
```

Default destructor.

14.114.2.3 ImageStatistics() [2/2]

```
ImageStatistics (
            const ImageStatistics & other )
```

Copy constructor.

14.114.3 Member Function Documentation

14.114.3.1 DisableAll()

```
virtual void DisableAll ( ) [virtual]
```

Disable all channels.

Implements IImageStatistics.

14.114.3.2 EnableAll()

```
virtual void EnableAll ( ) [virtual]
```

Enable all channels.

Implements IImageStatistics.

14.114.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [virtual]
```

Enable only the grey channel.

Implements IImageStatistics.

14.114.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [virtual]
```

Enable only the HSL channels.

Implements IImageStatistics.

14.114.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [virtual]
```

Enable only the RGB channels.

Implements IImageStatistics.

14.114.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
            StatisticsChannel channel,
            bool * pEnabled ) const [virtual]
```

Gets the status of a statistics channel.

Parameters

channel	The statistics channel.
pEnabled	Whether the channel is enabled.

See also

SetChannelStatus()

Implements IImageStatistics.

14.114.3.7 GetHistogram()

```
virtual void GetHistogram (
            StatisticsChannel channel,
            int ** ppHistogram ) const  [virtual]
```

Gets the histogram for the image.

Parameters

channel	The statistics channel.
ppHistogram	Pointer to an array containing the histogram.

Implements IImageStatistics.

14.114.3.8 GetMean()

```
virtual void GetMean (
            StatisticsChannel channel,
            float * pPixelValueMean ) const  [virtual]
```

Gets the mean of the image.

Parameters

channel	The statistics channel.
pPixelValueMean	The mean of the image.

Implements IImageStatistics.

14.114.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
            StatisticsChannel channel,
            unsigned int * pNumPixelValues ) const  [virtual]
```

Gets the number of unique pixel values in the image.

Parameters

channel	The statistics channel.
pNumPixelValues	The number of unique pixel values.

Implements IImageStatistics.

14.114.3.10 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

channel	The statistics channel.
pPixelValueMin	The minimum pixel value.
pPixelValueMax	The maximum pixel value.

Implements IImageStatistics.

14.114.3.11 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

channel	The statistics channel.
pMin	The minimum possible value.
pMax	The maximum possible value.

Implements IImageStatistics.

14.114.3.12 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

channel	The statistics channel.
pRangeMin	The minimum possible value.
pRangeMax	The maximum possible value.
pPixelValueMin	The minimum pixel value.
pPixelValueMax	The maximum pixel value.
pNumPixelValues	The number of unique pixel values.
pPixelValueMean	The mean of the image.
ppHistogram	Pointer to an array containing the histogram.

Implements IImageStatistics.

14.114.3.13 operator=()

```
ImageStatistics& operator= (
            const ImageStatistics & other )
```

Assignment operator.

Parameters

other	The ImageStatistics object to copy from.

14.114.3.14 SetChannelStatus()

```
virtual void SetChannelStatus (
            StatisticsChannel channel,
            bool enabled )  [virtual]
```

Sets the status of a statistics channel.

Parameters

channel	The statistics channel.
enabled	Whether the channel should be enabled.

See also

GetChannelStatus()

Implements IImageStatistics.

14.114.4 Friends And Related Function Documentation

14.114.4.1 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

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

- include/ImageStatistics.h

14.115 ImageUtility Class Reference

Static helper functions for the image object class.

Public Types

- enum ImageScalingAlgorithm { NEAREST_NEIGHBOR }

  Image scaling algorithms.
- enum SourceDataRange {
  IMAGE_DATA_RANGE,
  ABSOLUTE_DATA_RANGE,
  IMAGE_MIN_ABSOLUTE_MAX,
  ABSOLUTE_MIN_IMAGE_MAX }

  Image normalization source data options.

Static Public Member Functions

- static ImagePtr CreateScaled (const ImagePtr &srcImage, ImageScalingAlgorithm scalingAlg, double scalingFactor)

  Computes a scaled image using the specified parameters.
- static void CreateScaled (const ImagePtr &srcImage, ImagePtr &destImage, ImageScalingAlgorithm scalingAlg, double scalingFactor)

  Computes a scaled image using the specified parameters.
- static ImagePtr CreateNormalized (const ImagePtr &srcImage, const PixelFormatEnums destPixelFormat, SourceDataRange srcDataRange=IMAGE_DATA_RANGE)

  Computes a normalized image.
- static ImagePtr CreateNormalized (const ImagePtr &srcImage, const double min, const double max, SourceDataRange srcDataRange=IMAGE_DATA_RANGE)

  Computes a normalized image.
- static ImagePtr CreateNormalized (const ImagePtr &srcImage, const double min, const double max, const PixelFormatEnums destPixelFormat, SourceDataRange srcDataRange=IMAGE_DATA_RANGE)

  Computes a normalized image.
- static void CreateNormalized (const ImagePtr &srcImage, ImagePtr &destImage, SourceDataRange srcDataRange=IMAGE_DATA_RANGE)

  Computes a normalized image.
- static void CreateNormalized (const ImagePtr &srcImage, ImagePtr &destImage, const double min, const double max, SourceDataRange srcDataRange=IMAGE_DATA_RANGE)

  Computes a normalized image.

14.115.1 Detailed Description

Static helper functions for the image object class.

14.115.2 Member Enumeration Documentation

14.115.2.1 ImageScalingAlgorithm

```
enum ImageScalingAlgorithm
```

Image scaling algorithms.

Enumerator

NEAREST_NEIGHBOR	

14.115.2.2 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 abosolute max and min image data values for the image type (absolute data). By default the abosolute 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: NormalizedValue = ((maxDest - minDest) $*$ (PixelValue - minSrc) / (maxSrc - minSrc)) + 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.

14.115.3 Member Function Documentation

14.115.3.1 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

srcImage	The source image from which to create normalized image
destPixelFormat	The desired pixel format for the normalized image
srcDataRange	The desired option for the source data range to normalize from

Returns

The normalized image

14.115.3.2 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

srcImage	The source image from which to create normalized image
min	The lower bound of the normalization range
max	The upper bound of the normalization range
srcDataRange	The desired option for the source data range to normalize from

Returns

The normalized image

14.115.3.3 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

srcImage	The source image from which to create normalized image
min	The lower bound of the normalization range
max	The upper bound of the normalization range
destPixelFormat	The desired pixel format for the normalized image
srcDataRange	The desired option for the source data range to normalize from

Returns

The normalized image

14.115.3.4 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

srcImage	The source image from which to create normalized image
destImage	The destination image in which to store the normalized image
srcDataRange	The desired option for the source data range to normalize from

14.115.3.5 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

srcImage	The source image from which to create normalized image
destImage	The destination image in which to store the normalized image
min	The lower bound of the normalization range
max	The upper bound of the normalization range
srcDataRange	The desired option for the source data range to normalize from

14.115.3.6 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

srcImage	The source image from which to create scaled image
scalingAlg	The desired image scaling algorithm to use
scalingFactor	The desired image scaling factor to use

Returns

The scaled image

14.115.3.7 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

srcImage	The source image from which to create scaled image
destImage	An image object in which to store the scaled data
scalingAlg	The desired image scaling algorithm to use
scalingFactor	The desired image scaling factor to use

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

- include/ImageUtility.h

14.116 ImageUtilityHeatmap Class Reference

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

Public Types

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

  Color specifiers for the heatmap color gradient.

Static Public Member Functions

- static ImagePtr CreateHeatmap (const ImagePtr &srcImage)

  Computes a heatmap image.
- static void CreateHeatmap (const ImagePtr &srcImage, ImagePtr &destImage)

  Computes a heatmap image.
- static void SetHeatmapColorGradient (const HeatmapColor newLowColor, const HeatmapColor newHigh↩Color)

Sets the heatmap gradient color vector to the new desired range between HEATMAP_BLACK and HEATMAP_W↩ HITE.

- static void GetHeatmapColorGradient (HeatmapColor &currentLowColor, HeatmapColor &currentHighColor)

  Returns the current heatmap gradient color range.
- static void SetHeatmapRange (const unsigned int newLowValue, const unsigned int newHighValue)

  Sets the high and low values used to determine which grayscale values are converted to a color 'heatmap' representation.
- static void GetHeatmapRange (unsigned int &currentLowValue, unsigned int &currentHighValue)

  Returns the current high and low values used in heatmap representations.

14.116.1 Detailed Description

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

14.116.2 Member Enumeration Documentation

14.116.2.1 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	

14.116.3 Member Function Documentation

14.116.3.1 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

srcImage	The source image from which to create the heatmap

See also

SetHeatmapRange()
SetHeatmapColorGradient()

Returns

The heatmap image

14.116.3.2 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

srcImage	The source image from which to create the heatmap
destImage	The destination image in which to store the created heatmap

See also

SetHeatmapRange()
SetHeatmapColorGradient()

14.116.3.3 GetHeatmapColorGradient()

```
static void GetHeatmapColorGradient (
            HeatmapColor & currentLowColor,
            HeatmapColor & currentHighColor )  [static]
```

Returns the current heatmap gradient color range.

Parameters

currentLowColor	Current color at which the gradient begins.
currentHighColor	Current color at which the gradient ends.

See also

SetHeatmapColorGradient()

14.116.3.4 GetHeatmapRange()

```
static void GetHeatmapRange (
            unsigned int & currentLowValue,
            unsigned int & currentHighValue )  [static]
```

Returns the current high and low values used in heatmap representations.

Parameters

currentLowValue	Current value at which color representation begins.
currentHighValue	Current value at which color representation ends.

See also

SetHeatmapRange()

14.116.3.5 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

newLowColor	New color at which to begin the gradient.
newHighColor	New color at which to end the gradient.

14.116.3.6 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

newLowValue	New value at which to begin color representation.
newHighValue	New value at which to end color representation.

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

- include/ImageUtilityHeatmap.h

14.117 ImageUtilityPolarization Class Reference

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

Public Types

- enum PolarizationQuadrant {
  QUADRANT_I0,
  QUADRANT_I45,
  QUADRANT_I90,
  QUADRANT_I135 }

  Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

Static Public Member Functions

- static ImagePtr ExtractPolarQuadrant (const ImagePtr &srcImage, const PolarizationQuadrant desiredQuadrant)

  Extracts all pixels of a specified degree of linear polarization into a new image object.
- static void ExtractPolarQuadrant (const ImagePtr &srcImage, ImagePtr &destQuadImage, const PolarizationQuadrant desiredQuadrant)

  Extracts all pixels of a specified degree of linear polarization into the provided image object.
- static ImagePtr CreateGlareReduced (const ImagePtr &srcImage)

  Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant The source image pixel format must be Polarized8 or BayerRGPolarized8.
- static void CreateGlareReduced (const ImagePtr &srcImage, ImagePtr &destGlareReducedImage)

  Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant The source image pixel format must be Polarized8 or BayerRGPolarized8.
- static ImagePtr CreateStokesS0 (const ImagePtr &srcImage, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

Computes an image representing the overall intensity of light from a polarized image.

- static void CreateStokesS0 (const ImagePtr &srcImage, ImagePtr &destStokesS0Image, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the overall intensity of light from a polarized image.

- static ImagePtr CreateStokesS1 (const ImagePtr &srcImage, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

- static void CreateStokesS1 (const ImagePtr &srcImage, ImagePtr &destStokesS1Image, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

- static ImagePtr CreateStokesS2 (const ImagePtr &srcImage, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the difference in intensity accepted through the polarizers at 45 and -45 to the horizontal.

- static void CreateStokesS2 (const ImagePtr &srcImage, ImagePtr &destStokesS2Image, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the difference in intensity accepted through the polarizers.

- static ImagePtr CreateDoLp (const ImagePtr &srcImage, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the fraction of incident light intensity in the linear polarization states.

- static void CreateDoLp (const ImagePtr &srcImage, ImagePtr &destDoLpImage, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the fraction of incident light intensity in the linear polarization states.

- static ImagePtr CreateAoLp (const ImagePtr &srcImage, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

- static void CreateAoLp (const ImagePtr &srcImage, ImagePtr &destAoLpImg, const ColorProcessingAlgorithm colorProcessingAlg=DEFAULT)

  Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

14.117.1 Detailed Description

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

14.117.2 Member Enumeration Documentation

14.117.2.1 PolarizationQuadrant

```
enum PolarizationQuadrant
```

Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

Enumerator

QUADRANT_I0	The 0 degree of polarization.
QUADRANT_I45	The 45 degree of polarization.
QUADRANT_I90	The 90 degree of polarization.
QUADRANT_I135	The 135 degree of polarization.

14.117.3 Member Function Documentation

14.117.3.1 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

srcImage	The source image from which to extract polarization data
colorProcessingAlg	The color processing algorithm to use for color images

Returns

The angle of linear polarization (aolp) image

14.117.3.2 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

srcImage	The source image from which to extract polarization data
destAolpImg	The destination image in which to store the angle of linear polarization (aolp) image
colorProcessingAlg	The color processing algorithm to use for color images

14.117.3.3 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

srcImage	The source image from which to extract polarization data
colorProcessingAlg	The color processing algorithm to use for color images

Returns

The degree of linear polarization (dolp) image

14.117.3.4 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

srcImage	The source image from which to extract polarization data
destDolpImage	The destination image in which to store the degree of linear polarization (dolp) image
colorProcessingAlg	The color processing algorithm to use for color images

14.117.3.5 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

srcImage	The source image from which to extract polarization data

Returns

The reduced glare image

14.117.3.6 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

srcImage	The source image from which to apply glare reduction
destGlareReducedImage	The destination image in which to store the image with reduced glare

14.117.3.7 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

srcImage	The source image from which to extract polarization data
colorProcessingAlg	The color processing algorithm to use for color images

Returns

The Stokes' S0 image

14.117.3.8 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

srcImage	The source image from which to extract polarization data
destStokesS0Image	The destination image in which to store the Stokes' S0 image
colorProcessingAlg	The color processing algorithm to use for color images

14.117.3.9 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

srcImage	The source image from which to extract polarization data
colorProcessingAlg	The color processing algorithm to use for color images

Returns

The Stokes' S1 image

14.117.3.10 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

srcImage	The source image from which to extract polarization data
destStokesS1Image	The destination image in which to store the Stokes' S1 image
colorProcessingAlg	The color processing algorithm to use for color images

14.117.3.11 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

srcImage	The source image from which to extract polarization data
colorProcessingAlg	The color processing algorithm to use for color images

Returns

The Stokes' S2 image

14.117.3.12 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

srcImage	The source image from which to extract polarization data
destStokesS2Image	The destination image in which to store the Stokes' S2 image
colorProcessingAlg	The color processing algorithm to use for color images

14.117.3.13 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

srcImage	The source image from which to extract polarization data
desiredQuadrant	The polarization quadrant to extract

Returns

The specified polarization quadrant image

14.117.3.14 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

srcImage	The source image from which to extract polarization data
destQuadImage	The destination image in which to store the extracted polarization quadrant
desiredQuadrant	The polarization quadrant to extract

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

- include/ImageUtilityPolarization.h

14.118 InferenceBoundingBox Struct Reference

Inference Bounding Boxes data structure.

Public Attributes

- InferenceBoxType boxType
- int16_t classId
- float32_t confidence
- InferenceBoxRect rect
- InferenceBoxCircle circle
- InferenceBoxRotatedRect rotatedRect

14.118.1 Detailed Description

Inference Bounding Boxes data structure.

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

- include/ChunkDataInference.h

14.119 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.

14.119.1 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

14.120 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

14.121 InferenceBoxRect Struct Reference

Inference Bounding Box Type Data Structures.

Public Attributes

- int16_t topLeftXCoord
- int16_t topLeftYCoord
- int16_t bottomRightXCoord
- int16_t bottomRightYCoord

14.121.1 Detailed Description

Inference Bounding Box Type Data Structures.

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

- include/ChunkDataInference.h

14.122 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

14.123 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

14.123.1 Detailed Description

Vector of integers with reference counting.

14.123.2 Constructor & Destructor Documentation

14.123.2.1 int64_autovector_t() [1/3]

```
int64_autovector_t ( )
```

14.123.2.2 int64_autovector_t() [2/3]

```
int64_autovector_t (
            const int64_autovector_t & obj )
```

14.123.2.3 int64_autovector_t() [3/3]

```
int64_autovector_t (
            size_t n )  [explicit]
```

14.123.2.4 ~int64_autovector_t()

```
virtual ~int64_autovector_t (
            void  )  [virtual]
```

14.123.3 Member Function Documentation

14.123.3.1 operator delete()

```
void operator delete (
            void * pWhere )
```

14.123.3.2 operator new()

```
void* operator new (
            size_t uiSize )
```

14.123.3.3 operator=()

```
int64_autovector_t& operator= (
            const int64_autovector_t & obj )
```

14.123.3.4 operator[]() [1/2]

```
int64_t& operator[] (
            size_t uiIndex )
```

14.123.3.5 operator[]() [2/2]

```
const int64_t& operator[] (
            size_t uiIndex ) const
```

14.123.3.6 size()

```
size_t size ( ) const
```

14.123.4 Member Data Documentation

14.123.4.1 _pCount

```
ATOMIC_VARIABLE* _pCount  [protected]
```

14.123.4.2 _pv

```
std::vector<int64_t>* _pv  [protected]
```

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

- include/SpinGenApi/Autovector.h

14.124 IntegerNode Class Reference

Interface for string properties.

Inheritance diagram for IntegerNode:

Collaboration diagram for IntegerNode:

Public Member Functions

- IntegerNode ()
- IntegerNode (std::shared_ptr< Node::NodeImpl > pInteger)
- virtual ~IntegerNode ()
- virtual void SetValue (int64_t Value, bool Verify=true)

  Set node value.
- virtual IInteger & operator= (int64_t Value)

  Set node value.
- virtual int64_t GetValue (bool Verify=false, bool IgnoreCache=false)

  Get node value.
- virtual int64_t operator() ()

  Get node value.
- virtual int64_t operator* ()

  Get node value.
- virtual int64_t GetMin ()

  Get minimum value allowed.
- virtual int64_t GetMax ()

  Get maximum value allowed.
- virtual EIncMode GetIncMode ()

  Get increment mode.
- virtual int64_t GetInc ()

  Get increment.
- virtual int64_autovector_t GetListOfValidValues (bool bounded=true)

  Get list of valid value.
- virtual ERepresentation GetRepresentation ()

  Get recommended representation.

- virtual GenICam::gcstring GetUnit ()

  Get the physical unit name.
- virtual IFloat * GetFloatAlias ()

  gets the interface of an alias node.
- virtual void ImposeMin (int64_t Value)

  Restrict minimum value.
- virtual void ImposeMax (int64_t Value)

  Restrict maximum value.
- virtual void SetReference (INode *pBase)

  overload SetReference for Integer

Additional Inherited Members

14.124.1 Detailed Description

Interface for string properties.

14.124.2 Constructor & Destructor Documentation

14.124.2.1 IntegerNode() [1/2]

```
IntegerNode ( )
```

14.124.2.2 IntegerNode() [2/2]

```
IntegerNode (
            std::shared_ptr< Node::NodeImpl > pInteger )
```

14.124.2.3 ~IntegerNode()

```
virtual ~IntegerNode ( ) [virtual]
```

14.124.3 Member Function Documentation

14.124.3.1 GetFloatAlias()

```
virtual IFloat* GetFloatAlias ( ) [virtual]
```

gets the interface of an alias node.

14.124.3.2 GetInc()

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

Get increment.

14.124.3.3 GetIncMode()

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

Get increment mode.

14.124.3.4 GetListOfValidValues()

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

Get list of valid value.

14.124.3.5 GetMax()

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

Get maximum value allowed.

14.124.3.6 GetMin()

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

Get minimum value allowed.

14.124.3.7 GetRepresentation()

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

Get recommended representation.

14.124.3.8 GetUnit()

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

Get the physical unit name.

14.124.3.9 GetValue()

```
virtual int64_t GetValue (
            bool Verify = false,
            bool IgnoreCache = false ) [virtual]
```

Get 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

14.124.3.10 ImposeMax()

```
virtual void ImposeMax (
            int64_t Value ) [virtual]
```

Restrict maximum value.

14.124.3.11 ImposeMin()

```
virtual void ImposeMin (
            int64_t Value ) [virtual]
```

Restrict minimum value.

14.124.3.12 operator()()

```
virtual int64_t operator() ( ) [virtual]
```

Get node value.

14.124.3.13 operator*()

```
virtual int64_t operator* ( ) [virtual]
```

Get node value.

14.124.3.14 operator=()

```
virtual IInteger& operator= (
            int64_t Value ) [virtual]
```

Set node value.

14.124.3.15 SetReference()

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

overload SetReference for Integer

Reimplemented from ValueNode.

Reimplemented in IntRegNode.

14.124.3.16 SetValue()

```
virtual void SetValue (
            int64_t 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/IntegerNode.h

14.125 Interface Class Reference

An interface object which holds a list of cameras.

Inheritance diagram for Interface:

Collaboration diagram for Interface:

Public Member Functions

- virtual ~Interface (void)

  Virtual Destructor.
- CameraList GetCameras (bool updateCameras=true) const

Returns a list of cameras available on this interface.

- bool UpdateCameras ()

  Updates the list of cameras on this interface.

- GenApi::INodeMap & GetTLNodeMap () const

  Gets a nodeMap that is generated from a GenICam XML file for the GenTL interface Module.

- void RegisterEventHandler (EventHandler &evtHandlerToRegister)

  Registers an event handler for the interface Event handlers are automatically cleaned up when an interface is removed, and must be registered to interfaces as they arrive.

- void UnregisterEventHandler (EventHandler &evtHandlerToUnregister)

  Unregisters an event handler for the interface.

- bool IsInUse () const

  Checks if the interface is in use by any camera objects.

- void SendActionCommand (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[ ]=NULL) const

  Broadcast an Action Command to all devices on interface.

- bool IsValid ()

  IsValid Checks a flag to determine if interface is still valid for use.

Friends

- class InterfaceInternal

Additional Inherited Members

14.125.1 Detailed Description

An interface object which holds a list of cameras.

14.125.2 Constructor & Destructor Documentation

14.125.2.1 ~Interface()

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

Virtual Destructor.

14.125.3 Member Function Documentation

14.125.3.1 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

updateCameras	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.

14.125.3.2 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.

14.125.3.3 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.

14.125.3.4 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.

14.125.3.5 RegisterEventHandler()

```
void RegisterEventHandler (
            EventHandler & evtHandlerToRegister ) [virtual]
```

Registers an event handler for the interface Event handlers are automatically cleaned up when an interface is removed, and must be registered to interfaces as they arrive.

Note that GEV interfaces experience arrival/removal events when the IP information changes, similar to GEV cameras. Please refer to the EnumerationEvents example for recommended use.

See also

InterfaceEventHandler

Parameters

evtHandlerToRegister	The event handler to register for the interface

Implements IInterface.

14.125.3.6 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

deviceKey	The Action Command's device key
groupKey	The Action Command's group key
groupMask	The Action Command's group mask
actionTime	(Optional) Time when to assert a future action. Zero means immediate action.
pResultSize	(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.
results	(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.

14.125.3.7 UnregisterEventHandler()

```
void UnregisterEventHandler (
            EventHandler & evtHandlerToUnregister )  [virtual]
```

Unregisters an event handler for the interface.

See also

InterfaceEventHandler

Parameters

evtHandlerToUnregister	The event handler to unregister from the interface

Implements IInterface.

14.125.3.8 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::Get↩Cameras() 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.

14.125.4 Friends And Related Function Documentation

14.125.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

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

- include/Interface.h

14.126 InterfaceArrivalEventHandler Class Reference

An event handler for capturing the interface arrival event.

Inheritance diagram for InterfaceArrivalEventHandler:

Collaboration diagram for InterfaceArrivalEventHandler:

Public Member Functions

- InterfaceArrivalEventHandler ()

  Default constructor.
- virtual ~InterfaceArrivalEventHandler ()

  Virtual destructor.
- virtual void OnInterfaceArrival (std::string interfaceID)=0

  Interface arrival event callback.

Protected Member Functions

- InterfaceArrivalEventHandler & operator= (const InterfaceArrivalEventHandler &)

  Assignment operator.

Additional Inherited Members

14.126.1 Detailed Description

An event handler for capturing the interface arrival event.

Note that only GEV interface arrivals are currently handled.

14.126.2 Constructor & Destructor Documentation

14.126.2.1 InterfaceArrivalEventHandler()

```
InterfaceArrivalEventHandler ( )
```

Default constructor.

14.126.2.2 ~InterfaceArrivalEventHandler()

```
virtual ~InterfaceArrivalEventHandler ( )  [virtual]
```

Virtual destructor.

14.126.3 Member Function Documentation

14.126.3.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
            std::string interfaceID )  [pure virtual]
```

Interface arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

interfaceID	The ID of the interface that arrived

Implements IInterfaceArrivalEventHandler.

14.126.3.2 operator=()

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

Assignment operator.

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

- include/InterfaceArrivalEventHandler.h

14.127 InterfaceEventHandler Class Reference

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

Inheritance diagram for InterfaceEventHandler:

Collaboration diagram for InterfaceEventHandler:

Public Member Functions

- InterfaceEventHandler ()

  Default constructor.
- virtual ~InterfaceEventHandler ()

  Virtual destructor.
- virtual void OnDeviceArrival (uint64_t serialNumber)=0

  Device arrival event callback.
- virtual void OnDeviceRemoval (uint64_t serialNumber)=0

  Callback to the device removal event.

Protected Member Functions

- InterfaceEventHandler & operator= (const InterfaceEventHandler &)

  Assignment operator.

Additional Inherited Members

14.127.1 Detailed Description

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

14.127.2 Constructor & Destructor Documentation

14.127.2.1 InterfaceEventHandler()

```
InterfaceEventHandler ( )
```

Default constructor.

14.127.2.2 ~InterfaceEventHandler()

```
virtual ~InterfaceEventHandler ( ) [virtual]
```

Virtual destructor.

14.127.3 Member Function Documentation

14.127.3.1 OnDeviceArrival()

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

Device arrival event callback.

Implements IInterfaceEventHandler.

Implemented in InterfaceEventHandlerImpl, and InterfaceEventHandlerImpl.

14.127.3.2 OnDeviceRemoval()

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

Callback to the device removal event.

Parameters

serialNumber	The serial number of the removed device

Implements IInterfaceEventHandler.

Implemented in InterfaceEventHandlerImpl, and InterfaceEventHandlerImpl.

14.127.3.3 operator=()

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

Assignment operator.

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

- include/InterfaceEventHandler.h

14.128 InterfaceEventHandlerImpl Class Reference

Inheritance diagram for InterfaceEventHandlerImpl:

Collaboration diagram for InterfaceEventHandlerImpl:

Public Member Functions

- InterfaceEventHandlerImpl (SystemPtr system)
- ~InterfaceEventHandlerImpl ()
- void OnDeviceArrival (uint64_t deviceSerialNumber)

  Device arrival event callback.
- void OnDeviceRemoval (uint64_t deviceSerialNumber)

  Callback to the device removal event.
- InterfaceEventHandlerImpl (SystemPtr system)
- InterfaceEventHandlerImpl (InterfacePtr iface, std::string interfaceID)
- ~InterfaceEventHandlerImpl ()
- void PrintGenericHandlerMessage (const unsigned long long deviceCount)
- void OnDeviceArrival (uint64_t deviceSerialNumber)

  Device arrival event callback.
- void OnDeviceRemoval (uint64_t deviceSerialNumber)

  Callback to the device removal event.
- std::string GetInterfaceId ()

Additional Inherited Members

14.128.1 Constructor & Destructor Documentation

14.128.1.1 InterfaceEventHandlerImpl() [1/3]

```
InterfaceEventHandlerImpl (
            SystemPtr system )  [inline]
```

14.128.1.2 ~InterfaceEventHandlerImpl() [1/2]

```
~InterfaceEventHandlerImpl ( )  [inline]
```

14.128.1.3 InterfaceEventHandlerImpl() [2/3]

```
InterfaceEventHandlerImpl (
            SystemPtr system )  [inline]
```

14.128.1.4 InterfaceEventHandlerImpl() [3/3]

```
InterfaceEventHandlerImpl (
            InterfacePtr iface,
            std::string interfaceID )  [inline]
```

14.128.1.5 ~InterfaceEventHandlerImpl() [2/2]

```
~InterfaceEventHandlerImpl ( )  [inline]
```

14.128.2 Member Function Documentation

14.128.2.1 GetInterfaceId()

```
std::string GetInterfaceId ( )  [inline]
```

14.128.2.2 OnDeviceArrival() [1/2]

```
void OnDeviceArrival (
            uint64_t serialNumber )  [inline], [virtual]
```

Device arrival event callback.

Implements InterfaceEventHandler.

14.128.2.3 OnDeviceArrival() [2/2]

```
void OnDeviceArrival (
            uint64_t serialNumber )  [inline], [virtual]
```

Device arrival event callback.

Implements InterfaceEventHandler.

14.128.2.4 OnDeviceRemoval() [1/2]

```
void OnDeviceRemoval (
            uint64_t serialNumber )  [inline], [virtual]
```

Callback to the device removal event.

Parameters

serialNumber	The serial number of the removed device

Implements InterfaceEventHandler.

14.128.2.5 OnDeviceRemoval() [2/2]

```
void OnDeviceRemoval (
            uint64_t serialNumber )  [inline], [virtual]
```

Callback to the device removal event.

Parameters

serialNumber	The serial number of the removed device

Implements InterfaceEventHandler.

14.128.2.6 PrintGenericHandlerMessage()

```
void PrintGenericHandlerMessage (
            const unsigned long long deviceCount )  [inline]
```

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

- src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp
- src/EnumerationEvents/EnumerationEvents.cpp

14.129 InterfaceList Class Reference

A list of the available interfaces on the system.

Inheritance diagram for InterfaceList:

Collaboration diagram for InterfaceList:

Public Member Functions

- InterfaceList (void)
- virtual ~InterfaceList (void)
- InterfaceList (const InterfaceList &iface)
- InterfaceList & operator= (const InterfaceList &iface)

  Assignment operator.
- InterfacePtr operator[] (unsigned int index)

  Array subscription operators.
- unsigned int GetSize () const

  Returns the size of the interface list.
- InterfacePtr GetByIndex (unsigned int index) const

  Returns a pointer to an Interface object at the "index".
- void Clear ()

  Clears the list of interfaces and destroys their corresponding objects.
- void Append (const InterfaceList *list)

  Appends a copy of the interfact list.

Friends

- class SystemImpl
- class ProducerImpl

Additional Inherited Members

14.129.1 Detailed Description

A list of the available interfaces on the system.

14.129.2 Constructor & Destructor Documentation

14.129.2.1 InterfaceList() [1/2]

```
InterfaceList (
            void  )
```

14.129.2.2 ~InterfaceList()

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

14.129.2.3 InterfaceList() [2/2]

```
InterfaceList (
            const InterfaceList & iface )
```

14.129.3 Member Function Documentation

14.129.3.1 Append()

```
void Append (
            const InterfaceList * list )  [virtual]
```

Appends a copy of the interfact list.

Parameters

list	Another InterfaceList object, whose elements are added to this list.

Implements IInterfaceList.

14.129.3.2 Clear()

```
void Clear ( )  [virtual]
```

Clears the list of interfaces and destroys their corresponding objects.

It is important to first make sure there are no referenced cameras still in use before calling Clear(). If a camera on any of the interfaces is still in use this function will throw an exception.

Implements IInterfaceList.

14.129.3.3 GetByIndex()

```
InterfacePtr GetByIndex (
            unsigned int index ) const  [virtual]
```

Returns a pointer to an Interface object at the "index".

Parameters

index	The index at which to retrieve the Interface object

Returns

A pointer to an Interface object.

Implements IInterfaceList.

14.129.3.4 GetSize()

```
unsigned int GetSize ( ) const  [virtual]
```

Returns the size of the interface list.

The size is the number of Interface objects stored in the list.

Returns

An integer that represents the list size.

Implements IInterfaceList.

14.129.3.5 operator=()

```
InterfaceList& operator= (
            const InterfaceList & iface )
```

Assignment operator.

14.129.3.6 operator[]()

```
InterfacePtr operator[] (
            unsigned int index )  [virtual]
```

Array subscription operators.

Implements IInterfaceList.

14.129.4 Friends And Related Function Documentation

14.129.4.1 ProducerImpl

```
friend class ProducerImpl [friend]
```

14.129.4.2 SystemImpl

```
friend class SystemImpl [friend]
```

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

- include/InterfaceList.h

14.130 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

14.130.1 Detailed Description

A reference tracked pointer to the interface object.

14.130.2 Constructor & Destructor Documentation

14.130.2.1 InterfacePtr() [1/4]

```
InterfacePtr ( ) throw )   [inline]
```

Default Constructor.

14.130.2.2 InterfacePtr() [2/4]

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

Default Constructor with argument.

14.130.2.3 InterfacePtr() [3/4]

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

14.130.2.4 InterfacePtr() [4/4]

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

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

- include/InterfacePtr.h

14.131 InterfaceRemovalEventHandler Class Reference

An event handler for capturing the interface removal event.

Inheritance diagram for InterfaceRemovalEventHandler:

Collaboration diagram for InterfaceRemovalEventHandler:

Public Member Functions

- InterfaceRemovalEventHandler ()

  Default Constructor.
- virtual ~InterfaceRemovalEventHandler ()

  Virtual Destructor.
- virtual void OnInterfaceRemoval (std::string interfaceID)=0

  Interface removal event callback.

Protected Member Functions

- InterfaceRemovalEventHandler & operator= (const InterfaceRemovalEventHandler &)

  Assignment operator.

Additional Inherited Members

14.131.1 Detailed Description

An event handler for capturing the interface removal event.

Note that only GEV interface removals are currently handled.

14.131.2 Constructor & Destructor Documentation

14.131.2.1 InterfaceRemovalEventHandler()

```
InterfaceRemovalEventHandler ( )
```

Default Constructor.

14.131.2.2 ~InterfaceRemovalEventHandler()

```
virtual ~InterfaceRemovalEventHandler ( )  [virtual]
```

Virtual Destructor.

14.131.3 Member Function Documentation

14.131.3.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
            std::string interfaceID )  [pure virtual]
```

Interface removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

interfaceID	The ID of the interface removed

Implements IInterfaceRemovalEventHandler.

14.131.3.2 operator=()

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

Assignment operator.

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

- include/InterfaceRemovalEventHandler.h

14.132 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

14.132.1 Detailed Description

Interface for string properties.

14.132.2 Constructor & Destructor Documentation

14.132.2.1 IntRegNode() [1/2]

```
IntRegNode ( )
```

14.132.2.2 IntRegNode() [2/2]

```
IntRegNode (
            std::shared_ptr< Node::NodeImpl > pInteger )
```

14.132.2.3 ~IntRegNode()

```
virtual ~IntRegNode ( ) [virtual]
```

14.132.3 Member Function Documentation

14.132.3.1 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

14.133 IpInfo Struct Reference

Public Member Functions

- IpInfo ()

Public Attributes

- std::string ipAddress
- std::string subnetMask
- std::string gateway
- unsigned int subnetLength

14.133.1 Constructor & Destructor Documentation

14.133.1.1 IpInfo()

```
IpInfo ( ) [inline]
```

14.133.2 Member Data Documentation

14.133.2.1 gateway

```
std::string gateway
```

14.133.2.2 ipAddress

```
std::string ipAddress
```

14.133.2.3 subnetLength

```
unsigned int subnetLength
```

14.133.2.4 subnetMask

```
std::string subnetMask
```

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

- include/AdapterConfig.h

14.134 ISystem Class Reference

The interface file for System.

Inheritance diagram for ISystem:

Collaboration diagram for ISystem:

Public Member Functions

- virtual ~ISystem ()
- virtual void ReleaseInstance ()=0
- virtual InterfaceList GetInterfaces (bool updateInterface=true)=0
- virtual void UpdateInterfaceList ()=0
- virtual CameraList GetCameras (bool updateInterfaces=true, bool updateCameras=true)=0
- virtual bool UpdateCameras (bool updateInterfaces=true)=0
- virtual void RegisterEventHandler (EventHandler &evtHandlerToRegister)=0
- virtual void UnregisterEventHandler (EventHandler &evtHandlerToUnregister)=0
- virtual void RegisterInterfaceEventHandler (EventHandler &evtHandlerToRegister, bool updateInterface=true)=0
- virtual void UnregisterInterfaceEventHandler (EventHandler &evtHandlerToUnregister)=0
- virtual void RegisterLoggingEventHandler (LoggingEventHandler &handler)=0
- virtual void UnregisterAllLoggingEventHandlers ()=0

- virtual void UnregisterLoggingEventHandler (LoggingEventHandler &handler)=0
- virtual void SetLoggingEventPriorityLevel (SpinnakerLogLevel level)=0
- virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ()=0
- virtual bool IsInUse ()=0
- virtual void SendActionCommand (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[ ]=NULL)=0
- virtual const LibraryVersion GetLibraryVersion ()=0
- virtual GenApi::INodeMap & GetTLNodeMap () const =0

Public Attributes

- TransportLayerSystem TLSystem

Protected Member Functions

- ISystem ()
- ISystem (const ISystem &)
- ISystem & operator= (const ISystem &)

Friends

- class SystemPtrInternal

14.134.1 Detailed Description

The interface file for System.

14.134.2 Constructor & Destructor Documentation

14.134.2.1 ~ISystem()

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

14.134.2.2 ISystem() [1/2]

```
ISystem ( )  [inline], [protected]
```

14.134.2.3 ISystem() [2/2]

```
ISystem (
            const ISystem &  )  [inline], [protected]
```

14.134.3 Member Function Documentation

14.134.3.1 GetCameras()

```
virtual CameraList GetCameras (
            bool updateInterfaces = true,
            bool updateCameras = true )  [pure virtual]
```

Implemented in System.

14.134.3.2 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
            bool updateInterface = true )  [pure virtual]
```

Implemented in System.

14.134.3.3 GetLibraryVersion()

```
virtual const LibraryVersion GetLibraryVersion ( )  [pure virtual]
```

Implemented in System.

14.134.3.4 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( )  [pure virtual]
```

Implemented in System.

14.134.3.5 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const  [pure virtual]
```

Implemented in System.

14.134.3.6 IsInUse()

```
virtual bool IsInUse ( )  [pure virtual]
```

Implemented in System.

14.134.3.7 operator=()

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

14.134.3.8 RegisterEventHandler()

```
virtual void RegisterEventHandler (
            EventHandler & evtHandlerToRegister )  [pure virtual]
```

Implemented in System.

14.134.3.9 RegisterInterfaceEventHandler()

```
virtual void RegisterInterfaceEventHandler (
            EventHandler & evtHandlerToRegister,
            bool updateInterface = true )  [pure virtual]
```

Implemented in System.

14.134.3.10 RegisterLoggingEventHandler()

```
virtual void RegisterLoggingEventHandler (
            LoggingEventHandler & handler )  [pure virtual]
```

Implemented in System.

14.134.3.11 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [pure virtual]
```

Implemented in System.

14.134.3.12 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.

14.134.3.13 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
            SpinnakerLogLevel level ) [pure virtual]
```

Implemented in System.

14.134.3.14 UnregisterAllLoggingEventHandlers()

```
virtual void UnregisterAllLoggingEventHandlers ( ) [pure virtual]
```

Implemented in System.

14.134.3.15 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
            EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in System.

14.134.3.16 UnregisterInterfaceEventHandler()

```
virtual void UnregisterInterfaceEventHandler (
            EventHandler & evtHandlerToUnregister )  [pure virtual]
```

Implemented in System.

14.134.3.17 UnregisterLoggingEventHandler()

```
virtual void UnregisterLoggingEventHandler (
            LoggingEventHandler & handler )  [pure virtual]
```

Implemented in System.

14.134.3.18 UpdateCameras()

```
virtual bool UpdateCameras (
            bool updateInterfaces = true )  [pure virtual]
```

Implemented in System.

14.134.3.19 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( )  [pure virtual]
```

Implemented in System.

14.134.4 Friends And Related Function Documentation

14.134.4.1 SystemPtrInternal

```
friend class SystemPtrInternal  [friend]
```

14.134.5 Member Data Documentation

14.134.5.1 TLSystem

`TransportLayerSystem TLSystem`

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

- include/Interface/ISystem.h

14.135 ISystemEventHandler Class Reference

Inheritance diagram for ISystemEventHandler:

Collaboration diagram for ISystemEventHandler:

Public Member Functions

- virtual ~ISystemEventHandler ()
- virtual void OnInterfaceArrival (std::string interfaceID)=0
- virtual void OnInterfaceRemoval (std::string interfaceID)=0

Protected Member Functions

- ISystemEventHandler ()
- ISystemEventHandler (const ISystemEventHandler &)
- ISystemEventHandler & operator= (const ISystemEventHandler &)

Additional Inherited Members

14.135.1 Constructor & Destructor Documentation

14.135.1.1 ~ISystemEventHandler()

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

14.135.1.2 ISystemEventHandler() [1/2]

```
ISystemEventHandler ( )  [inline], [protected]
```

14.135.1.3 ISystemEventHandler() [2/2]

```
ISystemEventHandler (
            const ISystemEventHandler &  )  [inline], [protected]
```

14.135.2 Member Function Documentation

14.135.2.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
            std::string interfaceID )  [pure virtual]
```

Implements IInterfaceArrivalEventHandler.

Implemented in SystemEventHandlerImpl, and SystemEventHandler.

14.135.2.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
            std::string interfaceID )  [pure virtual]
```

Implements IInterfaceRemovalEventHandler.

Implemented in SystemEventHandlerImpl, and SystemEventHandler.

14.135.2.3 operator=()

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

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

- include/Interface/ISystemEventHandler.h

14.136 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.

14.136.1 Detailed Description

Options for saving JPEG image.

14.136.2 Constructor & Destructor Documentation

14.136.2.1 JPEGOption()

```
JPEGOption ( ) [inline]
```

14.136.3 Member Data Documentation

14.136.3.1 progressive

```
bool progressive
```

Whether to save as a progressive JPEG file.

14.136.3.2 quality

```
unsigned int quality
```

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

14.136.3.3 reserved

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

Reserved for future use.

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

- include/SpinnakerDefs.h

14.137 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.

14.137.1 Detailed Description

Options for saving JPEG2000 image.

14.137.2 Constructor & Destructor Documentation

14.137.2.1 JPG2Option()

```
JPG2Option ( ) [inline]
```

14.137.3 Member Data Documentation

14.137.3.1 quality

```
unsigned int quality
```

JPEG saving quality in range (1-512).

14.137.3.2 reserved

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

Reserved for future use.

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

- include/SpinnakerDefs.h

14.138 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.

14.138.1 Detailed Description

Provides easier access to the current version of Spinnaker.

14.138.2 Member Data Documentation

14.138.2.1 build

```
unsigned int build
```

Build number of the library.

14.138.2.2 major

```
unsigned int major
```

Major version of the library.

14.138.2.3 minor

```
unsigned int minor
```

Minor version of the library.

14.138.2.4 type

```
unsigned int type
```

Version type of the library.

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

- include/SpinnakerDefs.h

14.139 LockableObject< Object >::Lock Class Reference

A scopelevel Lock class.

Public Member Functions

- Lock (const LockableObject< Object > &obj)
- ~Lock ()

14.139.1 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.

14.139.2 Constructor & Destructor Documentation

14.139.2.1 Lock()

```
Lock (
            const LockableObject< Object > & obj )  [inline]
```

14.139.2.2 ~Lock()

```
~Lock ( )  [inline]
```

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

- include/SpinGenApi/GCSynch.h

14.140 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

14.140.1 Detailed Description

template<class Object>
class Spinnaker::GenICam::LockableObject< Object >

Instance-Lock for an object.

14.140.2 Member Function Documentation

14.140.2.1 GetLock()

```
Lock GetLock ( ) const  [inline]
```

Get a new lock.

14.140.3 Friends And Related Function Documentation

14.140.3.1 Lock

```
friend class Lock  [friend]
```

14.140.4 Member Data Documentation

14.140.4.1 m_Lock

```
CLock m_Lock  [mutable]
```

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

- include/SpinGenApi/GCSynch.h

14.141 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

14.141.1 Detailed Description

The LoggingEventData object.

14.141.2 Constructor & Destructor Documentation

14.141.2.1 ~LoggingEventData()

```
~LoggingEventData ( )
```

Default Destructor.

14.141.2.2 LoggingEventData()

```
LoggingEventData (
            void * data )  [protected]
```

Default Constructor.

14.141.3 Member Function Documentation

14.141.3.1 GetCategoryName()

```
const char* GetCategoryName ( )
```

Gets the logging event category name.

Returns

The category name

14.141.3.2 GetLogMessage()

```
const char* GetLogMessage ( )
```

Gets the logging event message.

Returns

The log message

14.141.3.3 GetNDC()

```
const char* GetNDC ( )
```

Gets the logging event's Nested Diagnostic Context (NDC).

Returns

The log event's NDC

14.141.3.4 GetPriority()

```
const int GetPriority ( )
```

Gets the logging event priority.

Returns

The log priority

14.141.3.5 GetPriorityName()

```
const char* GetPriorityName ( )
```

Gets the logging event priority name.

Returns

The priority name of the log

14.141.3.6 GetThreadName()

```
const char* GetThreadName ( )
```

Gets the logging event thread name.

Returns

The thread name

14.141.3.7 GetTimestamp()

```
const char* GetTimestamp ( )
```

Gets the logging event time stamp.

Returns

The time stamp of the log

14.141.4 Friends And Related Function Documentation

14.141.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

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

- include/LoggingEventData.h

14.142 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

14.142.1 Detailed Description

A reference tracked pointer to the LoggingEvent object.

14.142.2 Constructor & Destructor Documentation

14.142.2.1 LoggingEventDataPtr() [1/4]

```
LoggingEventDataPtr ( ) throw )   [inline]
```

Default Constructor.

14.142.2.2 LoggingEventDataPtr() [2/4]

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

Default Constructor with argument.

14.142.2.3 LoggingEventDataPtr() [3/4]

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

Default Constructor with argument.

14.142.2.4 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

14.143 LoggingEventHandler Class Reference

An event handler for capturing the device logging event.

Inheritance diagram for LoggingEventHandler:

Collaboration diagram for LoggingEventHandler:

Public Member Functions

- LoggingEventHandler ()

Default constructor.

- ~LoggingEventHandler ()

  Virtual destructor.

- virtual void OnLogEvent (LoggingEventDataPtr eventPtr)=0

  The callback for the log event.

Protected Member Functions

- LoggingEventHandler & operator= (const LoggingEventHandler &)

  Assignment operator.

Additional Inherited Members

14.143.1 Detailed Description

An event handler for capturing the device logging event.

14.143.2 Constructor & Destructor Documentation

14.143.2.1 LoggingEventHandler()

```
LoggingEventHandler ( )
```

Default constructor.

14.143.2.2 ~LoggingEventHandler()

```
~LoggingEventHandler ( )
```

Virtual destructor.

14.143.3 Member Function Documentation

14.143.3.1 OnLogEvent()

```
virtual void OnLogEvent (
            LoggingEventDataPtr eventPtr )  [pure virtual]
```

The callback for the log event.

Parameters

eventPtr	The logging event pointer

Implements ILoggingEventHandler.

14.143.3.2 operator=()

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

Assignment operator.

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

- include/LoggingEventHandler.h

14.144 LoggingEventHandlerImpl Class Reference

Inheritance diagram for LoggingEventHandlerImpl:

Collaboration diagram for LoggingEventHandlerImpl:

Additional Inherited Members

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

- src/Logging/Logging.cpp

14.145 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

14.145.1 Detailed Description

template<class Client, class Member>
class Spinnaker::GenApi::Member_NodeCallback< Client, Member >

Container for a member function pointer.

14.145.2 Member Typedef Documentation

14.145.2.1 PMEMBERFUNC

```
typedef void(Client::* PMEMBERFUNC) (INode *)
```

Member function type.

14.145.3 Constructor & Destructor Documentation

14.145.3.1 Member_NodeCallback()

```
Member_NodeCallback (
            INode * pNode,
            Client & client,
            Member member,
            ECallbackType CallbackType )  [inline]
```

Constructor.

14.145.4 Member Function Documentation

14.145.4.1 Destroy()

```
virtual void Destroy ( )  [inline], [virtual]
```

destroys the object

Implements CNodeCallback.

14.145.4.2 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

14.146 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]

14.146.1 Detailed Description

Options for saving MJPG files.

14.146.2 Constructor & Destructor Documentation

14.146.2.1 MJPGOption()

```
MJPGOption ( ) [inline]
```

14.146.3 Member Data Documentation

14.146.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

14.146.3.2 quality

```
unsigned int quality
```

Image quality (1-100)

14.146.3.3 reserved

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

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

- include/SpinVideoDefs.h

14.147 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, GenI↩Cam::gcstring &AttributeStr)

Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.

- virtual void ImposeAccessMode (EAccessMode ImposedAccessMode)

  Imposes an access mode to the natural access mode of the node.
- virtual void ImposeVisibility (EVisibility ImposedVisibility)

  Imposes a visibility to the natural visibility of the node.
- virtual INode ∗ GetAlias () const

  Retrieves the a node which describes the same feature in a different way.
- virtual INode ∗ GetCastAlias () const

  Retrieves the a node which describes the same feature so that it can be casted.
- virtual GenICam::gcstring GetDocuURL () const

  Gets a URL pointing to the documentation of that feature.
- virtual bool IsDeprecated () const

  True if the node should not be used any more.
- virtual EInterfaceType GetPrincipalInterfaceType () const

  Get the type of the main interface of a node.
- virtual bool IsFeature () const

  True if the node can be reached via category nodes from a category node named "Root".
- void SetNodeHandle (std::shared_ptr< Node::NodeImpl > pNodeHandle)

  Set Node handle.
- std::shared_ptr< Node::NodeImpl > GetNodeHandle () const

  Get Node handle.
- virtual EAccessMode GetAccessMode () const

  Base interface overrides.
- virtual bool IsSelector () const

  Selector interface overrides.
- virtual void GetSelectedFeatures (FeatureList_t &) const

  retrieve the group of selected features
- virtual void GetSelectingFeatures (FeatureList_t &) const

  retrieve the group of features selecting this node
- virtual void SetReference (INode ∗pBase)

  Reference interface overrides \ingroup Spinnaker_GenApi_PublicImpl.
- virtual void SetReference (ISelector ∗pBase)
- void SetNodeMap (INodeMap ∗pNodeMap)
- virtual bool operator== (int nullPtr) const
- virtual bool operator!= (int nullPtr) const

Protected Attributes

- std::shared_ptr< Node::NodeImpl > m_pNodeData
- std::list< CallbackHandleType_t ∗ > m_Callbacks

  List of callbacks.
- INodeMap ∗ m_pNodeMap

14.147.1 Detailed Description

class common to all nodes

14.147.2 Constructor & Destructor Documentation

14.147.2.1 Node() [1/2]

```
Node ( )
```

Constructor.

14.147.2.2 Node() [2/2]

```
Node (
            std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Constructor.

14.147.2.3 ~Node()

```
~Node ( )
```

Destructor.

14.147.3 Member Function Documentation

14.147.3.1 DeregisterCallback()

```
virtual bool DeregisterCallback (
            CallbackHandleType hCallback )  [virtual]
```

De register change callback Destroys CNodeCallback object.

Returns

true if the callback handle was valid

14.147.3.2 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const  [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented in PortRecorder, and PortReplay.

14.147.3.3 GetAlias()

```
virtual INode* GetAlias ( ) const  [virtual]
```

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

14.147.3.4 GetCachingMode()

```
virtual ECachingMode GetCachingMode ( ) const  [virtual]
```

Get Caching Mode.

14.147.3.5 GetCastAlias()

```
virtual INode* GetCastAlias ( ) const  [virtual]
```

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

14.147.3.6 GetChildren()

```
virtual void GetChildren (
            GenApi::NodeList_t & Children,
            ELinkType LinkType = ctReadingChildren ) const  [virtual]
```

Get all nodes this node directly depends on.

Parameters

out	*Children*	List of children nodes
	LinkType	The link type

14.147.3.7 GetDescription()

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

Get a long description of the node.

14.147.3.8 GetDeviceName()

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

Get a name of the device.

14.147.3.9 GetDisplayName()

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

Get a name string for display.

14.147.3.10 GetDocuURL()

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

Gets a URL pointing to the documentation of that feature.

14.147.3.11 GetEventID()

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

Get the EventId of the node.

14.147.3.12 GetName()

```
virtual GenICam::gcstring GetName (
            bool FullQualified = false ) const  [virtual]
```

Get node name.

14.147.3.13 GetNameSpace()

```
virtual GenApi::ENameSpace GetNameSpace ( ) const  [virtual]
```

Get name space.

14.147.3.14 GetNodeHandle()

```
std::shared_ptr<Node::NodeImpl> GetNodeHandle ( ) const
```

Get Node handle.

14.147.3.15 GetNodeMap()

```
virtual INodeMap* GetNodeMap ( ) const  [virtual]
```

Retrieves the central node map.

14.147.3.16 GetParents()

```
virtual void GetParents (
            GenApi::NodeList_t & Parents ) const  [virtual]
```

Gets all nodes this node is directly depending on.

Parameters

out	*Parents*	List of parent nodes

14.147.3.17 GetPollingTime()

```
virtual int64_t GetPollingTime ( ) const  [virtual]
```

recommended polling time (for not cacheable nodes)

14.147.3.18 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const  [virtual]
```

Get the type of the main interface of a node.

14.147.3.19 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.

14.147.3.20 GetPropertyNames()

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

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

14.147.3.21 GetSelectedFeatures()

```
virtual void GetSelectedFeatures (
            FeatureList_t &  ) const  [virtual]
```

retrieve the group of selected features

14.147.3.22 GetSelectingFeatures()

```
virtual void GetSelectingFeatures (
            FeatureList_t &  ) const  [virtual]
```

retrieve the group of features selecting this node

14.147.3.23 GetToolTip()

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

Get a short description of the node.

14.147.3.24 GetVisibility()

```
virtual EVisibility GetVisibility ( ) const  [virtual]
```

Get the recommended visibility of the node.

14.147.3.25 ImposeAccessMode()

```
virtual void ImposeAccessMode (
            EAccessMode ImposedAccessMode )  [virtual]
```

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

14.147.3.26 ImposeVisibility()

```
virtual void ImposeVisibility (
            EVisibility ImposedVisibility )  [virtual]
```

Imposes a visibility to the natural visibility of the node.

14.147.3.27 InvalidateNode()

```
virtual void InvalidateNode ( )  [virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

14.147.3.28 IsAccessModeCacheable()

```
virtual EYesNo IsAccessModeCacheable ( ) const  [virtual]
```

True if the AccessMode can be cached.

14.147.3.29 IsCachable()

```
virtual bool IsCachable ( ) const  [virtual]
```

Is the node value cacheable.

14.147.3.30 IsDeprecated()

```
virtual bool IsDeprecated ( ) const  [virtual]
```

True if the node should not be used any more.

14.147.3.31 IsFeature()

```
virtual bool IsFeature ( ) const  [virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

14.147.3.32 IsSelector()

```
virtual bool IsSelector ( ) const  [virtual]
```

Selector interface overrides.

true if this feature selects a group of features

14.147.3.33 IsStreamable()

```
virtual bool IsStreamable ( ) const  [virtual]
```

True if the node is streamable.

14.147.3.34 operator"!=()

```
virtual bool operator!= (
            int nullPtr ) const  [virtual]
```

14.147.3.35 operator==()

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

14.147.3.36 RegisterCallback()

```
virtual CallbackHandleType RegisterCallback (
            CNodeCallback * pCallback )  [virtual]
```

Register change callback Takes ownership of the CNodeCallback object.

14.147.3.37 SetNodeHandle()

```
void SetNodeHandle (
            std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Set Node handle.

14.147.3.38 SetNodeMap()

```
void SetNodeMap (
            INodeMap * pNodeMap )
```

14.147.3.39 SetReference() [1/2]

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

Reference interface overrides \ingroup Spinnaker_GenApi_PublicImpl.

Reimplemented in FloatNode, PortNode, IntegerNode, EnumNode, CEnumerationTRef< EnumT >, String↩Node, ValueNode, RegisterNode, BooleanNode, CommandNode, EnumEntryNode, CategoryNode, StringReg↩Node, FloatRegNode, and IntRegNode.

14.147.3.40 SetReference() [2/2]

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

14.147.4 Member Data Documentation

14.147.4.1 m_Callbacks

```
std::list<CallbackHandleType_t*> m_Callbacks  [protected]
```

List of callbacks.

14.147.4.2 m_pNodeData

```
std::shared_ptr<Node::NodeImpl> m_pNodeData  [protected]
```

14.147.4.3 m_pNodeMap

```
INodeMap* m_pNodeMap  [protected]
```

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

- include/SpinGenApi/Node.h

14.148 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.

14.148.1 Detailed Description

Smart pointer template for NodeMaps with create function.

Parameters

TCameraParams	The camera specific parameter class (auto generated from camera xml file)

14.148.2 Constructor & Destructor Documentation

14.148.2.1 NodeMap()

```
NodeMap (
            GenICam::gcstring DeviceName = "Device" )
```

Constructor.

14.148.2.2 ~NodeMap()

```
virtual ~NodeMap ( )  [virtual]
```

Destructor.

14.148.3 Member Function Documentation

14.148.3.1 ClearXMLCache()

```
static bool ClearXMLCache ( )  [static]
```

Clears the cache of the camera description files.

14.148.3.2 Connect() [1/2]

```
virtual bool Connect (
            IPort * pPort,
            const GenICam::gcstring & PortName ) const  [virtual]
```

Connects a port to a port node with given name.

14.148.3.3 Connect() [2/2]

```
virtual bool Connect (
            IPort * pPort ) const  [virtual]
```

Connects a port to the standard port "Device".

14.148.3.4 Destroy()

```
void Destroy ( )
```

Destroys the node map.

14.148.3.5 GetDeviceName()

```
virtual GenICam::gcstring GetDeviceName ( )  [virtual]
```

Get device name.

14.148.3.6 GetDeviceVersion()

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

Get the version of the device description file.

14.148.3.7 GetGenApiVersion()

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

Get the version of the DLL's GenApi implementation.

14.148.3.8 GetLock()

```
virtual CLock& GetLock ( ) const  [virtual]
```

Returns the lock which guards the node map.

14.148.3.9 GetModelName()

```
virtual GenICam::gcstring GetModelName ( )  [virtual]
```

Get the model name.

14.148.3.10 GetNode()

```
virtual INode* GetNode (
            const GenICam::gcstring & key ) const  [virtual]
```

Retrieves the node from the central map by name.

14.148.3.11 GetNodeMapHandle()

```
void* GetNodeMapHandle ( ) const
```

14.148.3.12 GetNodes()

```
virtual void GetNodes (
            NodeList_t & Nodes ) const  [virtual]
```

Retrieves all nodes in the node map.

14.148.3.13 GetNumNodes()

```
virtual uint64_t GetNumNodes ( ) const  [virtual]
```

Get the number of nodes in the map.

14.148.3.14 GetProductGuid()

```
virtual GenICam::gcstring GetProductGuid ( )  [virtual]
```

Get the GUID describing the product.

14.148.3.15 GetSchemaVersion()

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

Get the schema version number.

14.148.3.16 GetStandardNameSpace()

```
virtual GenICam::gcstring GetStandardNameSpace ( ) [virtual]
```

Get the standard name space.

14.148.3.17 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 PreprocessXMLFromZI↩PFile(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

14.148.3.18 GetToolTip()

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

Get tool tip.

14.148.3.19 GetVendorName()

```
virtual GenICam::gcstring GetVendorName ( ) [virtual]
```

Get the vendor name.

14.148.3.20 GetVersionGuid()

```
virtual GenICam::gcstring GetVersionGuid ( )  [virtual]
```

Get the GUID describing the product version.

14.148.3.21 InvalidateNodes()

```
virtual void InvalidateNodes ( ) const  [virtual]
```

Invalidates all nodes.

14.148.3.22 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);

14.148.3.23 LoadXMLFromFileInject()

```
void LoadXMLFromFileInject (
            GenICam::gcstring TargetFileName,
            GenICam::gcstring InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

14.148.3.24 LoadXMLFromString()

```
void LoadXMLFromString (
            const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

14.148.3.25 LoadXMLFromStringInject()

```
void LoadXMLFromStringInject (
            const GenICam::gcstring & TargetXMLDataconst,
            const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

14.148.3.26 LoadXMLFromZIPData()

```
void LoadXMLFromZIPData (
            const void * zipData,
            size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

14.148.3.27 LoadXMLFromZIPFile()

```
void LoadXMLFromZIPFile (
            GenICam::gcstring ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

14.148.3.28 Poll()

```
virtual void Poll (
            int64_t ElapsedTime )  [virtual]
```

Fires nodes which have a polling time.

14.148.4 Member Data Documentation

14.148.4.1 _Ptr

```
INodeMap* _Ptr
```

Pointer to the NodeMap.

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

- include/SpinGenApi/NodeMap.h

14.149 CNodeMapFactory::NodeStatistics_t Struct Reference

Public Attributes

- uint32_t NumNodes
- uint32_t NumProperties
- uint32_t NumLinks
- uint32_t NumStrings

14.149.1 Member Data Documentation

14.149.1.1 NumLinks

```
uint32_t NumLinks
```

14.149.1.2 NumNodes

```
uint32_t NumNodes
```

14.149.1.3 NumProperties

```
uint32_t NumProperties
```

14.149.1.4 NumStrings

```
uint32_t NumStrings
```

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

- include/SpinGenApi/NodeMapFactory.h

14.150 ODevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBase< CharType, Traits >:

Collaboration diagram for ODevFileStreamBase< CharType, Traits >:

Public Types

- typedef ODevFileStreamBuf< CharType, Traits > filebuf_type
- typedef std::basic_ios< CharType, Traits > ios_type
- typedef std::basic_ostream< CharType, Traits > ostream_type

Public Member Functions

- filebuf_type * rdbuf () const
- bool is_open () const
- void open (INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base↩ ::out|std::ios_base::trunc)

  Open file on device in write mode.
- void close ()

  Close the file on device.

14.150.1 Member Typedef Documentation

14.150.1.1 filebuf_type

```
typedef ODevFileStreamBuf<CharType, Traits> filebuf_type
```

14.150.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

14.150.1.3 ostream_type

```
typedef std::basic_ostream<CharType, Traits> ostream_type
```

14.150.2 Member Function Documentation

14.150.2.1 close()

```
void close ( ) [inline]
```

Close the file on device.

14.150.2.2 is_open()

```
bool is_open ( ) const [inline]
```

14.150.2.3 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

pInterface	NodeMap of the device to which the FileProtocolAdapter is attached
pFileName	Name of the file to open
mode	open mode

14.150.2.4 rdbuf()

```
filebuf_type* rdbuf ( ) const [inline]
```

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

- include/SpinGenApi/Filestream.h

14.151 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 xsputn (const char_type *s, std::streamsize n)
- int_type overflow (int_type c=traits_type::eof())
- int sync ()

14.151.1 Constructor & Destructor Documentation

14.151.1.1 ODevFileStreamBuf()

```
ODevFileStreamBuf ( ) [inline]
```

14.151.1.2 ~ODevFileStreamBuf()

```
~ODevFileStreamBuf ( ) [inline]
```

14.151.2 Member Function Documentation

14.151.2.1 close()

```
filebuf_type* close ( ) [inline]
```

14.151.2.2 is_open()

```
bool is_open ( ) const [inline]
```

14.151.2.3 open()

```
filebuf_type* open (
            Spinnaker::GenApi::INodeMap * pInterface,
            const char * pFileName,
            std::ios_base::openmode mode )  [inline]
```

14.151.2.4 overflow()

```
int_type overflow (
            int_type c = traits_type::eof() )  [inline], [protected]
```

14.151.2.5 sync()

```
int sync ( )  [inline], [protected]
```

14.151.2.6 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

14.152 PGMOption Struct Reference

Options for saving PGM images.

Public Member Functions

- PGMOption ()

Public Attributes

- bool binaryFile

  Whether to save the PPM as a binary file.
- unsigned int reserved [16]

  Reserved for future use.

14.152.1 Detailed Description

Options for saving PGM images.

14.152.2 Constructor & Destructor Documentation

14.152.2.1 PGMOption()

```
PGMOption ( ) [inline]
```

14.152.3 Member Data Documentation

14.152.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

14.152.3.2 reserved

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

Reserved for future use.

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

- include/SpinnakerDefs.h

14.153 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.

14.153.1 Detailed Description

Options for saving PNG images.

14.153.2 Constructor & Destructor Documentation

14.153.2.1 PNGOption()

```
PNGOption ( ) [inline]
```

14.153.3 Member Data Documentation

14.153.3.1 compressionLevel

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

14.153.3.2 interlaced

```
bool interlaced
```

Whether to save the PNG as interlaced.

14.153.3.3 reserved

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

Reserved for future use.

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

- include/SpinnakerDefs.h

14.154 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

14.154.1 Detailed Description

Interface for value properties.

14.154.2 Constructor & Destructor Documentation

14.154.2.1 PortNode() [1/2]

```
PortNode ( )
```

Constructor.

14.154.2.2 PortNode() [2/2]

```
PortNode (
            std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with GenICam IValue

14.154.2.3 ~PortNode()

```
~PortNode ( )
```

Destructor.

14.154.3 Member Function Documentation

14.154.3.1 CacheChunkData()

```
virtual EYesNo CacheChunkData ( ) const  [virtual]
```

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

14.154.3.2 GetChunkID()

```
virtual Spinnaker::GenICam::gcstring GetChunkID ( ) const  [virtual]
```

Get the Id of the chunk the port should be attached to.

14.154.3.3 GetPortHandle()

```
std::shared_ptr<Node::NodeImpl> GetPortHandle ( )  [inline]
```

14.154.3.4 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( )  [virtual]
```

Determines if the port adapter must perform an endianess swap.

14.154.3.5 Read()

```
virtual void Read (
            void * pBuffer,
            int64_t Address,
            int64_t Length )  [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented in PortRecorder, and PortReplay.

14.154.3.6 Replay()

```
virtual void Replay (
            IPortWriteList * pPortRecorder,
            bool Invalidate = true )  [virtual]
```

Sends the commands to the camera.

The default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented in PortRecorder, and PortReplay.

14.154.3.7 SetPortImpl()

```
void SetPortImpl (
            IPort * pPort )
```

Sets pointer the real port implementation; this function may called only once.

14.154.3.8 SetReference() [1/3]

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

overload SetReference for Value

Reimplemented from Node.

14.154.3.9 SetReference() [2/3]

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

overload SetReference for Value

Reimplemented in PortRecorder, and PortReplay.

14.154.3.10 SetReference() [3/3]

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

overload SetReference for Value

14.154.3.11 StartRecording()

```
virtual void StartRecording (
            IPortWriteList * pPortRecorder )  [virtual]
```

Starts logging all WriteRegister commands to a list.

Reimplemented in PortRecorder.

14.154.3.12 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

Stops recording.

Reimplemented in PortRecorder.

14.154.3.13 Write()

```
virtual void Write (
            const void * pBuffer,
            int64_t Address,
            int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented in PortRecorder, and PortReplay.

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

- include/SpinGenApi/PortNode.h

14.155 PortRecorder Class Reference

Interface for recording write commands on a port.

Inheritance diagram for PortRecorder:

Collaboration diagram for PortRecorder:

Public Member Functions

- PortRecorder ()
- virtual ~PortRecorder ()
- virtual void StartRecording (IPortWriteList *pPortRecorder)

  starts logging all WriteRegister commands to a list
- virtual void StopRecording ()

  stops recording
- virtual EAccessMode GetAccessMode () const

  Get the access mode of the node.
- virtual void SetReference (IPort *pBase)

  overload SetReference for Value
- virtual void Replay (IPortWriteList *pPortRecorder, bool Invalidate=true)

  sends the commands to the camera.
- virtual void Read (void *pBuffer, int64_t Address, int64_t Length)

  Reads a chunk of bytes from the port.
- virtual void Write (const void *pBuffer, int64_t Address, int64_t Length)

  Writes a chunk of bytes to the port.

Additional Inherited Members

14.155.1 Detailed Description

Interface for recording write commands on a port.

14.155.2 Constructor & Destructor Documentation

14.155.2.1 PortRecorder()

```
PortRecorder ( )
```

14.155.2.2 ~PortRecorder()

```
virtual ~PortRecorder ( )  [virtual]
```

14.155.3 Member Function Documentation

14.155.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const  [virtual]
```

Get the access mode of the node.

Reimplemented from PortReplay.

14.155.3.2 Read()

```
virtual void Read (
            void * pBuffer,
            int64_t Address,
            int64_t Length )  [inline], [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented from PortReplay.

14.155.3.3 Replay()

```
virtual void Replay (
            IPortWriteList * pPortRecorder,
            bool Invalidate = true )  [inline], [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented from PortReplay.

14.155.3.4 SetReference()

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

overload SetReference for Value

Reimplemented from PortReplay.

14.155.3.5 StartRecording()

```
virtual void StartRecording (
            IPortWriteList * pPortRecorder )  [virtual]
```

starts logging all WriteRegister commands to a list

Reimplemented from PortNode.

14.155.3.6 StopRecording()

```
virtual void StopRecording ( )  [virtual]
```

stops recording

Reimplemented from PortNode.

14.155.3.7 Write()

```
virtual void Write (
            const void * pBuffer,
            int64_t Address,
            int64_t Length )  [inline], [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented from PortReplay.

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

- include/SpinGenApi/PortRecorder.h

14.156 PortReplay Class Reference

Interface for replaying write commands on a port.

Inheritance diagram for PortReplay:

Collaboration diagram for PortReplay:

Public Member Functions

- PortReplay ()
- virtual ~PortReplay ()
- virtual void Replay (IPortWriteList *pPortRecorder, bool Invalidate=true)

  sends the commands to the camera.
- virtual void SetReference (IPort *pBase)

  overload SetReference for Value
- void * GetPortReplayHandle ()
- virtual EAccessMode GetAccessMode () const

  Base interface overrides.
- virtual void Read (void *pBuffer, int64_t Address, int64_t Length)

  Reads a chunk of bytes from the port.
- virtual void Write (const void *pBuffer, int64_t Address, int64_t Length)

  Writes a chunk of bytes to the port.

Additional Inherited Members

14.156.1 Detailed Description

Interface for replaying write commands on a port.

14.156.2 Constructor & Destructor Documentation

14.156.2.1 PortReplay()

```
PortReplay ( )
```

14.156.2.2 ~PortReplay()

```
virtual ~PortReplay ( )  [virtual]
```

14.156.3 Member Function Documentation

14.156.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const  [inline], [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented from Node.

Reimplemented in PortRecorder.

14.156.3.2 GetPortReplayHandle()

```
void* GetPortReplayHandle ( )
```

14.156.3.3 Read()

```
virtual void Read (
            void * pBuffer,
            int64_t Address,
            int64_t Length )  [inline], [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented from PortNode.

Reimplemented in PortRecorder.

14.156.3.4 Replay()

```
virtual void Replay (
            IPortWriteList * pPortRecorder,
            bool Invalidate = true )  [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented from PortNode.

Reimplemented in PortRecorder.

14.156.3.5 SetReference()

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

overload SetReference for Value

Reimplemented from PortNode.

Reimplemented in PortRecorder.

14.156.3.6 Write()

```
virtual void Write (
            const void * pBuffer,
            int64_t Address,
            int64_t Length )  [inline], [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented from PortNode.

Reimplemented in PortRecorder.

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

- include/SpinGenApi/PortReplay.h

14.157 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.

14.157.1 Detailed Description

Options for saving PPM images.

14.157.2 Constructor & Destructor Documentation

14.157.2.1 PPMOption()

```
PPMOption ( ) [inline]
```

14.157.3 Member Data Documentation

14.157.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

14.157.3.2 reserved

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

Reserved for future use.

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

- include/SpinnakerDefs.h

14.158 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

14.158.1 Detailed Description

Interface for string properties.

14.158.2 Constructor & Destructor Documentation

14.158.2.1 RegisterNode() [1/2]

```
RegisterNode ( )
```

14.158.2.2 RegisterNode() [2/2]

```
RegisterNode (
            std::shared_ptr< Node::NodeImpl > pRegister )
```

14.158.2.3 ~RegisterNode()

```
virtual ~RegisterNode ( ) [virtual]
```

14.158.3 Member Function Documentation

14.158.3.1 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

pBuffer	The buffer receiving the data to read
Length	The number of bytes to retrieve
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

14.158.3.2 GetAddress()

```
virtual int64_t GetAddress ( ) [virtual]
```

Retrieves the Address of the register.

14.158.3.3 GetLength()

```
virtual int64_t GetLength ( ) [virtual]
```

Retrieves the Length of the register [Bytes].

14.158.3.4 Set()

```
virtual void Set (
            const uint8_t * pBuffer,
            int64_t Length,
            bool Verify = true ) [virtual]
```

Set the register's contents.

Parameters

pBuffer	The buffer containing the data to set
Length	The number of bytes in pBuffer
Verify	Enables AccessMode and Range verification (default = true)

14.158.3.5 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

14.159 SingleChunkData_t Struct Reference

Public Attributes

- uint64_t ChunkID
- ptrdiff_t ChunkOffset
- size_t ChunkLength

14.159.1 Member Data Documentation

14.159.1.1 ChunkID

```
uint64_t ChunkID
```

14.159.1.2 ChunkLength

```
size_t ChunkLength
```

14.159.1.3 ChunkOffset

```
ptrdiff_t ChunkOffset
```

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

- include/SpinGenApi/ChunkAdapterGeneric.h

14.160 SingleChunkDataStr_t Struct Reference

Public Attributes

- GenICam::gcstring ChunkID
- ptrdiff_t ChunkOffset
- size_t ChunkLength

14.160.1 Member Data Documentation

14.160.1.1 ChunkID

```
GenICam::gcstring ChunkID
```

14.160.1.2 ChunkLength

```
size_t ChunkLength
```

14.160.1.3 ChunkOffset

```
ptrdiff_t ChunkOffset
```

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

- include/SpinGenApi/ChunkAdapterGeneric.h

14.161 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

14.162 SpinVideo Class Reference

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

Public Member Functions

- SpinVideo ()

  Default constructor.
- virtual ~SpinVideo ()

  Default destructor.
- virtual void Open (const char *pFileName, AVIOption &pOption)

  Open an video file in preparation for writing Images to disk.
- virtual void Open (const char *pFileName, MJPGOption &pOption)

  Open an MJPEG video file in preparation for writing Images to disk.
- virtual void Open (const char *pFileName, H264Option &pOption)

  Open an H264 MP4 video file in preparation for writing Images to disk.
- virtual void Append (ImagePtr pImage)

  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.

14.162.1 Detailed Description

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

14.162.2 Constructor & Destructor Documentation

14.162.2.1 SpinVideo()

```
SpinVideo ( )
```

Default constructor.

14.162.2.2 ~SpinVideo()

```
virtual ~SpinVideo ( ) [virtual]
```

Default destructor.

14.162.3 Member Function Documentation

14.162.3.1 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

pImage	The image to append.

14.162.3.2 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)

14.162.3.3 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

pFileName	The filename of the video file.
pOption	Options to apply to the video file.

See also

Close()

14.162.3.4 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

pFileName	The filename of the video file.
pOption	MJPEG options to apply to the video file.

See also

Close()
MJPGOption

14.162.3.5 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

pFileName	The filename of the MP4 video file.
pOption	H264 options to apply to the MP4 video file.

See also

Close()
H264Option

14.162.3.6 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

size	The maximum video file size in MB.

See also

Append(ImagePtr pImage)

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

- include/SpinVideo.h

14.163 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

14.163.1 Detailed Description

Interface for string properties.

14.163.2 Constructor & Destructor Documentation

14.163.2.1 StringNode() [1/2]

```
StringNode ( )
```

14.163.2.2 StringNode() [2/2]

```
StringNode (
            std::shared_ptr< Node::NodeImpl > pString )
```

14.163.2.3 ~StringNode()

```
virtual ~StringNode ( ) [virtual]
```

14.163.3 Member Function Documentation

14.163.3.1 GetMaxLength()

```
virtual int64_t GetMaxLength ( ) [virtual]
```

Retrieves the maximum length of the string in bytes.

14.163.3.2 GetValue()

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

Get 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

14.163.3.3 operator()()

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

Get node value.

14.163.3.4 operator*()

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

Get node value.

14.163.3.5 operator=()

```
virtual IString& operator= (
            const GenICam::gcstring & Value )  [virtual]
```

Set node value.

14.163.3.6 SetReference()

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

overload SetReference for Value

Reimplemented from ValueNode.

Reimplemented in StringRegNode.

14.163.3.7 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

14.164 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

14.164.1 Detailed Description

Interface for string properties.

14.164.2 Constructor & Destructor Documentation

14.164.2.1 StringRegNode() [1/2]

```
StringRegNode ( )
```

14.164.2.2 StringRegNode() [2/2]

```
StringRegNode (
            std::shared_ptr< Node::NodeImpl > pString )
```

14.164.2.3 ~StringRegNode()

```
virtual ~StringRegNode ( ) [virtual]
```

14.164.3 Member Function Documentation

14.164.3.1 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

14.165 System Class Reference

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

Inheritance diagram for System:

Collaboration diagram for System:

Public Member Functions

- virtual ~System ()

  Default destructor.
- virtual void ReleaseInstance ()

  This call releases the instance of the System Singleton for this process.
- virtual InterfaceList GetInterfaces (bool updateInterface=true)

  Returns a list of interfaces available on the system.
- virtual void UpdateInterfaceList ()

  Updates the list of interfaces on the system.

- virtual CameraList GetCameras (bool updateInterfaces=true, bool updateCameras=true)

  Returns a list of cameras that are available on the system.
- virtual bool UpdateCameras (bool updateInterfaces=true)

  Updates the list of cameras on the system.
- void RegisterEventHandler (EventHandler &evtHandlerToRegister)

  Registers an event handler for the system.
- void UnregisterEventHandler (EventHandler &evtHandlerToUnregister)

  Unregisters an event handler for the system.
- virtual void RegisterInterfaceEventHandler (EventHandler &evtHandlerToRegister, bool updateInterface=true)

  Registers event handlers for all available interfaces that are found on the system If new interfaces are detected by the system after RegisterInterfaceEventHandler() is called, those interfaces will be automatically registered with this event.
- void UnregisterInterfaceEventHandler (EventHandler &evtHandlerToUnregister)

  Unregisters event handlers for all available interfaces that are found on the system.
- virtual void RegisterLoggingEventHandler (LoggingEventHandler &handler)

  Registers a logging event.
- virtual void UnregisterAllLoggingEventHandlers ()

  Unregisters all previously registered logging events.
- virtual void UnregisterLoggingEventHandler (LoggingEventHandler &handler)

  Unregisters a logging event.
- virtual void SetLoggingEventPriorityLevel (SpinnakerLogLevel level)

  Sets a threshold priority level for logging event.
- virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ()

  Retrieves the current logging event priority level.
- virtual bool IsInUse ()

  Checks if the system is in use by any interface or camera objects.
- virtual void SendActionCommand (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[ ]=NULL)

  Broadcast an Action Command to all devices on system.
- virtual const LibraryVersion GetLibraryVersion ()

  Get current library version of Spinnaker.
- virtual GenApi::INodeMap & GetTLNodeMap () const

  Gets a reference to the system node map.

Static Public Member Functions

- static SystemPtr GetInstance ()

  Returns a pointer to a Singleton instance of a System object.

Protected Member Functions

- System ()

  Default constructor.

Additional Inherited Members

14.165.1 Detailed Description

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

14.165.2 Constructor & Destructor Documentation

14.165.2.1 ~System()

```
virtual ~System ( ) [virtual]
```

Default destructor.

14.165.2.2 System()

```
System ( ) [protected]
```

Default constructor.

14.165.3 Member Function Documentation

14.165.3.1 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

updateInterfaces	Determines whether or not updateInterfaceList() is called before getting cameras from available interfaces on the system
updateCameras	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.

14.165.3.2 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.

14.165.3.3 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

updateInterface	Determines whether or not UpdateInterfaceList() is called before getting available interfaces

Returns

An InterfaceList object that contains a list of all interfaces.

Implements ISystem.

14.165.3.4 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.

14.165.3.5 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.

14.165.3.6 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.

14.165.3.7 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.

14.165.3.8 RegisterEventHandler()

```
void RegisterEventHandler (
            EventHandler & evtHandlerToRegister )  [virtual]
```

Registers an event handler for the system.

See also

SystemEventHandler

Parameters

evtHandlerToRegister	The event handler to register for the system

Implements ISystem.

14.165.3.9 RegisterInterfaceEventHandler()

```
virtual void RegisterInterfaceEventHandler (
            EventHandler & evtHandlerToRegister,
            bool updateInterface = true )  [virtual]
```

Registers event handlers for all available interfaces that are found on the system If new interfaces are detected by the system after RegisterInterfaceEventHandler() is called, those interfaces will be automatically registered with this event.

Note that only GEV interface arrivals and removals are currently handled.

See also

InterfaceEventHandler

Parameters

evtHandlerToRegister	The event handler to register for the available interfaces
updateInterface	Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system

Implements ISystem.

14.165.3.10 RegisterLoggingEventHandler()

```
virtual void RegisterLoggingEventHandler (
            LoggingEventHandler & handler )  [virtual]
```

Registers a logging event.

Parameters

handler	The logging event handler to register

Implements ISystem.

14.165.3.11 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.

14.165.3.12 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

deviceKey	The Action Command's device key
groupKey	The Action Command's group key
groupMask	The Action Command's group mask
actionTime	(Optional) Time when to assert a future action. Zero means immediate action.
pResultSize	(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.
results	(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.

14.165.3.13 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

level	The threshold level

Implements ISystem.

14.165.3.14 UnregisterAllLoggingEventHandlers()

```
virtual void UnregisterAllLoggingEventHandlers ( ) [virtual]
```

Unregisters all previously registered logging events.

Implements ISystem.

14.165.3.15 UnregisterEventHandler()

```
void UnregisterEventHandler (
            EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters an event handler for the system.

See also

SystemEventHandler

Parameters

evtHandlerToUnregister	The event handler to unregister from the system

Implements ISystem.

14.165.3.16 UnregisterInterfaceEventHandler()

```
void UnregisterInterfaceEventHandler (
            EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters event handlers for all available interfaces that are found on the system.

See also

InterfaceEventHandler

Parameters

evtHandlerToUnregister	The event handler to unregister from the available interfaces

Implements ISystem.

14.165.3.17 UnregisterLoggingEventHandler()

```
virtual void UnregisterLoggingEventHandler (
            LoggingEventHandler & handler )  [virtual]
```

Unregisters a logging event.

Parameters

handler	The logging event handler to unregister

Implements ISystem.

14.165.3.18 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

updateInterfaces	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.

14.165.3.19 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

14.166 SystemEventHandler Class Reference

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

Inheritance diagram for SystemEventHandler:

Collaboration diagram for SystemEventHandler:

Public Member Functions

- SystemEventHandler ()

  Default constructor.
- virtual ~SystemEventHandler ()

  Virtual destructor.
- virtual void OnInterfaceArrival (std::string interfaceID)=0

  Interface arrival event callback.
- virtual void OnInterfaceRemoval (std::string interfaceID)=0

  Interface removal event callback.

Protected Member Functions

- SystemEventHandler & operator= (const SystemEventHandler &)

  Assignment operator.

Additional Inherited Members

14.166.1 Detailed Description

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

Note that only GEV interface arrivals and removals are currently handled.

14.166.2 Constructor & Destructor Documentation

14.166.2.1 SystemEventHandler()

```
SystemEventHandler ( )
```

Default constructor.

14.166.2.2 ~SystemEventHandler()

```
virtual ~SystemEventHandler ( )  [virtual]
```

Virtual destructor.

14.166.3 Member Function Documentation

14.166.3.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
            std::string interfaceID )  [pure virtual]
```

Interface arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

interfaceID	The ID of the arrived interface

Implements ISystemEventHandler.

Implemented in SystemEventHandlerImpl.

14.166.3.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
            std::string interfaceID )  [pure virtual]
```

Interface removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

interfaceID	The ID of the removed interface

Implements ISystemEventHandler.

Implemented in SystemEventHandlerImpl.

14.166.3.3 operator=()

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

Assignment operator.

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

- include/SystemEventHandler.h

14.167 SystemEventHandlerImpl Class Reference

Inheritance diagram for SystemEventHandlerImpl:

Collaboration diagram for SystemEventHandlerImpl:

Public Member Functions

- SystemEventHandlerImpl (SystemPtr system)
- ~SystemEventHandlerImpl ()
- void LockEventHandlerMutex ()
- void UnlockEventHandlerMutex ()
- void OnInterfaceArrival (std::string interfaceID)

  Interface arrival event callback.
- void OnInterfaceRemoval (std::string interfaceID)

  Interface removal event callback.
- void RegisterInterfaceEventToSystem ()
- void UnregisterInterfaceEventFromSystem ()
- void RegisterAllInterfaceEvents ()
- void UnregisterAllInterfaceEvents ()

Additional Inherited Members

14.167.1 Constructor & Destructor Documentation

14.167.1.1 SystemEventHandlerImpl()

```
SystemEventHandlerImpl (
            SystemPtr system ) [inline]
```

14.167.1.2 ~SystemEventHandlerImpl()

```
~SystemEventHandlerImpl ( ) [inline]
```

14.167.2 Member Function Documentation

14.167.2.1 LockEventHandlerMutex()

```
void LockEventHandlerMutex ( ) [inline]
```

14.167.2.2 OnInterfaceArrival()

```
void OnInterfaceArrival (
            std::string interfaceID ) [inline], [virtual]
```

Interface arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

interfaceID	The ID of the arrived interface

Implements SystemEventHandler.

14.167.2.3 OnInterfaceRemoval()

```
void OnInterfaceRemoval (
            std::string interfaceID ) [inline], [virtual]
```

Interface removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

interfaceID	The ID of the removed interface

Implements SystemEventHandler.

14.167.2.4 RegisterAllInterfaceEvents()

```
void RegisterAllInterfaceEvents ( ) [inline]
```

14.167.2.5 RegisterInterfaceEventToSystem()

```
void RegisterInterfaceEventToSystem ( ) [inline]
```

14.167.2.6 UnlockEventHandlerMutex()

```
void UnlockEventHandlerMutex ( ) [inline]
```

14.167.2.7 UnregisterAllInterfaceEvents()

```
void UnregisterAllInterfaceEvents ( ) [inline]
```

14.167.2.8 UnregisterInterfaceEventFromSystem()

```
void UnregisterInterfaceEventFromSystem ( ) [inline]
```

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

- src/EnumerationEvents/EnumerationEvents.cpp

14.168 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

14.168.1 Detailed Description

A reference tracked pointer to a system object.

14.168.2 Constructor & Destructor Documentation

14.168.2.1 SystemPtr() [1/4]

```
SystemPtr ( ) [inline]
```

Default constructor.

14.168.2.2 SystemPtr() [2/4]

```
SystemPtr (
            const int ) [inline]
```

Copy constructor.

14.168.2.3 SystemPtr() [3/4]

```
SystemPtr (
            const long ) [inline]
```

Copy constructor.

14.168.2.4 SystemPtr() [4/4]

```
SystemPtr (
            const std::nullptr_t ) [inline]
```

Copy constructor.

14.168.2.5 ~SystemPtr()

```
virtual ~SystemPtr (
            void  )  [inline], [virtual]
```

Virtual destructor.

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

- include/SystemPtr.h

14.169 TIFFOption Struct Reference

Options for saving TIFF images.

Public Types

- enum CompressionMethod {
  NONE = 1,
  PACKBITS,
  DEFLATE,
  ADOBE_DEFLATE,
  CCITTFAX3,
  CCITTFAX4,
  LZW,
  JPEG }

Public Member Functions

- TIFFOption ()

Public Attributes

- CompressionMethod compression

  Compression method to use for encoding TIFF images.
- unsigned int reserved [16]

  Reserved for future use.

14.169.1 Detailed Description

Options for saving TIFF images.

14.169.2 Member Enumeration Documentation

14.169.2.1 CompressionMethod

```
enum CompressionMethod
```

Enumerator

NONE	Save without any compression.
PACKBITS	Save using PACKBITS compression.
DEFLATE	Save using DEFLATE compression (ZLIB compression).
ADOBE_DEFLATE	Save using ADOBE DEFLATE compression.
CCITTFAX3	Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
CCITTFAX4	Save using CCITT Group 4 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
LZW	Save using LZW compression.
JPEG	Save using JPEG compression. This is only valid for 8-bit greyscale and 24-bit only. Default to LZW for other bit depths.

14.169.3 Constructor & Destructor Documentation

14.169.3.1 TIFFOption()

```
TIFFOption ( ) [inline]
```

14.169.4 Member Data Documentation

14.169.4.1 compression

```
CompressionMethod compression
```

Compression method to use for encoding TIFF images.

14.169.4.2 reserved

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

Reserved for future use.

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

- include/SpinnakerDefs.h

14.170 TransportLayerDevice Class Reference

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

Public Member Functions

- TransportLayerDevice (GenApi::INodeMap *nodeMapTLDevice)
- ~TransportLayerDevice ()

Public Attributes

- GenApi::IString & DeviceID

  Description: Interface-wide unique identifier of this device.
- GenApi::IString & DeviceSerialNumber

  Description: Serial number of the remote device.
- GenApi::IString & DeviceVendorName

  Description: Name of the remote device vendor.
- GenApi::IString & DeviceModelName

  Description: Name of the remote device model.
- GenApi::IEnumerationT< DeviceTypeEnum > & DeviceType

  Description: Transport layer type of the device.
- GenApi::IString & DeviceDisplayName

  Description: User readable name of the device.
- GenApi::IEnumerationT< DeviceAccessStatusEnum > & DeviceAccessStatus

  Description: Gets the access status the transport layer Producer has on the device.
- GenApi::IString & DeviceVersion

  Description: Version of the device.
- GenApi::IString & DeviceUserID

  Description: User Defined Name.
- GenApi::IString & DeviceDriverVersion

  Description: Version of the device driver.
- GenApi::IBoolean & DeviceIsUpdater

  Description: Indicates whether the device is in updater mode.
- GenApi::IEnumerationT< GevCCPEnum > & GevCCP

  Description: Controls the device access privilege of an application.
- GenApi::IEnumerationT< GUIXMLLocationEnum > & GUIXMLLocation

  Description: Sets the location to load GUI XML.
- GenApi::IString & GUIXMLPath

  Description: GUI XML Path.
- GenApi::IEnumerationT< GenICamXMLLocationEnum > & GenICamXMLLocation

  Description: Sets the location to load GenICam XML.
- GenApi::IString & GenICamXMLPath

  Description: GenICam XML Path.
- GenApi::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 & GevDeviceModeIsBigEndian

  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 & GevDeviceAutoForceIP

  Description: Forces the camera to be on the same subnet as its corresponding interface.
- GenApi::ICommand & GevDeviceForceIP

  Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDevice↩ ForceGateway) 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 CameraInternal

14.170.1 Detailed Description

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

14.170.2 Constructor & Destructor Documentation

14.170.2.1 TransportLayerDevice() [1/2]

```
TransportLayerDevice (
            GenApi::INodeMap * nodeMapTLDevice )
```

14.170.2.2 ~TransportLayerDevice()

```
~TransportLayerDevice ( )
```

14.170.2.3 TransportLayerDevice() [2/2]

```
TransportLayerDevice ( )  [protected]
```

14.170.3 Friends And Related Function Documentation

14.170.3.1 CameraBase

```
friend class CameraBase  [friend]
```

14.170.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

14.170.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

14.170.4 Member Data Documentation

14.170.4.1 DeviceAccessStatus

```
GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus
```

Description: Gets the access status the transport layer Producer has on the device.

Visibility: Beginner

14.170.4.2 DeviceCurrentSpeed

```
GenApi::IEnumerationT<DeviceCurrentSpeedEnum>& DeviceCurrentSpeed
```

Description: The USB Speed that the device is currently operating at.

Visibility: Expert

14.170.4.3 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

14.170.4.4 DeviceDriverVersion

```
GenApi::IString& DeviceDriverVersion
```

Description: Version of the device driver.

Visibility: Expert

14.170.4.5 DeviceEndianessMechanism

```
GenApi::IEnumerationT<DeviceEndianessMechanismEnum>& DeviceEndianessMechanism
```

Description: Identifies the endianness handling mode.

Visibility: Expert

14.170.4.6 DeviceID

```
GenApi::IString& DeviceID
```

Description: Interface-wide unique identifier of this device.

Visibility: Expert

14.170.4.7 DeviceInstanceId

```
GenApi::IString& DeviceInstanceId
```

Description: Visibility: Invisible.

14.170.4.8 DeviceIsUpdater

```
GenApi::IBoolean& DeviceIsUpdater
```

Description: Indicates whether the device is in updater mode.

Visibility: Expert

14.170.4.9 DeviceLinkSpeed

```
GenApi::IInteger& DeviceLinkSpeed
```

Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.

Visibility: Expert

14.170.4.10 DeviceLocation

```
GenApi::IString& DeviceLocation
```

Description: Device Location.

Visibility: Expert

14.170.4.11 DeviceModelName

```
GenApi::IString& DeviceModelName
```

Description: Name of the remote device model.

Visibility: Beginner

14.170.4.12 DeviceMulticastMonitorMode

```
GenApi::IBoolean& DeviceMulticastMonitorMode
```

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

Visibility: Expert

14.170.4.13 DeviceSerialNumber

```
GenApi::IString& DeviceSerialNumber
```

Description: Serial number of the remote device.

Visibility: Expert

14.170.4.14 DeviceType

```
GenApi::IEnumerationT<DeviceTypeEnum>& DeviceType
```

Description: Transport layer type of the device.

Visibility: Expert

14.170.4.15 DeviceU3VProtocol

```
GenApi::IBoolean& DeviceU3VProtocol
```

Description: Indicates whether the device is communicating in U3V Protocol.

Visibility: Expert

14.170.4.16 DeviceUserID

```
GenApi::IString& DeviceUserID
```

Description: User Defined Name.

Visibility: Expert

14.170.4.17 DeviceVendorName

```
GenApi::IString& DeviceVendorName
```

Description: Name of the remote device vendor.

Visibility: Beginner

14.170.4.18 DeviceVersion

```
GenApi::IString& DeviceVersion
```

Description: Version of the device.

Visibility: Beginner

14.170.4.19 GenICamXMLLocation

```
GenApi::IEnumerationT<GenICamXMLLocationEnum>& GenICamXMLLocation
```

Description: Sets the location to load GenICam XML.

Visibility: Beginner

14.170.4.20 GenICamXMLPath

```
GenApi::IString& GenICamXMLPath
```

Description: GenICam XML Path.

Visibility: Beginner

14.170.4.21 GevCCP

```
GenApi::IEnumerationT<GevCCPEnum>& GevCCP
```

Description: Controls the device access privilege of an application.

Visibility: Beginner

14.170.4.22 GevDeviceAutoForceIP

```
GenApi::ICommand& GevDeviceAutoForceIP
```

Description: Forces the camera to be on the same subnet as its corresponding interface.

Visibility: Expert

14.170.4.23 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

14.170.4.24 GevDeviceForceGateway

```
GenApi::IInteger& GevDeviceForceGateway
```

Description: Static gateway IP address to set for the GVCP interface of the remote device.

Visibility: Expert

14.170.4.25 GevDeviceForceIP

```
GenApi::ICommand& GevDeviceForceIP
```

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and Gev↩ DeviceForceGateway) in the Device using ForceIP command.

Visibility: Expert

14.170.4.26 GevDeviceForceIPAddress

```
GenApi::IInteger& GevDeviceForceIPAddress
```

Description: Static IP address to set for the GVCP interface of the remote device.

Visibility: Expert

14.170.4.27 GevDeviceForceSubnetMask

```
GenApi::IInteger& GevDeviceForceSubnetMask
```

Description: Static subnet mask to set for GVCP interface of the remote device.

Visibility: Expert

14.170.4.28 GevDeviceGateway

```
GenApi::IInteger& GevDeviceGateway
```

Description: Current gateway IP address of the GVCP interface of the remote device.

Visibility: Expert

14.170.4.29 GevDeviceIPAddress

```
GenApi::IInteger& GevDeviceIPAddress
```

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

14.170.4.30 GevDeviceIsWrongSubnet

```
GenApi::IBoolean& GevDeviceIsWrongSubnet
```

Description: Indicates whether the device is on the wrong subnet.

Visibility: Expert

14.170.4.31 GevDeviceMACAddress

```
GenApi::IInteger& GevDeviceMACAddress
```

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

14.170.4.32 GevDeviceMaximumPacketSize

```
GenApi::IInteger& GevDeviceMaximumPacketSize
```

Description: The maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

14.170.4.33 GevDeviceMaximumRetryCount

```
GenApi::IInteger& GevDeviceMaximumRetryCount
```

Description: Maximum number of times to retry a read/write operation.

Visibility: Expert

14.170.4.34 GevDeviceModeIsBigEndian

```
GenApi::IBoolean& GevDeviceModeIsBigEndian
```

Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).

Visibility: Expert

14.170.4.35 GevDevicePort

```
GenApi::IInteger& GevDevicePort
```

Description: Current IP port of the GVCP interface of the selected remote device.

Visibility: Expert

14.170.4.36 GevDeviceReadAndWriteTimeout

```
GenApi::IInteger& GevDeviceReadAndWriteTimeout
```

Description: The timeout in us for read/write operations to the camera.

Visibility: Expert

14.170.4.37 GevDeviceSubnetMask

```
GenApi::IInteger& GevDeviceSubnetMask
```

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

14.170.4.38 GevVersionMajor

```
GenApi::IInteger& GevVersionMajor
```

Description: Major version of the specification.

Visibility: Expert

14.170.4.39 GevVersionMinor

```
GenApi::IInteger& GevVersionMinor
```

Description: Minor version of the specification.

Visibility: Expert

14.170.4.40 GUIXMLLocation

```
GenApi::IEnumerationT<GUIXMLLocationEnum>& GUIXMLLocation
```

Description: Sets the location to load GUI XML.

Visibility: Beginner

14.170.4.41 GUIXMLPath

```
GenApi::IString& GUIXMLPath
```

Description: GUI XML Path.

Visibility: Beginner

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

- include/TransportLayerDevice.h

14.171 TransportLayerInterface Class Reference

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

Public Member Functions

- TransportLayerInterface (GenApi::INodeMap *nodeMapTLDevice)
- ~TransportLayerInterface ()

Public Attributes

- GenApi::IString & InterfaceID

  Description: Transport layer Producer wide unique identifier of the selected interface.
- GenApi::IString & InterfaceDisplayName

  Description: User readable name of the selected interface.
- GenApi::IEnumerationT< InterfaceTypeEnum > & InterfaceType

  Description: Transport layer type of the interface.
- GenApi::IInteger & GevInterfaceGatewaySelector

  Description: Selector for the different gateway entries for this 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 & GevInterfaceSubnetSelector

  Description: Selector for the subnet of this interface.
- GenApi::IInteger & GevInterfaceSubnetIPAddress

  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::IString & DeviceSerialNumber

  Description: Serial number of the selected remote device.
- 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 & GevDeviceGateway

  Description: Current gateway IP address 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 & 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::ICommand & GevDeviceForceIP

  Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDevice↩ ForceGateway) in the selected remote device using ForceIP command.
- GenApi::IInteger & GevDeviceForceIPAddress

  Description: Static IP address to set for the GVCP interface of the selected remote device.
- GenApi::IInteger & GevDeviceForceSubnetMask

  Description: Static subnet mask to set for GVCP interface of the selected remote device.
- GenApi::IInteger & GevDeviceForceGateway

  Description: Static gateway IP address to set for the GVCP interface of the selected remote device.
- GenApi::ICommand & GevDeviceAutoForceIP

  Description: Automatically forces the selected remote device to an IP Address on the same subnet as the GVCP interface.
- GenApi::IString & HostAdapterName

  Description: User readable name of the interface's host adapter.
- GenApi::IString & HostAdapterVendor

  Description: User readable name of the host adapter's vendor.
- GenApi::IString & HostAdapterDriverVersion

  Description: Driver version of the interface's host adapter.

Protected Member Functions

- TransportLayerInterface ()

Friends

- class Interface
- class IInterface
- class InterfaceInternal

14.171.1 Detailed Description

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

14.171.2 Constructor & Destructor Documentation

14.171.2.1 TransportLayerInterface() [1/2]

```
TransportLayerInterface (
            GenApi::INodeMap * nodeMapTLDevice )
```

14.171.2.2 ~TransportLayerInterface()

```
~TransportLayerInterface ( )
```

14.171.2.3 TransportLayerInterface() [2/2]

```
TransportLayerInterface ( ) [protected]
```

14.171.3 Friends And Related Function Documentation

14.171.3.1 IInterface

```
friend class IInterface [friend]
```

14.171.3.2 Interface

```
friend class Interface [friend]
```

14.171.3.3 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

14.171.4 Member Data Documentation

14.171.4.1 ActionCommand

```
GenApi::ICommand& ActionCommand
```

Description: Issues an Action Command to attached GEV devices on interface.

Visibility: Expert

14.171.4.2 DeviceAccessStatus

```
GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus
```

Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

14.171.4.3 DeviceCount

```
GenApi::IInteger& DeviceCount
```

Description: Number of compatible devices detected on current interface.

Visibility: Expert

14.171.4.4 DeviceID

```
GenApi::IString& DeviceID
```

Description: Interface wide unique identifier of the selected device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

14.171.4.5 DeviceModelName

```
GenApi::IString& DeviceModelName
```

Description: Name of the device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

14.171.4.6 DeviceSelector

```
GenApi::IInteger& DeviceSelector
```

Description: Selector for the different devices on this interface.

This value only changes on execution of "DeviceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Expert

14.171.4.7 DeviceSerialNumber

```
GenApi::IString& DeviceSerialNumber
```

Description: Serial number of the selected remote device.

Visibility: Expert

14.171.4.8 DeviceUnlock

```
GenApi::IString& DeviceUnlock
```

Description: Unlocks devices for internal use.

Visibility: Expert

14.171.4.9 DeviceUpdateList

```
GenApi::ICommand& DeviceUpdateList
```

Description: Updates the internal device list.

Visibility: Expert

14.171.4.10 DeviceVendorName

```
GenApi::IString& DeviceVendorName
```

Description: Name of the device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

14.171.4.11 FilterDriverStatus

```
GenApi::IEnumerationT<FilterDriverStatusEnum>& FilterDriverStatus
```

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

Visibility: Expert

14.171.4.12 GevActionDeviceKey

```
GenApi::IInteger& GevActionDeviceKey
```

Description: Key to authorize the action for the device.

Visibility: Expert

14.171.4.13 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

14.171.4.14 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

14.171.4.15 GevActionTime

```
GenApi::IInteger& GevActionTime
```

Description: Provides the time in nanoseconds when the action is to be executed.

Visibility: Expert

14.171.4.16 GevDeviceAutoForceIP

```
GenApi::ICommand& GevDeviceAutoForceIP
```

Description: Automatically forces the selected remote device to an IP Address on the same subnet as the GVCP interface.

Visibility: Expert

14.171.4.17 GevDeviceForceGateway

```
GenApi::IInteger& GevDeviceForceGateway
```

Description: Static gateway IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

14.171.4.18 GevDeviceForceIP

```
GenApi::ICommand& GevDeviceForceIP
```

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and Gev↩DeviceForceGateway) in the selected remote device using ForceIP command.

Visibility: Expert

14.171.4.19 GevDeviceForceIPAddress

```
GenApi::IInteger& GevDeviceForceIPAddress
```

Description: Static IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

14.171.4.20 GevDeviceForceSubnetMask

```
GenApi::IInteger& GevDeviceForceSubnetMask
```

Description: Static subnet mask to set for GVCP interface of the selected remote device.

Visibility: Expert

14.171.4.21 GevDeviceGateway

```
GenApi::IInteger& GevDeviceGateway
```

Description: Current gateway IP address of the GVCP interface of the selected remote device.

Visibility: Expert

14.171.4.22 GevDeviceIPAddress

```
GenApi::IInteger& GevDeviceIPAddress
```

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

14.171.4.23 GevDeviceMACAddress

```
GenApi::IInteger& GevDeviceMACAddress
```

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

14.171.4.24 GevDeviceSubnetMask

```
GenApi::IInteger& GevDeviceSubnetMask
```

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

14.171.4.25 GevInterfaceGateway

```
GenApi::IInteger& GevInterfaceGateway
```

Description: IP address of the selected gateway entry of this interface.

Visibility: Expert

14.171.4.26 GevInterfaceGatewaySelector

```
GenApi::IInteger& GevInterfaceGatewaySelector
```

Description: Selector for the different gateway entries for this interface.

Visibility: Expert

14.171.4.27 GevInterfaceMACAddress

```
GenApi::IInteger& GevInterfaceMACAddress
```

Description: 48-bit MAC address of this interface.

Visibility: Expert

14.171.4.28 GevInterfaceMTU

```
GenApi::IInteger& GevInterfaceMTU
```

Description: Maximum transmission unit of this interface.

Visibility: Expert

14.171.4.29 GevInterfaceReceiveLinkSpeed

```
GenApi::IInteger& GevInterfaceReceiveLinkSpeed
```

Description: Receive link speed of this interface in bits per second.

Visibility: Expert

14.171.4.30 GevInterfaceSubnetIPAddress

```
GenApi::IInteger& GevInterfaceSubnetIPAddress
```

Description: IP address of the selected subnet of this interface.

Visibility: Expert

14.171.4.31 GevInterfaceSubnetMask

```
GenApi::IInteger& GevInterfaceSubnetMask
```

Description: Subnet mask of the selected subnet of this interface.

Visibility: Expert

14.171.4.32 GevInterfaceSubnetSelector

```
GenApi::IInteger& GevInterfaceSubnetSelector
```

Description: Selector for the subnet of this interface.

Visibility: Expert

14.171.4.33 GevInterfaceTransmitLinkSpeed

```
GenApi::IInteger& GevInterfaceTransmitLinkSpeed
```

Description: Transmit link speed of this interface in bits per second.

Visibility: Expert

14.171.4.34 HostAdapterDriverVersion

```
GenApi::IString& HostAdapterDriverVersion
```

Description: Driver version of the interface's host adapter.

Visibility: Expert

14.171.4.35 HostAdapterName

```
GenApi::IString& HostAdapterName
```

Description: User readable name of the interface's host adapter.

Visibility: Expert

14.171.4.36 HostAdapterVendor

```
GenApi::IString& HostAdapterVendor
```

Description: User readable name of the host adapter's vendor.

Visibility: Expert

14.171.4.37 IncompatibleDeviceCount

```
GenApi::IInteger& IncompatibleDeviceCount
```

Description: Number of incompatible devices detected on current interface.

Visibility: Expert

14.171.4.38 IncompatibleDeviceID

```
GenApi::IString& IncompatibleDeviceID
```

Description: Interface wide unique identifier of the selected incompatible device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

14.171.4.39 IncompatibleDeviceModelName

```
GenApi::IString& IncompatibleDeviceModelName
```

Description: Name of the incompatible device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

14.171.4.40 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

14.171.4.41 IncompatibleDeviceVendorName

```
GenApi::IString& IncompatibleDeviceVendorName
```

Description: Name of the incompatible device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

14.171.4.42 IncompatibleGevDeviceIPAddress

```
GenApi::IInteger& IncompatibleGevDeviceIPAddress
```

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

14.171.4.43 IncompatibleGevDeviceMACAddress

```
GenApi::IInteger& IncompatibleGevDeviceMACAddress
```

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

14.171.4.44 IncompatibleGevDeviceSubnetMask

```
GenApi::IInteger& IncompatibleGevDeviceSubnetMask
```

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

14.171.4.45 InterfaceDisplayName

```
GenApi::IString& InterfaceDisplayName
```

Description: User readable name of the selected interface.

Visibility: Expert

14.171.4.46 InterfaceID

```
GenApi::IString& InterfaceID
```

Description: Transport layer Producer wide unique identifier of the selected interface.

Visibility: Expert

14.171.4.47 InterfaceType

```
GenApi::IEnumerationT<InterfaceTypeEnum>& InterfaceType
```

Description: Transport layer type of the interface.

Visibility: Expert

14.171.4.48 POEStatus

```
GenApi::IEnumerationT<POEStatusEnum>& POEStatus
```

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

Visibility: Expert

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

- include/TransportLayerInterface.h

14.172 TransportLayerStream Class Reference

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

Public Member Functions

- TransportLayerStream (GenApi::INodeMap *nodeMapTLDevice)
- ~TransportLayerStream ()

Public Attributes

- GenApi::IString & StreamID

  Description: Device unique ID for the data stream, e.g.
- GenApi::IEnumerationT< StreamTypeEnum > & StreamType

  Description: Stream type of the device.
- GenApi::IInteger & StreamBufferCountManual

  Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.
- GenApi::IInteger & StreamBufferCountResult

  Description: Displays the number of buffers to be used on this stream upon acquisition start.
- GenApi::IInteger & StreamBufferCountMax

  Description: Controls the maximum number of buffers that should be used on this stream.
- GenApi::IEnumerationT< StreamBufferCountModeEnum > & StreamBufferCountMode

  Description: Controls access to setting the number of buffers used for the stream.
- GenApi::IEnumerationT< StreamBufferHandlingModeEnum > & StreamBufferHandlingMode

  Description: Available buffer handling modes of this data stream: Visibility: Beginner.
- GenApi::IInteger & StreamAnnounceBufferMinimum

  Description: Minimal number of buffers to announce to enable selected buffer handling mode.
- GenApi::IInteger & StreamAnnouncedBufferCount

  Description: Number of announced (known) buffers on this stream.
- GenApi::IInteger & StreamStartedFrameCount

  Description: Number of frames started in the acquisition engine.
- GenApi::IInteger & StreamDeliveredFrameCount

  Description: Number of delivered frames since last acquisition start.
- GenApi::IInteger & StreamLostFrameCount

  Description: Number of lost frames due to queue underrun.
- GenApi::IInteger & StreamInputBufferCount

  Description: Number of buffers in the input buffer pool plus the buffers(s) currently being filled.
- GenApi::IInteger & StreamOutputBufferCount

  Description: Number of buffers in the output buffer queue.
- GenApi::IBoolean & StreamCRCCheckEnable

  Description: Enables or disables CRC checks on received images.
- GenApi::IBoolean & GevPacketResendMode

  Description: Enables or disables the packet resend mechanism.
- GenApi::IInteger & GevMaximumNumberResendRequests

  Description: Maximum number of resend requests per image.
- GenApi::IInteger & GevPacketResendTimeout

  Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.
- GenApi::IBoolean & StreamIsGrabbing

  Description: Flag indicating whether the acquisition engine is started or not.
- GenApi::IInteger & StreamChunkCountMaximum

  Description: Maximum number of chunks to be expected in a buffer.
- GenApi::IInteger & StreamBufferAlignment

  Description: Alignment size in bytes of the buffer passed to DSAnnounceBuffer.
- 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 images with missing packet.

- 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 CameraInternal

14.172.1 Detailed Description

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

14.172.2 Constructor & Destructor Documentation

14.172.2.1 TransportLayerStream() [1/2]

```
TransportLayerStream (
            GenApi::INodeMap * nodeMapTLDevice )
```

14.172.2.2 ~TransportLayerStream()

```
~TransportLayerStream ( )
```

14.172.2.3 TransportLayerStream() [2/2]

```
TransportLayerStream ( ) [protected]
```

14.172.3 Friends And Related Function Documentation

14.172.3.1 CameraBase

```
friend class CameraBase [friend]
```

14.172.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

14.172.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

14.172.4 Member Data Documentation

14.172.4.1 GevFailedPacketCount

```
GenApi::IInteger& GevFailedPacketCount
```

Description: Displays number of packets missed on this stream.

Successful resent packets are not counted as a missed packet. Visibility: Expert

14.172.4.2 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

14.172.4.3 GevPacketResendMode

```
GenApi::IBoolean& GevPacketResendMode
```

Description: Enables or disables the packet resend mechanism.

Visibility: Expert

14.172.4.4 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

14.172.4.5 GevResendPacketCount

```
GenApi::IInteger& GevResendPacketCount
```

Description: Displays number of packets received after retransmit request on this stream.

Visibility: Expert

14.172.4.6 GevResendRequestCount

```
GenApi::IInteger& GevResendRequestCount
```

Description: Displays number of packets requested to be retransmitted on this stream.

Visibility: Expert

14.172.4.7 GevTotalPacketCount

```
GenApi::IInteger& GevTotalPacketCount
```

Description: Displays number of packets received on this stream.

Visibility: Expert

14.172.4.8 StreamAnnounceBufferMinimum

```
GenApi::IInteger& StreamAnnounceBufferMinimum
```

Description: Minimal number of buffers to announce to enable selected buffer handling mode.

Visibility: Expert

14.172.4.9 StreamAnnouncedBufferCount

```
GenApi::IInteger& StreamAnnouncedBufferCount
```

Description: Number of announced (known) buffers on this stream.

This value is volatile. It may change if additional buffers are announced and/or buffers are revoked by the GenTL Consumer. Visibility: Expert

14.172.4.10 StreamBlockTransferSize

```
GenApi::IInteger& StreamBlockTransferSize
```

Description: Controls the image breakup size that should be used on this stream.

Visibility: Expert

14.172.4.11 StreamBufferAlignment

```
GenApi::IInteger& StreamBufferAlignment
```

Description: Alignment size in bytes of the buffer passed to DSAnnounceBuffer.

Visibility: Expert

14.172.4.12 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

14.172.4.13 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

14.172.4.14 StreamBufferCountMode

```
GenApi::IEnumerationT<StreamBufferCountModeEnum>& StreamBufferCountMode
```

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

Visibility: Expert

14.172.4.15 StreamBufferCountResult

```
GenApi::IInteger& StreamBufferCountResult
```

Description: Displays the number of buffers to be used on this stream upon acquisition start.

Recalculated on acquisition start if in auto mode. Visibility: Expert

14.172.4.16 StreamBufferHandlingMode

```
GenApi::IEnumerationT<StreamBufferHandlingModeEnum>& StreamBufferHandlingMode
```

Description: Available buffer handling modes of this data stream: Visibility: Beginner.

14.172.4.17 StreamChunkCountMaximum

```
GenApi::IInteger& StreamChunkCountMaximum
```

Description: Maximum number of chunks to be expected in a buffer.

Visibility: Expert

14.172.4.18 StreamCRCCheckEnable

```
GenApi::IBoolean& StreamCRCCheckEnable
```

Description: Enables or disables CRC checks on received images.

Visibility: Expert

14.172.4.19 StreamDeliveredFrameCount

```
GenApi::IInteger& StreamDeliveredFrameCount
```

Description: Number of delivered frames since last acquisition start.

It is not reset until the stream is closed. Visibility: Expert

14.172.4.20 StreamFailedBufferCount

```
GenApi::IInteger& StreamFailedBufferCount
```

Description: Displays number of images with missing packet.

Visibility: Expert

14.172.4.21 StreamID

```
GenApi::IString& StreamID
```

Description: Device unique ID for the data stream, e.g.

a GUID. Visibility: Expert

14.172.4.22 StreamInputBufferCount

```
GenApi::IInteger& StreamInputBufferCount
```

Description: Number of buffers in the input buffer pool plus the buffers(s) currently being filled.

Visibility: Expert

14.172.4.23 StreamIsGrabbing

```
GenApi::IBoolean& StreamIsGrabbing
```

Description: Flag indicating whether the acquisition engine is started or not.

Visibility: Expert

14.172.4.24 StreamLostFrameCount

```
GenApi::IInteger& StreamLostFrameCount
```

Description: Number of lost frames due to queue underrun.

This number is initialized with zero at the time the stream is opened and incremented every time the data could not be acquired because there was no buffer in the input buffer pool. It is not reset until the stream is closed. Visibility: Expert

14.172.4.25 StreamOutputBufferCount

```
GenApi::IInteger& StreamOutputBufferCount
```

Description: Number of buffers in the output buffer queue.

Visibility: Expert

14.172.4.26 StreamStartedFrameCount

```
GenApi::IInteger& StreamStartedFrameCount
```

Description: Number of frames started in the acquisition engine.

This number is incremented every time in case of a new buffer is started and then to be filled (data written to) regardless even if the buffer is later delivered to the user or discarded for any reason. This number is initialized with 0 at at the time of the stream is opened. It is not reset until the stream is closed. Visibility: Expert

14.172.4.27 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

14.173 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::IString & TLID

  Description: Unique identifier of the GenTL Producer like a GUID.
- GenApi::IString & TLVendorName

  Description: Name of the GenTL Producer vendor.
- GenApi::IString & TLModelName

  Description: Name of the GenTL Producer to distinguish different kinds of GenTL Producer implementations from one vendor.
- GenApi::IString & TLVersion

  Description: Vendor specific version string.
- GenApi::IString & TLFileName

  Description: Filename including extension of the GenTL Producer.
- GenApi::IString & TLDisplayName

  Description: User readable name of the GenTL Producer.
- GenApi::IString & TLPath

  Description: Full path to the GenTL Producer including filename and extension.
- GenApi::IEnumerationT< TLTypeEnum > & TLType

  Description: Transport layer type of the GenTL Producer implementation.
- GenApi::IInteger & GenTLVersionMajor

  Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.
- GenApi::IInteger & GenTLVersionMinor

  Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.
- GenApi::IInteger & GenTLSFNCVersionMajor

Description: Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

- GenApi::IInteger & GenTLSFNCVersionMinor

  Description: Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.
- GenApi::IInteger & GenTLSFNCVersionSubMinor

  Description: Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.
- GenApi::IInteger & GevVersionMajor

  Description: Major version number of the GigE Vision specification the GenTL Producer implementation complies to.
- GenApi::IInteger & GevVersionMinor

  Description: Minor version number of the GigE Vision specification the GenTL Producer implementation complies to.
- GenApi::ICommand & InterfaceUpdateList

  Description: Updates the internal list of the interfaces.
- GenApi::IInteger & InterfaceSelector

  Description: Selector for the different GenTL Producer interfaces.
- GenApi::IString & InterfaceID

  Description: GenTL Producer wide unique identifier of the selected interface.
- GenApi::IString & InterfaceDisplayName

  Description: A user-friendly name of the selected Interface.
- GenApi::IInteger & GevInterfaceMACAddress

  Description: 48-bit MAC address of the selected interface.
- GenApi::IInteger & GevInterfaceDefaultIPAddress

  Description: IP address of the first subnet of the selected interface.
- GenApi::IInteger & GevInterfaceDefaultSubnetMask

  Description: Subnet mask of the first subnet of the selected interface.
- GenApi::IInteger & GevInterfaceDefaultGateway

  Description: Gateway of the selected interface.

Protected Member Functions

- TransportLayerSystem ()

Friends

- class System
- class ISystem
- class SystemPtrInternal

14.173.1 Detailed Description

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

14.173.2 Constructor & Destructor Documentation

14.173.2.1 TransportLayerSystem() [1/2]

```
TransportLayerSystem (
            GenApi::INodeMap * nodeMapTLDevice )
```

14.173.2.2 ~TransportLayerSystem()

```
~TransportLayerSystem ( )
```

14.173.2.3 TransportLayerSystem() [2/2]

```
TransportLayerSystem ( ) [protected]
```

14.173.3 Friends And Related Function Documentation

14.173.3.1 ISystem

```
friend class ISystem [friend]
```

14.173.3.2 System

```
friend class System [friend]
```

14.173.3.3 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

14.173.4 Member Data Documentation

14.173.4.1 EnumerateGEVInterfaces

```
GenApi::IBoolean& EnumerateGEVInterfaces
```

Description: Enables or disables enumeration of GEV Interfaces.

Visibility: Expert

14.173.4.2 GenTLSFNCVersionMajor

```
GenApi::IInteger& GenTLSFNCVersionMajor
```

Description: Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer‘s XML.

Visibility: Expert

14.173.4.3 GenTLSFNCVersionMinor

```
GenApi::IInteger& GenTLSFNCVersionMinor
```

Description: Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer‘s XML.

Visibility: Expert

14.173.4.4 GenTLSFNCVersionSubMinor

```
GenApi::IInteger& GenTLSFNCVersionSubMinor
```

Description: Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer‘s XML.

Visibility: Expert

14.173.4.5 GenTLVersionMajor

```
GenApi::IInteger& GenTLVersionMajor
```

Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

14.173.4.6 GenTLVersionMinor

```
GenApi::IInteger& GenTLVersionMinor
```

Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

14.173.4.7 GevInterfaceDefaultGateway

```
GenApi::IInteger& GevInterfaceDefaultGateway
```

Description: Gateway of the selected interface.

Visibility: Expert

14.173.4.8 GevInterfaceDefaultIPAddress

```
GenApi::IInteger& GevInterfaceDefaultIPAddress
```

Description: IP address of the first subnet of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

14.173.4.9 GevInterfaceDefaultSubnetMask

```
GenApi::IInteger& GevInterfaceDefaultSubnetMask
```

Description: Subnet mask of the first subnet of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

14.173.4.10 GevInterfaceMACAddress

```
GenApi::IInteger& GevInterfaceMACAddress
```

Description: 48-bit MAC address of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

14.173.4.11 GevVersionMajor

```
GenApi::IInteger& GevVersionMajor
```

Description: Major version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

14.173.4.12 GevVersionMinor

```
GenApi::IInteger& GevVersionMinor
```

Description: Minor version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

14.173.4.13 InterfaceDisplayName

```
GenApi::IString& InterfaceDisplayName
```

Description: A user-friendly name of the selected Interface.

Visibility: Beginner

14.173.4.14 InterfaceID

```
GenApi::IString& InterfaceID
```

Description: GenTL Producer wide unique identifier of the selected interface.

Visibility: Beginner

14.173.4.15 InterfaceSelector

```
GenApi::IInteger& InterfaceSelector
```

Description: Selector for the different GenTL Producer interfaces.

This interface list only changes on execution of "InterfaceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Beginner

14.173.4.16 InterfaceUpdateList

```
GenApi::ICommand& InterfaceUpdateList
```

Description: Updates the internal list of the interfaces.

This feature is readable even if the execution cannot be performed immediately. The command then returns and the status can be polled. This function interacts with the TLUpdateInterfaceList function of the GenTL producer. It is up to the GenTL consumer to handle access in case both methods are used. Visibility: Beginner

14.173.4.17 TLDisplayName

```
GenApi::IString& TLDisplayName
```

Description: User readable name of the GenTL Producer.

Visibility: Expert

14.173.4.18 TLFileName

```
GenApi::IString& TLFileName
```

Description: Filename including extension of the GenTL Producer.

Visibility: Expert

14.173.4.19 TLID

```
GenApi::IString& TLID
```

Description: Unique identifier of the GenTL Producer like a GUID.

Visibility: Expert

14.173.4.20 TLModelName

```
GenApi::IString& TLModelName
```

Description: Name of the GenTL Producer to distinguish different kinds of GenTL Producer implementations from one vendor.

Visibility: Beginner

14.173.4.21 TLPath

```
GenApi::IString& TLPath
```

Description: Full path to the GenTL Producer including filename and extension.

Visibility: Expert

14.173.4.22 TLType

```
GenApi::IEnumerationT<TLTypeEnum>& TLType
```

Description: Transport layer type of the GenTL Producer implementation.

Visibility: Expert

14.173.4.23 TLVendorName

```
GenApi::IString& TLVendorName
```

Description: Name of the GenTL Producer vendor.

Visibility: Beginner

14.173.4.24 TLVersion

```
GenApi::IString& TLVersion
```

Description: Vendor specific version string.

Visibility: Beginner

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

- include/TransportLayerSystem.h

14.174 U3V_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t ChunkID
- uint32_t ChunkLength

14.174.1 Detailed Description

header of a GVCP request packet

14.174.2 Member Data Documentation

14.174.2.1 ChunkID

```
uint32_t ChunkID
```

14.174.2.2 ChunkLength

```
uint32_t ChunkLength
```

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

- include/SpinGenApi/ChunkAdapterU3V.h

14.175 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

14.175.1 Detailed Description

U3V/GenCP command header.

14.175.2 Member Data Documentation

14.175.2.1 CommandId

```
uint16_t CommandId
```

14.175.2.2 Flags

```
uint16_t Flags
```

14.175.2.3 Length

```
uint16_t Length
```

14.175.2.4 Prefix

```
uint32_t Prefix
```

14.175.2.5 ReqId

```
uint16_t ReqId
```

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

- include/SpinGenApi/EventAdapterU3V.h

14.176 U3V_EVENT_DATA Struct Reference

U3V/GenCP EVENT_CMD specific command data.

Public Attributes

- uint16_t Reserved
- uint16_t EventId
- uint64_t Timestamp

14.176.1 Detailed Description

U3V/GenCP EVENT_CMD specific command data.

14.176.2 Member Data Documentation

14.176.2.1 EventId

```
uint16_t EventId
```

14.176.2.2 Reserved

```
uint16_t Reserved
```

14.176.2.3 Timestamp

```
uint64_t Timestamp
```

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

- include/SpinGenApi/EventAdapterU3V.h

14.177 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

14.177.1 Detailed Description

Entire event data message (without the variable-sized data field)

14.177.2 Member Data Documentation

14.177.2.1 CommandHeader

```
U3V_COMMAND_HEADER CommandHeader
```

14.177.2.2 EventData

```
U3V_EVENT_DATA EventData
```

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

- include/SpinGenApi/EventAdapterU3V.h

14.178 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

14.178.1 Detailed Description

Interface for value properties.

14.178.2 Constructor & Destructor Documentation

14.178.2.1 ValueNode() [1/2]

```
ValueNode ( )
```

Constructor.

14.178.2.2 ValueNode() [2/2]

```
ValueNode (
            std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with GenICam IValue

14.178.2.3 ~ValueNode()

```
~ValueNode ( )
```

Destructor.

14.178.3 Member Function Documentation

14.178.3.1 FromString()

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

Set content of the node as string.

Parameters

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

14.178.3.2 GetNode()

```
virtual INode* GetNode ( )  [virtual]
```

14.178.3.3 IsValueCacheValid()

```
virtual bool IsValueCacheValid ( ) const  [virtual]
```

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

14.178.3.4 SetReference()

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

overload SetReference for Value

Reimplemented from Node.

Reimplemented in FloatNode, IntegerNode, EnumNode, CEnumerationTRef< EnumT >, StringNode, RegisterNode, BooleanNode, CommandNode, EnumEntryNode, CategoryNode, StringRegNode, FloatRegNode, and IntRegNode.

14.178.3.5 ToString()

```
virtual GenICam::gcstring ToString (
            bool Verify = false,
            bool IgnoreCache = false )  [virtual]
```

Get content of the node as string.

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

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

- include/SpinGenApi/ValueNode.h

14.179 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

14.179.1 Detailed Description

Version.

14.179.2 Member Data Documentation

14.179.2.1 Major

```
uint16_t Major
```

14.179.2.2 Minor

```
uint16_t Minor
```

a is incompatible with b if a != b

14.179.2.3 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 15

File Documentation

15.1 doc/spindocs/C++/GettingStarted.dox File Reference

15.2 doc/spindocs/C++/ProgrammerGuide.dox File Reference

15.3 doc/spindocs/shared/Benefits.dox File Reference

15.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference

15.5 doc/spindocs/shared/GenICamGenTL.dox File Reference

15.6 doc/spindocs/shared/Licensing.dox File Reference

15.7 include/AdapterConfig.h File Reference

Include dependency graph for AdapterConfig.h:

Classes

- struct IpInfo
- struct AdapterInfo
- class AdapterConfigException

Namespaces

- AdapterConfig

Macros

- #define ADAPTERCONFIG_API __declspec(dllimport)

Enumerations

- enum AdapterConfigErr {
  IP_ADDRESS_INVALID,
  IP_ADDRESS_IS_NOT_V4,
  IP_ADDRESS_TOO_LARGE,
  IP_ADDRESS_TOO_SMALL,
  HOST_ADDRESS_ZERO,
  SUBNET_MASK_INVALID,
  VALID_SUBNET_NOT_FOUND }

Functions

- ADAPTERCONFIG_API std::vector< AdapterInfo > RetrieveAllAdapters ()
- ADAPTERCONFIG_API void AutoPopulateAdapterInfo (std::vector< AdapterInfo > &adaptersToConfigure, const std::vector< AdapterInfo > &allAdapters)
- ADAPTERCONFIG_API void AutoPopulateAdvancedProperties (std::vector< AdapterInfo > &adaptersTo↩Configure)
- 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 GetEnumerationLogFileName ()
- ADAPTERCONFIG_API std::string GetConfigLogFileName ()
- ADAPTERCONFIG_API void ConfigureAdapter (AdapterInfo &adapter, bool configureIP, bool configure↩AdvancedProperties)
- 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 ()

15.7.1 Macro Definition Documentation

15.7.1.1 ADAPTERCONFIG_API

```
#define ADAPTERCONFIG_API __declspec(dllimport)
```

15.8 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.

15.9 include/BasePtr.h File Reference

Include dependency graph for BasePtr.h:

This graph shows which files directly or indirectly include this file:

Classes

- class BasePtr< T, B >

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

Namespaces

- Spinnaker

Functions

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

  Pointer equal.

15.10 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

15.11 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

15.12 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_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_DEVICETLTYPE }
- enum DevicePowerSupplySelectorEnums {
DevicePowerSupplySelector_External,
NUM_DEVICEPOWERSUPPLYSELECTOR }
- enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICETEMPERATURESELECTOR }
- enum DeviceIndicatorModeEnums {
DeviceIndicatorMode_Inactive,
DeviceIndicatorMode_Active,
DeviceIndicatorMode_ErrorStatus,
NUM_DEVICEINDICATORMODE }
- enum AutoExposureControlPriorityEnums {
AutoExposureControlPriority_Gain,
AutoExposureControlPriority_ExposureTime,
NUM_AUTOEXPOSURECONTROLPRIORITY }
- enum AutoExposureMeteringModeEnums {
AutoExposureMeteringMode_Average,
AutoExposureMeteringMode_Spot,
AutoExposureMeteringMode_Partial,
AutoExposureMeteringMode_CenterWeighted,
AutoExposureMeteringMode_HistgramPeak,
NUM_AUTOEXPOSUREMETERINGMODE }
- enum BalanceWhiteAutoProfileEnums {
BalanceWhiteAutoProfile_Indoor,
BalanceWhiteAutoProfile_Outdoor,
NUM_BALANCEWHITEAUTOPROFILE }
- enum AutoAlgorithmSelectorEnums {
AutoAlgorithmSelector_Awb,
AutoAlgorithmSelector_Ae,
NUM_AUTOALGORITHMSELECTOR }
- enum AutoExposureTargetGreyValueAutoEnums {
AutoExposureTargetGreyValueAuto_Off,
AutoExposureTargetGreyValueAuto_Continuous,
NUM_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum AutoExposureLightingModeEnums {
AutoExposureLightingMode_AutoDetect,
AutoExposureLightingMode_Backlight,
AutoExposureLightingMode_Frontlight,
AutoExposureLightingMode_Normal,
NUM_AUTOEXPOSURELIGHTINGMODE }
- enum GevIEEE1588StatusEnums {
GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,

GevIEEE1588Status_Listening,
GevIEEE1588Status_PreMaster,
GevIEEE1588Status_Master,
GevIEEE1588Status_Passive,
GevIEEE1588Status_Uncalibrated,
GevIEEE1588Status_Slave,
NUM_GEVIEEE1588STATUS }

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

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

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

ChunkSelector_PixelFormat,
ChunkSelector_Timestamp,
ChunkSelector_SequencerSetActive,
ChunkSelector_SerialData,
ChunkSelector_ExposureEndLineStatusAll,
NUM_CHUNKSELECTOR }

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

TestPattern_SensorTestPattern,
NUM_TESTPATTERN }

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

PixelFormat_BayerGB8,
PixelFormat_BayerBG8,
PixelFormat_BayerGR16,
PixelFormat_BayerRG16,
PixelFormat_BayerGB16,
PixelFormat_BayerBG16,
PixelFormat_Mono12Packed,
PixelFormat_BayerGR12Packed,
PixelFormat_BayerRG12Packed,
PixelFormat_BayerGB12Packed,
PixelFormat_BayerBG12Packed,
PixelFormat_YUV411Packed,
PixelFormat_YUV422Packed,
PixelFormat_YUV444Packed,
PixelFormat_Mono12p,
PixelFormat_BayerGR12p,
PixelFormat_BayerRG12p,
PixelFormat_BayerGB12p,
PixelFormat_BayerBG12p,
PixelFormat_YCbCr8,
PixelFormat_YCbCr422_8,
PixelFormat_YCbCr411_8,
PixelFormat_BGR8,
PixelFormat_BGRa8,
PixelFormat_Mono10Packed,
PixelFormat_BayerGR10Packed,
PixelFormat_BayerRG10Packed,
PixelFormat_BayerGB10Packed,
PixelFormat_BayerBG10Packed,
PixelFormat_Mono10p,
PixelFormat_BayerGR10p,
PixelFormat_BayerRG10p,
PixelFormat_BayerGB10p,
PixelFormat_BayerBG10p,
PixelFormat_Mono1p,
PixelFormat_Mono2p,
PixelFormat_Mono4p,
PixelFormat_Mono8s,
PixelFormat_Mono10,
PixelFormat_Mono12,
PixelFormat_Mono14,
PixelFormat_Mono16s,
PixelFormat_Mono32f,
PixelFormat_BayerBG10,
PixelFormat_BayerBG12,
PixelFormat_BayerGB10,
PixelFormat_BayerGB12,
PixelFormat_BayerGR10,
PixelFormat_BayerGR12,
PixelFormat_BayerRG10,
PixelFormat_BayerRG12,
PixelFormat_RGBa8,
PixelFormat_RGBa10,
PixelFormat_RGBa10p,
PixelFormat_RGBa12,
PixelFormat_RGBa12p,
PixelFormat_RGBa14,
PixelFormat_RGBa16,

PixelFormat_RGB8,
PixelFormat_RGB8_Planar,
PixelFormat_RGB10,
PixelFormat_RGB10_Planar,
PixelFormat_RGB10p,
PixelFormat_RGB10p32,
PixelFormat_RGB12,
PixelFormat_RGB12_Planar,
PixelFormat_RGB12p,
PixelFormat_RGB14,
PixelFormat_RGB16,
PixelFormat_RGB16s,
PixelFormat_RGB32f,
PixelFormat_RGB16_Planar,
PixelFormat_RGB565p,
PixelFormat_BGRa10,
PixelFormat_BGRa10p,
PixelFormat_BGRa12,
PixelFormat_BGRa12p,
PixelFormat_BGRa14,
PixelFormat_BGRa16,
PixelFormat_RGBa32f,
PixelFormat_BGR10,
PixelFormat_BGR10p,
PixelFormat_BGR12,
PixelFormat_BGR12p,
PixelFormat_BGR14,
PixelFormat_BGR16,
PixelFormat_BGR565p,
PixelFormat_R8,
PixelFormat_R10,
PixelFormat_R12,
PixelFormat_R16,
PixelFormat_G8,
PixelFormat_G10,
PixelFormat_G12,
PixelFormat_G16,
PixelFormat_B8,
PixelFormat_B10,
PixelFormat_B12,
PixelFormat_B16,
PixelFormat_Coord3D_ABC8,
PixelFormat_Coord3D_ABC8_Planar,
PixelFormat_Coord3D_ABC10p,
PixelFormat_Coord3D_ABC10p_Planar,
PixelFormat_Coord3D_ABC12p,
PixelFormat_Coord3D_ABC12p_Planar,
PixelFormat_Coord3D_ABC16,
PixelFormat_Coord3D_ABC16_Planar,
PixelFormat_Coord3D_ABC32f,
PixelFormat_Coord3D_ABC32f_Planar,
PixelFormat_Coord3D_AC8,
PixelFormat_Coord3D_AC8_Planar,
PixelFormat_Coord3D_AC10p,
PixelFormat_Coord3D_AC10p_Planar,
PixelFormat_Coord3D_AC12p,
PixelFormat_Coord3D_AC12p_Planar,
PixelFormat_Coord3D_AC16,

PixelFormat_Coord3D_AC16_Planar,
PixelFormat_Coord3D_AC32f,
PixelFormat_Coord3D_AC32f_Planar,
PixelFormat_Coord3D_A8,
PixelFormat_Coord3D_A10p,
PixelFormat_Coord3D_A12p,
PixelFormat_Coord3D_A16,
PixelFormat_Coord3D_A32f,
PixelFormat_Coord3D_B8,
PixelFormat_Coord3D_B10p,
PixelFormat_Coord3D_B12p,
PixelFormat_Coord3D_B16,
PixelFormat_Coord3D_B32f,
PixelFormat_Coord3D_C8,
PixelFormat_Coord3D_C10p,
PixelFormat_Coord3D_C12p,
PixelFormat_Coord3D_C16,
PixelFormat_Coord3D_C32f,
PixelFormat_Confidence1,
PixelFormat_Confidence1p,
PixelFormat_Confidence8,
PixelFormat_Confidence16,
PixelFormat_Confidence32f,
PixelFormat_BiColorBGRG8,
PixelFormat_BiColorBGRG10,
PixelFormat_BiColorBGRG10p,
PixelFormat_BiColorBGRG12,
PixelFormat_BiColorBGRG12p,
PixelFormat_BiColorRGBG8,
PixelFormat_BiColorRGBG10,
PixelFormat_BiColorRGBG10p,
PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,
PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,
PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,

PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,
PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,
PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,
PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,
PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,

PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

- enum DecimationVerticalModeEnums {
  DecimationVerticalMode_Discard,
  NUM_DECIMATIONVERTICALMODE }
- enum LineModeEnums {
  LineMode_Input,
  LineMode_Output,
  NUM_LINEMODE }
- enum LineSourceEnums {
  LineSource_Off,
  LineSource_Line0,
  LineSource_Line1,
  LineSource_Line2,
  LineSource_Line3,
  LineSource_UserOutput0,
  LineSource_UserOutput1,
  LineSource_UserOutput2,
  LineSource_UserOutput3,
  LineSource_Counter0Active,
  LineSource_Counter1Active,
  LineSource_LogicBlock0,
  LineSource_LogicBlock1,
  LineSource_ExposureActive,
  LineSource_FrameTriggerWait,
  LineSource_SerialPort0,
  LineSource_PPSSignal,
  LineSource_AllPixel,
  LineSource_AnyPixel,
  NUM_LINESOURCE }
- enum LineInputFilterSelectorEnums {
  LineInputFilterSelector_Deglitch,
  LineInputFilterSelector_Debounce,
  NUM_LINEINPUTFILTERSELECTOR }
- enum UserOutputSelectorEnums {
  UserOutputSelector_UserOutput0,
  UserOutputSelector_UserOutput1,
  UserOutputSelector_UserOutput2,
  UserOutputSelector_UserOutput3,
  NUM_USEROUTPUTSELECTOR }
- enum LineFormatEnums {
  LineFormat_NoConnect,
  LineFormat_TriState,
  LineFormat_TTL,

LineFormat_LVDS,
LineFormat_RS422,
LineFormat_OptoCoupled,
LineFormat_OpenDrain,
NUM_LINEFORMAT }

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

CounterResetSource_Line2,
CounterResetSource_Line3,
CounterResetSource_UserOutput0,
CounterResetSource_UserOutput1,
CounterResetSource_UserOutput2,
CounterResetSource_UserOutput3,
CounterResetSource_Counter0Start,
CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }

- enum CounterEventSourceEnums {
  CounterEventSource_Off,
  CounterEventSource_MHzTick,
  CounterEventSource_Line0,
  CounterEventSource_Line1,
  CounterEventSource_Line2,
  CounterEventSource_Line3,
  CounterEventSource_UserOutput0,
  CounterEventSource_UserOutput1,
  CounterEventSource_UserOutput2,
  CounterEventSource_UserOutput3,
  CounterEventSource_Counter0Start,
  CounterEventSource_Counter1Start,
  CounterEventSource_Counter0End,
  CounterEventSource_Counter1End,
  CounterEventSource_LogicBlock0,
  CounterEventSource_LogicBlock1,
  CounterEventSource_ExposureStart,
  CounterEventSource_ExposureEnd,
  CounterEventSource_FrameTriggerWait,
  NUM_COUNTEREVENTSOURCE }
- enum CounterEventActivationEnums {
  CounterEventActivation_LevelLow,
  CounterEventActivation_LevelHigh,
  CounterEventActivation_FallingEdge,
  CounterEventActivation_RisingEdge,
  CounterEventActivation_AnyEdge,
  NUM_COUNTEREVENTACTIVATION }
- enum CounterResetActivationEnums {
  CounterResetActivation_LevelLow,
  CounterResetActivation_LevelHigh,
  CounterResetActivation_FallingEdge,
  CounterResetActivation_RisingEdge,
  CounterResetActivation_AnyEdge,
  NUM_COUNTERRESETACTIVATION }
- enum DeviceTypeEnums {
  DeviceType_Transmitter,
  DeviceType_Receiver,
  DeviceType_Transceiver,
  DeviceType_Peripheral,
  NUM_DEVICETYPE }
- enum DeviceConnectionStatusEnums {

DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }

- enum DeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }
- enum DeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }
- enum DeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }
- enum DeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,
DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }
- enum DeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }
- enum DeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICESERIALPORTSELECTOR }
- enum DeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,
DeviceSerialPortBaudRate_Baud19200,
DeviceSerialPortBaudRate_Baud38400,
DeviceSerialPortBaudRate_Baud57600,
DeviceSerialPortBaudRate_Baud115200,
DeviceSerialPortBaudRate_Baud230400,
DeviceSerialPortBaudRate_Baud460800,
DeviceSerialPortBaudRate_Baud921600,
NUM_DEVICESERIALPORTBAUDRATE }
- enum SensorTapsEnums {
SensorTaps_One,
SensorTaps_Two,
SensorTaps_Three,
SensorTaps_Four,
SensorTaps_Eight,
SensorTaps_Ten,
NUM_SENSORTAPS }
- enum SensorDigitizationTapsEnums {
SensorDigitizationTaps_One,
SensorDigitizationTaps_Two,
SensorDigitizationTaps_Three,
SensorDigitizationTaps_Four,
SensorDigitizationTaps_Eight,
SensorDigitizationTaps_Ten,
NUM_SENSORDIGITIZATIONTAPS }
- enum RegionSelectorEnums {
RegionSelector_Region0,
RegionSelector_Region1,
RegionSelector_Region2,

RegionSelector_All,
NUM_REGIONSELECTOR }

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

PixelFormatInfoSelector_BayerRG16,
PixelFormatInfoSelector_RGBa8,
PixelFormatInfoSelector_RGBa10,
PixelFormatInfoSelector_RGBa10p,
PixelFormatInfoSelector_RGBa12,
PixelFormatInfoSelector_RGBa12p,
PixelFormatInfoSelector_RGBa14,
PixelFormatInfoSelector_RGBa16,
PixelFormatInfoSelector_RGB8,
PixelFormatInfoSelector_RGB8_Planar,
PixelFormatInfoSelector_RGB10,
PixelFormatInfoSelector_RGB10_Planar,
PixelFormatInfoSelector_RGB10p,
PixelFormatInfoSelector_RGB10p32,
PixelFormatInfoSelector_RGB12,
PixelFormatInfoSelector_RGB12_Planar,
PixelFormatInfoSelector_RGB12p,
PixelFormatInfoSelector_RGB14,
PixelFormatInfoSelector_RGB16,
PixelFormatInfoSelector_RGB16s,
PixelFormatInfoSelector_RGB32f,
PixelFormatInfoSelector_RGB16_Planar,
PixelFormatInfoSelector_RGB565p,
PixelFormatInfoSelector_BGRa8,
PixelFormatInfoSelector_BGRa10,
PixelFormatInfoSelector_BGRa10p,
PixelFormatInfoSelector_BGRa12,
PixelFormatInfoSelector_BGRa12p,
PixelFormatInfoSelector_BGRa14,
PixelFormatInfoSelector_BGRa16,
PixelFormatInfoSelector_RGBa32f,
PixelFormatInfoSelector_BGR8,
PixelFormatInfoSelector_BGR10,
PixelFormatInfoSelector_BGR10p,
PixelFormatInfoSelector_BGR12,
PixelFormatInfoSelector_BGR12p,
PixelFormatInfoSelector_BGR14,
PixelFormatInfoSelector_BGR16,
PixelFormatInfoSelector_BGR565p,
PixelFormatInfoSelector_R8,
PixelFormatInfoSelector_R10,
PixelFormatInfoSelector_R12,
PixelFormatInfoSelector_R16,
PixelFormatInfoSelector_G8,
PixelFormatInfoSelector_G10,
PixelFormatInfoSelector_G12,
PixelFormatInfoSelector_G16,
PixelFormatInfoSelector_B8,
PixelFormatInfoSelector_B10,
PixelFormatInfoSelector_B12,
PixelFormatInfoSelector_B16,
PixelFormatInfoSelector_Coord3D_ABC8,
PixelFormatInfoSelector_Coord3D_ABC8_Planar,
PixelFormatInfoSelector_Coord3D_ABC10p,
PixelFormatInfoSelector_Coord3D_ABC10p_Planar,
PixelFormatInfoSelector_Coord3D_ABC12p,
PixelFormatInfoSelector_Coord3D_ABC12p_Planar,
PixelFormatInfoSelector_Coord3D_ABC16,

PixelFormatInfoSelector_Coord3D_ABC16_Planar,
PixelFormatInfoSelector_Coord3D_ABC32f,
PixelFormatInfoSelector_Coord3D_ABC32f_Planar,
PixelFormatInfoSelector_Coord3D_AC8,
PixelFormatInfoSelector_Coord3D_AC8_Planar,
PixelFormatInfoSelector_Coord3D_AC10p,
PixelFormatInfoSelector_Coord3D_AC10p_Planar,
PixelFormatInfoSelector_Coord3D_AC12p,
PixelFormatInfoSelector_Coord3D_AC12p_Planar,
PixelFormatInfoSelector_Coord3D_AC16,
PixelFormatInfoSelector_Coord3D_AC16_Planar,
PixelFormatInfoSelector_Coord3D_AC32f,
PixelFormatInfoSelector_Coord3D_AC32f_Planar,
PixelFormatInfoSelector_Coord3D_A8,
PixelFormatInfoSelector_Coord3D_A10p,
PixelFormatInfoSelector_Coord3D_A12p,
PixelFormatInfoSelector_Coord3D_A16,
PixelFormatInfoSelector_Coord3D_A32f,
PixelFormatInfoSelector_Coord3D_B8,
PixelFormatInfoSelector_Coord3D_B10p,
PixelFormatInfoSelector_Coord3D_B12p,
PixelFormatInfoSelector_Coord3D_B16,
PixelFormatInfoSelector_Coord3D_B32f,
PixelFormatInfoSelector_Coord3D_C8,
PixelFormatInfoSelector_Coord3D_C10p,
PixelFormatInfoSelector_Coord3D_C12p,
PixelFormatInfoSelector_Coord3D_C16,
PixelFormatInfoSelector_Coord3D_C32f,
PixelFormatInfoSelector_Confidence1,
PixelFormatInfoSelector_Confidence1p,
PixelFormatInfoSelector_Confidence8,
PixelFormatInfoSelector_Confidence16,
PixelFormatInfoSelector_Confidence32f,
PixelFormatInfoSelector_BiColorBGRG8,
PixelFormatInfoSelector_BiColorBGRG10,
PixelFormatInfoSelector_BiColorBGRG10p,
PixelFormatInfoSelector_BiColorBGRG12,
PixelFormatInfoSelector_BiColorBGRG12p,
PixelFormatInfoSelector_BiColorRGBG8,
PixelFormatInfoSelector_BiColorRGBG10,
PixelFormatInfoSelector_BiColorRGBG10p,
PixelFormatInfoSelector_BiColorRGBG12,
PixelFormatInfoSelector_BiColorRGBG12p,
PixelFormatInfoSelector_SCF1WBWG8,
PixelFormatInfoSelector_SCF1WBWG10,
PixelFormatInfoSelector_SCF1WBWG10p,
PixelFormatInfoSelector_SCF1WBWG12,
PixelFormatInfoSelector_SCF1WBWG12p,
PixelFormatInfoSelector_SCF1WBWG14,
PixelFormatInfoSelector_SCF1WBWG16,
PixelFormatInfoSelector_SCF1WGWB8,
PixelFormatInfoSelector_SCF1WGWB10,
PixelFormatInfoSelector_SCF1WGWB10p,
PixelFormatInfoSelector_SCF1WGWB12,
PixelFormatInfoSelector_SCF1WGWB12p,
PixelFormatInfoSelector_SCF1WGWB14,
PixelFormatInfoSelector_SCF1WGWB16,
PixelFormatInfoSelector_SCF1WGWR8,

PixelFormatInfoSelector_SCF1WGWR10,
PixelFormatInfoSelector_SCF1WGWR10p,
PixelFormatInfoSelector_SCF1WGWR12,
PixelFormatInfoSelector_SCF1WGWR12p,
PixelFormatInfoSelector_SCF1WGWR14,
PixelFormatInfoSelector_SCF1WGWR16,
PixelFormatInfoSelector_SCF1WRWG8,
PixelFormatInfoSelector_SCF1WRWG10,
PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,
PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,

PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

- enum DeinterlacingEnums {
  Deinterlacing_Off,
  Deinterlacing_LineDuplication,
  Deinterlacing_Weave,
  NUM_DEINTERLACING }
- enum ImageCompressionRateOptionEnums {
  ImageCompressionRateOption_FixBitrate,
  ImageCompressionRateOption_FixQuality,
  NUM_IMAGECOMPRESSIONRATEOPTION }
- enum ImageCompressionJPEGFormatOptionEnums {
  ImageCompressionJPEGFormatOption_Lossless,
  ImageCompressionJPEGFormatOption_BaselineStandard,
  ImageCompressionJPEGFormatOption_BaselineOptimized,
  ImageCompressionJPEGFormatOption_Progressive,
  NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }
- enum AcquisitionStatusSelectorEnums {
  AcquisitionStatusSelector_AcquisitionTriggerWait,
  AcquisitionStatusSelector_AcquisitionActive,
  AcquisitionStatusSelector_AcquisitionTransfer,
  AcquisitionStatusSelector_FrameTriggerWait,
  AcquisitionStatusSelector_FrameActive,
  AcquisitionStatusSelector_ExposureActive,
  NUM_ACQUISITIONSTATUSSELECTOR }
- enum ExposureTimeModeEnums {
  ExposureTimeMode_Common,
  ExposureTimeMode_Individual,
  NUM_EXPOSURETIMEMODE }
- enum ExposureTimeSelectorEnums {
  ExposureTimeSelector_Common,
  ExposureTimeSelector_Red,
  ExposureTimeSelector_Green,
  ExposureTimeSelector_Blue,
  ExposureTimeSelector_Cyan,
  ExposureTimeSelector_Magenta,
  ExposureTimeSelector_Yellow,
  ExposureTimeSelector_Infrared,

ExposureTimeSelector_Ultraviolet,
ExposureTimeSelector_Stage1,
ExposureTimeSelector_Stage2,
NUM_EXPOSURETIMESELECTOR }

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

TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

- enum TimerTriggerActivationEnums {
  TimerTriggerActivation_RisingEdge,
  TimerTriggerActivation_FallingEdge,
  TimerTriggerActivation_AnyEdge,
  TimerTriggerActivation_LevelHigh,
  TimerTriggerActivation_LevelLow,
  NUM_TIMERTRIGGERACTIVATION }
- enum EncoderSelectorEnums {
  EncoderSelector_Encoder0,
  EncoderSelector_Encoder1,
  EncoderSelector_Encoder2,
  NUM_ENCODERSELECTOR }
- enum EncoderSourceAEnums {
  EncoderSourceA_Off,
  EncoderSourceA_Line0,
  EncoderSourceA_Line1,
  EncoderSourceA_Line2,
  NUM_ENCODERSOURCEA }
- enum EncoderSourceBEnums {
  EncoderSourceB_Off,
  EncoderSourceB_Line0,
  EncoderSourceB_Line1,
  EncoderSourceB_Line2,
  NUM_ENCODERSOURCEB }
- enum EncoderModeEnums {
  EncoderMode_FourPhase,
  EncoderMode_HighResolution,
  NUM_ENCODERMODE }
- enum EncoderOutputModeEnums {

EncoderOutputMode_Off,
EncoderOutputMode_PositionUp,
EncoderOutputMode_PositionDown,
EncoderOutputMode_DirectionUp,
EncoderOutputMode_DirectionDown,
EncoderOutputMode_Motion,
NUM_ENCODEROUTPUTMODE }

- enum EncoderStatusEnums {
  EncoderStatus_EncoderUp,
  EncoderStatus_EncoderDown,
  EncoderStatus_EncoderIdle,
  EncoderStatus_EncoderStatic,
  NUM_ENCODERSTATUS }
- enum EncoderResetSourceEnums {
  EncoderResetSource_Off,
  EncoderResetSource_AcquisitionTrigger,
  EncoderResetSource_AcquisitionStart,
  EncoderResetSource_AcquisitionEnd,
  EncoderResetSource_FrameTrigger,
  EncoderResetSource_FrameStart,
  EncoderResetSource_FrameEnd,
  EncoderResetSource_ExposureStart,
  EncoderResetSource_ExposureEnd,
  EncoderResetSource_Line0,
  EncoderResetSource_Line1,
  EncoderResetSource_Line2,
  EncoderResetSource_Counter0Start,
  EncoderResetSource_Counter1Start,
  EncoderResetSource_Counter2Start,
  EncoderResetSource_Counter0End,
  EncoderResetSource_Counter1End,
  EncoderResetSource_Counter2End,
  EncoderResetSource_Timer0Start,
  EncoderResetSource_Timer1Start,
  EncoderResetSource_Timer2Start,
  EncoderResetSource_Timer0End,
  EncoderResetSource_Timer1End,
  EncoderResetSource_Timer2End,
  EncoderResetSource_UserOutput0,
  EncoderResetSource_UserOutput1,
  EncoderResetSource_UserOutput2,
  EncoderResetSource_SoftwareSignal0,
  EncoderResetSource_SoftwareSignal1,
  EncoderResetSource_SoftwareSignal2,
  EncoderResetSource_Action0,
  EncoderResetSource_Action1,
  EncoderResetSource_Action2,
  EncoderResetSource_LinkTrigger0,
  EncoderResetSource_LinkTrigger1,
  EncoderResetSource_LinkTrigger2,
  NUM_ENCODERRESETSOURCE }
- enum EncoderResetActivationEnums {
  EncoderResetActivation_RisingEdge,
  EncoderResetActivation_FallingEdge,
  EncoderResetActivation_AnyEdge,
  EncoderResetActivation_LevelHigh,
  EncoderResetActivation_LevelLow,
  NUM_ENCODERRESETACTIVATION }

- enum SoftwareSignalSelectorEnums {
  SoftwareSignalSelector_SoftwareSignal0,
  SoftwareSignalSelector_SoftwareSignal1,
  SoftwareSignalSelector_SoftwareSignal2,
  NUM_SOFTWARESIGNALSELECTOR }
- enum ActionUnconditionalModeEnums {
  ActionUnconditionalMode_Off,
  ActionUnconditionalMode_On,
  NUM_ACTIONUNCONDITIONALMODE }
- enum SourceSelectorEnums {
  SourceSelector_Source0,
  SourceSelector_Source1,
  SourceSelector_Source2,
  SourceSelector_All,
  NUM_SOURCESELECTOR }
- enum TransferSelectorEnums {
  TransferSelector_Stream0,
  TransferSelector_Stream1,
  TransferSelector_Stream2,
  TransferSelector_All,
  NUM_TRANSFERSELECTOR }
- enum TransferTriggerSelectorEnums {
  TransferTriggerSelector_TransferStart,
  TransferTriggerSelector_TransferStop,
  TransferTriggerSelector_TransferAbort,
  TransferTriggerSelector_TransferPause,
  TransferTriggerSelector_TransferResume,
  TransferTriggerSelector_TransferActive,
  TransferTriggerSelector_TransferBurstStart,
  TransferTriggerSelector_TransferBurstStop,
  NUM_TRANSFERTRIGGERSELECTOR }
- enum TransferTriggerModeEnums {
  TransferTriggerMode_Off,
  TransferTriggerMode_On,
  NUM_TRANSFERTRIGGERMODE }
- enum TransferTriggerSourceEnums {
  TransferTriggerSource_Line0,
  TransferTriggerSource_Line1,
  TransferTriggerSource_Line2,
  TransferTriggerSource_Counter0Start,
  TransferTriggerSource_Counter1Start,
  TransferTriggerSource_Counter2Start,
  TransferTriggerSource_Counter0End,
  TransferTriggerSource_Counter1End,
  TransferTriggerSource_Counter2End,
  TransferTriggerSource_Timer0Start,
  TransferTriggerSource_Timer1Start,
  TransferTriggerSource_Timer2Start,
  TransferTriggerSource_Timer0End,
  TransferTriggerSource_Timer1End,
  TransferTriggerSource_Timer2End,
  TransferTriggerSource_SoftwareSignal0,
  TransferTriggerSource_SoftwareSignal1,
  TransferTriggerSource_SoftwareSignal2,
  TransferTriggerSource_Action0,
  TransferTriggerSource_Action1,
  TransferTriggerSource_Action2,
  NUM_TRANSFERTRIGGERSOURCE }

- enum TransferTriggerActivationEnums {
  TransferTriggerActivation_RisingEdge,
  TransferTriggerActivation_FallingEdge,
  TransferTriggerActivation_AnyEdge,
  TransferTriggerActivation_LevelHigh,
  TransferTriggerActivation_LevelLow,
  NUM_TRANSFERTRIGGERACTIVATION }
- enum TransferStatusSelectorEnums {
  TransferStatusSelector_Streaming,
  TransferStatusSelector_Paused,
  TransferStatusSelector_Stopping,
  TransferStatusSelector_Stopped,
  TransferStatusSelector_QueueOverflow,
  NUM_TRANSFERSTATUSSELECTOR }
- enum TransferComponentSelectorEnums {
  TransferComponentSelector_Red,
  TransferComponentSelector_Green,
  TransferComponentSelector_Blue,
  TransferComponentSelector_All,
  NUM_TRANSFERCOMPONENTSELECTOR }
- enum Scan3dDistanceUnitEnums {
  Scan3dDistanceUnit_Millimeter,
  Scan3dDistanceUnit_Inch,
  NUM_SCAN3DDISTANCEUNIT }
- enum Scan3dCoordinateSystemEnums {
  Scan3dCoordinateSystem_Cartesian,
  Scan3dCoordinateSystem_Spherical,
  Scan3dCoordinateSystem_Cylindrical,
  NUM_SCAN3DCOORDINATESYSTEM }
- enum Scan3dOutputModeEnums {
  Scan3dOutputMode_UncalibratedC,
  Scan3dOutputMode_CalibratedABC_Grid,
  Scan3dOutputMode_CalibratedABC_PointCloud,
  Scan3dOutputMode_CalibratedAC,
  Scan3dOutputMode_CalibratedAC_Linescan,
  Scan3dOutputMode_CalibratedC,
  Scan3dOutputMode_CalibratedC_Linescan,
  Scan3dOutputMode_RectifiedC,
  Scan3dOutputMode_RectifiedC_Linescan,
  Scan3dOutputMode_DisparityC,
  Scan3dOutputMode_DisparityC_Linescan,
  NUM_SCAN3DOUTPUTMODE }
- enum Scan3dCoordinateSystemReferenceEnums {
  Scan3dCoordinateSystemReference_Anchor,
  Scan3dCoordinateSystemReference_Transformed,
  NUM_SCAN3DCOORDINATESYSTEMREFERENCE }
- enum Scan3dCoordinateSelectorEnums {
  Scan3dCoordinateSelector_CoordinateA,
  Scan3dCoordinateSelector_CoordinateB,
  Scan3dCoordinateSelector_CoordinateC,
  NUM_SCAN3DCOORDINATESELECTOR }
- enum Scan3dCoordinateTransformSelectorEnums {
  Scan3dCoordinateTransformSelector_RotationX,
  Scan3dCoordinateTransformSelector_RotationY,
  Scan3dCoordinateTransformSelector_RotationZ,
  Scan3dCoordinateTransformSelector_TranslationX,
  Scan3dCoordinateTransformSelector_TranslationY,

Scan3dCoordinateTransformSelector_TranslationZ,
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }

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

ChunkSourceID_Source2,
NUM_CHUNKSOURCEID }

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

ChunkScan3dCoordinateReferenceSelector_TranslationZ,
NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }

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

GevCurrentPhysicalLinkConfiguration_StaticLAG,
GevCurrentPhysicalLinkConfiguration_DynamicLAG,
NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

- enum GevIPConfigurationStatusEnums {
GevIPConfigurationStatus_None,
GevIPConfigurationStatus_PersistentIP,
GevIPConfigurationStatus_DHCP,
GevIPConfigurationStatus_LLA,
GevIPConfigurationStatus_ForceIP,
NUM_GEVIPCONFIGURATIONSTATUS }
- enum GevGVCPExtendedStatusCodesSelectorEnums {
GevGVCPExtendedStatusCodesSelector_Version1_1,
GevGVCPExtendedStatusCodesSelector_Version2_0,
NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }
- enum GevGVSPExtendedIDModeEnums {
GevGVSPExtendedIDMode_Off,
GevGVSPExtendedIDMode_On,
NUM_GEVGVSPEXTENDEDIDMODE }
- enum ClConfigurationEnums {
ClConfiguration_Base,
ClConfiguration_Medium,
ClConfiguration_Full,
ClConfiguration_DualBase,
ClConfiguration_EightyBit,
NUM_CLCONFIGURATION }
- enum ClTimeSlotsCountEnums {
ClTimeSlotsCount_One,
ClTimeSlotsCount_Two,
ClTimeSlotsCount_Three,
NUM_CLTIMESLOTSCOUNT }
- enum CxpLinkConfigurationStatusEnums {
CxpLinkConfigurationStatus_None,
CxpLinkConfigurationStatus_Pending,
CxpLinkConfigurationStatus_CXP1_X1,
CxpLinkConfigurationStatus_CXP2_X1,
CxpLinkConfigurationStatus_CXP3_X1,
CxpLinkConfigurationStatus_CXP5_X1,
CxpLinkConfigurationStatus_CXP6_X1,
CxpLinkConfigurationStatus_CXP1_X2,
CxpLinkConfigurationStatus_CXP2_X2,
CxpLinkConfigurationStatus_CXP3_X2,
CxpLinkConfigurationStatus_CXP5_X2,
CxpLinkConfigurationStatus_CXP6_X2,
CxpLinkConfigurationStatus_CXP1_X3,
CxpLinkConfigurationStatus_CXP2_X3,
CxpLinkConfigurationStatus_CXP3_X3,
CxpLinkConfigurationStatus_CXP5_X3,
CxpLinkConfigurationStatus_CXP6_X3,
CxpLinkConfigurationStatus_CXP1_X4,
CxpLinkConfigurationStatus_CXP2_X4,
CxpLinkConfigurationStatus_CXP3_X4,
CxpLinkConfigurationStatus_CXP5_X4,
CxpLinkConfigurationStatus_CXP6_X4,
CxpLinkConfigurationStatus_CXP1_X5,
CxpLinkConfigurationStatus_CXP2_X5,
CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,

CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

- enum CxpLinkConfigurationPreferredEnums {
  CxpLinkConfigurationPreferred_CXP1_X1,
  CxpLinkConfigurationPreferred_CXP2_X1,
  CxpLinkConfigurationPreferred_CXP3_X1,
  CxpLinkConfigurationPreferred_CXP5_X1,
  CxpLinkConfigurationPreferred_CXP6_X1,
  CxpLinkConfigurationPreferred_CXP1_X2,
  CxpLinkConfigurationPreferred_CXP2_X2,
  CxpLinkConfigurationPreferred_CXP3_X2,
  CxpLinkConfigurationPreferred_CXP5_X2,
  CxpLinkConfigurationPreferred_CXP6_X2,
  CxpLinkConfigurationPreferred_CXP1_X3,
  CxpLinkConfigurationPreferred_CXP2_X3,
  CxpLinkConfigurationPreferred_CXP3_X3,
  CxpLinkConfigurationPreferred_CXP5_X3,
  CxpLinkConfigurationPreferred_CXP6_X3,
  CxpLinkConfigurationPreferred_CXP1_X4,
  CxpLinkConfigurationPreferred_CXP2_X4,
  CxpLinkConfigurationPreferred_CXP3_X4,
  CxpLinkConfigurationPreferred_CXP5_X4,
  CxpLinkConfigurationPreferred_CXP6_X4,
  CxpLinkConfigurationPreferred_CXP1_X5,
  CxpLinkConfigurationPreferred_CXP2_X5,
  CxpLinkConfigurationPreferred_CXP3_X5,
  CxpLinkConfigurationPreferred_CXP5_X5,
  CxpLinkConfigurationPreferred_CXP6_X5,
  CxpLinkConfigurationPreferred_CXP1_X6,
  CxpLinkConfigurationPreferred_CXP2_X6,
  CxpLinkConfigurationPreferred_CXP3_X6,
  CxpLinkConfigurationPreferred_CXP5_X6,
  CxpLinkConfigurationPreferred_CXP6_X6,
  NUM_CXPLINKCONFIGURATIONPREFERRED }
- enum CxpLinkConfigurationEnums {
  CxpLinkConfiguration_Auto,
  CxpLinkConfiguration_CXP1_X1,
  CxpLinkConfiguration_CXP2_X1,
  CxpLinkConfiguration_CXP3_X1,
  CxpLinkConfiguration_CXP5_X1,
  CxpLinkConfiguration_CXP6_X1,
  CxpLinkConfiguration_CXP1_X2,
  CxpLinkConfiguration_CXP2_X2,
  CxpLinkConfiguration_CXP3_X2,
  CxpLinkConfiguration_CXP5_X2,
  CxpLinkConfiguration_CXP6_X2,
  CxpLinkConfiguration_CXP1_X3,
  CxpLinkConfiguration_CXP2_X3,
  CxpLinkConfiguration_CXP3_X3,
  CxpLinkConfiguration_CXP5_X3,
  CxpLinkConfiguration_CXP6_X3,
  CxpLinkConfiguration_CXP1_X4,
  CxpLinkConfiguration_CXP2_X4,
  CxpLinkConfiguration_CXP3_X4,

CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,
CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }

- enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
- enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

15.13 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

15.14 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

15.15 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

15.16 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.

15.17 include/DeviceArrivalEventHandler.h File Reference

Include dependency graph for DeviceArrivalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class DeviceArrivalEventHandler

  An event handler for capturing the device arrival event.

Namespaces

- Spinnaker

15.18 include/DeviceEventHandler.h File Reference

Include dependency graph for DeviceEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class DeviceEventHandler

  A handler to device events.

Namespaces

- Spinnaker

15.19 include/DeviceRemovalEventHandler.h File Reference

Include dependency graph for DeviceRemovalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class DeviceRemovalEventHandler

  An event handler for capturing the device removal event.

Namespaces

- Spinnaker

15.20 include/EventHandler.h File Reference

Include dependency graph for EventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class EventHandler

  The base class for all event handler types.

Namespaces

- Spinnaker

15.21 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

15.22 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

15.23 include/ImageEventHandler.h File Reference

Include dependency graph for ImageEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class ImageEventHandler

  A handler for capturing image arrival events.

Namespaces

- Spinnaker

15.24 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

15.25 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

15.26 include/ImageUtility.h File Reference

Include dependency graph for ImageUtility.h:

This graph shows which files directly or indirectly include this file:

Classes

- class ImageUtility

  Static helper functions for the image object class.

Namespaces

- Spinnaker

15.27 include/ImageUtilityHeatmap.h File Reference

Include dependency graph for ImageUtilityHeatmap.h:

This graph shows which files directly or indirectly include this file:

Classes

- class ImageUtilityHeatmap

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

Namespaces

- Spinnaker

15.28 include/ImageUtilityPolarization.h File Reference

Include dependency graph for ImageUtilityPolarization.h:

This graph shows which files directly or indirectly include this file:

Classes

- class ImageUtilityPolarization

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

Namespaces

- Spinnaker

15.29 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

15.30 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

15.31 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

15.32 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

15.33 include/Interface/IDeviceArrivalEventHandler.h File Reference

Include dependency graph for IDeviceArrivalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class IDeviceArrivalEventHandler

Namespaces

- Spinnaker

15.34 include/Interface/IDeviceEventHandler.h File Reference

Include dependency graph for IDeviceEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class IDeviceEventHandler

Namespaces

- Spinnaker

15.35 include/Interface/IDeviceRemovalEventHandler.h File Reference

Include dependency graph for IDeviceRemovalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class IDeviceRemovalEventHandler

Namespaces

- Spinnaker

15.36 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

15.37 include/Interface/IImageEventHandler.h File Reference

Include dependency graph for IImageEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class IImageEventHandler

Namespaces

- Spinnaker

15.38 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

15.39 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

15.40 include/Interface/IInterfaceArrivalEventHandler.h File Reference

Include dependency graph for IInterfaceArrivalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class IInterfaceArrivalEventHandler

Namespaces

- Spinnaker

15.41 include/Interface/IInterfaceEventHandler.h File Reference

Include dependency graph for IInterfaceEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class IInterfaceEventHandler

Namespaces

- Spinnaker

15.42 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

15.43 include/Interface/IInterfaceRemovalEventHandler.h File Reference

Include dependency graph for IInterfaceRemovalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class IInterfaceRemovalEventHandler

Namespaces

- Spinnaker

15.44 include/Interface/ILoggingEventHandler.h File Reference

Include dependency graph for ILoggingEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class ILoggingEventHandler

Namespaces

- Spinnaker

15.45 include/Interface/IStream.h File Reference

Include dependency graph for IStream.h:

Classes

- class IDataStream

Namespaces

- Spinnaker

15.46 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

15.47 include/Interface/ISystemEventHandler.h File Reference

Include dependency graph for ISystemEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class ISystemEventHandler

Namespaces

- Spinnaker

15.48 include/InterfaceArrivalEventHandler.h File Reference

Include dependency graph for InterfaceArrivalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class InterfaceArrivalEventHandler

  An event handler for capturing the interface arrival event.

Namespaces

- Spinnaker

15.49 include/InterfaceEventHandler.h File Reference

Include dependency graph for InterfaceEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class InterfaceEventHandler

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

Namespaces

- Spinnaker

15.50 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

15.51 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

15.52 include/InterfaceRemovalEventHandler.h File Reference

Include dependency graph for InterfaceRemovalEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class InterfaceRemovalEventHandler

  An event handler for capturing the interface removal event.

Namespaces

- Spinnaker

15.53 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

15.54 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

15.55 include/LoggingEventHandler.h File Reference

Include dependency graph for LoggingEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class LoggingEventHandler

  An event handler for capturing the device logging event.

Namespaces

- Spinnaker

15.56 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

15.57 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.

15.58 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

15.59 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

15.60 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

15.61 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)

15.62 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

15.63 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

15.64 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

15.65 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

15.66 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

15.67 include/SpinGenApi/Compatibility.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define FMT_I64 "ll"

15.67.1 Macro Definition Documentation

15.67.1.1 FMT_I64

```
#define FMT_I64 "ll"
```

15.68 include/SpinGenApi/Container.h File Reference

Include dependency graph for Container.h:

This graph shows which files directly or indirectly include this file:

15.69 include/SpinGenApi/Counter.h File Reference

Classes

- class Counter

  Definition of a simple Counter class.

Namespaces

- Spinnaker
- Spinnaker::GenApi

15.70 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 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

Namespaces

- Spinnaker
- Spinnaker::GenApi

15.71 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

15.72 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

15.73 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

15.74 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

15.75 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

15.76 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

15.77 include/SpinGenApi/EventAdapterGEV.h File Reference

Include dependency graph for EventAdapterGEV.h:

This graph shows which files directly or indirectly include this file:

Classes

- struct GVCP_REQUEST_HEADER

  header of a GVCP request packet
- struct GVCP_EVENT_ITEM_BASIC

  layout of a GVCP event item (common to all types)
- struct GVCP_EVENT_ITEM

  layout of a GVCP event item (Extended ID flag not set)
- struct GVCP_EVENT_REQUEST

  Layout of a GVCP event request packet (Extended ID flag not set)
- struct GVCP_EVENTDATA_REQUEST

  Layout of a GVCP event data request packet (Extended ID flag not set)
- struct GVCP_EVENT_ITEM_EXTENDED_ID

  layout of a GVCP event item (Extended ID flag set)
- struct GVCP_EVENT_REQUEST_EXTENDED_ID

  Layout of a GVCP event request packet (Extended ID flag set)
- struct GVCP_EVENTDATA_REQUEST_EXTENDED_ID

  Layout of a GVCP event data request packet (Extended ID flag set)
- class CEventAdapterGEV

  Connects a GigE Event to a node map.

Namespaces

- Spinnaker
- Spinnaker::GenApi

Enumerations

- enum GVCP_MESSAGE_TAGS {
  TAG_EVENT_CMD = 0xc0,
  TAG_EVENTDATA_CMD = 0xc2 }

Variables

- const uint8_t COMMAND_MAGIC = 0x42

15.78 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)

15.79 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

15.80 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

15.81 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

15.82 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

15.83 include/SpinGenApi/GCBase.h File Reference

Include dependency graph for GCBase.h:

This graph shows which files directly or indirectly include this file:

15.84 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.

15.84.1 Macro Definition Documentation

15.84.1.1 GCSTRING_NPOS

```
#define GCSTRING_NPOS size_t(-1)
```

15.84.2 Function Documentation

15.84.2.1 operator<<()

```
std::ostream& operator<< (
            std::ostream & ostr,
            const Spinnaker::GenICam::gcstring & str )  [inline]
```

STL operator out.

15.84.2.2 operator>>()

```
std::istream& operator>> (
            std::istream & istr,
            Spinnaker::GenICam::gcstring & str )  [inline]
```

STL operator in.

15.85 include/SpinGenApi/GCStringVector.h File Reference

Include dependency graph for GCStringVector.h:

This graph shows which files directly or indirectly include this file:

15.86 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

15.87 include/SpinGenApi/GCTypes.h File Reference

Include dependency graph for GCTypes.h:

This graph shows which files directly or indirectly include this file:

Classes

- struct Version_t

  Version.

Namespaces

- Spinnaker
- Spinnaker::GenICam

Macros

- #define __STDC_LIMIT_MACROS
- #define __STDC_CONSTANT_MACROS
- #define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /∗ maximum signed int64 value ∗/
- #define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /∗ minimum signed int64 value ∗/
- #define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /∗ maximum unsigned int64 value ∗/
- #define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /∗ maximum signed int32 value ∗/
- #define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /∗ minimum signed int32 value ∗/
- #define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /∗ maximum unsigned int32 value ∗/
- #define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /∗ maximum signed int8 value ∗/
- #define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /∗ minimum signed int8 value ∗/
- #define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /∗ maximum unsigned int8 value ∗/

Typedefs

- typedef float float32_t

  32 bit floating point
- typedef double float64_t

  64 bit floating point

15.87.1 Macro Definition Documentation

15.87.1.1 __STDC_CONSTANT_MACROS

```
#define __STDC_CONSTANT_MACROS
```

15.87.1.2 __STDC_LIMIT_MACROS

```
#define __STDC_LIMIT_MACROS
```

15.87.1.3 GC_INT32_MAX

```
#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value
*/
```

15.87.1.4 GC_INT32_MIN

```
#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value
*/
```

15.87.1.5 GC_INT64_MAX

```
#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value
*/
```

15.87.1.6 GC_INT64_MIN

```
#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value
*/
```

15.87.1.7 GC_INT8_MAX

```
#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */
```

15.87.1.8 GC_INT8_MIN

```
#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /* minimum signed int8 value */
```

15.87.1.9 GC_UINT32_MAX

```
#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32
value */
```

15.87.1.10 GC_UINT64_MAX

```
#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64
value */
```

15.87.1.11 GC_UINT8_MAX

```
#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8
value */
```

15.88 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 &Variable↩Name)

  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 unsave 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 SetGenICam↩CacheFolder().
- 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 SetGenICam↩CLProtocolFolder().
- SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)

  Stores the path of the GenICam cache folder.
- SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)

  Stores the path of the GenICam logging properties file.
- SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)

  Stores the path of the CLProtocol folder.
- SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")

  splits str input string into a list of tokens using the delimiter
- SPINNAKER_API void GetFiles (const gcstring &FileTemplate, gcstring_vector &FileNames, const bool DirectoriesOnly=false)

  Gets a list of files or directories matching a given FileTemplate.
- SPINNAKER_API gcstring GetModulePathFromFunction (void *pFunction)

  Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

15.88.1 Macro Definition Documentation

15.88.1.1 __ERR__

```
#define __ERR__ __OUTPUT_FORMATER__("ERROR")
```

15.88.1.2 __LINE_STR__

```
#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)
```

15.88.1.3 __LOCATION__

```
#define __LOCATION__ __FILE__ "(" __LINE_STR__ ")"
```

15.88.1.4 __OUTPUT_FORMATER__

```
#define __OUTPUT_FORMATER__(
            _type ) __LOCATION__ " :  " _type " :  "
```

15.88.1.5 __TODO__

```
#define __TODO__ __OUTPUT_FORMATER__("TBD")
```

15.88.1.6 __WARN__

```
#define __WARN__ __OUTPUT_FORMATER__("WARNING")
```

15.88.1.7 _TO_STRING

```
#define _TO_STRING(
            __stN ) #__stN
```

15.88.1.8 EXPAND_TO_STRINGISE

```
#define EXPAND_TO_STRINGISE(
            __stN ) _TO_STRING(__stN)
```

15.88.1.9 GC_COUNTOF

```
#define GC_COUNTOF(
            arr ) (sizeof(arr) / sizeof(arr)[0])
```

15.88.1.10 GENICAM_DEPRECATED

```
#define GENICAM_DEPRECATED(
            FUNCTION ) FUNCTION
```

15.88.1.11 GENICAM_UNUSED

```
#define GENICAM_UNUSED(
            unused_var ) ((void)(unused_var))
```

15.88.1.12 USE_TEMP_CACHE_FILE [1/2]

```
#define USE_TEMP_CACHE_FILE 1
```

15.88.1.13 USE_TEMP_CACHE_FILE [2/2]

```
#define USE_TEMP_CACHE_FILE 1
```

15.89 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

15.90 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.

15.91 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.

15.92 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.

15.93 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.

15.94 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)

15.95 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.

15.96 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.

15.97 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.

15.98 include/SpinGenApi/IFloat.h File Reference

Include dependency graph for IFloat.h:

This graph shows which files directly or indirectly include this file:

Namespaces

- Spinnaker
- Spinnaker::GenApi

Functions

- virtual IFloat & operator= (double Value)=0

  Set node value.
- virtual bool GetValue (bool Verify=false, bool IgnoreCache=false) const =0

  Get node value.
- virtual bool operator() () const

  Get node value.
- virtual GenICam::gcstring operator∗ ()=0

  Get string node value.
- virtual double GetMin ()=0

  Get minimum value allowed.
- virtual double GetMax ()=0

  Get maximum value allowed.
- virtual bool HasInc ()=0

  True if the float has a constant increment.
- virtual EIncMode GetIncMode ()=0

  Get increment mode.
- virtual double GetInc ()=0

  Get the constant increment if there is any.
- virtual double_autovector_t GetListOfValidValues (bool bounded=true)=0

  Get list of valid value.
- virtual ERepresentation GetRepresentation ()=0

  Get recommended representation.
- virtual GenICam::gcstring GetUnit () const =0

  Get the physical unit name.
- virtual EDisplayNotation GetDisplayNotation () const =0

  Get the way the float should be converted to a string.
- virtual int64_t GetDisplayPrecision () const =0

  Get the precision to be used when converting the float to a string.
- virtual void ImposeMin (double Value)=0

  Restrict minimum value.
- virtual void ImposeMax (double Value)=0

  Restrict maximum value.

Variables

- interface SPINNAKER_API_ABSTRACT IFloat

  Interface for float properties.

15.99 include/SpinGenApi/IInteger.h File Reference

Include dependency graph for IInteger.h:

This graph shows which files directly or indirectly include this file:

Namespaces

- Spinnaker
- Spinnaker::GenApi

Functions

- virtual IInteger & operator= (int64_t Value)=0

  Set node value.
- virtual bool GetValue (bool Verify=false, bool IgnoreCache=false) const =0

  Get node value.
- virtual bool operator() () const

Get node value.

- virtual GenICam::gcstring operator∗ ()=0

  Get string node value.

- virtual double GetMin ()=0

  Get minimum value allowed.

- virtual double GetMax ()=0

  Get maximum value allowed.

- virtual EIncMode GetIncMode ()=0

  Get increment mode.

- virtual double GetInc ()=0

  Get the constant increment if there is any.

- virtual double_autovector_t GetListOfValidValues (bool bounded=true)=0

  Get list of valid value.

- virtual ERepresentation GetRepresentation ()=0

  Get recommended representation.

- virtual GenICam::gcstring GetUnit () const =0

  Get the physical unit name.

- virtual void ImposeMin (int64_t Value)=0

  Restrict minimum value.

- virtual void ImposeMax (int64_t Value)=0

  Restrict maximum value.

Variables

- interface SPINNAKER_API_ABSTRACT IInteger

  Interface for integer properties.

15.100 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=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, GenI↩Cam::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 MaxVisiblity)

  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.

15.101 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.

15.102 include/SpinGenApi/INodeMapDyn.h File Reference

Include dependency graph for INodeMapDyn.h:

This graph shows which files directly or indirectly include this file:

Namespaces

- Spinnaker
- Spinnaker::GenApi

Functions

- virtual void LoadXMLFromFile (const GenICam::gcstring &FileName)=0

  Loads an XML from a file.
- virtual void LoadXMLFromFileInject (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)=0

  Loads an XML from a file with injection.
- virtual void LoadXMLFromString (const GenICam::gcstring &XMLData)=0

  Loads an XML from a string.
- virtual void LoadXMLFromStringInject (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)=0

  Loads an XML from a string with injection.
- virtual void PreprocessXMLFromFile (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xv↩Default)=0

Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

- virtual void MergeXMLFiles (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectedFileName, const GenICam::gcstring &OutputFileName)=0

  Injects an XML file into a target file.

- virtual void ExtractIndependentSubtree (const GenICam::gcstring &XMLData, const GenICam::gcstring &InjectXMLData, const GenICam::gcstring &SubTreeRootNodeName, GenICam::gcstring &Extracted↩ Subtree)=0

  Extract independent subtree.

- virtual void GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions)=0

  Gets a list of supported schema versions.

- virtual void LoadXMLFromZIPFile (const GenICam::gcstring &ZipFileName)=0

  Loads an XML from a ZIP file.

- virtual void LoadXMLFromZIPData (const void ∗zipData, size_t zipSize)=0

  Loads an XML from a ZIP data buffer.

- virtual void PreprocessXMLFromZIPFile (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xv↩ Default)=0

  Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

Variables

- interface SPINNAKER_API_ABSTRACT INodeMapDyn

  Interface to access the node map.

15.103 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

15.104 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

15.105 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

15.106 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.

15.107 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.

15.108 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.

15.109 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.

15.110 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.

15.111 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.

15.112 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.

15.113 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

15.114 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.

15.115 include/SpinGenApi/NodeCallbackImpl.h File Reference

Include dependency graph for NodeCallbackImpl.h:

15.116 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

15.117 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.

15.118 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 ()

15.119 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.

15.120 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.

15.121 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

15.122 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

15.123 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.

15.124 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

15.125 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

15.126 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

15.127 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

15.128 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

15.129 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

15.130 include/SpinGenApi/SpinnakerGenApi.h File Reference

Include dependency graph for SpinnakerGenApi.h:

This graph shows which files directly or indirectly include this file:

15.131 include/SpinGenApi/SpinTestCamera.h File Reference

Include dependency graph for SpinTestCamera.h:

Classes

- class SpinTestCamera

Namespaces

- Spinnaker
- Spinnaker::GenApi

15.132 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

15.133 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

15.134 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

15.135 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

15.136 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 _UndefindedRepresentation _UndefinedRepresentation

Typedefs

- typedef GenICam::gcstring_vector StringList_t

  A list of strings.

Enumerations

- enum ESign {
  Signed,
  Unsigned,
  _UndefinedSign }

  signed or unsigned integers
- enum EAccessMode {
  NI,
  NA,
  WO,
  RO,
  RW,
  _UndefinedAccesMode,
  _CycleDetectAccesMode }

  access mode of a node
- enum EVisibility {
  Beginner = 0,
  Expert = 1,
  Guru = 2,
  Invisible = 3,
  _UndefinedVisibility = 99 }

  recommended visibility of a node
- enum ECachingMode {
  NoCache,
  WriteThrough,
  WriteAround,
  _UndefinedCachingMode }

  caching mode of a register
- enum ERepresentation {
  Linear,
  Logarithmic,
  Boolean,
  PureNumber,
  HexNumber,
  IPV4Address,
  MACAddress,
  _UndefinedRepresentation }

  recommended representation of a node value
- enum EEndianess {
  BigEndian,
  LittleEndian,
  _UndefinedEndian }

  Endianess of a value in a register.
- enum ENameSpace {
  Custom,
  Standard,
  _UndefinedNameSpace }

  Defines if a node name is standard or custom.
- enum EStandardNameSpace {
  None,
  GEV,
  IIDC,
  CL,
  USB,
  _UndefinedStandardNameSpace }

Defines from which standard namespace a node name comes from.

- enum EYesNo {
  Yes = 1,
  No = 0,
  _UndefinedYesNo = 2 }

  Defines the choices of a Yes/No alternative.
- enum ESlope {
  Increasing,
  Decreasing,
  Varying,
  Automatic,
  _UndefinedESlope }

  typedef for formula type
- enum EXMLValidation {
  xvLoad = 0x00000001L,
  xvCycles = 0x00000002L,
  xvSFNC = 0x00000004L,
  xvDefault = 0x00000000L,
  xvAll = 0xffffffffL,
  _UndefinedEXMLValidation = 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,
  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.

15.136.1 Macro Definition Documentation

15.136.1.1 interface

```
#define interface struct
```

15.137 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

15.138 include/Spinnaker.h File Reference

Include dependency graph for Spinnaker.h:

This graph shows which files directly or indirectly include this file:

15.139 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 = 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 }

Variables

- const uint64_t EVENT_TIMEOUT_NONE = 0

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

- const uint64_t EVENT_TIMEOUT_INFINITE = 0xFFFFFFFFFFFFFFFF

15.140 include/SpinnakerPlatform.h File Reference

Macros

- #define SPINNAKER_API_ABSTRACT /∗nothing∗/
- #define SPINNAKER_API __attribute__((visibility("default")))
- #define SPINNAKER_LOCAL __attribute__((visibility("hidden")))

15.141 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)

15.141.1 Macro Definition Documentation

15.141.1.1 SPINUPDATE_API

```
#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT
```

15.141.2 Function Documentation

15.141.2.1 GetErrorMessage()

```
SPINUPDATE_API const char* GetErrorMessage ( )
```

15.141.2.2 SetMessageCallback()

```
SPINUPDATE_API void SetMessageCallback (
            UpdatorMessageCallback messageCallbackFunction )
```

15.141.2.3 SetProgressCallback()

```
SPINUPDATE_API void SetProgressCallback (
            UpdatorProgressCallback progressCallbackFunction )
```

15.141.2.4 UpdateFirmware()

```
SPINUPDATE_API int UpdateFirmware (
            std::vector< std::string > args )
```

15.141.2.5 UpdateFirmwareConsole()

```
SPINUPDATE_API int UpdateFirmwareConsole (
            unsigned int numArgs,
            char ** argList )
```

Updates the firmware for the device.

Parameters

numArgs	Number of strings pointed to by argv
argList	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.

15.141.2.6 UpdateFirmwareGUI()

```
SPINUPDATE_API int UpdateFirmwareGUI (
            std::string args )
```

15.141.3 Variable Documentation

15.141.3.1 UpdatorMessageCallback

```
SPINUPDATE_API typedef int(* UpdatorMessageCallback) (const char *message)
```

15.141.3.2 UpdatorProgressCallback

```
SPINUPDATE_API typedef int( * UpdatorProgressCallback) (const char *action, unsigned int address,
int globalPercent, int currPercent)
```

15.142 include/SpinVideo.h File Reference

Include dependency graph for SpinVideo.h:

This graph shows which files directly or indirectly include this file:

Classes

- class SpinVideo

  Provides the functionality for the user to record images to an AVI/MP4 file.

Namespaces

- Spinnaker
- Spinnaker::Video

15.143 include/SpinVideoDefs.h File Reference

Include dependency graph for SpinVideoDefs.h:

This graph shows which files directly or indirectly include this file:

Classes

- struct MJPGOption

  Options for saving MJPG files.
- struct H264Option

  Options for saving H264 files.
- struct AVIOption

  Options for saving AVI files.

Namespaces

- Spinnaker
- Spinnaker::Video

15.144 include/System.h File Reference

Include dependency graph for System.h:

This graph shows which files directly or indirectly include this file:

Classes

- class System

  The system object is used to retrieve the list of interfaces and cameras available.

Namespaces

- Spinnaker

Macros

- #define FLIR_SPINNAKER_VERSION_MAJOR 2
- #define FLIR_SPINNAKER_VERSION_MINOR 3
- #define FLIR_SPINNAKER_VERSION_TYPE 0
- #define FLIR_SPINNAKER_VERSION_BUILD 77

15.144.1 Macro Definition Documentation

15.144.1.1 FLIR_SPINNAKER_VERSION_BUILD

```
#define FLIR_SPINNAKER_VERSION_BUILD 77
```

15.144.1.2 FLIR_SPINNAKER_VERSION_MAJOR

```
#define FLIR_SPINNAKER_VERSION_MAJOR 2
```

15.144.1.3 FLIR_SPINNAKER_VERSION_MINOR

```
#define FLIR_SPINNAKER_VERSION_MINOR 3
```

15.144.1.4 FLIR_SPINNAKER_VERSION_TYPE

```
#define FLIR_SPINNAKER_VERSION_TYPE 0
```

15.145 include/SystemEventHandler.h File Reference

Include dependency graph for SystemEventHandler.h:

This graph shows which files directly or indirectly include this file:

Classes

- class SystemEventHandler

  A handler to interface arrival and removal events on the system.

Namespaces

- Spinnaker

15.146 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

15.147 include/TransportLayerDefs.h File Reference

This graph shows which files directly or indirectly include this file:

Namespaces

- Spinnaker

Enumerations

- enum StreamTypeEnum {
  StreamType_GigEVision,
  StreamType_CameraLink,
  StreamType_CameraLinkHS,
  StreamType_CoaXPress,
  StreamType_USB3Vision,
  StreamType_Custom,
  NUMSTREAMTYPE }

  The enum definitions for TL Device nodes from the transport layer .xml files.
- enum StreamBufferCountModeEnum {
  StreamBufferCountMode_Manual,
  StreamBufferCountMode_Auto,
  NUMSTREAMBUFFERCOUNTMODE }
- enum StreamBufferHandlingModeEnum {
  StreamBufferHandlingMode_OldestFirst,
  StreamBufferHandlingMode_OldestFirstOverwrite,
  StreamBufferHandlingMode_NewestOnly,
  StreamBufferHandlingMode_NewestFirst,
  NUMSTREAMBUFFERHANDLINGMODE }
- enum DeviceTypeEnum {
  DeviceType_GigEVision,
  DeviceType_CameraLink,
  DeviceType_CameraLinkHS,
  DeviceType_CoaXPress,
  DeviceType_USB3Vision,
  DeviceType_Custom,
  NUMDEVICETYPE }
- enum DeviceAccessStatusEnum {
  DeviceAccessStatus_Unknown,
  DeviceAccessStatus_ReadWrite,
  DeviceAccessStatus_ReadOnly,
  DeviceAccessStatus_NoAccess,
  DeviceAccessStatus_Busy,
  DeviceAccessStatus_OpenReadWrite,
  DeviceAccessStatus_OpenReadOnly,
  NUMDEVICEACCESSSTATUS }
- enum GevCCPEnum {
  GevCCP_EnumEntry_GevCCP_OpenAccess,
  GevCCP_EnumEntry_GevCCP_ExclusiveAccess,
  GevCCP_EnumEntry_GevCCP_ControlAccess,
  NUMGEVCCP }
- enum GUIXMLLocationEnum {
  GUIXMLLocation_Device,
  GUIXMLLocation_Host,
  NUMGUIXMLLOCATION }
- enum GenICamXMLLocationEnum {
  GenICamXMLLocation_Device,
  GenICamXMLLocation_Host,
  NUMGENICAMXMLLOCATION }
- enum DeviceEndianessMechanismEnum {
  DeviceEndianessMechanism_Legacy,
  DeviceEndianessMechanism_Standard,
  NUMDEVICEENDIANESSMECHANISM }
- enum DeviceCurrentSpeedEnum {
  DeviceCurrentSpeed_UnknownSpeed,
  DeviceCurrentSpeed_LowSpeed,

DeviceCurrentSpeed_FullSpeed,
DeviceCurrentSpeed_HighSpeed,
DeviceCurrentSpeed_SuperSpeed,
NUMDEVICECURRENTSPEED }

- enum InterfaceTypeEnum {
  InterfaceType_GigEVision,
  InterfaceType_CameraLink,
  InterfaceType_CameraLinkHS,
  InterfaceType_CoaXPress,
  InterfaceType_USB3Vision,
  InterfaceType_Custom,
  NUMINTERFACETYPE }
- enum POEStatusEnum {
  POEStatus_NotSupported,
  POEStatus_PowerOff,
  POEStatus_PowerOn,
  NUMPOESTATUS }
- enum FilterDriverStatusEnum {
  FilterDriverStatus_NotSupported,
  FilterDriverStatus_Disabled,
  FilterDriverStatus_Enabled,
  NUMFILTERDRIVERSTATUS }
- enum TLTypeEnum {
  TLType_GigEVision,
  TLType_CameraLink,
  TLType_CameraLinkHS,
  TLType_CoaXPress,
  TLType_USB3Vision,
  TLType_Mixed,
  TLType_Custom,
  NUMTLTYPE }

15.148 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

15.149 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

15.150 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

15.151 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

15.152 src/Acquisition/Acquisition.cpp File Reference

Include dependency graph for Acquisition.cpp:

Functions

- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.152.1 Function Documentation

15.152.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.152.1.2 main()

```
int main (
            int ,
            char **  )
```

15.152.1.3 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.152.1.4 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.153 src/Acquisition/resource.h File Reference

15.154 src/AcquisitionMultipleCameraRecovery/resource.h File Reference

15.155 src/AcquisitionMultipleThread/resource.h File Reference

15.156 src/ActionCommand/resource.h File Reference

15.157 src/BufferHandling/resource.h File Reference

15.158 src/ChunkData/resource.h File Reference

15.159 src/CounterAndTimer/resource.h File Reference

15.160 src/DeviceEvents/resource.h File Reference

15.161 src/Enumeration/resource.h File Reference

15.162 src/Enumeration_QuickSpin/resource.h File Reference

15.163 src/EnumerationEvents/resource.h File Reference

15.164 src/ExceptionHandling/resource.h File Reference

15.165 src/Exposure/resource.h File Reference

15.166 src/Exposure_QuickSpin/resource.h File Reference

15.167 src/FileAccess_QuickSpin/resource.h File Reference

15.168 src/GigEVisionPerformance/resource.h File Reference

15.169 src/HighDynamicRange/resource.h File Reference

15.170 src/GenTLInfo_QuickSpin/resource.h File Reference

15.171 src/ImageEvents/resource.h File Reference

15.172 src/ImageFormatControl/resource.h File Reference

15.173 src/ImageFormatControl_QuickSpin/resource.h File Reference

15.174 src/Inference/resource.h File Reference

15.175 src/Logging/resource.h File Reference

15.176 src/LogicBlock/resource.h File Reference

15.177 src/LookupTable/resource.h File Reference

15.178 src/NodeMapCallback/resource.h File Reference

15.179 src/NodeMapInfo/resource.h File Reference

15.180 src/Polarization/resource.h File Reference

15.181 src/SaveToAvi/resource.h File Reference

15.182 src/Sequencer/resource.h File Reference

15.183 src/SerialRxTx/resource.h File Reference

15.184 src/Trigger/resource.h File Reference

15.185 src/Trigger_QuickSpin/resource.h File Reference

15.186 src/Acquisition/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.187 src/ActionCommand/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.188 src/BufferHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.189 src/CounterAndTimer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.190 src/DeviceEvents/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.191 src/Enumeration/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.192 src/Enumeration_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.193 src/ExceptionHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.194 src/Exposure/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.195 src/Exposure_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.196 src/FileAccess_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.197 src/GigEVisionPerformance/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.198 src/NodeMapInfo/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.199 src/Sequencer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.200 src/SerialRxTx/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

15.201 src/Acquisition/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.202 src/ActionCommand/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.203 src/BufferHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.204 src/CounterAndTimer/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.205 src/DeviceEvents/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.206 src/Enumeration/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.207 src/Enumeration_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.208 src/ExceptionHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.209 src/Exposure/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.210 src/Exposure_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.211 src/FileAccess_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.212 src/GigEVisionPerformance/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.213 src/ImageFormatControl/stdafx.h File Reference

Include dependency graph for stdafx.h:

15.214 src/ImageFormatControl_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

15.215 src/NodeMapInfo/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.216 src/Polarization/stdafx.h File Reference

Include dependency graph for stdafx.h:

15.217 src/Sequencer/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.218 src/SerialRxTx/stdafx.h File Reference

Include dependency graph for stdafx.h:

This graph shows which files directly or indirectly include this file:

15.219 src/Acquisition/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.220 src/ActionCommand/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.221 src/BufferHandling/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.222 src/CounterAndTimer/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.223 src/DeviceEvents/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.224 src/Enumeration/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.225 src/Enumeration_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.226 src/ExceptionHandling/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.227 src/Exposure/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.228 src/Exposure_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.229 src/FileAccess_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.230 src/GigEVisionPerformance/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.231 src/GenTLInfo_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

15.232 src/NodeMapInfo/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.233 src/Sequencer/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.234 src/SerialRxTx/targetver.h File Reference

Include dependency graph for targetver.h:

This graph shows which files directly or indirectly include this file:

15.235 src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp File Reference

Include dependency graph for AcquisitionMultipleCameraRecovery.cpp:

Classes

- class ImageEventHandlerImpl
- struct GrabInfo
- class InterfaceEventHandlerImpl

Functions

- void SleepyWrapper (int milliseconds)
- void RefreshCameraList (SystemPtr system)
- bool ConfigureCamera (CameraPtr pCam)
- bool ConfigureUserSet1 (CameraPtr pCam)
- void ResetCameraUserSetToDefault (CameraPtr pCam)
- string GetDeviceSerial (CameraPtr pCam)
- void PrintExampleStatistics ()
- int main (int, char **)

Variables

- std::map< std::string, GrabInfo > cameraGrabInfoMap
- CameraList globalCamList

15.235.1 Function Documentation

15.235.1.1 ConfigureCamera()

```
bool ConfigureCamera (
            CameraPtr pCam )
```

15.235.1.2 ConfigureUserSet1()

```
bool ConfigureUserSet1 (
            CameraPtr pCam )
```

15.235.1.3 GetDeviceSerial()

```
string GetDeviceSerial (
            CameraPtr pCam )
```

15.235.1.4 main()

```
int main (
            int ,
            char **  )
```

15.235.1.5 PrintExampleStatistics()

```
void PrintExampleStatistics ( )
```

15.235.1.6 RefreshCameraList()

```
void RefreshCameraList (
            SystemPtr system )
```

15.235.1.7 ResetCameraUserSetToDefault()

```
void ResetCameraUserSetToDefault (
            CameraPtr pCam )
```

15.235.1.8 SleepyWrapper()

```
void SleepyWrapper (
            int milliseconds )
```

15.235.2 Variable Documentation

15.235.2.1 cameraGrabInfoMap

```
std::map<std::string, GrabInfo> cameraGrabInfoMap
```

15.235.2.2 globalCamList

```
CameraList globalCamList
```

15.236 src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference

Include dependency graph for AcquisitionMultipleThread.cpp:

Functions

- int PrintDeviceInfo (INodeMap &nodeMap, std::string camSerial)
- void ∗ AcquireImages (void ∗arg)
- int RunMultipleCameras (CameraList camList)
- int main (int, char ∗∗)

15.236.1 Function Documentation

15.236.1.1 AcquireImages()

```
void* AcquireImages (
            void * arg )
```

15.236.1.2 main()

```
int main (
            int ,
            char **  )
```

15.236.1.3 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap,
            std::string camSerial )
```

15.236.1.4 RunMultipleCameras()

```
int RunMultipleCameras (
            CameraList camList )
```

15.237 src/ActionCommand/ActionCommand.cpp File Reference

Include dependency graph for ActionCommand.cpp:

Functions

- void SleepyWrapper (int milliseconds)
- int PrintDeviceInfo (INodeMap &nodeMap, const unsigned int camNum)
- int ConfigureInterface (const InterfaceList &interfaceList)
- int ConfigureIEEE1588 (const CameraList &camList)
- int ConfigureActionControl (const CameraList &camList)
- int ConfigureOtherNodes (const CameraList &camList)
- int ConfigureTrigger (const CameraList camList)
- int ConfigureChunkData (const CameraList &camList)
- int AcquireImages (const SystemPtr &system, const InterfaceList &interfaceList, CameraList camList)
- int RunMultipleCameras (const SystemPtr &system, const InterfaceList &interfaceList, const CameraList &camList)
- int main (int, char **)

15.237.1 Function Documentation

15.237.1.1 AcquireImages()

```
int AcquireImages (
            const SystemPtr & system,
            const InterfaceList & interfaceList,
            CameraList camList )
```

15.237.1.2 ConfigureActionControl()

```
int ConfigureActionControl (
            const CameraList & camList )
```

15.237.1.3 ConfigureChunkData()

```
int ConfigureChunkData (
            const CameraList & camList )
```

15.237.1.4 ConfigureIEEE1588()

```
int ConfigureIEEE1588 (
            const CameraList & camList )
```

15.237.1.5 ConfigureInterface()

```
int ConfigureInterface (
            const InterfaceList & interfaceList )
```

15.237.1.6 ConfigureOtherNodes()

```
int ConfigureOtherNodes (
            const CameraList & camList )
```

15.237.1.7 ConfigureTrigger()

```
int ConfigureTrigger (
            const CameraList camList )
```

15.237.1.8 main()

```
int main (
            int ,
            char **  )
```

15.237.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap,
            const unsigned int camNum )
```

15.237.1.10 RunMultipleCameras()

```
int RunMultipleCameras (
            const SystemPtr & system,
            const InterfaceList & interfaceList,
            const CameraList & camList )
```

15.237.1.11 SleepyWrapper()

```
void SleepyWrapper (
            int milliseconds )
```

15.238 src/BufferHandling/BufferHandling.cpp File Reference

Include dependency graph for BufferHandling.cpp:

Macros

- #define numBuffers 3
- #define z_numTriggers 6
- #define k_numLoops 9

Functions

- void SleepyWrapper (int milliseconds)
- int ConfigureTrigger (INodeMap &nodeMap)
- int GrabNextImageByTrigger (INodeMap &nodeMap)
- int ResetTrigger (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.238.1 Macro Definition Documentation

15.238.1.1 k_numLoops

```
#define k_numLoops 9
```

15.238.1.2 numBuffers

```
#define numBuffers 3
```

15.238.1.3 z_numTriggers

```
#define z_numTriggers 6
```

15.238.2 Function Documentation

15.238.2.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.238.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
            INodeMap & nodeMap )
```

15.238.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
            INodeMap & nodeMap )
```

15.238.2.4 main()

```
int main (
            int ,
            char **  )
```

15.238.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.238.2.6 ResetTrigger()

```
int ResetTrigger (
            INodeMap & nodeMap )
```

15.238.2.7 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.238.2.8 SleepyWrapper()

```
void SleepyWrapper (
            int milliseconds )
```

15.239 src/ChunkData/ChunkData.cpp File Reference

Include dependency graph for ChunkData.cpp:

Enumerations

- enum chunkDataType {
  IMAGE,
  NODEMAP }

Functions

- int ConfigureChunkData (INodeMap &nodeMap)
- int DisplayChunkData (ImagePtr pImage)
- int DisplayChunkData (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int DisableChunkData (INodeMap &nodeMap)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const chunkDataType chosenChunkData = IMAGE

15.239.1 Enumeration Type Documentation

15.239.1.1 chunkDataType

```
enum chunkDataType
```

Enumerator

IMAGE	
NODEMAP	

15.239.2 Function Documentation

15.239.2.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.239.2.2 ConfigureChunkData()

```
int ConfigureChunkData (
            INodeMap & nodeMap )
```

15.239.2.3 DisableChunkData()

```
int DisableChunkData (
            INodeMap & nodeMap )
```

15.239.2.4 DisplayChunkData() [1/2]

```
int DisplayChunkData (
            ImagePtr pImage )
```

15.239.2.5 DisplayChunkData() [2/2]

```
int DisplayChunkData (
            INodeMap & nodeMap )
```

15.239.2.6 main()

```
int main (
            int ,
            char **  )
```

15.239.2.7 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.239.2.8 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.239.3 Variable Documentation

15.239.3.1 chosenChunkData

```
const chunkDataType chosenChunkData = IMAGE
```

15.240 src/CounterAndTimer/CounterAndTimer.cpp File Reference

Include dependency graph for CounterAndTimer.cpp:

Functions

- int PrintDeviceInfo (INodeMap &nodeMap)
- int SetupCounterAndTimer (INodeMap &nodeMap)
- int ConfigureDigitalIO (INodeMap &nodeMap)
- int ConfigureExposureandTrigger (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int ResetTrigger (INodeMap &nodeMap)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.240.1 Function Documentation

15.240.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.240.1.2 ConfigureDigitalIO()

```
int ConfigureDigitalIO (
            INodeMap & nodeMap )
```

15.240.1.3 ConfigureExposureandTrigger()

```
int ConfigureExposureandTrigger (
            INodeMap & nodeMap )
```

15.240.1.4 main()

```
int main (
            int ,
            char **  )
```

15.240.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.240.1.6 ResetTrigger()

```
int ResetTrigger (
            INodeMap & nodeMap )
```

15.240.1.7 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.240.1.8 SetupCounterAndTimer()

```
int SetupCounterAndTimer (
            INodeMap & nodeMap )
```

15.241 src/DeviceEvents/DeviceEvents.cpp File Reference

Include dependency graph for DeviceEvents.cpp:

Classes

- class DeviceEventHandlerImpl

Enumerations

- enum eventType {
  GENERIC,
  SPECIFIC }

Functions

- int ConfigureDeviceEvents (INodeMap &nodeMap, CameraPtr pCam, DeviceEventHandlerImpl *&device↩EventHandler)
- int ResetDeviceEvents (CameraPtr pCam, DeviceEventHandlerImpl *&deviceEventHandler)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const eventType chosenEvent = GENERIC

15.241.1 Enumeration Type Documentation

15.241.1.1 eventType

```
enum eventType
```

Enumerator

GENERIC	
SPECIFIC	

15.241.2 Function Documentation

15.241.2.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.241.2.2 ConfigureDeviceEvents()

```
int ConfigureDeviceEvents (
            INodeMap & nodeMap,
            CameraPtr pCam,
            DeviceEventHandlerImpl *& deviceEventHandler )
```

15.241.2.3 main()

```
int main (
            int ,
            char **  )
```

15.241.2.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.241.2.5 ResetDeviceEvents()

```
int ResetDeviceEvents (
            CameraPtr pCam,
            DeviceEventHandlerImpl *& deviceEventHandler )
```

15.241.2.6 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.241.3 Variable Documentation

15.241.3.1 chosenEvent

```
const eventType chosenEvent = GENERIC
```

15.242 src/Enumeration/Enumeration.cpp File Reference

Include dependency graph for Enumeration.cpp:

Functions

- int QueryInterface (InterfacePtr pInterface)
- int main (int, char **)

15.242.1 Function Documentation

15.242.1.1 main()

```
int main (
            int ,
            char **  )
```

15.242.1.2 QueryInterface()

```
int QueryInterface (
            InterfacePtr pInterface )
```

15.243 src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp File Reference

Include dependency graph for Enumeration_QuickSpin.cpp:

Functions

- int QueryInterface (InterfacePtr pInterface)
- int main (int, char **)

15.243.1 Function Documentation

15.243.1.1 main()

```
int main (
            int ,
            char **  )
```

15.243.1.2 QueryInterface()

```
int QueryInterface (
            InterfacePtr pInterface )
```

15.244 src/EnumerationEvents/EnumerationEvents.cpp File Reference

Include dependency graph for EnumerationEvents.cpp:

Classes

- class InterfaceEventHandlerImpl
- class SystemEventHandlerImpl

Functions

- void CheckGevEnabled (SystemPtr &pSystem)
- int main (int, char **)

15.244.1 Function Documentation

15.244.1.1 CheckGevEnabled()

```
void CheckGevEnabled (
            SystemPtr & pSystem )
```

15.244.1.2 main()

```
int main (
            int ,
            char **  )
```

15.245 src/ExceptionHandling/ExceptionHandling.cpp File Reference

Include dependency graph for ExceptionHandling.cpp:

Enumerations

- enum exceptionType {
  SPINNAKER_EXCEPTION,
  STANDARD_EXCEPTION,
  STANDARD_CAST_TO_SPINNAKER }

Functions

- void causeSpinnakerException ()
- void causeStandardException ()
- int main (int, char **)

Variables

- const exceptionType chosenException = SPINNAKER_EXCEPTION

15.245.1 Enumeration Type Documentation

15.245.1.1 exceptionType

```
enum exceptionType
```

Enumerator

SPINNAKER_EXCEPTION	
STANDARD_EXCEPTION	
STANDARD_CAST_TO_SPINNAKER	

15.245.2 Function Documentation

15.245.2.1 causeSpinnakerException()

```
void causeSpinnakerException ( )
```

15.245.2.2 causeStandardException()

```
void causeStandardException ( )
```

15.245.2.3 main()

```
int main (
            int ,
            char **  )
```

15.245.3 Variable Documentation

15.245.3.1 chosenException

```
const exceptionType chosenException = SPINNAKER_EXCEPTION
```

15.246 src/Exposure/Exposure.cpp File Reference

Include dependency graph for Exposure.cpp:

Functions

- int ConfigureExposure (INodeMap &nodeMap)
- int ResetExposure (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.246.1 Function Documentation

15.246.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.246.1.2 ConfigureExposure()

```
int ConfigureExposure (
            INodeMap & nodeMap )
```

15.246.1.3 main()

```
int main (
            int ,
            char **  )
```

15.246.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.246.1.5 ResetExposure()

```
int ResetExposure (
            INodeMap & nodeMap )
```

15.246.1.6 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.247 src/Exposure_QuickSpin/Exposure_QuickSpin.cpp File Reference

Include dependency graph for Exposure_QuickSpin.cpp:

Functions

- int ConfigureExposure (CameraPtr pCam)
- int ResetExposure (CameraPtr pCam)
- int PrintDeviceInfo (CameraPtr pCam)
- int AcquireImages (CameraPtr pCam)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.247.1 Function Documentation

15.247.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam )
```

15.247.1.2 ConfigureExposure()

```
int ConfigureExposure (
            CameraPtr pCam )
```

15.247.1.3 main()

```
int main (
            int ,
            char **  )
```

15.247.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            CameraPtr pCam )
```

15.247.1.5 ResetExposure()

```
int ResetExposure (
            CameraPtr pCam )
```

15.247.1.6 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.248 src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp File Reference

Include dependency graph for FileAccess_QuickSpin.cpp:

Functions

- static void PrintResultMessage (bool result)
- int PrintDeviceInfo (INodeMap &nodeMap)
- bool InitializeSystem (SystemPtr &system, CameraList &camList, CameraPtr &pCam)
- static void PrintDebugMessage (string msg)
- bool AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice, ImagePtr pReferenceImage)
- bool ExecuteDeleteCommand (CameraPtr pCam)
- bool OpenFileToWrite (CameraPtr pCam)
- bool ExecuteWriteCommand (CameraPtr pCam)
- bool CloseFile (CameraPtr pCam)
- bool UploadImage ()
- bool OpenFileToRead (CameraPtr pCamera)
- bool ExecuteReadCommand (CameraPtr pCamera)
- bool DownloadImage ()
- void PrintUsage ()
- int main (int argc, char *argv[ ])

Variables

- static bool _enableDebug = false
- static gcstring _fileSelector = "UserFile1"

15.248.1 Function Documentation

15.248.1.1 AcquireImages()

```
bool AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice,
            ImagePtr pReferenceImage )
```

15.248.1.2 CloseFile()

```
bool CloseFile (
            CameraPtr pCam )
```

15.248.1.3 DownloadImage()

```
bool DownloadImage ( )
```

15.248.1.4 ExecuteDeleteCommand()

```
bool ExecuteDeleteCommand (
            CameraPtr pCam )
```

15.248.1.5 ExecuteReadCommand()

```
bool ExecuteReadCommand (
            CameraPtr pCamera )
```

15.248.1.6 ExecuteWriteCommand()

```
bool ExecuteWriteCommand (
            CameraPtr pCam )
```

15.248.1.7 InitializeSystem()

```
bool InitializeSystem (
            SystemPtr & system,
            CameraList & camList,
            CameraPtr & pCam )
```

15.248.1.8 main()

```
int main (
            int argc,
            char * argv[] )
```

15.248.1.9 OpenFileToRead()

```
bool OpenFileToRead (
            CameraPtr pCamera )
```

15.248.1.10 OpenFileToWrite()

```
bool OpenFileToWrite (
            CameraPtr pCam )
```

15.248.1.11 PrintDebugMessage()

```
static void PrintDebugMessage (
            string msg )  [static]
```

15.248.1.12 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.248.1.13 PrintResultMessage()

```
static void PrintResultMessage (
            bool result )  [static]
```

15.248.1.14 PrintUsage()

```
void PrintUsage ( )
```

15.248.1.15 UploadImage()

```
bool UploadImage ( )
```

15.248.2 Variable Documentation

15.248.2.1 _enableDebug

```
bool _enableDebug = false  [static]
```

15.248.2.2 _fileSelector

```
gcstring _fileSelector = "UserFile1"  [static]
```

15.249 src/GenTLInfo_QuickSpin/GenTLInfo_QuickSpin.cpp File Reference

Include dependency graph for GenTLInfo_QuickSpin.cpp:

Functions

- int PrintTransportLayerDeviceInfo (CameraPtr pCamera)
- int PrintTransportLayerStreamInfo (CameraPtr pCamera)
- int PrintTransportLayerInterfaceInfo (InterfacePtr pInterface)
- int PrintApplicationLayerDeviceInfo (CameraPtr pCamera)
- int main (int, char **)

15.249.1 Function Documentation

15.249.1.1 main()

```
int main (
            int ,
            char **  )
```

15.249.1.2 PrintApplicationLayerDeviceInfo()

```
int PrintApplicationLayerDeviceInfo (
            CameraPtr pCamera )
```

15.249.1.3 PrintTransportLayerDeviceInfo()

```
int PrintTransportLayerDeviceInfo (
            CameraPtr pCamera )
```

15.249.1.4 PrintTransportLayerInterfaceInfo()

```
int PrintTransportLayerInterfaceInfo (
            InterfacePtr pInterface )
```

15.249.1.5 PrintTransportLayerStreamInfo()

```
int PrintTransportLayerStreamInfo (
            CameraPtr pCamera )
```

15.250 src/GigEVisionPerformance/CpuUtil.cpp File Reference

Include dependency graph for CpuUtil.cpp:

Namespaces

- CpuUtil
- PerformanceCounter
- SecondsCounter
- Conversion

Functions

- bool StartCpuTracing (CpuUsageInfo *cpuUsage)
- bool StopCpuTracing (CpuUsageInfo *cpuUsage)
- std::string GetCpuStats (CpuUsageInfo *cpuUsage)
- void StartPerformanceCounter ()
- double GetPerformanceCounter ()
- void StartSecondsCounter ()
- int GetSecondsCounter ()
- string NumToCString (int number)
- string NumToCString (double number)
- string NumToCString (float number)

Variables

- double PCFreq
- __int64 CounterStart
- time_t startTime
- time_t endTime
- double timeDiff

15.251 src/GigEVisionPerformance/CpuUtil.h File Reference

Include dependency graph for CpuUtil.h:

This graph shows which files directly or indirectly include this file:

Classes

- struct CpuUsageInfo

Namespaces

- CpuUtil
- PerformanceCounter
- SecondsCounter
- Conversion

Functions

- bool StartCpuTracing (CpuUsageInfo ∗cpuUsage)
- bool StopCpuTracing (CpuUsageInfo ∗cpuUsage)
- std::string GetCpuStats (CpuUsageInfo ∗cpuUsage)
- void StartPerformanceCounter ()
- double GetPerformanceCounter ()
- void StartSecondsCounter ()
- int GetSecondsCounter ()
- string NumToCString (int number)
- string NumToCString (double number)

15.252 src/GigEVisionPerformance/GigEVisionPerformance.cpp File Reference

Include dependency graph for GigEVisionPerformance.cpp:

Functions

- void PrintUsage ()
- bool ParseArguments (int argc, char ∗argv[ ])
- void getCameraCategory (INodeMap &nodeMap, string categoryString)
- void PrintDataStreamInfo (const Spinnaker::CameraPtr pCamera)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapGenTL, int numImages↩ ToAcquire, int iteration)
- int PrintDeviceInfo (INodeMap &nodeMap)
- void PrintCPUUsage ()
- void PrintAllNodes (CameraPtr pCam)
- bool EnableManualFramerate (CameraPtr pCam)
- bool SetFrameRate (CameraPtr pCam)
- int RunSingleCamera (CameraPtr pCam)
- int main (int argc, char ∗argv[ ])

Variables

- CpuUtil::CpuUsageInfo cpuUsageInfo
- int TestDuration = 0
- char * PixelFormatToSet = nullptr
- int PacketSizeToSet = 9000
- int PacketDelayToSet = 0
- bool IsRelease = false
- bool UseDuration = false
- bool UseMaxFramerate = false
- float UserSetFramerate = 0.0
- int NumImagesToGrab = 100
- const char * argNumImages = "-numimages"
- const char * argDuration = "-duration"
- const char * argRelease = "-callrelease"
- const char * argBayerRG = "-bayerrg"
- const char * argPacketSize = "-packetsize"
- const char * argPacketDelay = "-packetdelay"
- const char * argMaxFrames = "-maxfps"
- const char * argUserSetFrames = "-fps"
- const char * argPrintUsage = "-?"

15.252.1 Function Documentation

15.252.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapGenTL,
            int numImagesToAcquire,
            int iteration )
```

15.252.1.2 EnableManualFramerate()

```
bool EnableManualFramerate (
            CameraPtr pCam )
```

15.252.1.3 getCameraCategory()

```
void getCameraCategory (
            INodeMap & nodeMap,
            string categoryString )
```

15.252.1.4 main()

```
int main (
            int argc,
            char * argv[] )
```

15.252.1.5 ParseArguments()

```
bool ParseArguments (
            int argc,
            char * argv[] )
```

15.252.1.6 PrintAllNodes()

```
void PrintAllNodes (
            CameraPtr pCam )
```

15.252.1.7 PrintCPUUsage()

```
void PrintCPUUsage ( )
```

15.252.1.8 PrintDataStreamInfo()

```
void PrintDataStreamInfo (
            const Spinnaker::CameraPtr pCamera )
```

15.252.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.252.1.10 PrintUsage()

```
void PrintUsage ( )
```

15.252.1.11 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.252.1.12 SetFrameRate()

```
bool SetFrameRate (
            CameraPtr pCam )
```

15.252.2 Variable Documentation

15.252.2.1 argBayerRG

```
const char* argBayerRG = "-bayerrg"
```

15.252.2.2 argDuration

```
const char* argDuration = "-duration"
```

15.252.2.3 argMaxFrames

```
const char* argMaxFrames = "-maxfps"
```

15.252.2.4 argNumImages

```
const char* argNumImages = "-numimages"
```

15.252.2.5 argPacketDelay

```
const char* argPacketDelay = "-packetdelay"
```

15.252.2.6 argPacketSize

```
const char* argPacketSize = "-packetsize"
```

15.252.2.7 argPrintUsage

```
const char* argPrintUsage = "-?"
```

15.252.2.8 argRelease

```
const char* argRelease = "-callrelease"
```

15.252.2.9 argUserSetFrames

```
const char* argUserSetFrames = "-fps"
```

15.252.2.10 cpuUsageInfo

```
CpuUtil::CpuUsageInfo cpuUsageInfo
```

15.252.2.11 IsRelease

```
bool IsRelease = false
```

15.252.2.12 NumImagesToGrab

```
int NumImagesToGrab = 100
```

15.252.2.13 PacketDelayToSet

```
int PacketDelayToSet = 0
```

15.252.2.14 PacketSizeToSet

```
int PacketSizeToSet = 9000
```

15.252.2.15 PixelFormatToSet

```
char* PixelFormatToSet = nullptr
```

15.252.2.16 TestDuration

```
int TestDuration = 0
```

15.252.2.17 UseDuration

```
bool UseDuration = false
```

15.252.2.18 UseMaxFramerate

```
bool UseMaxFramerate = false
```

15.252.2.19 UserSetFramerate

```
float UserSetFramerate = 0.0
```

15.253 src/GigEVisionPerformance/GigEVisionPerformance.h File Reference

This graph shows which files directly or indirectly include this file:

15.254 src/HighDynamicRange/HighDynamicRange.cpp File Reference

Include dependency graph for HighDynamicRange.cpp:

Functions

- void PrintBuildInfo ()
- void PrintDeviceInfo (INodeMap &nodeMap)
- bool CheckNodeAccessibility (const CNodePtr nodePtr)
- bool ToggleHDRMode (INodeMap &nodeMap, bool hdrOn)
- bool InitializeHDRImages (INodeMap &nodeMap)
- int RunSingleCamera (CameraPtr cam)
- int main (int, char **)

Variables

- const unsigned int k_HDRShutter1 = 1000
- const unsigned int k_HDRShutter2 = 5000
- const unsigned int k_HDRShutter3 = 15000
- const unsigned int k_HDRShutter4 = 30000
- const unsigned int k_HDRGain1 = 0
- const unsigned int k_HDRGain2 = 5
- const unsigned int k_HDRGain3 = 10
- const unsigned int k_HDRGain4 = 15

15.254.1 Function Documentation

15.254.1.1 CheckNodeAccessibility()

```
bool CheckNodeAccessibility (
            const CNodePtr nodePtr )
```

15.254.1.2 InitializeHDRImages()

```
bool InitializeHDRImages (
            INodeMap & nodeMap )
```

15.254.1.3 main()

```
int main (
            int ,
            char **  )
```

15.254.1.4 PrintBuildInfo()

```
void PrintBuildInfo ( )
```

15.254.1.5 PrintDeviceInfo()

```
void PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.254.1.6 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr cam )
```

15.254.1.7 ToggleHDRMode()

```
bool ToggleHDRMode (
            INodeMap & nodeMap,
            bool hdrOn )
```

15.254.2 Variable Documentation

15.254.2.1 k_HDRGain1

```
const unsigned int k_HDRGain1 = 0
```

15.254.2.2 k_HDRGain2

```
const unsigned int k_HDRGain2 = 5
```

15.254.2.3 k_HDRGain3

```
const unsigned int k_HDRGain3 = 10
```

15.254.2.4 k_HDRGain4

```
const unsigned int k_HDRGain4 = 15
```

15.254.2.5 k_HDRShutter1

```
const unsigned int k_HDRShutter1 = 1000
```

15.254.2.6 k_HDRShutter2

```
const unsigned int k_HDRShutter2 = 5000
```

15.254.2.7 k_HDRShutter3

```
const unsigned int k_HDRShutter3 = 15000
```

15.254.2.8 k_HDRShutter4

```
const unsigned int k_HDRShutter4 = 30000
```

15.255 src/ImageEvents/ImageEvents.cpp File Reference

Include dependency graph for ImageEvents.cpp:

Classes

- class ImageEventHandlerImpl

Functions

- void SleepyWrapper (int milliseconds)
- int ConfigureImageEvents (CameraPtr pCam, ImageEventHandlerImpl ∗&imageEventHandler)
- int WaitForImages (ImageEventHandlerImpl ∗&imageEventHandler)
- int ResetImageEvents (CameraPtr pCam, ImageEventHandlerImpl ∗&imageEventHandler)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice, ImageEvent↩ HandlerImpl ∗&imageEventHandler)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char ∗∗)

15.255.1 Function Documentation

15.255.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice,
            ImageEventHandlerImpl *& imageEventHandler )
```

15.255.1.2 ConfigureImageEvents()

```
int ConfigureImageEvents (
            CameraPtr pCam,
            ImageEventHandlerImpl *& imageEventHandler )
```

15.255.1.3 main()

```
int main (
            int ,
            char **  )
```

15.255.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.255.1.5 ResetImageEvents()

```
int ResetImageEvents (
            CameraPtr pCam,
            ImageEventHandlerImpl *& imageEventHandler )
```

15.255.1.6 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.255.1.7 SleepyWrapper()

```
void SleepyWrapper (
            int milliseconds )
```

15.255.1.8 WaitForImages()

```
int WaitForImages (
            ImageEventHandlerImpl *& imageEventHandler )
```

15.256 src/ImageFormatControl/ImageFormatControl.cpp File Reference

Include dependency graph for ImageFormatControl.cpp:

Functions

- int ConfigureCustomImageSettings (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.256.1 Function Documentation

15.256.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.256.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (
            INodeMap & nodeMap )
```

15.256.1.3 main()

```
int main (
            int ,
            char **  )
```

15.256.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.256.1.5 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.257 src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp File Reference

Include dependency graph for ImageFormatControl_QuickSpin.cpp:

Functions

- int ConfigureCustomImageSettings (CameraPtr pCam)
- int PrintDeviceInfo (CameraPtr pCam)
- int AcquireImages (CameraPtr pCam)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.257.1 Function Documentation

15.257.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam )
```

15.257.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (
            CameraPtr pCam )
```

15.257.1.3 main()

```
int main (
            int ,
            char **  )
```

15.257.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            CameraPtr pCam )
```

15.257.1.5 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.258 src/Inference/Inference.cpp File Reference

Include dependency graph for Inference.cpp:

Enumerations

- enum InferenceNetworkType {
  DETECTION,
  CLASSIFICATION }
- enum FileUploadPersistence {
  FLASH,
  DDR }

Functions

- const std::vector< std::string > labelClassification (arrayLabelClassification, end(arrayLabelClassification))
- const std::vector< std::string > labelDetection (arrayLabelDetection, end(arrayLabelDetection))
- int PrintDeviceInfo (INodeMap &nodeMap)
- bool CameraDeleteFile (INodeMap &nodeMap)
- bool CameraOpenFile (INodeMap &nodeMap)
- bool CameraWriteToFile (INodeMap &nodeMap)
- bool CameraCloseFile (INodeMap &nodeMap)
- std::vector< char > LoadFileIntoMemory (const string &filename)
- int UploadFileToCamera (INodeMap &nodeMap, const std::string &fileSelectorEntryName, const std::string &filePath)
- int DeleteFileOnCamera (INodeMap &nodeMap, const std::string &fileSelectorEntryName)
- int SetChunkEnable (INodeMap &nodeMap, const gcstring &entryName, const bool enable)
- int ConfigureChunkData (INodeMap &nodeMap)
- int DisableChunkData (INodeMap &nodeMap)
- int DisplayChunkData (const ImagePtr pImage)
- int DisableTrigger (INodeMap &nodeMap)
- int ConfigureTrigger (INodeMap &nodeMap)
- int ConfigureInference (INodeMap &nodeMap, bool isEnabled)
- int ConfigureTestPattern (INodeMap &nodeMap, bool isEnabled)
- int AcquireImages (const CameraPtr &pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (const CameraPtr &pCam)
- int main (int, char **)

Variables

- const InferenceNetworkType chosenInferenceNetworkType = CLASSIFICATION
- const FileUploadPersistence chosenFileUploadPersistence = DDR
- const std::string networkFilePath
- const std::string injectedImageFilePath
- const unsigned int injectedImageWidth = (chosenInferenceNetworkType == CLASSIFICATION ? 640 : 720)
- const unsigned int injectedImageHeight = (chosenInferenceNetworkType == CLASSIFICATION ? 400 : 540)
- const char * arrayLabelClassification [ ] = {"daisy", "dandelion", "roses", "sunflowers", "tulips"}
- const char * arrayLabelDetection [ ]

15.258.1 Enumeration Type Documentation

15.258.1.1 FileUploadPersistence

```
enum FileUploadPersistence
```

Enumerator

FLASH	
DDR	

15.258.1.2 InferenceNetworkType

```
enum InferenceNetworkType
```

Enumerator

DETECTION	This network determines the most likely class given a set of predetermined, trained options. Object detection can also provide a location within the image (in the form of a "bounding box" surrounding the class), and can detect multiple objects.
CLASSIFICATION	This network determines the best option from a list of predetermined options; the camera gives a percentage that determines the likelihood of the currently perceived image being one of the classes it has been trained to recognize.

15.258.2 Function Documentation

15.258.2.1 AcquireImages()

```
int AcquireImages (
            const CameraPtr & pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.258.2.2 CameraCloseFile()

```
bool CameraCloseFile (
            INodeMap & nodeMap )
```

15.258.2.3 CameraDeleteFile()

```
bool CameraDeleteFile (
            INodeMap & nodeMap )
```

15.258.2.4 CameraOpenFile()

```
bool CameraOpenFile (
            INodeMap & nodeMap )
```

15.258.2.5 CameraWriteToFile()

```
bool CameraWriteToFile (
            INodeMap & nodeMap )
```

15.258.2.6 ConfigureChunkData()

```
int ConfigureChunkData (
            INodeMap & nodeMap )
```

15.258.2.7 ConfigureInference()

```
int ConfigureInference (
            INodeMap & nodeMap,
            bool isEnabled )
```

15.258.2.8 ConfigureTestPattern()

```
int ConfigureTestPattern (
            INodeMap & nodeMap,
            bool isEnabled )
```

15.258.2.9 ConfigureTrigger()

```
int ConfigureTrigger (
            INodeMap & nodeMap )
```

15.258.2.10 DeleteFileOnCamera()

```
int DeleteFileOnCamera (
            INodeMap & nodeMap,
            const std::string & fileSelectorEntryName )
```

15.258.2.11 DisableChunkData()

```
int DisableChunkData (
            INodeMap & nodeMap )
```

15.258.2.12 DisableTrigger()

```
int DisableTrigger (
            INodeMap & nodeMap )
```

15.258.2.13 DisplayChunkData()

```
int DisplayChunkData (
            const ImagePtr pImage )
```

15.258.2.14 labelClassification()

```
const std::vector<std::string> labelClassification (
            arrayLabelClassification ,
            end(arrayLabelClassification)  )
```

15.258.2.15 labelDetection()

```
const std::vector<std::string> labelDetection (
            arrayLabelDetection ,
            end(arrayLabelDetection)  )
```

15.258.2.16 LoadFileIntoMemory()

```
std::vector<char> LoadFileIntoMemory (
            const string & filename )
```

15.258.2.17 main()

```
int main (
            int ,
            char **  )
```

15.258.2.18 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.258.2.19 RunSingleCamera()

```
int RunSingleCamera (
            const CameraPtr & pCam )
```

15.258.2.20 SetChunkEnable()

```
int SetChunkEnable (
            INodeMap & nodeMap,
            const gcstring & entryName,
            const bool enable )
```

15.258.2.21 UploadFileToCamera()

```
int UploadFileToCamera (
            INodeMap & nodeMap,
            const std::string & fileSelectorEntryName,
            const std::string & filePath )
```

15.258.3 Variable Documentation

15.258.3.1 arrayLabelClassification

```
const char* arrayLabelClassification[] = {"daisy", "dandelion", "roses", "sunflowers", "tulips"}
```

15.258.3.2 arrayLabelDetection

```
const char* arrayLabelDetection[]
```

Initial value:

```
= {"background", "aeroplane", "bicycle",   "bird",   "boat",        "bottle",
                                  "bus",        "car",       "cat",       "chair",  "cow",          "
      diningtable",
                                  "dog",        "horse",     "motorbike", "person", "pottedplant", "
      sheep",
                                  "sofa",       "train",     "monitor"}
```

15.258.3.3 chosenFileUploadPersistence

```
const FileUploadPersistence chosenFileUploadPersistence = DDR
```

15.258.3.4 chosenInferenceNetworkType

```
const InferenceNetworkType chosenInferenceNetworkType = CLASSIFICATION
```

15.258.3.5 injectedImageFilePath

```
const std::string injectedImageFilePath
```

Initial value:

```
=
    (chosenInferenceNetworkType == CLASSIFICATION ? "
      Injected_Image_Classification.raw"
                                                  : "Injected_Image_Detection.raw")
```

15.258.3.6 injectedImageHeight

```
const unsigned int injectedImageHeight = (chosenInferenceNetworkType == CLASSIFICATION ? 400
:  540)
```

15.258.3.7 injectedImageWidth

```
const unsigned int injectedImageWidth = (chosenInferenceNetworkType == CLASSIFICATION ? 640 :
720)
```

15.258.3.8 networkFilePath

```
const std::string networkFilePath
```

Initial value:

```
=
    (chosenInferenceNetworkType == CLASSIFICATION ? "
      Network_Classification" : "Network_Detection")
```

15.259 src/Logging/Logging.cpp File Reference

Include dependency graph for Logging.cpp:

Classes

- class LoggingEventHandlerImpl

Functions

- int main (int, char **)

Variables

- const SpinnakerLogLevel k_LoggingLevel = LOG_LEVEL_DEBUG

15.259.1 Function Documentation

15.259.1.1 main()

```
int main (
            int ,
            char **  )
```

15.259.2 Variable Documentation

15.259.2.1 k_LoggingLevel

```
const SpinnakerLogLevel k_LoggingLevel = LOG_LEVEL_DEBUG
```

15.260 src/LogicBlock/LogicBlock.cpp File Reference

Include dependency graph for LogicBlock.cpp:

Functions

- int ConfigureTrigger (INodeMap &nodeMap)
- int ConfigureLogicBlock (INodeMap &nodeMap)
- int GrabTwoImages (INodeMap &nodeMap, CameraPtr pCam)
- int ResetTrigger (INodeMap &nodeMap)
- int ResetExposure (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.260.1 Function Documentation

15.260.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.260.1.2 ConfigureLogicBlock()

```
int ConfigureLogicBlock (
            INodeMap & nodeMap )
```

15.260.1.3 ConfigureTrigger()

```
int ConfigureTrigger (
            INodeMap & nodeMap )
```

15.260.1.4 GrabTwoImages()

```
int GrabTwoImages (
            INodeMap & nodeMap,
            CameraPtr pCam )
```

15.260.1.5 main()

```
int main (
            int ,
            char **  )
```

15.260.1.6 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.260.1.7 ResetExposure()

```
int ResetExposure (
            INodeMap & nodeMap )
```

15.260.1.8 ResetTrigger()

```
int ResetTrigger (
            INodeMap & nodeMap )
```

15.260.1.9 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.261 src/LookupTable/LookupTable.cpp File Reference

Include dependency graph for LookupTable.cpp:

Functions

- void PrintRetrieveNodeFailure (string node, string name)
- int ConfigureLookupTables (INodeMap &nodeMap)
- int ResetLookupTables (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.261.1 Function Documentation

15.261.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.261.1.2 ConfigureLookupTables()

```
int ConfigureLookupTables (
            INodeMap & nodeMap )
```

15.261.1.3 main()

```
int main (
            int ,
            char **  )
```

15.261.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.261.1.5 PrintRetrieveNodeFailure()

```
void PrintRetrieveNodeFailure (
            string node,
            string name )
```

15.261.1.6 ResetLookupTables()

```
int ResetLookupTables (
            INodeMap & nodeMap )
```

15.261.1.7 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.262 src/NodeMapCallback/NodeMapCallback.cpp File Reference

Include dependency graph for NodeMapCallback.cpp:

Functions

- void OnHeightNodeUpdate (INode *node)
- void OnGainNodeUpdate (INode *node)
- int ConfigureCallbacks (INodeMap &nodeMap, int64_t &callbackHeight, int64_t &callbackGain)
- int ChangeHeightAndGain (INodeMap &nodeMap)
- int ResetCallbacks (INodeMap &nodeMap, int64_t &callbackHeight, int64_t &callbackGain)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.262.1 Function Documentation

15.262.1.1 ChangeHeightAndGain()

```
int ChangeHeightAndGain (
            INodeMap & nodeMap )
```

15.262.1.2 ConfigureCallbacks()

```
int ConfigureCallbacks (
            INodeMap & nodeMap,
            int64_t & callbackHeight,
            int64_t & callbackGain )
```

15.262.1.3 main()

```
int main (
            int ,
            char **  )
```

15.262.1.4 OnGainNodeUpdate()

```
void OnGainNodeUpdate (
            INode * node )
```

15.262.1.5 OnHeightNodeUpdate()

```
void OnHeightNodeUpdate (
            INode * node )
```

15.262.1.6 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.262.1.7 ResetCallbacks()

```
int ResetCallbacks (
            INodeMap & nodeMap,
            int64_t & callbackHeight,
            int64_t & callbackGain )
```

15.262.1.8 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.263 src/NodeMapInfo/NodeMapInfo.cpp File Reference

Include dependency graph for NodeMapInfo.cpp:

Enumerations

- enum readType {
  VALUE,
  INDIVIDUAL }

Functions

- int PrintEnumerationSelector (CNodePtr node, unsigned int level)
- void Indent (unsigned int level)
- int PrintValueNode (CNodePtr node, unsigned int level)
- int PrintStringNode (CNodePtr node, unsigned int level)
- int PrintIntegerNode (CNodePtr node, unsigned int level)
- int PrintFloatNode (CNodePtr node, unsigned int level)
- int PrintBooleanNode (CNodePtr node, unsigned int level)
- int PrintCommandNode (CNodePtr node, unsigned int level)
- int PrintEnumerationNodeAndCurrentEntry (CNodePtr node, unsigned int level)
- int PrintNode (CNodePtr node, unsigned int level)
- int PrintCategoryNodeAndAllFeatures (CNodePtr node, unsigned int level)
- int RunSingleCamera (CameraPtr cam)
- int main (int, char **)

Variables

- const unsigned int maxChars = 35
- const readType chosenRead = VALUE

15.263.1 Enumeration Type Documentation

15.263.1.1 readType

```
enum readType
```

Enumerator

VALUE	
INDIVIDUAL	

15.263.2 Function Documentation

15.263.2.1 Indent()

```
void Indent (
            unsigned int level )
```

15.263.2.2 main()

```
int main (
            int ,
            char **  )
```

15.263.2.3 PrintBooleanNode()

```
int PrintBooleanNode (
            CNodePtr node,
            unsigned int level )
```

15.263.2.4 PrintCategoryNodeAndAllFeatures()

```
int PrintCategoryNodeAndAllFeatures (
            CNodePtr node,
            unsigned int level )
```

15.263.2.5 PrintCommandNode()

```
int PrintCommandNode (
            CNodePtr node,
            unsigned int level )
```

15.263.2.6 PrintEnumerationNodeAndCurrentEntry()

```
int PrintEnumerationNodeAndCurrentEntry (
            CNodePtr node,
            unsigned int level )
```

15.263.2.7 PrintEnumerationSelector()

```
int PrintEnumerationSelector (
            CNodePtr node,
            unsigned int level )
```

15.263.2.8 PrintFloatNode()

```
int PrintFloatNode (
            CNodePtr node,
            unsigned int level )
```

15.263.2.9 PrintIntegerNode()

```
int PrintIntegerNode (
            CNodePtr node,
            unsigned int level )
```

15.263.2.10 PrintNode()

```
int PrintNode (
            CNodePtr node,
            unsigned int level )
```

15.263.2.11 PrintStringNode()

```
int PrintStringNode (
            CNodePtr node,
            unsigned int level )
```

15.263.2.12 PrintValueNode()

```
int PrintValueNode (
            CNodePtr node,
            unsigned int level )
```

15.263.2.13 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr cam )
```

15.263.3 Variable Documentation

15.263.3.1 chosenRead

```
const readType chosenRead = VALUE
```

15.263.3.2 maxChars

```
const unsigned int maxChars = 35
```

15.264 src/Polarization/Polarization.cpp File Reference

Include dependency graph for Polarization.cpp:

Functions

- int PrintDeviceInfo (INodeMap &nodeMap)
- int ConfigureStream (INodeMap &nodeMap)
- int SaveImage (const ImagePtr &pImage, const string filename, gcstring &serialNumber)
- std::string GetQuadFileNameAppendage (const ImageUtilityPolarization::PolarizationQuadrant quadrant)
- int CreateHeatmapImages (const ImagePtr &mono8Image, const string baseFilename, gcstring &device↩ SerialNumber)
- int ExtractAndSavePolarQuadImages (const ImagePtr &pRawPolarizedImage, gcstring &deviceSerial↩ Number)
- int CreateAndSaveGlareReducedImage (const ImagePtr &pRawPolarizedImage, gcstring &deviceSerial↩ Number)
- int CreateNormalizedImage (const ImagePtr &imageToNormalize, const string baseFilename, gcstring &deviceSerialNumber, ImageUtility::SourceDataRange srcDataRange=ImageUtility::IMAGE_DATA_RAN↩ GE)
- int CreateAndSaveStokesImages (const ImagePtr &pRawPolarizedImage, gcstring &deviceSerialNumber)
- int CreateAndSaveAolpDolpImages (const ImagePtr &pRawPolarizedImage, gcstring &deviceSerialNumber)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- static bool isPixelFormatColor = false

15.264.1 Function Documentation

15.264.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMapTLDevice )
```

15.264.1.2 ConfigureStream()

```
int ConfigureStream (
            INodeMap & nodeMap )
```

15.264.1.3 CreateAndSaveAolpDolpImages()

```
int CreateAndSaveAolpDolpImages (
            const ImagePtr & pRawPolarizedImage,
            gcstring & deviceSerialNumber )
```

15.264.1.4 CreateAndSaveGlareReducedImage()

```
int CreateAndSaveGlareReducedImage (
            const ImagePtr & pRawPolarizedImage,
            gcstring & deviceSerialNumber )
```

15.264.1.5 CreateAndSaveStokesImages()

```
int CreateAndSaveStokesImages (
            const ImagePtr & pRawPolarizedImage,
            gcstring & deviceSerialNumber )
```

15.264.1.6 CreateHeatmapImages()

```
int CreateHeatmapImages (
            const ImagePtr & mono8Image,
            const string baseFilename,
            gcstring & deviceSerialNumber )
```

15.264.1.7 CreateNormalizedImage()

```
int CreateNormalizedImage (
            const ImagePtr & imageToNormalize,
            const string baseFilename,
            gcstring & deviceSerialNumber,
            ImageUtility::SourceDataRange srcDataRange = ImageUtility::IMAGE_DATA_RANGE )
```

15.264.1.8 ExtractAndSavePolarQuadImages()

```
int ExtractAndSavePolarQuadImages (
            const ImagePtr & pRawPolarizedImage,
            gcstring & deviceSerialNumber )
```

15.264.1.9 GetQuadFileNameAppendage()

```
std::string GetQuadFileNameAppendage (
            const ImageUtilityPolarization::PolarizationQuadrant quadrant )
```

15.264.1.10 main()

```
int main (
            int ,
            char **  )
```

15.264.1.11 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.264.1.12 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.264.1.13 SaveImage()

```
int SaveImage (
            const ImagePtr & pImage,
            const string filename,
            gcstring & serialNumber )
```

15.264.2 Variable Documentation

15.264.2.1 isPixelFormatColor

```
bool isPixelFormatColor = false  [static]
```

15.265 src/SaveToAvi/SaveToAvi.cpp File Reference

Include dependency graph for SaveToAvi.cpp:

Enumerations

- enum videoType {
  UNCOMPRESSED,
  MJPG,
  H264 }

Functions

- int SaveVectorToVideo (INodeMap &nodeMap, INodeMap &nodeMapTLDevice, vector< ImagePtr > &images)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, vector< ImagePtr > &images)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const videoType chosenVideoType = UNCOMPRESSED

15.265.1 Enumeration Type Documentation

15.265.1.1 videoType

```
enum videoType
```

Enumerator

UNCOMPRESSED	
MJPG	
H264	

15.265.2 Function Documentation

15.265.2.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            vector< ImagePtr > & images )
```

15.265.2.2 main()

```
int main (
            int ,
            char **  )
```

15.265.2.3 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.265.2.4 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.265.2.5 SaveVectorToVideo()

```
int SaveVectorToVideo (
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice,
            vector< ImagePtr > & images )
```

15.265.3 Variable Documentation

15.265.3.1 chosenVideoType

```
const videoType chosenVideoType = UNCOMPRESSED
```

15.266 src/Sequencer/Sequencer.cpp File Reference

Include dependency graph for Sequencer.cpp:

Functions

- void PrintRetrieveNodeFailure (string node, string name)
- int ConfigureSequencerPartOne (INodeMap &nodeMap)
- int SetSingleState (INodeMap &nodeMap, unsigned int sequenceNumber, int64_t widthToSet, int64_↩ t heightToSet, double exposureTimeToSet, double gainToSet)
- int ConfigureSequencerPartTwo (INodeMap &nodeMap)
- int ResetSequencer (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapGenTL, uint64_t timeout)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.266.1 Function Documentation

15.266.1.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapGenTL,
            uint64_t timeout )
```

15.266.1.2 ConfigureSequencerPartOne()

```
int ConfigureSequencerPartOne (
            INodeMap & nodeMap )
```

15.266.1.3 ConfigureSequencerPartTwo()

```
int ConfigureSequencerPartTwo (
            INodeMap & nodeMap )
```

15.266.1.4 main()

```
int main (
            int ,
            char **  )
```

15.266.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.266.1.6 PrintRetrieveNodeFailure()

```
void PrintRetrieveNodeFailure (
            string node,
            string name )
```

15.266.1.7 ResetSequencer()

```
int ResetSequencer (
            INodeMap & nodeMap )
```

15.266.1.8 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.266.1.9 SetSingleState()

```
int SetSingleState (
            INodeMap & nodeMap,
            unsigned int sequenceNumber,
            int64_t widthToSet,
            int64_t heightToSet,
            double exposureTimeToSet,
            double gainToSet )
```

15.267 src/SerialRxTx/SerialRxTx.cpp File Reference

Include dependency graph for SerialRxTx.cpp:

Macros

- #define COM_PORT_COUNT_MAX 256
- #define TWO_SECOND_DELAY 2000
- #define SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND 1000
- #define SERIAL_PORT_BAUD_RATE 19200
- #define SERIAL_PORT_STOP_BITS 0
- #define SERIAL_PORT_PARITY_BITS 0
- #define SERIAL_PORT_DELAY 1500
- #define DATA_BITS 8
- #define MILLISECOND 1000

Functions

- int PrintDeviceInfo (INodeMap &nodeMap)
- int ConfigureDevice (CameraPtr pCam, HANDLE &hFileHandle)
- int SerialRx (CameraPtr pCam, INodeMap &nodeMap, HANDLE &hFileHandle)
- int SerialTx (CameraPtr pCam, INodeMap &nodeMap, HANDLE &hFileHandle)
- int CleanUp (INodeMap &nodeMap, HANDLE &hFileHandle)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

15.267.1 Macro Definition Documentation

15.267.1.1 COM_PORT_COUNT_MAX

```
#define COM_PORT_COUNT_MAX 256
```

15.267.1.2 DATA_BITS

```
#define DATA_BITS 8
```

15.267.1.3 MILLISECOND

```
#define MILLISECOND 1000
```

15.267.1.4 SERIAL_PORT_BAUD_RATE

```
#define SERIAL_PORT_BAUD_RATE 19200
```

15.267.1.5 SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND

```
#define SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND 1000
```

15.267.1.6 SERIAL_PORT_DELAY

```
#define SERIAL_PORT_DELAY 1500
```

15.267.1.7 SERIAL_PORT_PARITY_BITS

```
#define SERIAL_PORT_PARITY_BITS 0
```

15.267.1.8 SERIAL_PORT_STOP_BITS

```
#define SERIAL_PORT_STOP_BITS 0
```

15.267.1.9 TWO_SECOND_DELAY

```
#define TWO_SECOND_DELAY 2000
```

15.267.2 Function Documentation

15.267.2.1 CleanUp()

```
int CleanUp (
            INodeMap & nodeMap,
            HANDLE & hFileHandle )
```

15.267.2.2 ConfigureDevice()

```
int ConfigureDevice (
            CameraPtr pCam,
            HANDLE & hFileHandle )
```

15.267.2.3 main()

```
int main (
            int ,
            char **  )
```

15.267.2.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.267.2.5 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.267.2.6 SerialRx()

```
int SerialRx (
            CameraPtr pCam,
            INodeMap & nodeMap,
            HANDLE & hFileHandle )
```

15.267.2.7 SerialTx()

```
int SerialTx (
            CameraPtr pCam,
            INodeMap & nodeMap,
            HANDLE & hFileHandle )
```

15.268 src/Trigger/Trigger.cpp File Reference

Include dependency graph for Trigger.cpp:

Enumerations

- enum triggerType {
  SOFTWARE,
  HARDWARE,
  SOFTWARE,
  HARDWARE }

Functions

- int ConfigureTrigger (INodeMap &nodeMap)
- int GrabNextImageByTrigger (INodeMap &nodeMap, CameraPtr pCam)
- int ResetTrigger (INodeMap &nodeMap)
- int PrintDeviceInfo (INodeMap &nodeMap)
- int AcquireImages (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const triggerType chosenTrigger = SOFTWARE

15.268.1 Enumeration Type Documentation

15.268.1.1 triggerType

```
enum triggerType
```

Enumerator

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

15.268.2 Function Documentation

15.268.2.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam,
            INodeMap & nodeMap,
            INodeMap & nodeMapTLDevice )
```

15.268.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
            INodeMap & nodeMap )
```

15.268.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
            INodeMap & nodeMap,
            CameraPtr pCam )
```

15.268.2.4 main()

```
int main (
            int ,
            char **  )
```

15.268.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
            INodeMap & nodeMap )
```

15.268.2.6 ResetTrigger()

```
int ResetTrigger (
            INodeMap & nodeMap )
```

15.268.2.7 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.268.3 Variable Documentation

15.268.3.1 chosenTrigger

```
const triggerType chosenTrigger = SOFTWARE
```

15.269 src/Trigger_QuickSpin/Trigger_QuickSpin.cpp File Reference

Include dependency graph for Trigger_QuickSpin.cpp:

Enumerations

- enum triggerType {
  SOFTWARE,
  HARDWARE,
  SOFTWARE,
  HARDWARE }

Functions

- int ConfigureTrigger (CameraPtr pCam)
- int GrabNextImageByTrigger (CameraPtr pCam, ImagePtr &pResultImage)
- int ResetTrigger (CameraPtr pCam)
- int PrintDeviceInfo (CameraPtr pCam)
- int AcquireImages (CameraPtr pCam)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const triggerType chosenTrigger = SOFTWARE

15.269.1 Enumeration Type Documentation

15.269.1.1 triggerType

```
enum triggerType
```

Enumerator

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

15.269.2 Function Documentation

15.269.2.1 AcquireImages()

```
int AcquireImages (
            CameraPtr pCam )
```

15.269.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
            CameraPtr pCam )
```

15.269.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
            CameraPtr pCam,
            ImagePtr & pResultImage )
```

15.269.2.4 main()

```
int main (
            int ,
            char **  )
```

15.269.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
            CameraPtr pCam )
```

15.269.2.6 ResetTrigger()

```
int ResetTrigger (
            CameraPtr pCam )
```

15.269.2.7 RunSingleCamera()

```
int RunSingleCamera (
            CameraPtr pCam )
```

15.269.3 Variable Documentation

15.269.3.1 chosenTrigger

```
const triggerType chosenTrigger = SOFTWARE
```

Chapter 16

Example Documentation

16.1 Acquisition.cpp

Acquisition.cpp shows how to acquire images. It relies on information provided in the Enumeration example. Also, check out the ExceptionHandling and NodeMapInfo examples if you haven't already. ExceptionHandling shows the handling of standard and Spinnaker exceptions while NodeMapInfo explores retrieving information from various node types.

This example touches on the preparation and cleanup of a camera just before and just after the acquisition of images. Image retrieval and conversion, grabbing image data, and saving images are all covered as well.

Once comfortable with Acquisition, we suggest checking out AcquisitionMultipleCamera, NodeMapCallback, or SaveToAvi. AcquisitionMultipleCamera demonstrates simultaneously acquiring images from a number of cameras, NodeMapCallback serves as a good introduction to programming with callbacks and events, and SaveToAvi exhibits video creation.

16.2 AcquisitionMultipleCameraRecovery.cpp

AcquisitionMultipleCameraRecovery.cpp shows how to continously acquire images from multiple cameras using image events. It demonstrates the use of User Set Control to save persistent camera configurations, allowing for smooth camera recovery through interface events. This example relies on information provided in the ImageEvents, EnumerationEvents, ImageFormatControl, and Acquisition examples.

This example uses a global map to retain image information, including the number of images grabbed, the number of incomplete images and the number of removals for each camera over the duration of the example. Cameras may be added or removed after the example has started.

The example assumes each camera has a unique serial number and is capable of configuring User Set 1. Note that if a camera was configured and is disconnected before the example ends, it will not be reconfigured to use the default User Set.

16.3 AcquisitionMultipleThread.cpp

AcquisitionMultipleThread.cpp shows how to capture images from multiple cameras simultaneously using threads. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example is similar to the Acquisition example, except that threads are used to allow for simultaneous acquisitions.

16.4 ActionCommand.cpp

ActionCommand.cpp shows how to send action commands to the camera(s).And it also can perform frame synchronization using multiple cameras simultaneously. It uses IEEE 1588 timestamp, triggers and action commands.

One camera will be master and the other camera(s) will be slave(s). All cameras have to be on the same network.

16.5 BufferHandling.cpp

BufferHandling.cpp shows how the different buffer handling modes work.It relies on information provided in the Acquisition and Trigger examples.

Buffer handling determines the ordering in which images are retrieved, and what occurs when an image is transmitted while the buffer is full. There are four different buffer handling modes available; NewestFirst, NewestOnly, OldestFirst and OldestFirstOverwrite.

This example explores retrieving images in a set pattern; triggering the camera while not retrieving an image (letting the buffer fill up), and retrieving images while not triggering. We cycle through the different buffer handling modes to see which images are retrieved, confirming their identities via their Frame ID values.

16.6 ChunkData.cpp

ChunkData.cpp shows how to get chunk data on an image, either from the nodemap or from the image itself. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

Chunk data provides information on various traits of an image. This includes identifiers such as frame ID, properties such as black level, and more. This information can be acquired from either the nodemap or the image itself.

It may be preferable to grab chunk data from each individual image, as it can be hard to verify whether data is coming from the correct image when using the nodemap. This is because chunk data retrieved from the nodemap is only valid for the current image; when GetNextImage() is called, chunk data will be updated to that of the new current image.

16.7 CounterAndTimer.cpp

CounterAndTimer.cpp shows how to setup a Pulse Width Modulation (PWM) signal using counters and timers. The camera will output the PWM signal via strobe, and capture images at a rate defined by the PWM signal as well. Users should take care to use a PWM signal within the camera's max framerate (by default, the PWM signal is set to 50 Hz).

Counter and Timer functionality is only available for BFS and Oryx Cameras. For details on the hardware setup, see our kb article, "Using Counter and Timer Control"; https://www.flir.↩com/support-center/iis/machine-vision/application-note/using-counter-and-timer-control

16.8 DeviceEvents.cpp

DeviceEvents.cpp shows how to create a handler to access device events. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

Device events can be thought of as camera-related events. This example creates a user-defined class, Device↩EventHandlerImpl, which allows the user to define any properties, parameters, and the event handler itself while DeviceEventHandler, the parent class, allows the child class to appropriately interface with the Spinnaker SDK.

16.9 Enumeration.cpp

Enumeration.cpp shows how to enumerate interfaces and cameras. Knowing this is mandatory for doing anything with the Spinnaker SDK, and is therefore the best place to start learning how to use the SDK.

This example introduces the preparation, use, and cleanup of the system object, interface and camera lists, interfaces, and cameras. It also touches on retrieving both nodes from nodemaps and information from nodes.

Once comfortable with enumeration, we suggest checking out the Acquisition, ExceptionHandling, or NodeMap↩Info examples. Acquisition demonstrates using a camera to acquire images, ExceptionHandling explores the use of standard and Spinnaker exceptions, and NodeMapInfo demonstrates retrieving information from various node types.

16.10 Enumeration_QuickSpin.cpp

Enumeration_QuickSpin.cpp shows how to enumerate interfaces and cameras using the QuickSpin API. QuickSpin is a subset of the Spinnaker library that allows for simpler node access and control. This is a great example to start learning about QuickSpin.

```
This example introduces the preparation, use, and cleanup of the system
object, interface and camera lists, interfaces, and cameras. It also
touches on retrieving information from pre-fetched nodes using QuickSpin.
Retrieving node information is the only portion of the example that
differs from Enumeration.
```

A much wider range of topics is covered in the full Spinnaker examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full Spinnaker examples for further or specific knowledge on a topic.

16.11 EnumerationEvents.cpp

EnumerationEvents.cpp explores arrival and removal events on interfaces and the system. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates two user-defined classes: InterfaceEventHandlerImpl and SystemEventHandlerImpl. These child classes allow the user to define properties, parameters, and the event itself while the parent classes - DeviceArrivalEventHandler, DeviceRemovalEventHandler, and InterfaceEventHandler - allow the child classes to interface with Spinnaker.

16.12 ExceptionHandling.cpp

ExceptionHandling.cpp shows the catching of an exception in Spinnaker. Following this, check out the Acquisition or NodeMapInfo examples if you haven't already. Acquisition demonstrates image acquisition while NodeMapInfo explores retrieving information from various node types.

This example shows three typical paths of exception handling in Spinnaker: catching the exception as a Spinnaker exception, as a standard exception, or as a standard exception which is then cast to a Spinnaker exception.

Once comfortable with Acquisition, ExceptionHandling, and NodeMapInfo, we suggest checking out AcquisitionMultipleCamera, NodeMapCallback, or SaveToAvi. AcquisitionMultipleCamera demonstrates simultaneously acquiring images from a number of cameras, NodeMapCallback serves as a good introduction to programming with callbacks and events, and SaveToAvi exhibits video creation.

16.13 Exposure.cpp

Exposure.cpp shows how to set a custom exposure time on a device. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example shows the processes of preparing the camera, setting a custom exposure time, and restoring the camera to its default state (without power cycling). Ensuring custom values do not fall out of range is also touched on.

Following this, we suggest familiarizing yourself with the ImageFormatControl example if you haven't already. ImageFormatControl is another example on camera customization that is shorter and simpler than many of the others. Once comfortable with Exposure and ImageFormatControl, we suggest checking out any of the longer, more complicated examples related to camera configuration: ChunkData, LookupTable, Sequencer, or Trigger.

16.14 Exposure_QuickSpin.cpp

Exposure_QuickSpin.cpp shows how to customize image exposure time using the QuickSpin API. QuickSpin is a subset of the Spinnaker library that allows for simpler node access and control.

This example prepares the camera, sets a new exposure time, and restores the camera to its default state. Ensure custom values fall within an acceptable range is also touched on. Retrieving and setting information is the only portion of hte example that differs from Exposure.

A much wider range of topics is covered in the full Spinnaker examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full Spinnaker examples for further or specific knowledge on a topic.

16.15 FileAccess_Quickspin.cpp

FileAccess_Quickspin.cpp shows how to read and write images using camera File Access function.This example uploads an image to the camera File Access storage and also download the image from the camera File Access storage and saves it to the disk. It also provides debug message when debug mode is turned on giving more detail status of the progress and error messages to the users.

It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

16.16 GenTLInfo_QuickSpin.cpp

GenTLInfo_QuickSpin.cpp shows how to access node information from interfaces and cameras in C++ with the QuickSpin API. QuickSpin is a subset of Spinnaker that eases access to camera information via direct node access. If you're not already familiar with the basics of Spinnaker, we suggest starting with the Enumeration_QuickSpin example.

The example demonstrates the retrieval of information from interface, transport layer device, transport layer stream, and application layer nodes. The retrieval of information of different data types is also touched on.

A much wider range of topics is covered in the full Spinnaker examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full Spinnaker examples for further or specific knowledge on a topic.

16.17 GigEVisionPerformance.cpp

GigEVisionPerformance.cpp measures GigE Vision performance. It is built on top of Acquisition example.

This example measures CPU related performance statistics and print them out at the end.

16.18 HighDynamicRange.cpp

This example shows how to set High Dynamic Range (HDR) if it is available on the camera.

16.19 ImageEvents.cpp

ImageEvents.cpp shows how to acquire images using the image event handler. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates a user-defined class, ImageEventHandlerImpl, that inherits from the Spinnaker class, Image↩EventHandler. ImageEventHandlerImpl allows the user to define any properties, parameters, and the event itself while ImageEventHandler allows the child class to appropriately interface with Spinnaker.

16.20 ImageFormatControl.cpp

ImageFormatControl.cpp shows how to apply custom image settings to the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example demonstrates setting minimums to offsets, X and Y, and maximums to width and height. It also shows the setting of a new pixel format, which is an enumeration type node.

Following this, we suggest familiarizing yourself with the Exposure example if you haven't already. Exposure is another example on camera customization that is shorter and simpler than many of the others. Once comfortable with Exposure and ImageFormatControl, we suggest checking out any of the longer, more complicated examples related to camera configuration: ChunkData, LookupTable, Sequencer, or Trigger.

16.21 ImageFormatControl_QuickSpin.cpp

ImageFormatControl_QuickSpin.cpp shows how to apply custom image settings to the camera using the QuickSpin API. QuickSpin is a subset of the Spinnaker library that allows for simpler node access and control.

This example demonstrates customizing offsets X and Y, width and height, and the pixel format. Ensuring custom values fall within an acceptable range is also touched on. Retrieving and setting node values using QuickSpin is the only portion of the example that differs from ImageFormatControl.

A much wider range of topics is covered in the full Spinnaker examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full Spinnaker examples for further or specific knowledge on a topic.

16.22 Inference.cpp

Inference.cpp shows how to perform the following:

- Upload custom inference neural networks to the camera (DDR or Flash)
- Inject sample test image
- Enable/Configure chunk data
- Enable/Configure trigger inference ready sync
- Acquire images
- Display inference data from acquired image chunk data
- Disable previously configured camera configurations

Inference is only available for Firefly deep learning cameras. See the related content section on the Firefly DL product page for relevant documentation.

https://www.flir.com/products/firefly-dl/

It can also be helpful to familiarize yourself with the Acquisition, ChunkData and FileAccess_QuickSpin examples.

16.23 Logging.cpp

Logging.cpp shows how to create a handler to access logging events.It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates a user-defined class, LoggingEventHandlerImpl, that inherits from the Spinnaker class, LoggingEventHandler. The child class allows the user to define any properties, parameters, and the event itself while LoggingEventHandler allows the child class to appropriately interface with the Spinnaker SDK.

16.24 LogicBlock.cpp

LogicBlock.cpp shows how to use logic blocks to detect missing triggers and refire. It relies on information provided in the Acquisition and Trigger examples.

A logic block is a collection of combinatorial logic and latches that allows users to create new, custom signals inside the camera. These custom signals can be used by the camera (for example to trigger exposure) or sent out to integrate with external systems.

Logic Block functionality is only available for BFS and Oryx Cameras. For details on logic blocks and how this example works, see our kb article, "Using Logic Blocks with Blackfly S and Oryx"; https://www.flir.↩com/support-center/iis/machine-vision/application-note/using-logic-blocks-with-blackfly-

16.25 LookupTable.cpp

LookupTable.cpp shows how to configure lookup tables on the camera.It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

Lookup tables allow for the customization and control of individual pixels. This can be a very powerful and deeply useful tool; however, because use cases are context dependent, this example only explores lookup table configuration.

16.26 NodeMapCallback.cpp

NodeMapCallback.cpp shows how to use nodemap callbacks. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples. As callbacks are very similar to events, it may be a good idea to explore this example prior to tackling the events examples.

This example focuses on creating, registering, using, and unregistering callbacks. A callback requires a function signature, which allows it to be registered to and access a node. Events, while slightly more complex, follow this same pattern.

Once comfortable with NodeMapCallback, we suggest checking out any of the events examples: DeviceEvents, EnumerationEvents, ImageEvents, or Logging.

16.27 NodeMapInfo.cpp

NodeMapInfo.cpp shows how to retrieve node map information. It relies on information provided in the Enumeration example. Also, check out the Acquisition and ExceptionHandling examples if you haven't already. Acquisition demonstrates image acquisition while ExceptionHandling shows the handling of standard and Spinnaker exceptions.

This example explores retrieving information from all major node types on the camera. This includes string, integer, float, boolean, command, enumeration, category, and value types. Looping through multiple child nodes is also covered. A few node types are not covered - base, port, and register - as they are not fundamental. The final node type - enumeration entry - is explored and printed for nodes whose parent node is a selector node.

Once comfortable with NodeMapInfo, we suggest checking out ImageFormatControl and Exposure. ImageFormat↩ Control explores customizing image settings on a camera while Exposure introduces the standard structure of configuring a device, acquiring some images, and then returning the device to a default state.

16.28 Polarization.cpp

Polarization.cpp shows how to extract and create images from a source image of Polarized8 or BayerRGPolarized8 pixel format using methods from the ImageUtilityPolarization, ImageUtility and ImageUtilityHeatmap classes. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example demonstrates some of the methods that can be used to extract polarization quadrant images and create Stokes', AoLP, and DoLP images from the ImageUtilityPolarization class. It then demonstrates how to use some of the available methods in the ImageUtility and ImageUtilityHeatmap classes to create normalized and heatmap images.

Polarization is only available for polarized cameras. For more information please visit our website; https↩ ://www.flir.com/discover/iis/machine-vision/imaging-reflective-surfaces-sonys-first-pola

16.29 SaveToAvi.cpp

SaveToAvi.cpp shows how to create an video from a vector of images.It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example introduces the SpinVideo class, which is used to quickly and easily create various types of video files. It demonstrates the creation of three types: uncompressed, MJPG, and H264.

16.30 Sequencer.cpp

Sequencer.cpp shows how to use the sequencer to grab images with various settings. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples as these examples provide a strong introduction to camera customization.

The sequencer is another very powerful tool, which can be used to create and store multiple states of customized image settings. A very useful application of the sequencer is creating high dynamic range images.

This example is probably the most complex and definitely the longest. As such, the configuration has been split between three functions. The first prepares the camera to set the sequences, the second sets the settings for a single state (it is run five times), and the third configures the camera to use the sequencer when it acquires images.

16.31 SerialRxTx.cpp

SerialRxTx.cpp shows how to communicate using Serial ports.It sets serial port settings in Spinnaker, open and operate File access and create Com Port handle After the setup, it transmits and received simple data. It verifies the transmission by reading data and transmitting data to COM Port.

THIS EXAMPLE ONLY WORKS IN WINDOWS OS

16.32 Trigger.cpp

Trigger.cpp shows how to trigger the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

This example shows the process of configuring, using, and cleaning up a camera for use with both a software and a hardware trigger.

16.33 Trigger_QuickSpin.cpp

Trigger_QuickSpin.cpp shows how to capture images with the trigger using the QuickSpin API. QuickSpin is a subset of the Spinnaker library that allows for simpler node access and control.

This example demonstrates how to prepare, execute, and clean up the camera in regards to using both software and hardware triggers. Retrieving and setting node values using QuickSpin is the only portion of the example that differs from Trigger.

```
A much wider range of topics is covered in the full Spinnaker examples than
in the QuickSpin ones. There are only enough QuickSpin examples to
demonstrate node access and to get started with the API; please see full
Spinnaker examples for further or specific knowledge on a topic.
```

Index

_ClearXMLCache
Spinnaker GenApi Classes, 243
Spinnaker::GenApi::CNodeMapRefT, 733
_Connect
Spinnaker GenApi Classes, 243, 244
Spinnaker::GenApi::CNodeMapRefT, 733, 734
_Destroy
Spinnaker GenApi Classes, 244
_GetDeviceName
Spinnaker GenApi Classes, 244
Spinnaker::GenApi::CNodeMapRefT, 734
_GetNode
Spinnaker GenApi Classes, 244
Spinnaker::GenApi::CNodeMapRefT, 734
_GetNodes
Spinnaker GenApi Classes, 244
Spinnaker::GenApi::CNodeMapRefT, 734
_GetSupportedSchemaVersions
Spinnaker GenApi Classes, 244
Spinnaker::GenApi::CNodeMapRefT, 734
_Initialize
Spinnaker::GenApi::CGeneric_XMLLoader↩
Params, 693
_InvalidateNodes
Spinnaker GenApi Classes, 244
Spinnaker::GenApi::CNodeMapRefT, 735
_LoadXMLFromFile
Spinnaker GenApi Classes, 245
Spinnaker::GenApi::CNodeMapRefT, 735
_LoadXMLFromFileInject
Spinnaker GenApi Classes, 245
Spinnaker::GenApi::CNodeMapRefT, 735
_LoadXMLFromString
Spinnaker GenApi Classes, 245
Spinnaker::GenApi::CNodeMapRefT, 735
_LoadXMLFromStringInject
Spinnaker GenApi Classes, 245
Spinnaker::GenApi::CNodeMapRefT, 735
_LoadXMLFromZIPData
Spinnaker GenApi Classes, 245
Spinnaker::GenApi::CNodeMapRefT, 736
_LoadXMLFromZIPFile
Spinnaker GenApi Classes, 245
Spinnaker::GenApi::CNodeMapRefT, 736
_Poll
Spinnaker GenApi Classes, 245
Spinnaker::GenApi::CNodeMapRefT, 736
_Ptr
Spinnaker::GenApi::CNodeMapRefT, 736
Spinnaker::GenApi::NodeMap, 1060
_TO_STRING
GCUtilities.h, 1268
_UndefindedRepresentation
Types Enums, 377
__ERR__
GCUtilities.h, 1267
__LINE_STR__
GCUtilities.h, 1268
__LOCATION__
GCUtilities.h, 1268
__OUTPUT_FORMATER__
GCUtilities.h, 1268
__STDC_CONSTANT_MACROS
GCTypes.h, 1264
__STDC_LIMIT_MACROS
GCTypes.h, 1264
__TODO__
GCUtilities.h, 1268
__WARN__
GCUtilities.h, 1268
_enableDebug
FileAccess_QuickSpin.cpp, 1406
_fileSelector
FileAccess_QuickSpin.cpp, 1406
_npos
Spinnaker::GenICam::gcstring, 826
_pCount
Spinnaker::GenApi::double_autovector_t, 777
Spinnaker::GenApi::int64_autovector_t, 980
_pv
Spinnaker::GenApi::double_autovector_t, 777
Spinnaker::GenApi::int64_autovector_t, 980
~AutoLock
Spinnaker::GenApi::AutoLock, 463
Spinnaker::GenICam::AutoLock, 462
~BasePtr
Spinnaker::BasePtr, 465
~BooleanNode
Spinnaker::GenApi::BooleanNode, 471
~CChunkAdapter
Spinnaker::GenApi::CChunkAdapter, 650
~CChunkAdapterDcam
Spinnaker::GenApi::CChunkAdapterDcam, 653
~CChunkAdapterGEV
Spinnaker::GenApi::CChunkAdapterGEV, 658
~CChunkAdapterGeneric
Spinnaker::GenApi::CChunkAdapterGeneric, 656
~CChunkAdapterU3V

Spinnaker::GenApi::CChunkAdapterU3V, 660
~CChunkPort
Spinnaker::GenApi::CChunkPort, 663
~CEnumerationTRef
Spinnaker::GenApi::CEnumerationTRef, 669
~CEventAdapter
Spinnaker::GenApi::CEventAdapter, 673
~CEventAdapter1394
Spinnaker::GenApi::CEventAdapter1394, 675
~CEventAdapterGEV
Spinnaker::GenApi::CEventAdapterGEV, 680
~CEventAdapterGeneric
Spinnaker::GenApi::CEventAdapterGeneric, 677
~CEventAdapterU3V
Spinnaker::GenApi::CEventAdapterU3V, 682
~CEventPort
Spinnaker::GenApi::CEventPort, 685
~CFeatureBag
Spinnaker::GenApi::CFeatureBag, 689
~CGlobalLock
Spinnaker::GenICam::CGlobalLock, 695
~CGlobalLockUnlocker
Spinnaker::GenICam::CGlobalLockUnlocker, 697
~CLock
Spinnaker::GenApi::CLock, 713
Spinnaker::GenICam::CLock, 711
~CNodeCallback
Spinnaker::GenApi::CNodeCallback, 718
~CNodeMapFactory
Spinnaker::GenApi::CNodeMapFactory, 722
~CNodeMapRefT
Spinnaker GenApi Classes, 247
~CPointer
Spinnaker::GenApi::CPointer, 743
~CPortImpl
Spinnaker::GenApi::CPortImpl, 748
~CPortWriteList
Spinnaker::GenApi::CPortWriteList, 751
~CRegisterPortImpl
Spinnaker::GenApi::CRegisterPortImpl, 755
~CSelectorSet
Spinnaker::GenApi::CSelectorSet, 758
~Camera
Spinnaker::Camera, 503
~CameraBase
Spinnaker::CameraBase, 627
~CameraList
Spinnaker::CameraList, 641
~CategoryNode
Spinnaker::GenApi::CategoryNode, 648
~ChunkData
Spinnaker::ChunkData, 701
~CommandNode
Spinnaker::GenApi::CommandNode, 738
~DeviceArrivalEventHandler
Spinnaker::DeviceArrivalEventHandler, 766
~DeviceEventHandler
Spinnaker::DeviceEventHandler, 769
~DeviceEventHandlerImpl
DeviceEventHandlerImpl, 771
~DeviceRemovalEventHandler
Spinnaker::DeviceRemovalEventHandler, 774
~EnumEntryNode
Spinnaker::GenApi::EnumEntryNode, 786
~EnumNode
Spinnaker::GenApi::EnumNode, 790
~EventHandler
Spinnaker::EventHandler, 799
~Exception
Spinnaker::Exception, 805
~FileProtocolAdapter
Spinnaker::GenApi::FileProtocolAdapter, 809
~FloatNode
Spinnaker::GenApi::FloatNode, 815
~FloatRegNode
Spinnaker::GenApi::FloatRegNode, 821
~ICameraBase
Spinnaker::ICameraBase, 848
~ICameraList
Spinnaker::ICameraList, 856
~IChunkData
Spinnaker::IChunkData, 860
~IDataStream
Spinnaker::IDataStream, 868
~IDevFileStreamBuf
Spinnaker::GenApi::IDevFileStreamBuf, 878
~IDeviceArrivalEventHandler
Spinnaker::IDeviceArrivalEventHandler, 880
~IDeviceEventHandler
Spinnaker::IDeviceEventHandler, 882
~IDeviceRemovalEventHandler
Spinnaker::IDeviceRemovalEventHandler, 884
~IImage
Spinnaker::IImage, 887
~IImageEventHandler
Spinnaker::IImageEventHandler, 898
~IImageStatistics
Spinnaker::IImageStatistics, 900
~IInterface
Spinnaker::IInterface, 904
~IInterfaceArrivalEventHandler
Spinnaker::IInterfaceArrivalEventHandler, 909
~IInterfaceEventHandler
Spinnaker::IInterfaceEventHandler, 911
~IInterfaceList
Spinnaker::IInterfaceList, 914
~IInterfaceRemovalEventHandler
Spinnaker::IInterfaceRemovalEventHandler, 917
~ILoggingEventHandler
Spinnaker::ILoggingEventHandler, 919
~ISystem
Spinnaker::ISystem, 1015
~ISystemEventHandler
Spinnaker::ISystemEventHandler, 1021
~Image
Spinnaker::Image, 924

~ImageEventHandler
Spinnaker::ImageEventHandler, 946
~ImageEventHandlerImpl
ImageEventHandlerImpl, 948
~ImagePtr
Spinnaker::ImagePtr, 951
~ImageStatistics
Spinnaker::ImageStatistics, 954
~InferenceBoundingBoxResult
Chunk Data Inference Class, 171
~IntRegNode
Spinnaker::GenApi::IntRegNode, 1012
~IntegerNode
Spinnaker::GenApi::IntegerNode, 983
~Interface
Spinnaker::Interface, 988
~InterfaceArrivalEventHandler
Spinnaker::InterfaceArrivalEventHandler, 994
~InterfaceEventHandler
Spinnaker::InterfaceEventHandler, 996
~InterfaceEventHandlerImpl
InterfaceEventHandlerImpl, 1000
~InterfaceList
Spinnaker::InterfaceList, 1003
~InterfaceRemovalEventHandler
Spinnaker::InterfaceRemovalEventHandler, 1009
~Lock
Spinnaker::GenICam::LockableObject::Lock, 1026
~LoggingEventData
Spinnaker::LoggingEventData, 1029
~LoggingEventHandler
Spinnaker::LoggingEventHandler, 1035
~Node
Spinnaker::GenApi::Node, 1044
~NodeMap
Spinnaker::GenApi::NodeMap, 1055
~ODevFileStreamBuf
Spinnaker::GenApi::ODevFileStreamBuf, 1065
~PortNode
Spinnaker::GenApi::PortNode, 1071
~PortRecorder
Spinnaker::GenApi::PortRecorder, 1076
~PortReplay
Spinnaker::GenApi::PortReplay, 1080
~RegisterNode
Spinnaker::GenApi::RegisterNode, 1085
~SpinVideo
Spinnaker::Video::SpinVideo, 1089
~StringNode
Spinnaker::GenApi::StringNode, 1094
~StringRegNode
Spinnaker::GenApi::StringRegNode, 1099
~System
Spinnaker::System, 1102
~SystemEventHandler
Spinnaker::SystemEventHandler, 1112
~SystemEventHandlerImpl
SystemEventHandlerImpl, 1114
~SystemPtr
Spinnaker::SystemPtr, 1118
~TransportLayerDevice
Spinnaker::TransportLayerDevice, 1123
~TransportLayerInterface
Spinnaker::TransportLayerInterface, 1133
~TransportLayerStream
Spinnaker::TransportLayerStream, 1144
~TransportLayerSystem
Spinnaker::TransportLayerSystem, 1152
~ValueNode
Spinnaker::GenApi::ValueNode, 1162
~double_autovector_t
Spinnaker::GenApi::double_autovector_t, 775
~gcstring
Spinnaker::GenICam::gcstring, 826
~int64_autovector_t
Spinnaker::GenApi::int64_autovector_t, 979

ADAPTERCONFIG_API
AdapterConfig.h, 1167
aPAUSEMACCtrlFramesReceived
Spinnaker::Camera, 508
aPAUSEMACCtrlFramesTransmitted
Spinnaker::Camera, 508
AVI Recorder Class, 44
DEPRECATED_CLASS, 44
AVIOption, 463
Spinnaker::Video::AVIOption, 463
AasRoiEnable
Spinnaker::Camera, 503
AasRoiHeight
Spinnaker::Camera, 503
AasRoiOffsetX
Spinnaker::Camera, 504
AasRoiOffsetY
Spinnaker::Camera, 504
AasRoiWidth
Spinnaker::Camera, 504
AcquireImages
Acquisition.cpp, 1339
AcquisitionMultipleThread.cpp, 1384
ActionCommand.cpp, 1386
BufferHandling.cpp, 1388
ChunkData.cpp, 1391
CounterAndTimer.cpp, 1392
DeviceEvents.cpp, 1395
Exposure.cpp, 1400
Exposure_QuickSpin.cpp, 1402
FileAccess_QuickSpin.cpp, 1403
GigEVisionPerformance.cpp, 1410
ImageEvents.cpp, 1418
ImageFormatControl.cpp, 1420
ImageFormatControl_QuickSpin.cpp, 1421
Inference.cpp, 1424
LogicBlock.cpp, 1430
LookupTable.cpp, 1432
Polarization.cpp, 1439
SaveToAvi.cpp, 1443

Sequencer.cpp, 1444
Trigger.cpp, 1451
Trigger_QuickSpin.cpp, 1453
Acquisition.cpp
AcquireImages, 1339
main, 1339
PrintDeviceInfo, 1339
RunSingleCamera, 1339
AcquisitionAbort
Spinnaker::Camera, 504
AcquisitionArm
Spinnaker::Camera, 505
AcquisitionBurstFrameCount
Spinnaker::Camera, 505
AcquisitionFrameCount
Spinnaker::Camera, 505
AcquisitionFrameRate
Spinnaker::Camera, 505
AcquisitionFrameRateEnable
Spinnaker::Camera, 505
AcquisitionLineRate
Spinnaker::Camera, 506
AcquisitionMode
Spinnaker::Camera, 506
AcquisitionModeEnums
CameraDefs Class, 82
AcquisitionMultipleCameraRecovery.cpp
cameraGrabInfoMap, 1384
ConfigureCamera, 1382
ConfigureUserSet1, 1382
GetDeviceSerial, 1383
globalCamList, 1384
main, 1383
PrintExampleStatistics, 1383
RefreshCameraList, 1383
ResetCameraUserSetToDefault, 1383
SleepyWrapper, 1383
AcquisitionMultipleThread.cpp
AcquireImages, 1384
main, 1384
PrintDeviceInfo, 1385
RunMultipleCameras, 1385
AcquisitionResultingFrameRate
Spinnaker::Camera, 506
AcquisitionStart
Spinnaker::Camera, 506
AcquisitionStatus
Spinnaker::Camera, 506
AcquisitionStatusSelector
Spinnaker::Camera, 506
AcquisitionStatusSelectorEnums
CameraDefs Class, 82
AcquisitionStop
Spinnaker::Camera, 507
ActionCommand
Spinnaker::TransportLayerInterface, 1134
ActionCommand.cpp
AcquireImages, 1386
ConfigureActionControl, 1386
ConfigureChunkData, 1386
ConfigureIEEE1588, 1386
ConfigureInterface, 1386
ConfigureOtherNodes, 1386
ConfigureTrigger, 1386
main, 1387
PrintDeviceInfo, 1387
RunMultipleCameras, 1387
SleepyWrapper, 1387
ActionCommandResult, 455
ActionCommandStatus
Spinnaker Definitions, 204
ActionDeviceKey
Spinnaker::Camera, 507
ActionGroupKey
Spinnaker::Camera, 507
ActionGroupMask
Spinnaker::Camera, 507
ActionQueueSize
Spinnaker::Camera, 507
ActionSelector
Spinnaker::Camera, 507
ActionUnconditionalMode
Spinnaker::Camera, 508
ActionUnconditionalModeEnums
CameraDefs Class, 83
AdapterConfig, 387
AdapterConfigErr, 388
AutoPopulateAdapterInfo, 388
AutoPopulateAdvancedProperties, 388
ConfigureAdapter, 388
GetAuto10GDesc, 389
GetAutoGigabitDesc, 389
GetAutoStartIp, 389
GetAutoSubnetMask, 389
GetAutoSubnetMaskLength, 389
GetConfigLogFileName, 389
GetEnumerationLogFileName, 389
GetMaxIpAddress, 390
GetMinIpAddress, 390
GetSubnetMaskLength, 390
IsOnSameSubnet, 390
IsValidIpAddress, 390
IsValidSubnetMask, 390
PopulateAdapterIpInfo, 390
RetrieveAllAdapters, 391
ValidateIpAddress, 391
AdapterConfig.h
ADAPTERCONFIG_API, 1167
AdapterConfig::AdapterInfo
adapterDescription, 458
adapterGUID, 458
AdapterInfo, 458
adapterMACAddress, 458
adapterName, 459
dhcpEnabled, 459
ipInfo, 459

jumboPacketValidValues, 459
jumboPackets, 459
jumboPacketsRegKey, 459
receiveBuffers, 459
receiveBuffersMax, 459
receiveBuffersMin, 460
receiveBuffersRegKey, 460
receiveBuffersStep, 460
transmitBuffers, 460
transmitBuffersMax, 460
transmitBuffersMin, 460
transmitBuffersRegKey, 460
transmitBuffersStep, 460
AdapterConfig::IpInfo
gateway, 1013
ipAddress, 1013
IpInfo, 1013
subnetLength, 1013
subnetMask, 1013
AdapterConfigErr
AdapterConfig, 388
AdapterConfigException, 456
AdapterConfigException, 456
ErrCode, 457
GetParamStr, 457
adapterDescription
AdapterConfig::AdapterInfo, 458
adapterGUID
AdapterConfig::AdapterInfo, 458
AdapterInfo, 457
AdapterConfig::AdapterInfo, 458
adapterMACAddress
AdapterConfig::AdapterInfo, 458
adapterName
AdapterConfig::AdapterInfo, 459
AdaptiveCompressionEnable
Spinnaker::Camera, 508
AdcBitDepth
Spinnaker::Camera, 508
AdcBitDepthEnums
CameraDefs Class, 83
AddInjectionData
Spinnaker::GenApi::CNodeMapFactory, 724
Address
IPort Interface, 331
AnnounceImage
Spinnaker::IDataStream, 869
Append
Spinnaker::CameraList, 642
Spinnaker::ICameraList, 856
Spinnaker::IInterfaceList, 914
Spinnaker::InterfaceList, 1004
Spinnaker::Video::SpinVideo, 1090
append
Spinnaker::GenICam::gcstring, 826
ApplyStyleSheet
Spinnaker::GenApi::CNodeMapFactory, 725
argBayerRG
GigEVisionPerformance.cpp, 1412
argDuration
GigEVisionPerformance.cpp, 1412
argMaxFrames
GigEVisionPerformance.cpp, 1412
argNumImages
GigEVisionPerformance.cpp, 1412
argPacketDelay
GigEVisionPerformance.cpp, 1412
argPacketSize
GigEVisionPerformance.cpp, 1412
argPrintUsage
GigEVisionPerformance.cpp, 1413
argRelease
GigEVisionPerformance.cpp, 1413
argUserSetFrames
GigEVisionPerformance.cpp, 1413
arrayLabelClassification
Inference.cpp, 1427
arrayLabelDetection
Inference.cpp, 1427
assign
Spinnaker::GenICam::gcstring, 826, 827
attach
Spinnaker::GenApi::FileProtocolAdapter, 809
AttachBuffer
Spinnaker::GenApi::CChunkAdapter, 650
Spinnaker::GenApi::CChunkAdapterDcam, 653
Spinnaker::GenApi::CChunkAdapterGEV, 658
Spinnaker::GenApi::CChunkAdapterGeneric, 656
Spinnaker::GenApi::CChunkAdapterU3V, 661
Spinnaker::IDataStream, 869
AttachChunk
Spinnaker::GenApi::CChunkPort, 664
AttachEvent
Spinnaker::GenApi::CEventPort, 685
AttachNode
Spinnaker::GenApi::CEventPort, 685
AttachNodeMap
Spinnaker::GenApi::CChunkAdapter, 650
Spinnaker::GenApi::CEventAdapter, 673
AttachPort
Spinnaker::GenApi::CChunkPort, 664
AttachStatistics_t, 461
NumAttachedChunks, 461
NumChunkPorts, 461
NumChunks, 461
AutoAlgorithmSelector
Spinnaker::Camera, 509
AutoAlgorithmSelectorEnums
CameraDefs Class, 83
AutoExposureControlLoopDamping
Spinnaker::Camera, 509
AutoExposureControlPriority
Spinnaker::Camera, 509
AutoExposureControlPriorityEnums
CameraDefs Class, 84
AutoExposureEVCompensation

Spinnaker::Camera, 509
AutoExposureExposureTimeLowerLimit
Spinnaker::Camera, 510
AutoExposureExposureTimeUpperLimit
Spinnaker::Camera, 510
AutoExposureGainLowerLimit
Spinnaker::Camera, 510
AutoExposureGainUpperLimit
Spinnaker::Camera, 510
AutoExposureGreyValueLowerLimit
Spinnaker::Camera, 511
AutoExposureGreyValueUpperLimit
Spinnaker::Camera, 511
AutoExposureLightingMode
Spinnaker::Camera, 511
AutoExposureLightingModeEnums
CameraDefs Class, 84
AutoExposureMeteringMode
Spinnaker::Camera, 511
AutoExposureMeteringModeEnums
CameraDefs Class, 84
AutoExposureTargetGreyValue
Spinnaker::Camera, 512
AutoExposureTargetGreyValueAuto
Spinnaker::Camera, 512
AutoExposureTargetGreyValueAutoEnums
CameraDefs Class, 85
AutoLock, 462
Spinnaker::GenApi::AutoLock, 463
Spinnaker::GenICam::AutoLock, 462
AutoPopulateAdapterInfo
AdapterConfig, 388
AutoPopulateAdvancedProperties
AdapterConfig, 388
AutoVector Class, 248
Automatic
Types Enums, 383

BMPOption, 468
Spinnaker::BMPOption, 469
BalanceRatio
Spinnaker::Camera, 512
BalanceRatioSelector
Spinnaker::Camera, 513
BalanceRatioSelectorEnums
CameraDefs Class, 85
BalanceWhiteAuto
Spinnaker::Camera, 513
BalanceWhiteAutoDamping
Spinnaker::Camera, 513
BalanceWhiteAutoEnums
CameraDefs Class, 86
BalanceWhiteAutoLowerLimit
Spinnaker::Camera, 513
BalanceWhiteAutoProfile
Spinnaker::Camera, 514
BalanceWhiteAutoProfileEnums
CameraDefs Class, 86
BalanceWhiteAutoUpperLimit
Spinnaker::Camera, 514
BasePtr
Spinnaker::BasePtr, 465
BasePtr Class, 47
operator==, 47
BasePtr< T, B >, 464
BeginAcquisition
Spinnaker::CameraBase, 628
Spinnaker::ICameraBase, 848
Beginner
Types Enums, 384
binaryFile
Spinnaker::PGMOption, 1067
Spinnaker::PPMOption, 1082
BinningHorizontal
Spinnaker::Camera, 514
BinningHorizontalMode
Spinnaker::Camera, 514
BinningHorizontalModeEnums
CameraDefs Class, 86
BinningSelector
Spinnaker::Camera, 515
BinningSelectorEnums
CameraDefs Class, 87
BinningVertical
Spinnaker::Camera, 515
BinningVerticalMode
Spinnaker::Camera, 515
BinningVerticalModeEnums
CameraDefs Class, 87
bitrate
Spinnaker::Video::H264Option, 845
BlackLevel
Spinnaker::Camera, 515
BlackLevelAuto
Spinnaker::Camera, 515
BlackLevelAutoBalance
Spinnaker::Camera, 516
BlackLevelAutoBalanceEnums
CameraDefs Class, 87
BlackLevelAutoEnums
CameraDefs Class, 88
BlackLevelClampingEnable
Spinnaker::Camera, 516
BlackLevelRaw
Spinnaker::Camera, 516
BlackLevelSelector
Spinnaker::Camera, 516
BlackLevelSelectorEnums
CameraDefs Class, 88
BlockId
GVCP_EVENT_ITEM_EXTENDED_ID, 837
GVCP_EVENT_ITEM, 835
BlockId64High
GVCP_EVENT_ITEM_EXTENDED_ID, 837
BlockId64Low
GVCP_EVENT_ITEM_EXTENDED_ID, 838
Boolean

Types Enums, 382
BooleanNode, 470
Spinnaker::GenApi::BooleanNode, 471
BooleanNode Class, 252
CBooleanRef, 252
bottomRightXCoord
Chunk Data Inference Class, 172
bottomRightYCoord
Chunk Data Inference Class, 172
boxType
Chunk Data Inference Class, 172
BufferHandling.cpp
AcquireImages, 1388
ConfigureTrigger, 1388
GrabNextImageByTrigger, 1389
k_numLoops, 1388
main, 1389
numBuffers, 1388
PrintDeviceInfo, 1389
ResetTrigger, 1389
RunSingleCamera, 1389
SleepyWrapper, 1389
z_numTriggers, 1388
BufferOwnership
Spinnaker Definitions, 205
build
Spinnaker::LibraryVersion, 1025

c_str
Spinnaker::GenICam::gcstring, 827
CBasePtr
Pointer Class, 355
CBooleanPtr
Pointer Class, 355
CBooleanRef
BooleanNode Class, 252
CCategoryPtr
Pointer Class, 356
CCategoryRef
CategoryNode Class, 253
CChunkAdapter, 649
Spinnaker::GenApi::CChunkAdapter, 650
CChunkAdapterDcam, 652
Spinnaker::GenApi::CChunkAdapterDcam, 653
CChunkAdapterGEV, 657
Spinnaker::GenApi::CChunkAdapterGEV, 658
CChunkAdapterGeneric, 655
Spinnaker::GenApi::CChunkAdapterGeneric, 655
CChunkAdapterU3V, 659
Spinnaker::GenApi::CChunkAdapterU3V, 660
CChunkPort, 662
Spinnaker::GenApi::CChunkPort, 663
CChunkPortPtr
Pointer Class, 356
CCommandPtr
Pointer Class, 356
CCommandRef
CommandNode Class, 259
CDeviceInfoPtr
Pointer Class, 356
CEnumEntryPtr
Pointer Class, 356
CEnumEntryRef
EnumEntryNode Class, 264
CEnumerationPtr
Pointer Class, 356
CEnumerationRef
EnumNode Class, 265
CEnumerationTRef
Spinnaker::GenApi::CEnumerationTRef, 669
CEnumerationTRef< EnumT >, 667
CEventAdapter, 672
Spinnaker::GenApi::CEventAdapter, 673
CEventAdapter1394, 674
Spinnaker::GenApi::CEventAdapter1394, 675
CEventAdapterGEV, 679
Spinnaker::GenApi::CEventAdapterGEV, 680
CEventAdapterGeneric, 676
Spinnaker::GenApi::CEventAdapterGeneric, 677
CEventAdapterU3V, 681
Spinnaker::GenApi::CEventAdapterU3V, 682
CEventPort, 683
Spinnaker::GenApi::CEventPort, 684
CFeatureBag, 688
Spinnaker::GenApi::CFeatureBag, 689
CFloatPtr, 691
Spinnaker::GenApi::CFloatPtr, 692
CFloatRef
FloatNode Class, 274
CGeneric_XMLLoaderParams, 693
CGlobalLock, 694
Spinnaker::GenICam::CGlobalLock, 694, 695
CGlobalLockUnlocker, 696
Spinnaker::GenICam::CGlobalLockUnlocker, 697
CHUNK_BASE_ADDRESS_REGISTER_LEN
IChunkPort Interface, 289
CHUNK_BASE_ADDRESS_REGISTER
IChunkPort Interface, 289
CHUNK_LENGTH_REGISTER_LEN
IChunkPort Interface, 290
CHUNK_LENGTH_REGISTER
IChunkPort Interface, 290
CIntegerPtr
Pointer Class, 357
CIntegerRef
IntegerNode Class, 329
CLock, 711, 712
Spinnaker::GenApi::CLock, 713
Spinnaker::GenICam::CLock, 711
CLockEx, 715, 716
CNodeCallback, 717
Spinnaker::GenApi::CNodeCallback, 718
CNodeMapDynPtr
Pointer Class, 357
CNodeMapFactory, 720
Spinnaker::GenApi::CNodeMapFactory, 722–724
CNodeMapFactory::NodeStatistics_t, 1061

CNodeMapPtr
Pointer Class, 357
CNodeMapRef, 729
Spinnaker GenApi Classes, 243
Spinnaker::GenApi::CNodeMapRef, 730
CNodeMapRefT< TCameraParams >, 731
CNodeMapRefT
Spinnaker GenApi Classes, 246
CNodePtr
Pointer Class, 357
CNodeRef
Spinnaker GenApi Classes, 243
COM_PORT_COUNT_MAX
SerialRxTx.cpp, 1446
COMMAND_MAGIC
Spinnaker::GenApi, 450
CPointer
Spinnaker::GenApi::CPointer, 743
CPointer< T, B >, 741
CPortConstructPtr
Pointer Class, 357
CPortImpl, 746
Spinnaker::GenApi::CPortImpl, 747
CPortPtr
Pointer Class, 357
CPortRecorderPtr
Pointer Class, 358
CPortRecorderRef
PortRecorder Class, 362
CPortRef
PortNode Class, 361
CPortReplayPtr
Pointer Class, 358
CPortWriteList, 750
Spinnaker::GenApi::CPortWriteList, 751
CPortWriteListPtr
Pointer Class, 358
CRCChecksum
DCAM_CHECKSUM, 764
CRegisterPortImpl, 753
Spinnaker::GenApi::CRegisterPortImpl, 754
CRegisterPtr
Pointer Class, 358
CRegisterRef
RegisterNode Class, 366
CSelectorPtr
Pointer Class, 358
CSelectorRef
Spinnaker GenApi Classes, 243
CSelectorSet, 757
Spinnaker::GenApi::CSelectorSet, 758
CStringPtr
Pointer Class, 358
CStringRef
StringNode Class, 370
CTestPortStruct
Spinnaker::GenApi::CTestPortStruct, 761
CTestPortStruct< CDataStruct >, 760
CValuePtr
Pointer Class, 359
CValueRef
ValueNode Class, 385
CacheChunkData
IChunkPort Interface, 290
Spinnaker::GenApi::PortNode, 1071
CalculateStatistics
Spinnaker::IImage, 887
Spinnaker::Image, 924
CallbackHandleType
Spinnaker GenApi Interfaces, 250
Camera, 473
Spinnaker::Camera, 503
Camera Base Class, 49
Camera Base Interface Class, 230
Camera Class, 48
Camera List Class, 165
CameraBase, 625
Spinnaker::CameraBase, 627, 628
Spinnaker::TransportLayerDevice, 1123
Spinnaker::TransportLayerStream, 1145
CameraCloseFile
Inference.cpp, 1424
CameraDefs Class, 50
AcquisitionModeEnums, 82
AcquisitionStatusSelectorEnums, 82
ActionUnconditionalModeEnums, 83
AdcBitDepthEnums, 83
AutoAlgorithmSelectorEnums, 83
AutoExposureControlPriorityEnums, 84
AutoExposureLightingModeEnums, 84
AutoExposureMeteringModeEnums, 84
AutoExposureTargetGreyValueAutoEnums, 85
BalanceRatioSelectorEnums, 85
BalanceWhiteAutoEnums, 86
BalanceWhiteAutoProfileEnums, 86
BinningHorizontalModeEnums, 86
BinningSelectorEnums, 87
BinningVerticalModeEnums, 87
BlackLevelAutoBalanceEnums, 87
BlackLevelAutoEnums, 88
BlackLevelSelectorEnums, 88
ChunkBlackLevelSelectorEnums, 88
ChunkCounterSelectorEnums, 88
ChunkEncoderSelectorEnums, 89
ChunkEncoderStatusEnums, 89
ChunkExposureTimeSelectorEnums, 89
ChunkGainSelectorEnums, 90
ChunkImageComponentEnums, 90
ChunkPixelFormatEnums, 91
ChunkRegionIDEnums, 91
ChunkScan3dCoordinateReferenceSelector↩
Enums, 91
ChunkScan3dCoordinateSelectorEnums, 92
ChunkScan3dCoordinateSystemEnums, 92
ChunkScan3dCoordinateSystemReferenceEnums,
92

ChunkScan3dCoordinateTransformSelector↩ Enums, 93
ChunkScan3dDistanceUnitEnums, 93
ChunkScan3dOutputModeEnums, 94
ChunkSelectorEnums, 94
ChunkSourceIDEnums, 95
ChunkTimerSelectorEnums, 95
ChunkTransferStreamIDEnums, 96
ClConfigurationEnums, 96
ClTimeSlotsCountEnums, 96
ColorTransformationSelectorEnums, 97
ColorTransformationValueSelectorEnums, 97
CounterEventActivationEnums, 98
CounterEventSourceEnums, 98
CounterResetActivationEnums, 99
CounterResetSourceEnums, 99
CounterSelectorEnums, 99
CounterStatusEnums, 100
CounterTriggerActivationEnums, 100
CounterTriggerSourceEnums, 100
CxpConnectionTestModeEnums, 101
CxpLinkConfigurationEnums, 101
CxpLinkConfigurationPreferredEnums, 102
CxpLinkConfigurationStatusEnums, 103
CxpPoCxpStatusEnums, 104
DecimationHorizontalModeEnums, 105
DecimationSelectorEnums, 105
DecimationVerticalModeEnums, 105
DefectCorrectionModeEnums, 105
DeinterlacingEnums, 106
DeviceCharacterSetEnums, 106
DeviceClockSelectorEnums, 106
DeviceConnectionStatusEnums, 107
DeviceIndicatorModeEnums, 107
DeviceLinkHeartbeatModeEnums, 107
DeviceLinkThroughputLimitModeEnums, 109
DevicePowerSupplySelectorEnums, 109
DeviceRegistersEndiannessEnums, 109
DeviceScanTypeEnums, 110
DeviceSerialPortBaudRateEnums, 110
DeviceSerialPortSelectorEnums, 110
DeviceStreamChannelEndiannessEnums, 110
DeviceStreamChannelTypeEnums, 111
DeviceTLTypeEnums, 113
DeviceTapGeometryEnums, 111
DeviceTemperatureSelectorEnums, 112
DeviceTypeEnums, 113
EncoderModeEnums, 113
EncoderOutputModeEnums, 114
EncoderResetActivationEnums, 114
EncoderResetSourceEnums, 115
EncoderSelectorEnums, 116
EncoderSourceAEnums, 116
EncoderSourceBEnums, 116
EncoderStatusEnums, 117
EventNotificationEnums, 117
EventSelectorEnums, 117
ExposureActiveModeEnums, 118
ExposureAutoEnums, 118
ExposureModeEnums, 118
ExposureTimeModeEnums, 119
ExposureTimeSelectorEnums, 119
FileOpenModeEnums, 120
FileOperationSelectorEnums, 120
FileOperationStatusEnums, 120
FileSelectorEnums, 121
GainAutoBalanceEnums, 121
GainAutoEnums, 121
GainSelectorEnums, 122
GevCCPEnums, 122
GevCurrentPhysicalLinkConfigurationEnums, 122
GevGVCPExtendedStatusCodesSelectorEnums, 122
GevGVSPExtendedIDModeEnums, 123
GevIEEE1588ClockAccuracyEnums, 123
GevIEEE1588ModeEnums, 123
GevIEEE1588StatusEnums, 124
GevIPConfigurationStatusEnums, 124
GevPhysicalLinkConfigurationEnums, 124
GevSupportedOptionSelectorEnums, 125
ImageComponentSelectorEnums, 126
ImageCompressionJPEGFormatOptionEnums, 126
ImageCompressionModeEnums, 127
ImageCompressionRateOptionEnums, 127
LUTSelectorEnums, 131
LineFormatEnums, 127
LineInputFilterSelectorEnums, 128
LineModeEnums, 128
LineSelectorEnums, 128
LineSourceEnums, 129
LogicBlockLUTInputActivationEnums, 129
LogicBlockLUTInputSelectorEnums, 130
LogicBlockLUTInputSourceEnums, 130
LogicBlockLUTSelectorEnums, 131
LogicBlockSelectorEnums, 131
PixelColorFilterEnums, 132
PixelFormatEnums, 132
PixelFormatInfoSelectorEnums, 138
PixelSizeEnums, 143
RegionDestinationEnums, 144
RegionModeEnums, 144
RegionSelectorEnums, 145
RgbTransformLightSourceEnums, 145
Scan3dCoordinateReferenceSelectorEnums, 145
Scan3dCoordinateSelectorEnums, 146
Scan3dCoordinateSystemEnums, 146
Scan3dCoordinateSystemReferenceEnums, 146
Scan3dCoordinateTransformSelectorEnums, 147
Scan3dDistanceUnitEnums, 147
Scan3dOutputModeEnums, 148
SensorDigitizationTapsEnums, 148
SensorShutterModeEnums, 149
SensorTapsEnums, 149
SequencerConfigurationModeEnums, 150
SequencerConfigurationValidEnums, 150

SequencerModeEnums, 150
SequencerSetValidEnums, 150
SequencerTriggerActivationEnums, 151
SequencerTriggerSourceEnums, 151
SerialPortBaudRateEnums, 151
SerialPortParityEnums, 152
SerialPortSelectorEnums, 152
SerialPortSourceEnums, 153
SerialPortStopBitsEnums, 153
SoftwareSignalSelectorEnums, 153
SourceSelectorEnums, 154
TestPatternEnums, 154
TestPatternGeneratorSelectorEnums, 154
TimerSelectorEnums, 155
TimerStatusEnums, 155
TimerTriggerActivationEnums, 155
TimerTriggerSourceEnums, 156
TransferComponentSelectorEnums, 157
TransferControlModeEnums, 157
TransferOperationModeEnums, 158
TransferQueueModeEnums, 158
TransferSelectorEnums, 158
TransferStatusSelectorEnums, 159
TransferTriggerActivationEnums, 159
TransferTriggerModeEnums, 159
TransferTriggerSelectorEnums, 160
TransferTriggerSourceEnums, 160
TriggerActivationEnums, 161
TriggerModeEnums, 162
TriggerOverlapEnums, 162
TriggerSelectorEnums, 162
TriggerSourceEnums, 162
UserOutputSelectorEnums, 163
UserSetDefaultEnums, 163
UserSetSelectorEnums, 164
WhiteClipSelectorEnums, 164
CameraDeleteFile
Inference.cpp, 1424
cameraGrabInfoMap
AcquisitionMultipleCameraRecovery.cpp, 1384
CameraInternal
Spinnaker::ICameraBase, 854
Spinnaker::TransportLayerDevice, 1123
Spinnaker::TransportLayerStream, 1145
CameraList, 640
Spinnaker::CameraList, 641
CameraListImpl
Spinnaker::ICameraList, 858
CameraOpenFile
Inference.cpp, 1424
CameraPtr, 645
CameraPtr Class, 166, 167
CameraPtr Class, 166
CameraPtr, 166, 167
CameraWriteToFile
Inference.cpp, 1424
CastToIDestroy
Spinnaker GenApi Classes, 246
CategoryNode, 647
Spinnaker::GenApi::CategoryNode, 648
CategoryNode Class, 253
CCategoryRef, 253
causeSpinnakerException
ExceptionHandling.cpp, 1399
causeStandardException
ExceptionHandling.cpp, 1399
centerXCoord
Chunk Data Inference Class, 172
centerYCoord
Chunk Data Inference Class, 172
ChangeHeightAndGain
NodeMapCallback.cpp, 1434
CheckBufferLayout
Spinnaker::GenApi::CChunkAdapter, 651
Spinnaker::GenApi::CChunkAdapterDcam, 654
Spinnaker::GenApi::CChunkAdapterGEV, 659
Spinnaker::GenApi::CChunkAdapterGeneric, 656
Spinnaker::GenApi::CChunkAdapterU3V, 661
CheckCRC
Spinnaker::GenApi::CChunkAdapterDcam, 654
Spinnaker::IImage, 887
Spinnaker::Image, 925
CheckChunkID
Spinnaker::GenApi::CChunkPort, 664
CheckEventID
Spinnaker::GenApi::CEventPort, 685
CheckGevEnabled
EnumerationEvents.cpp, 1398
CheckNodeAccessibility
HighDynamicRange.cpp, 1415
chosenChunkData
ChunkData.cpp, 1392
chosenEvent
DeviceEvents.cpp, 1396
chosenException
ExceptionHandling.cpp, 1399
chosenFileUploadPersistence
Inference.cpp, 1428
chosenInferenceNetworkType
Inference.cpp, 1428
chosenRead
NodeMapInfo.cpp, 1438
chosenTrigger
Trigger.cpp, 1452
Trigger_QuickSpin.cpp, 1454
chosenVideoType
SaveToAvi.cpp, 1444
Chunk Data Inference Class, 169
~InferenceBoundingBoxResult, 171
bottomRightXCoord, 172
bottomRightYCoord, 172
boxType, 172
centerXCoord, 172
centerYCoord, 172
circle, 173
classId, 173

confidence, 173
GetBoxAt, 170
GetBoxCount, 170
GetBoxSize, 170
GetVersion, 170
InferenceBoundingBoxResult, 171
operator=, 171
radius, 173
rect, 173
rotatedRect, 173
rotationAngle, 173
topLeftXCoord, 173, 174
topLeftYCoord, 174
ChunkAdapter Class, 254
ChunkAdapterDcam Class, 255
ChunkAdapterGEV Class, 257
ChunkAdapterGeneric Class, 256
ChunkAdapterU3V Class, 386
ChunkBlackLevel
Spinnaker::Camera, 516
ChunkBlackLevelSelector
Spinnaker::Camera, 517
ChunkBlackLevelSelectorEnums
CameraDefs Class, 88
ChunkCRC
Spinnaker::Camera, 517
ChunkCounterSelector
Spinnaker::Camera, 517
ChunkCounterSelectorEnums
CameraDefs Class, 88
ChunkCounterValue
Spinnaker::Camera, 517
ChunkData, 698
Spinnaker::ChunkData, 700
ChunkData Class, 168
ChunkData.cpp
AcquireImages, 1391
chosenChunkData, 1392
chunkDataType, 1390
ConfigureChunkData, 1391
DisableChunkData, 1391
DisplayChunkData, 1391
main, 1391
PrintDeviceInfo, 1391
RunSingleCamera, 1392
chunkDataType
ChunkData.cpp, 1390
ChunkEnable
Spinnaker::Camera, 517
ChunkEncoderSelector
Spinnaker::Camera, 517
ChunkEncoderSelectorEnums
CameraDefs Class, 89
ChunkEncoderStatus
Spinnaker::Camera, 518
ChunkEncoderStatusEnums
CameraDefs Class, 89
ChunkEncoderValue
Spinnaker::Camera, 518
ChunkExposureEndLineStatusAll
Spinnaker::Camera, 518
ChunkExposureTime
Spinnaker::Camera, 518
ChunkExposureTimeSelector
Spinnaker::Camera, 518
ChunkExposureTimeSelectorEnums
CameraDefs Class, 89
ChunkFrameID
Spinnaker::Camera, 518
ChunkGain
Spinnaker::Camera, 519
ChunkGainSelector
Spinnaker::Camera, 519
ChunkGainSelectorEnums
CameraDefs Class, 90
ChunkHeight
Spinnaker::Camera, 519
ChunkID
DCAM_CHUNK_TRAILER, 764
GVCP_CHUNK_TRAILER, 834
SingleChunkData_t, 1087
SingleChunkDataStr_t, 1087
U3V_CHUNK_TRAILER, 1157
ChunkImage
Spinnaker::Camera, 519
ChunkImageComponent
Spinnaker::Camera, 519
ChunkImageComponentEnums
CameraDefs Class, 90
ChunkInferenceBoundingBoxResult
Spinnaker::Camera, 519
ChunkInferenceConfidence
Spinnaker::Camera, 520
ChunkInferenceFrameId
Spinnaker::Camera, 520
ChunkInferenceResult
Spinnaker::Camera, 520
ChunkLength
DCAM_CHUNK_TRAILER, 764
GVCP_CHUNK_TRAILER, 834
SingleChunkData_t, 1087
SingleChunkDataStr_t, 1087
U3V_CHUNK_TRAILER, 1157
ChunkLinePitch
Spinnaker::Camera, 520
ChunkLineStatusAll
Spinnaker::Camera, 520
ChunkModeActive
Spinnaker::Camera, 520
ChunkOffset
SingleChunkData_t, 1087
SingleChunkDataStr_t, 1088
ChunkOffsetX
Spinnaker::Camera, 521
ChunkOffsetY
Spinnaker::Camera, 521

ChunkPartSelector
Spinnaker::Camera, 521
ChunkPixelDynamicRangeMax
Spinnaker::Camera, 521
ChunkPixelDynamicRangeMin
Spinnaker::Camera, 521
ChunkPixelFormat
Spinnaker::Camera, 521
ChunkPixelFormatEnums
CameraDefs Class, 91
ChunkPort Class, 258
ChunkRegionIDEnums
CameraDefs Class, 91
ChunkRegionID
Spinnaker::Camera, 522
ChunkScan3dAxisMax
Spinnaker::Camera, 522
ChunkScan3dAxisMin
Spinnaker::Camera, 522
ChunkScan3dCoordinateOffset
Spinnaker::Camera, 522
ChunkScan3dCoordinateReferenceSelector
Spinnaker::Camera, 522
ChunkScan3dCoordinateReferenceSelectorEnums
CameraDefs Class, 91
ChunkScan3dCoordinateReferenceValue
Spinnaker::Camera, 522
ChunkScan3dCoordinateScale
Spinnaker::Camera, 523
ChunkScan3dCoordinateSelector
Spinnaker::Camera, 523
ChunkScan3dCoordinateSelectorEnums
CameraDefs Class, 92
ChunkScan3dCoordinateSystem
Spinnaker::Camera, 523
ChunkScan3dCoordinateSystemEnums
CameraDefs Class, 92
ChunkScan3dCoordinateSystemReference
Spinnaker::Camera, 523
ChunkScan3dCoordinateSystemReferenceEnums
CameraDefs Class, 92
ChunkScan3dCoordinateTransformSelector
Spinnaker::Camera, 523
ChunkScan3dCoordinateTransformSelectorEnums
CameraDefs Class, 93
ChunkScan3dDistanceUnit
Spinnaker::Camera, 523
ChunkScan3dDistanceUnitEnums
CameraDefs Class, 93
ChunkScan3dInvalidDataFlag
Spinnaker::Camera, 524
ChunkScan3dInvalidDataValue
Spinnaker::Camera, 524
ChunkScan3dOutputMode
Spinnaker::Camera, 524
ChunkScan3dOutputModeEnums
CameraDefs Class, 94
ChunkScan3dTransformValue
Spinnaker::Camera, 524
ChunkScanLineSelector
Spinnaker::Camera, 524
ChunkSelector
Spinnaker::Camera, 524
ChunkSelectorEnums
CameraDefs Class, 94
ChunkSequencerSetActive
Spinnaker::Camera, 525
ChunkSerialData
Spinnaker::Camera, 525
ChunkSerialDataLength
Spinnaker::Camera, 525
ChunkSerialReceiveOverflow
Spinnaker::Camera, 525
ChunkSourceIDEnums
CameraDefs Class, 95
ChunkSourceID
Spinnaker::Camera, 525
ChunkStreamChannelID
Spinnaker::Camera, 525
ChunkTimerSelector
Spinnaker::Camera, 526
ChunkTimerSelectorEnums
CameraDefs Class, 95
ChunkTimerValue
Spinnaker::Camera, 526
ChunkTimestamp
Spinnaker::Camera, 526
ChunkTimestampLatchValue
Spinnaker::Camera, 526
ChunkTransferBlockID
Spinnaker::Camera, 526
ChunkTransferQueueCurrentBlockCount
Spinnaker::Camera, 526
ChunkTransferStreamIDEnums
CameraDefs Class, 96
ChunkTransferStreamID
Spinnaker::Camera, 527
ChunkWidth
Spinnaker::Camera, 527
circle
Chunk Data Inference Class, 173
CL
Types Enums, 383
ClConfiguration
Spinnaker::Camera, 527
ClConfigurationEnums
CameraDefs Class, 96
ClTimeSlotsCount
Spinnaker::Camera, 527
ClTimeSlotsCountEnums
CameraDefs Class, 96
classId
Chunk Data Inference Class, 173
CleanUp
SerialRxTx.cpp, 1448
CleanupChunkAdapter

Spinnaker::IDataStream, 869
Clear
Spinnaker::CameraList, 642
Spinnaker::ICameraList, 856
Spinnaker::IInterfaceList, 914
Spinnaker::InterfaceList, 1004
ClearCache
Spinnaker::GenApi::CChunkPort, 664
Spinnaker::GenApi::CNodeMapFactory, 725
ClearCaches
Spinnaker::GenApi::CChunkAdapter, 651
ClearXMLCache
Spinnaker::GenApi::NodeMap, 1055
Close
Spinnaker::Video::SpinVideo, 1090
close
Spinnaker::GenApi::IDevFileStreamBase, 876
Spinnaker::GenApi::IDevFileStreamBuf, 878
Spinnaker::GenApi::ODevFileStreamBase, 1063
Spinnaker::GenApi::ODevFileStreamBuf, 1065
CloseFile
FileAccess_QuickSpin.cpp, 1403
closeFile
Spinnaker::GenApi::FileProtocolAdapter, 810
ColorProcessingAlgorithm
Spinnaker Definitions, 205
ColorTransformationEnable
Spinnaker::Camera, 527
ColorTransformationSelector
Spinnaker::Camera, 528
ColorTransformationSelectorEnums
CameraDefs Class, 97
ColorTransformationValue
Spinnaker::Camera, 528
ColorTransformationValueSelector
Spinnaker::Camera, 528
ColorTransformationValueSelectorEnums
CameraDefs Class, 97
Combine
INode Interface, 312
Command
GVCP_REQUEST_HEADER, 843
CommandHeader
U3V_EVENT_MESSAGE, 1160
CommandId
U3V_COMMAND_HEADER, 1158
CommandNode, 737
Spinnaker::GenApi::CommandNode, 738
CommandNode Class, 259
CCommandRef, 259
compare
Spinnaker::GenICam::gcstring, 827
Compatibility.h
FMT_I64, 1246
compression
Spinnaker::TIFFOption, 1120
compressionLevel
Spinnaker::PNGOption, 1068
CompressionMethod
Spinnaker::TIFFOption, 1119
CompressionRatio
Spinnaker::Camera, 528
confidence
Chunk Data Inference Class, 173
ConfigureActionControl
ActionCommand.cpp, 1386
ConfigureAdapter
AdapterConfig, 388
ConfigureCallbacks
NodeMapCallback.cpp, 1434
ConfigureCamera
AcquisitionMultipleCameraRecovery.cpp, 1382
ConfigureChunkData
ActionCommand.cpp, 1386
ChunkData.cpp, 1391
Inference.cpp, 1425
ConfigureCustomImageSettings
ImageFormatControl.cpp, 1420
ImageFormatControl_QuickSpin.cpp, 1421
ConfigureDevice
SerialRxTx.cpp, 1448
ConfigureDeviceEvents
DeviceEvents.cpp, 1395
ConfigureDigitalIO
CounterAndTimer.cpp, 1393
ConfigureExposure
Exposure.cpp, 1400
Exposure_QuickSpin.cpp, 1402
ConfigureExposureandTrigger
CounterAndTimer.cpp, 1393
ConfigureIEEE1588
ActionCommand.cpp, 1386
ConfigureImageEvents
ImageEvents.cpp, 1418
ConfigureInference
Inference.cpp, 1425
ConfigureInterface
ActionCommand.cpp, 1386
ConfigureLogicBlock
LogicBlock.cpp, 1430
ConfigureLookupTables
LookupTable.cpp, 1432
ConfigureOtherNodes
ActionCommand.cpp, 1386
ConfigureSequencerPartOne
Sequencer.cpp, 1444
ConfigureSequencerPartTwo
Sequencer.cpp, 1445
ConfigureStream
Polarization.cpp, 1440
ConfigureTestPattern
Inference.cpp, 1425
ConfigureTrigger
ActionCommand.cpp, 1386
BufferHandling.cpp, 1388
Inference.cpp, 1425

LogicBlock.cpp, 1431
Trigger.cpp, 1451
Trigger_QuickSpin.cpp, 1453
ConfigureUserSet1
AcquisitionMultipleCameraRecovery.cpp, 1382
Connect
INodeMap Interface, 321, 322
Spinnaker::GenApi::NodeMap, 1055
Container Class, 260
Conversion, 391
NumToCString, 391
Convert
Spinnaker::IImage, 887, 888
Spinnaker::Image, 925, 926
Counter, 739
Spinnaker::GenApi::Counter, 740
Counter Class, 261
CounterAndTimer.cpp
AcquireImages, 1392
ConfigureDigitalIO, 1393
ConfigureExposureandTrigger, 1393
main, 1393
PrintDeviceInfo, 1393
ResetTrigger, 1393
RunSingleCamera, 1393
SetupCounterAndTimer, 1394
CounterDelay
Spinnaker::Camera, 528
CounterDuration
Spinnaker::Camera, 529
CounterEventActivation
Spinnaker::Camera, 529
CounterEventActivationEnums
CameraDefs Class, 98
CounterEventSource
Spinnaker::Camera, 529
CounterEventSourceEnums
CameraDefs Class, 98
CounterReset
Spinnaker::Camera, 529
CounterResetActivation
Spinnaker::Camera, 529
CounterResetActivationEnums
CameraDefs Class, 99
CounterResetSource
Spinnaker::Camera, 529
CounterResetSourceEnums
CameraDefs Class, 99
CounterSelector
Spinnaker::Camera, 530
CounterSelectorEnums
CameraDefs Class, 99
CounterStart
PerformanceCounter, 393
CounterStatus
Spinnaker::Camera, 530
CounterStatusEnums
CameraDefs Class, 100
CounterTriggerActivation
Spinnaker::Camera, 530
CounterTriggerActivationEnums
CameraDefs Class, 100
CounterTriggerSource
Spinnaker::Camera, 530
CounterTriggerSourceEnums
CameraDefs Class, 100
CounterValue
Spinnaker::Camera, 530
CounterValueAtReset
Spinnaker::Camera, 530
CpuUsageInfo, 753
cpuUsageInfo
GigEVisionPerformance.cpp, 1413
CpuUtil, 392
GetCpuStats, 392
StartCpuTracing, 392
StopCpuTracing, 392
CpuUtil::CpuUsageInfo
dummy, 753
Create
Spinnaker::Image, 926, 927
CreateAndSaveAolpDolpImages
Polarization.cpp, 1440
CreateAndSaveGlareReducedImage
Polarization.cpp, 1440
CreateAndSaveStokesImages
Polarization.cpp, 1440
CreateAolp
Spinnaker::ImageUtilityPolarization, 970
CreateDolp
Spinnaker::ImageUtilityPolarization, 970, 971
CreateEmptyNodeMap
Spinnaker::GenApi::CNodeMapFactory, 725
CreateGlareReduced
Spinnaker::ImageUtilityPolarization, 971, 972
CreateHeatmap
Spinnaker::ImageUtilityHeatmap, 965, 966
CreateHeatmapImages
Polarization.cpp, 1440
CreateNodeDataFromNodeMap
Spinnaker::GenApi::CNodeMapFactory, 725
CreateNodeMap
Spinnaker::GenApi::CNodeMapFactory, 725
CreateNormalized
Spinnaker::ImageUtility, 961–963
CreateNormalizedImage
Polarization.cpp, 1440
CreateScaled
Spinnaker::ImageUtility, 963
CreateShared
Spinnaker::Image, 927
CreateStokesS0
Spinnaker::ImageUtilityPolarization, 972
CreateStokesS1
Spinnaker::ImageUtilityPolarization, 973
CreateStokesS2

Spinnaker::ImageUtilityPolarization, 974
Custom
Types Enums, 382
CxpConnectionSelector
Spinnaker::Camera, 531
CxpConnectionTestErrorCount
Spinnaker::Camera, 531
CxpConnectionTestMode
Spinnaker::Camera, 531
CxpConnectionTestModeEnums
CameraDefs Class, 101
CxpConnectionTestPacketCount
Spinnaker::Camera, 531
CxpLinkConfiguration
Spinnaker::Camera, 531
CxpLinkConfigurationEnums
CameraDefs Class, 101
CxpLinkConfigurationPreferred
Spinnaker::Camera, 531
CxpLinkConfigurationPreferredEnums
CameraDefs Class, 102
CxpLinkConfigurationStatus
Spinnaker::Camera, 532
CxpLinkConfigurationStatusEnums
CameraDefs Class, 103
CxpPoCxpAuto
Spinnaker::Camera, 532
CxpPoCxpStatus
Spinnaker::Camera, 532
CxpPoCxpStatusEnums
CameraDefs Class, 104
CxpPoCxpTripReset
Spinnaker::Camera, 532
CxpPoCxpTurnOff
Spinnaker::Camera, 532

DATA_BITS
SerialRxTx.cpp, 1447
DCAM_CHECKSUM, 763
CRCChecksum, 764
DCAM_CHUNK_TRAILER, 764
ChunkID, 764
ChunkLength, 764
InverseChunkLength, 764
DEPRECATED_CLASS
AVI Recorder Class, 44
Data
GVCP_EVENTDATA_REQUEST_EXTENDED_↩
ID, 842
GVCP_EVENTDATA_REQUEST, 841
DeInit
Spinnaker::CameraBase, 628
Spinnaker::ICameraBase, 848
DecimationHorizontal
Spinnaker::Camera, 532
DecimationHorizontalMode
Spinnaker::Camera, 533
DecimationHorizontalModeEnums
CameraDefs Class, 105
DecimationSelector
Spinnaker::Camera, 533
DecimationSelectorEnums
CameraDefs Class, 105
DecimationVertical
Spinnaker::Camera, 533
DecimationVerticalMode
Spinnaker::Camera, 533
DecimationVerticalModeEnums
CameraDefs Class, 105
Decreasing
Types Enums, 383
DeepCopy
Spinnaker::IImage, 888
Spinnaker::Image, 927, 928
DefectCorrectStaticEnable
Spinnaker::Camera, 534
DefectCorrectionMode
Spinnaker::Camera, 534
DefectCorrectionModeEnums
CameraDefs Class, 105
DefectTableApply
Spinnaker::Camera, 534
DefectTableCoordinateX
Spinnaker::Camera, 534
DefectTableCoordinateY
Spinnaker::Camera, 534
DefectTableFactoryRestore
Spinnaker::Camera, 535
DefectTableIndex
Spinnaker::Camera, 535
DefectTablePixelCount
Spinnaker::Camera, 535
DefectTableSave
Spinnaker::Camera, 535
Deinterlacing
Spinnaker::Camera, 536
DeinterlacingEnums
CameraDefs Class, 106
deleteFile
Spinnaker::GenApi::FileProtocolAdapter, 810
DeleteFileOnCamera
Inference.cpp, 1425
DeliverEventMessage
Spinnaker::GenApi::CEventAdapter1394, 676
Spinnaker::GenApi::CEventAdapterGEV, 680
Spinnaker::GenApi::CEventAdapterU3V, 682
DeliverMessage
Spinnaker::GenApi::CEventAdapter, 673
Spinnaker::GenApi::CEventAdapter1394, 676
Spinnaker::GenApi::CEventAdapterGEV, 680
Spinnaker::GenApi::CEventAdapterGeneric, 678
Spinnaker::GenApi::CEventAdapterU3V, 682
Deregister
NodeCallback Class, 347
DeregisterCallback
INode Interface, 312
Spinnaker::GenApi::Node, 1044

Destroy
Spinnaker::GenApi::CNodeCallback, 718
Spinnaker::GenApi::Function_NodeCallback, 823
Spinnaker::GenApi::Member_NodeCallback, 1039
Spinnaker::GenApi::NodeMap, 1055
DetachBuffer
Spinnaker::GenApi::CChunkAdapter, 651
DetachChunk
Spinnaker::GenApi::CChunkPort, 665
DetachEvent
Spinnaker::GenApi::CEventPort, 686
DetachNode
Spinnaker::GenApi::CEventPort, 686
DetachNodeMap
Spinnaker::GenApi::CChunkAdapter, 651
Spinnaker::GenApi::CEventAdapter, 674
DetachPort
Spinnaker::GenApi::CChunkPort, 665
DeviceAccessStatus
Spinnaker::TransportLayerDevice, 1124
Spinnaker::TransportLayerInterface, 1134
DeviceAccessStatusEnum
TransportLayerDefs Class, 219
DeviceAddress
Spinnaker::ActionCommandResult, 455
DeviceArrivalEventHandler, 765
Spinnaker::DeviceArrivalEventHandler, 766
DeviceArrivalEventHandler Class, 177
DeviceCharacterSet
Spinnaker::Camera, 536
DeviceCharacterSetEnums
CameraDefs Class, 106
DeviceClockFrequency
Spinnaker::Camera, 536
DeviceClockSelector
Spinnaker::Camera, 536
DeviceClockSelectorEnums
CameraDefs Class, 106
DeviceConnectionSelector
Spinnaker::Camera, 536
DeviceConnectionSpeed
Spinnaker::Camera, 537
DeviceConnectionStatus
Spinnaker::Camera, 537
DeviceConnectionStatusEnums
CameraDefs Class, 107
DeviceCount
Spinnaker::TransportLayerInterface, 1135
DeviceCurrentSpeed
Spinnaker::TransportLayerDevice, 1124
DeviceCurrentSpeedEnum
TransportLayerDefs Class, 220
DeviceDisplayName
Spinnaker::TransportLayerDevice, 1124
DeviceDriverVersion
Spinnaker::TransportLayerDevice, 1124
DeviceEndianessMechanism
Spinnaker::TransportLayerDevice, 1124
DeviceEndianessMechanismEnum
TransportLayerDefs Class, 220
DeviceEventChannelCount
Spinnaker::Camera, 537
DeviceEventHandler, 767
Spinnaker::DeviceEventHandler, 768
DeviceEventHandler Class, 178
DeviceEventHandlerImpl, 770
~DeviceEventHandlerImpl, 771
DeviceEventHandlerImpl, 771
OnDeviceEvent, 772
DeviceEvents.cpp
AcquireImages, 1395
chosenEvent, 1396
ConfigureDeviceEvents, 1395
eventType, 1394
main, 1395
PrintDeviceInfo, 1395
ResetDeviceEvents, 1395
RunSingleCamera, 1395
DeviceFamilyName
Spinnaker::Camera, 537
DeviceFeaturePersistenceEnd
Spinnaker::Camera, 537
DeviceFeaturePersistenceStart
Spinnaker::Camera, 537
DeviceFirmwareVersion
Spinnaker::Camera, 538
DeviceGenCPVersionMajor
Spinnaker::Camera, 538
DeviceGenCPVersionMinor
Spinnaker::Camera, 538
DeviceID
Spinnaker::Camera, 538
Spinnaker::TransportLayerDevice, 1125
Spinnaker::TransportLayerInterface, 1135
DeviceIndicatorMode
Spinnaker::Camera, 538
DeviceIndicatorModeEnums
CameraDefs Class, 107
DeviceInstanceId
Spinnaker::TransportLayerDevice, 1125
DeviceIsUpdater
Spinnaker::TransportLayerDevice, 1125
DeviceLinkBandwidthReserve
Spinnaker::Camera, 538
DeviceLinkCommandTimeout
Spinnaker::Camera, 539
DeviceLinkConnectionCount
Spinnaker::Camera, 539
DeviceLinkCurrentThroughput
Spinnaker::Camera, 539
DeviceLinkHeartbeatMode
Spinnaker::Camera, 539
DeviceLinkHeartbeatModeEnums
CameraDefs Class, 107
DeviceLinkHeartbeatTimeout
Spinnaker::Camera, 539

DeviceLinkSelector
Spinnaker::Camera, 539
DeviceLinkSpeed
Spinnaker::Camera, 540
Spinnaker::TransportLayerDevice, 1125
DeviceLinkThroughputLimit
Spinnaker::Camera, 540
DeviceLinkThroughputLimitMode
Spinnaker::Camera, 540
DeviceLinkThroughputLimitModeEnums
CameraDefs Class, 109
DeviceLocation
Spinnaker::TransportLayerDevice, 1125
DeviceManifestEntrySelector
Spinnaker::Camera, 540
DeviceManifestPrimaryURL
Spinnaker::Camera, 541
DeviceManifestSchemaMajorVersion
Spinnaker::Camera, 541
DeviceManifestSchemaMinorVersion
Spinnaker::Camera, 541
DeviceManifestSecondaryURL
Spinnaker::Camera, 541
DeviceManifestXMLMajorVersion
Spinnaker::Camera, 541
DeviceManifestXMLMinorVersion
Spinnaker::Camera, 541
DeviceManifestXMLSubMinorVersion
Spinnaker::Camera, 542
DeviceManufacturerInfo
Spinnaker::Camera, 542
DeviceMaxThroughput
Spinnaker::Camera, 542
DeviceModelName
Spinnaker::Camera, 542
Spinnaker::TransportLayerDevice, 1125
Spinnaker::TransportLayerInterface, 1135
DeviceMulticastMonitorMode
Spinnaker::TransportLayerDevice, 1126
DevicePowerSupplySelector
Spinnaker::Camera, 542
DevicePowerSupplySelectorEnums
CameraDefs Class, 109
DeviceRegistersCheck
Spinnaker::Camera, 543
DeviceRegistersEndianness
Spinnaker::Camera, 543
DeviceRegistersEndiannessEnums
CameraDefs Class, 109
DeviceRegistersStreamingEnd
Spinnaker::Camera, 543
DeviceRegistersStreamingStart
Spinnaker::Camera, 543
DeviceRegistersValid
Spinnaker::Camera, 543
DeviceRemovalEventHandler, 772
Spinnaker::DeviceRemovalEventHandler, 773
DeviceRemovalEventHandler Class, 179
DeviceReset
Spinnaker::Camera, 543
DeviceSFNCVersionMajor
Spinnaker::Camera, 544
DeviceSFNCVersionMinor
Spinnaker::Camera, 545
DeviceSFNCVersionSubMinor
Spinnaker::Camera, 545
DeviceScanType
Spinnaker::Camera, 544
DeviceScanTypeEnums
CameraDefs Class, 110
DeviceSelector
Spinnaker::TransportLayerInterface, 1135
DeviceSerialNumber
Spinnaker::Camera, 544
Spinnaker::TransportLayerDevice, 1126
Spinnaker::TransportLayerInterface, 1135
DeviceSerialPortBaudRate
Spinnaker::Camera, 544
DeviceSerialPortBaudRateEnums
CameraDefs Class, 110
DeviceSerialPortSelector
Spinnaker::Camera, 544
DeviceSerialPortSelectorEnums
CameraDefs Class, 110
DeviceStreamChannelCount
Spinnaker::Camera, 545
DeviceStreamChannelEndianness
Spinnaker::Camera, 545
DeviceStreamChannelEndiannessEnums
CameraDefs Class, 110
DeviceStreamChannelLink
Spinnaker::Camera, 545
DeviceStreamChannelPacketSize
Spinnaker::Camera, 546
DeviceStreamChannelSelector
Spinnaker::Camera, 546
DeviceStreamChannelType
Spinnaker::Camera, 546
DeviceStreamChannelTypeEnums
CameraDefs Class, 111
DeviceTLType
Spinnaker::Camera, 547
DeviceTLTypeEnums
CameraDefs Class, 113
DeviceTLVersionMajor
Spinnaker::Camera, 547
DeviceTLVersionMinor
Spinnaker::Camera, 547
DeviceTLVersionSubMinor
Spinnaker::Camera, 547
DeviceTapGeometry
Spinnaker::Camera, 546
DeviceTapGeometryEnums
CameraDefs Class, 111
DeviceTemperature
Spinnaker::Camera, 546

DeviceTemperatureSelector
Spinnaker::Camera, 546
DeviceTemperatureSelectorEnums
CameraDefs Class, 112
DeviceType
Spinnaker::Camera, 547
Spinnaker::TransportLayerDevice, 1126
DeviceTypeEnum
TransportLayerDefs Class, 221
DeviceTypeEnums
CameraDefs Class, 113
DeviceU3VProtocol
Spinnaker::TransportLayerDevice, 1126
DeviceUnlock
Spinnaker::TransportLayerInterface, 1135
DeviceUpdateList
Spinnaker::TransportLayerInterface, 1136
DeviceUptime
Spinnaker::Camera, 548
DeviceUserID
Spinnaker::Camera, 548
Spinnaker::TransportLayerDevice, 1126
DeviceVendorName
Spinnaker::Camera, 548
Spinnaker::TransportLayerDevice, 1126
Spinnaker::TransportLayerInterface, 1136
DeviceVersion
Spinnaker::Camera, 548
Spinnaker::TransportLayerDevice, 1127
dhcpEnabled
AdapterConfig::AdapterInfo, 459
DisableAll
Spinnaker::IImageStatistics, 900
Spinnaker::ImageStatistics, 954
DisableChunkData
ChunkData.cpp, 1391
Inference.cpp, 1425
DisableTrigger
Inference.cpp, 1426
DiscoverMaxPacketSize
Spinnaker::CameraBase, 628
Spinnaker::ICameraBase, 849
DisplayChunkData
ChunkData.cpp, 1391
Inference.cpp, 1426
doc/spindocs/C++/GettingStarted.dox, 1165
doc/spindocs/C++/ProgrammerGuide.dox, 1165
doc/spindocs/shared/Benefits.dox, 1165
doc/spindocs/shared/FlyCapture2Comparison.dox, 1165
doc/spindocs/shared/GenICamGenTL.dox, 1165
doc/spindocs/shared/Licensing.dox, 1165
DoesEnvironmentVariableExist
GCUtilities Utility, 281
double_autovector_t, 774
Spinnaker::GenApi::double_autovector_t, 775
DownloadImage
FileAccess_QuickSpin.cpp, 1404
dummy
CpuUtil::CpuUsageInfo, 753

EAccessMode
Types Enums, 378
EAccessModeClass, 777
ECacheUsage_t
NodeMapFactory Class, 350
ECachingMode
Types Enums, 378
ECachingModeClass, 778
ECallbackType
NodeCallback Class, 347
EContentType_t
NodeMapFactory Class, 351
EDisplayNotation
Types Enums, 378
EDisplayNotationClass, 779
EEndianess
Types Enums, 380
EEndianessClass, 780
EGenApiSchemaVersion
Types Enums, 380
EGenApiSchemaVersionClass, 782
EIncMode
Types Enums, 380
EInputDirection
Types Enums, 381
EInputDirectionClass, 783
EInterfaceType
Types Enums, 381
ELinkType
Types Enums, 381
ENameSpace
Types Enums, 382
ENameSpaceClass, 784
ERepresentation
Types Enums, 382
ERepresentationClass, 793
ESign
Types Enums, 382
ESignClass, 794
ESlope
Types Enums, 383
ESlopeClass, 795
EStandardNameSpace
Types Enums, 383
EStandardNameSpaceClass, 796
EVENT_TIMEOUT_INFINITE
Spinnaker Headers, 199
EVENT_TIMEOUT_NONE
Spinnaker Headers, 199
EVisibility
Types Enums, 383
EVisibilityClass, 801
EXMLValidation
Types Enums, 384
EXPAND_TO_STRINGISE
GCUtilities.h, 1268

EYesNo
Types Enums, 384
EYesNoClass, 807
EatComments
Spinnaker GenApi Classes, 246
empty
Spinnaker::GenICam::gcstring, 827
EnableAll
Spinnaker::IImageStatistics, 900
Spinnaker::ImageStatistics, 954
EnableGreyOnly
Spinnaker::IImageStatistics, 900
Spinnaker::ImageStatistics, 954
EnableHSLOnly
Spinnaker::IImageStatistics, 900
Spinnaker::ImageStatistics, 955
EnableManualFramerate
GigEVisionPerformance.cpp, 1410
EnableRGBOnly
Spinnaker::IImageStatistics, 901
Spinnaker::ImageStatistics, 955
EncoderDivider
Spinnaker::Camera, 548
EncoderMode
Spinnaker::Camera, 548
EncoderModeEnums
CameraDefs Class, 113
EncoderOutputMode
Spinnaker::Camera, 549
EncoderOutputModeEnums
CameraDefs Class, 114
EncoderReset
Spinnaker::Camera, 549
EncoderResetActivation
Spinnaker::Camera, 549
EncoderResetActivationEnums
CameraDefs Class, 114
EncoderResetSource
Spinnaker::Camera, 549
EncoderResetSourceEnums
CameraDefs Class, 115
EncoderSelector
Spinnaker::Camera, 549
EncoderSelectorEnums
CameraDefs Class, 116
EncoderSourceAEnums
CameraDefs Class, 116
EncoderSourceBEnums
CameraDefs Class, 116
EncoderSourceA
Spinnaker::Camera, 549
EncoderSourceB
Spinnaker::Camera, 550
EncoderStatus
Spinnaker::Camera, 550
EncoderStatusEnums
CameraDefs Class, 117
EncoderTimeout
Spinnaker::Camera, 550
EncoderValue
Spinnaker::Camera, 550
EncoderValueAtReset
Spinnaker::Camera, 550
EndAcquisition
Spinnaker::CameraBase, 629
Spinnaker::ICameraBase, 849
endTime
SecondsCounter, 394
EnumClasses Class, 262
EnumEntryNode, 785
Spinnaker::GenApi::EnumEntryNode, 786
EnumEntryNode Class, 264
CEnumEntryRef, 264
EnumNode, 788
Spinnaker::GenApi::EnumNode, 790
EnumNode Class, 265
CEnumerationRef, 265
EnumNodeT Class, 266
EnumerateGEVInterfaces
Spinnaker::TransportLayerSystem, 1152
Enumeration.cpp
main, 1396
QueryInterface, 1396
Enumeration_QuickSpin.cpp
main, 1397
QueryInterface, 1397
EnumerationCount
Spinnaker::Camera, 550
EnumerationEvents.cpp
CheckGevEnabled, 1398
main, 1398
ErrCode
AdapterConfigException, 457
Error
Spinnaker Definitions, 205
Event
GVCP_EVENTDATA_REQUEST_EXTENDED_↩
ID, 842
GVCP_EVENTDATA_REQUEST, 841
EventAcquisitionEnd
Spinnaker::Camera, 551
EventAcquisitionEndFrameID
Spinnaker::Camera, 551
EventAcquisitionEndTimestamp
Spinnaker::Camera, 551
EventAcquisitionError
Spinnaker::Camera, 551
EventAcquisitionErrorFrameID
Spinnaker::Camera, 551
EventAcquisitionErrorTimestamp
Spinnaker::Camera, 551
EventAcquisitionStart
Spinnaker::Camera, 552
EventAcquisitionStartFrameID
Spinnaker::Camera, 552
EventAcquisitionStartTimestamp

Spinnaker::Camera, 552
EventAcquisitionTransferEnd
Spinnaker::Camera, 552
EventAcquisitionTransferEndFrameID
Spinnaker::Camera, 552
EventAcquisitionTransferEndTimestamp
Spinnaker::Camera, 552
EventAcquisitionTransferStart
Spinnaker::Camera, 553
EventAcquisitionTransferStartFrameID
Spinnaker::Camera, 553
EventAcquisitionTransferStartTimestamp
Spinnaker::Camera, 553
EventAcquisitionTrigger
Spinnaker::Camera, 553
EventAcquisitionTriggerFrameID
Spinnaker::Camera, 553
EventAcquisitionTriggerTimestamp
Spinnaker::Camera, 553
EventActionLate
Spinnaker::Camera, 554
EventActionLateFrameID
Spinnaker::Camera, 554
EventActionLateTimestamp
Spinnaker::Camera, 554
EventAdapter Class, 267
EventAdapter1394 Class, 268
EventAdapterGEV Class, 270
EventAdapterGeneric Class, 269
EventAdapterU3V Class, 271
EventCounter0End
Spinnaker::Camera, 554
EventCounter0EndFrameID
Spinnaker::Camera, 554
EventCounter0EndTimestamp
Spinnaker::Camera, 554
EventCounter0Start
Spinnaker::Camera, 555
EventCounter0StartFrameID
Spinnaker::Camera, 555
EventCounter0StartTimestamp
Spinnaker::Camera, 555
EventCounter1End
Spinnaker::Camera, 555
EventCounter1EndFrameID
Spinnaker::Camera, 555
EventCounter1EndTimestamp
Spinnaker::Camera, 555
EventCounter1Start
Spinnaker::Camera, 556
EventCounter1StartFrameID
Spinnaker::Camera, 556
EventCounter1StartTimestamp
Spinnaker::Camera, 556
EventData
U3V_EVENT_MESSAGE, 1160
EventEncoder0Restarted
Spinnaker::Camera, 556
EventEncoder0RestartedFrameID
Spinnaker::Camera, 556
EventEncoder0RestartedTimestamp
Spinnaker::Camera, 556
EventEncoder0Stopped
Spinnaker::Camera, 557
EventEncoder0StoppedFrameID
Spinnaker::Camera, 557
EventEncoder0StoppedTimestamp
Spinnaker::Camera, 557
EventEncoder1Restarted
Spinnaker::Camera, 557
EventEncoder1RestartedFrameID
Spinnaker::Camera, 557
EventEncoder1RestartedTimestamp
Spinnaker::Camera, 557
EventEncoder1Stopped
Spinnaker::Camera, 558
EventEncoder1StoppedFrameID
Spinnaker::Camera, 558
EventEncoder1StoppedTimestamp
Spinnaker::Camera, 558
EventError
Spinnaker::Camera, 558
EventErrorCode
Spinnaker::Camera, 558
EventErrorFrameID
Spinnaker::Camera, 558
EventErrorTimestamp
Spinnaker::Camera, 559
EventExposureEnd
Spinnaker::Camera, 559
EventExposureEndFrameID
Spinnaker::Camera, 559
EventExposureEndTimestamp
Spinnaker::Camera, 559
EventExposureStart
Spinnaker::Camera, 559
EventExposureStartFrameID
Spinnaker::Camera, 559
EventExposureStartTimestamp
Spinnaker::Camera, 560
EventFrameBurstEnd
Spinnaker::Camera, 560
EventFrameBurstEndFrameID
Spinnaker::Camera, 560
EventFrameBurstEndTimestamp
Spinnaker::Camera, 560
EventFrameBurstStart
Spinnaker::Camera, 560
EventFrameBurstStartFrameID
Spinnaker::Camera, 560
EventFrameBurstStartTimestamp
Spinnaker::Camera, 561
EventFrameEnd
Spinnaker::Camera, 561
EventFrameEndFrameID
Spinnaker::Camera, 561

EventFrameEndTimestamp
Spinnaker::Camera, 561
EventFrameStart
Spinnaker::Camera, 561
EventFrameStartFrameID
Spinnaker::Camera, 561
EventFrameStartTimestamp
Spinnaker::Camera, 562
EventFrameTransferEnd
Spinnaker::Camera, 562
EventFrameTransferEndFrameID
Spinnaker::Camera, 562
EventFrameTransferEndTimestamp
Spinnaker::Camera, 562
EventFrameTransferStart
Spinnaker::Camera, 562
EventFrameTransferStartFrameID
Spinnaker::Camera, 562
EventFrameTransferStartTimestamp
Spinnaker::Camera, 563
EventFrameTrigger
Spinnaker::Camera, 563
EventFrameTriggerFrameID
Spinnaker::Camera, 563
EventFrameTriggerTimestamp
Spinnaker::Camera, 563
EventHandler, 798
Spinnaker::EventHandler, 799
EventHandler Class, 180
EventId
GVCP_EVENT_ITEM_BASIC, 836
GVCP_EVENT_ITEM_EXTENDED_ID, 838
GVCP_EVENT_ITEM, 835
U3V_EVENT_DATA, 1159
EventLine0AnyEdge
Spinnaker::Camera, 563
EventLine0AnyEdgeFrameID
Spinnaker::Camera, 563
EventLine0AnyEdgeTimestamp
Spinnaker::Camera, 564
EventLine0FallingEdge
Spinnaker::Camera, 564
EventLine0FallingEdgeFrameID
Spinnaker::Camera, 564
EventLine0FallingEdgeTimestamp
Spinnaker::Camera, 564
EventLine0RisingEdge
Spinnaker::Camera, 564
EventLine0RisingEdgeFrameID
Spinnaker::Camera, 564
EventLine0RisingEdgeTimestamp
Spinnaker::Camera, 565
EventLine1AnyEdge
Spinnaker::Camera, 565
EventLine1AnyEdgeFrameID
Spinnaker::Camera, 565
EventLine1AnyEdgeTimestamp
Spinnaker::Camera, 565
EventLine1FallingEdge
Spinnaker::Camera, 565
EventLine1FallingEdgeFrameID
Spinnaker::Camera, 565
EventLine1FallingEdgeTimestamp
Spinnaker::Camera, 566
EventLine1RisingEdge
Spinnaker::Camera, 566
EventLine1RisingEdgeFrameID
Spinnaker::Camera, 566
EventLine1RisingEdgeTimestamp
Spinnaker::Camera, 566
EventLinkSpeedChange
Spinnaker::Camera, 566
EventLinkSpeedChangeFrameID
Spinnaker::Camera, 566
EventLinkSpeedChangeTimestamp
Spinnaker::Camera, 567
EventLinkTrigger0
Spinnaker::Camera, 567
EventLinkTrigger0FrameID
Spinnaker::Camera, 567
EventLinkTrigger0Timestamp
Spinnaker::Camera, 567
EventLinkTrigger1
Spinnaker::Camera, 567
EventLinkTrigger1FrameID
Spinnaker::Camera, 567
EventLinkTrigger1Timestamp
Spinnaker::Camera, 568
EventNotification
Spinnaker::Camera, 568
EventNotificationEnums
CameraDefs Class, 117
EventPort Class, 272
EventProcessor
Spinnaker::EventHandler, 800
EventSelector
Spinnaker::Camera, 568
EventSelectorEnums
CameraDefs Class, 117
EventSequencerSetChange
Spinnaker::Camera, 568
EventSequencerSetChangeFrameID
Spinnaker::Camera, 568
EventSequencerSetChangeTimestamp
Spinnaker::Camera, 568
EventSerialData
Spinnaker::Camera, 569
EventSerialDataLength
Spinnaker::Camera, 569
EventSerialPortReceive
Spinnaker::Camera, 569
EventSerialPortReceiveTimestamp
Spinnaker::Camera, 569
EventSerialReceiveOverflow
Spinnaker::Camera, 569
EventStream0TransferBlockEnd

Spinnaker::Camera, 569
EventStream0TransferBlockEndFrameID
Spinnaker::Camera, 570
EventStream0TransferBlockEndTimestamp
Spinnaker::Camera, 570
EventStream0TransferBlockStart
Spinnaker::Camera, 570
EventStream0TransferBlockStartFrameID
Spinnaker::Camera, 570
EventStream0TransferBlockStartTimestamp
Spinnaker::Camera, 570
EventStream0TransferBlockTrigger
Spinnaker::Camera, 570
EventStream0TransferBlockTriggerFrameID
Spinnaker::Camera, 571
EventStream0TransferBlockTriggerTimestamp
Spinnaker::Camera, 571
EventStream0TransferBurstEnd
Spinnaker::Camera, 571
EventStream0TransferBurstEndFrameID
Spinnaker::Camera, 571
EventStream0TransferBurstEndTimestamp
Spinnaker::Camera, 571
EventStream0TransferBurstStart
Spinnaker::Camera, 571
EventStream0TransferBurstStartFrameID
Spinnaker::Camera, 572
EventStream0TransferBurstStartTimestamp
Spinnaker::Camera, 572
EventStream0TransferEnd
Spinnaker::Camera, 572
EventStream0TransferEndFrameID
Spinnaker::Camera, 572
EventStream0TransferEndTimestamp
Spinnaker::Camera, 572
EventStream0TransferOverflow
Spinnaker::Camera, 572
EventStream0TransferOverflowFrameID
Spinnaker::Camera, 573
EventStream0TransferOverflowTimestamp
Spinnaker::Camera, 573
EventStream0TransferPause
Spinnaker::Camera, 573
EventStream0TransferPauseFrameID
Spinnaker::Camera, 573
EventStream0TransferPauseTimestamp
Spinnaker::Camera, 573
EventStream0TransferResume
Spinnaker::Camera, 573
EventStream0TransferResumeFrameID
Spinnaker::Camera, 574
EventStream0TransferResumeTimestamp
Spinnaker::Camera, 574
EventStream0TransferStart
Spinnaker::Camera, 574
EventStream0TransferStartFrameID
Spinnaker::Camera, 574
EventStream0TransferStartTimestamp
Spinnaker::Camera, 574
EventTest
Spinnaker::Camera, 574
EventTestTimestamp
Spinnaker::Camera, 575
EventTimer0End
Spinnaker::Camera, 575
EventTimer0EndFrameID
Spinnaker::Camera, 575
EventTimer0EndTimestamp
Spinnaker::Camera, 575
EventTimer0Start
Spinnaker::Camera, 575
EventTimer0StartFrameID
Spinnaker::Camera, 575
EventTimer0StartTimestamp
Spinnaker::Camera, 576
EventTimer1End
Spinnaker::Camera, 576
EventTimer1EndFrameID
Spinnaker::Camera, 576
EventTimer1EndTimestamp
Spinnaker::Camera, 576
EventTimer1Start
Spinnaker::Camera, 576
EventTimer1StartFrameID
Spinnaker::Camera, 576
EventTimer1StartTimestamp
Spinnaker::Camera, 577
EventType
Spinnaker Definitions, 207
eventType
DeviceEvents.cpp, 1394
Exception, 802
Spinnaker::Exception, 804, 805
Exception Class, 181
ExceptionHandling.cpp
causeSpinnakerException, 1399
causeStandardException, 1399
chosenException, 1399
exceptionType, 1399
main, 1399
exceptionType
ExceptionHandling.cpp, 1399
Execute
Spinnaker::GenApi::CommandNode, 738
ExecuteDeleteCommand
FileAccess_QuickSpin.cpp, 1404
ExecuteReadCommand
FileAccess_QuickSpin.cpp, 1404
ExecuteWriteCommand
FileAccess_QuickSpin.cpp, 1404
Expert
Types Enums, 384
Exposure.cpp
AcquireImages, 1400
ConfigureExposure, 1400
main, 1400

PrintDeviceInfo, 1401
ResetExposure, 1401
RunSingleCamera, 1401
Exposure_QuickSpin.cpp
AcquireImages, 1402
ConfigureExposure, 1402
main, 1402
PrintDeviceInfo, 1402
ResetExposure, 1402
RunSingleCamera, 1402
ExposureActiveMode
Spinnaker::Camera, 577
ExposureActiveModeEnums
CameraDefs Class, 118
ExposureAuto
Spinnaker::Camera, 577
ExposureAutoEnums
CameraDefs Class, 118
ExposureMode
Spinnaker::Camera, 577
ExposureModeEnums
CameraDefs Class, 118
ExposureTime
Spinnaker::Camera, 577
ExposureTimeMode
Spinnaker::Camera, 577
ExposureTimeModeEnums
CameraDefs Class, 119
ExposureTimeSelector
Spinnaker::Camera, 578
ExposureTimeSelectorEnums
CameraDefs Class, 119
ExtractAndSavePolarQuadImages
Polarization.cpp, 1441
ExtractIndependentSubtree
INodeMapDyn Interface, 325
ExtractPolarQuadrant
Spinnaker::ImageUtilityPolarization, 975
ExtractSubtree
Spinnaker::GenApi::CNodeMapFactory, 726

FLIR_SPINNAKER_VERSION_BUILD
System.h, 1332
FLIR_SPINNAKER_VERSION_MAJOR
System.h, 1332
FLIR_SPINNAKER_VERSION_MINOR
System.h, 1332
FLIR_SPINNAKER_VERSION_TYPE
System.h, 1332
FMT_I64
Compatibility.h, 1246
FactoryReset
Spinnaker::Camera, 578
FileAccess_QuickSpin.cpp
_enableDebug, 1406
_fileSelector, 1406
AcquireImages, 1403
CloseFile, 1403
DownloadImage, 1404
ExecuteDeleteCommand, 1404
ExecuteReadCommand, 1404
ExecuteWriteCommand, 1404
InitializeSystem, 1404
main, 1404
OpenFileToRead, 1404
OpenFileToWrite, 1405
PrintDebugMessage, 1405
PrintDeviceInfo, 1405
PrintResultMessage, 1405
PrintUsage, 1405
UploadImage, 1405
FileAccessBuffer
Spinnaker::Camera, 578
FileAccessLength
Spinnaker::Camera, 578
FileAccessOffset
Spinnaker::Camera, 578
FileOpenMode
Spinnaker::Camera, 578
FileOpenModeEnums
CameraDefs Class, 120
FileOperationExecute
Spinnaker::Camera, 579
FileOperationResult
Spinnaker::Camera, 579
FileOperationSelector
Spinnaker::Camera, 579
FileOperationSelectorEnums
CameraDefs Class, 120
FileOperationStatus
Spinnaker::Camera, 579
FileOperationStatusEnums
CameraDefs Class, 120
FileProtocolAdapter, 808
Spinnaker::GenApi::FileProtocolAdapter, 809
FileSelector
Spinnaker::Camera, 579
FileSelectorEnums
CameraDefs Class, 121
FileSize
Spinnaker::Camera, 580
FileUploadPersistence
Inference.cpp, 1423
filebuf_type
Spinnaker::GenApi::IDevFileStreamBase, 875
Spinnaker::GenApi::ODevFileStreamBase, 1063
Filestream Class, 273
FilterDriverStatus
Spinnaker::TransportLayerInterface, 1136
FilterDriverStatusEnum
TransportLayerDefs Class, 221
find
Spinnaker::GenICam::gcstring, 827, 828
find_first_not_of
Spinnaker::GenICam::gcstring, 828
find_first_of
Spinnaker::GenICam::gcstring, 828

Flags
GVCP_REQUEST_HEADER, 843
U3V_COMMAND_HEADER, 1158
float32_t
GCTypes Class, 278
float64_t
GCTypes Class, 278
FloatNode, 812
Spinnaker::GenApi::FloatNode, 815
FloatNode Class, 274
CFloatRef, 274
FloatRegNode, 819
Spinnaker::GenApi::FloatRegNode, 820, 821
FloatRegNode Class, 275
FlushQueueAllDiscard
Spinnaker::IDataStream, 870
ForceIP
Spinnaker::CameraBase, 629
Spinnaker::ICameraBase, 849
frameRate
Spinnaker::Video::AVIOption, 464
Spinnaker::Video::H264Option, 845
Spinnaker::Video::MJPGOption, 1040
FromString
IValue Class, 343
Spinnaker::GenApi::EAccessModeClass, 777
Spinnaker::GenApi::ECachingModeClass, 779
Spinnaker::GenApi::EDisplayNotationClass, 780
Spinnaker::GenApi::EEndianessClass, 781
Spinnaker::GenApi::EGenApiSchemaVersion↩ Class, 782
Spinnaker::GenApi::EInputDirectionClass, 783
Spinnaker::GenApi::ENameSpaceClass, 784
Spinnaker::GenApi::ERepresentationClass, 793
Spinnaker::GenApi::ESignClass, 795
Spinnaker::GenApi::ESlopeClass, 796
Spinnaker::GenApi::EStandardNameSpaceClass, 797
Spinnaker::GenApi::EVisibilityClass, 801
Spinnaker::GenApi::EYesNoClass, 808
Spinnaker::GenApi::ValueNode, 1162
Function_NodeCallback
Spinnaker::GenApi::Function_NodeCallback, 823
Function_NodeCallback< Function >, 822

GC_COUNTOF
GCUtilities.h, 1269
GC_INT32_MAX
GCTypes.h, 1264
GC_INT32_MIN
GCTypes.h, 1264
GC_INT64_MAX
GCTypes.h, 1264
GC_INT64_MIN
GCTypes.h, 1264
GC_INT8_MAX
GCTypes.h, 1265
GC_INT8_MIN
GCTypes.h, 1265
GC_UINT32_MAX
GCTypes.h, 1265
GC_UINT64_MAX
GCTypes.h, 1265
GC_UINT8_MAX
GCTypes.h, 1265
GCSTRING_NPOS
GCString.h, 1260
GCString Class, 276
GCString.h
GCSTRING_NPOS, 1260
operator<<, 1260
operator>>, 1261
GCSynch Class, 277
GCTypes Class, 278
float32_t, 278
float64_t, 278
GCTypes.h
__STDC_CONSTANT_MACROS, 1264
__STDC_LIMIT_MACROS, 1264
GC_INT32_MAX, 1264
GC_INT32_MIN, 1264
GC_INT64_MAX, 1264
GC_INT64_MIN, 1264
GC_INT8_MAX, 1265
GC_INT8_MIN, 1265
GC_UINT32_MAX, 1265
GC_UINT64_MAX, 1265
GC_UINT8_MAX, 1265
GCUtilities Utility, 280
DoesEnvironmentVariableExist, 281
GetFiles, 281
GetGenICamCLProtocolFolder, 282
GetGenICamCacheFolder, 281
GetGenICamLogConfig, 282
GetModulePathFromFunction, 282
GetValueOfEnvironmentVariable, 282, 283
INTEGRAL_CAST2, 283
INTEGRAL_CAST, 283
ReplaceEnvironmentVariables, 283
SetGenICamCLProtocolFolder, 284
SetGenICamCacheFolder, 283
SetGenICamLogConfig, 284
Tokenize, 284
UrlDecode, 284
UrlEncode, 285
GCUtilities.h
_TO_STRING, 1268
__ERR__, 1267
__LINE_STR__, 1268
__LOCATION__, 1268
__OUTPUT_FORMATER__, 1268
__TODO__, 1268
__WARN__, 1268
EXPAND_TO_STRINGISE, 1268
GC_COUNTOF, 1269
GENICAM_DEPRECATED, 1269
GENICAM_UNUSED, 1269

USE_TEMP_CACHE_FILE, 1269
GENCP_COMMAND_HEADER_SIZE
Spinnaker::GenApi, 450
GENCP_EVENT_BASIC_SIZE
Spinnaker::GenApi, 450
GENCP_EVENT_CMD_ID
Spinnaker::GenApi, 450
GENICAM_DEPRECATED
GCUtilities.h, 1269
GENICAM_UNUSED
GCUtilities.h, 1269
GUIXMLLocation
Spinnaker::TransportLayerDevice, 1130
GUIXMLLocationEnum
TransportLayerDefs Class, 222
GUIXMLPath
Spinnaker::TransportLayerDevice, 1130
GVCP_CHUNK_TRAILER, 834
ChunkID, 834
ChunkLength, 834
GVCP_EVENT_ITEM_BASIC, 836
EventId, 836
ReservedOrEventSize, 837
GVCP_EVENT_ITEM_EXTENDED_ID, 837
BlockId, 837
BlockId64High, 837
BlockId64Low, 838
EventId, 838
ReservedOrEventSize, 838
StreamChannelId, 838
TimestampHigh, 838
TimestampLow, 838
GVCP_EVENT_ITEM, 835
BlockId, 835
EventId, 835
ReservedOrEventSize, 835
StreamChannelId, 836
TimestampHigh, 836
TimestampLow, 836
GVCP_EVENT_REQUEST_EXTENDED_ID, 840
Header, 840
Items, 840
GVCP_EVENT_REQUEST, 839
Header, 839
Items, 839
GVCP_EVENTDATA_REQUEST_EXTENDED_ID, 842
Data, 842
Event, 842
Header, 843
GVCP_EVENTDATA_REQUEST, 841
Data, 841
Event, 841
Header, 841
GVCP_MESSAGE_TAGS
Spinnaker::GenApi, 449
GVCP_REQUEST_HEADER, 843
Command, 843
Flags, 843
Length, 843
Magic, 844
ReqId, 844
Gain
Spinnaker::Camera, 580
GainAuto
Spinnaker::Camera, 580
GainAutoBalance
Spinnaker::Camera, 580
GainAutoBalanceEnums
CameraDefs Class, 121
GainAutoEnums
CameraDefs Class, 121
GainSelector
Spinnaker::Camera, 581
GainSelectorEnums
CameraDefs Class, 122
Gamma
Spinnaker::Camera, 581
GammaEnable
Spinnaker::Camera, 581
gateway
AdapterConfig::IpInfo, 1013
gcstring, 824
Spinnaker::GenICam::gcstring, 825
GenICamXMLLocation
Spinnaker::TransportLayerDevice, 1127
GenICamXMLLocationEnum
TransportLayerDefs Class, 221
GenICamXMLPath
Spinnaker::TransportLayerDevice, 1127
GenTLInfo_QuickSpin.cpp
main, 1406
PrintApplicationLayerDeviceInfo, 1406
PrintTransportLayerDeviceInfo, 1407
PrintTransportLayerInterfaceInfo, 1407
PrintTransportLayerStreamInfo, 1407
GenTLSFNCVersionMajor
Spinnaker::TransportLayerSystem, 1153
GenTLSFNCVersionMinor
Spinnaker::TransportLayerSystem, 1153
GenTLSFNCVersionSubMinor
Spinnaker::TransportLayerSystem, 1153
GenTLVersionMajor
Spinnaker::TransportLayerSystem, 1153
GenTLVersionMinor
Spinnaker::TransportLayerSystem, 1153
Get
IRegister Interfaces, 336
Spinnaker::GenApi::RegisterNode, 1085
get
Spinnaker::BasePtr, 466
GetAccessMode
Spinnaker::CameraBase, 629
Spinnaker::GenApi::CChunkPort, 665
Spinnaker::GenApi::CEventPort, 686
Spinnaker::GenApi::CPortImpl, 748
Spinnaker::GenApi::CRegisterPortImpl, 755

Spinnaker::GenApi::CTestPortStruct, 761
Spinnaker::GenApi::Node, 1044
Spinnaker::GenApi::PortRecorder, 1076
Spinnaker::GenApi::PortReplay, 1080
Spinnaker::ICameraBase, 849
GetAddress
IRegister Interfaces, 337
Spinnaker::GenApi::RegisterNode, 1085
GetAlias
INode Interface, 313
Spinnaker::GenApi::Node, 1045
GetAuto10GDesc
AdapterConfig, 389
GetAutoGigabitDesc
AdapterConfig, 389
GetAutoStartIp
AdapterConfig, 389
GetAutoSubnetMask
AdapterConfig, 389
GetAutoSubnetMaskLength
AdapterConfig, 389
GetBitsPerPixel
Spinnaker::IImage, 888
Spinnaker::Image, 928
GetBlackLevel
Spinnaker::ChunkData, 701
Spinnaker::IChunkData, 861
GetBoxAt
Chunk Data Inference Class, 170
GetBoxCount
Chunk Data Inference Class, 170
GetBoxSize
Chunk Data Inference Class, 170
getBufSize
Spinnaker::GenApi::FileProtocolAdapter, 810
GetBufferChunkData
Spinnaker::IDataStream, 870
GetBufferInfoBool8Type
Spinnaker::IDataStream, 870
GetBufferInfoPtrType
Spinnaker::IDataStream, 870
GetBufferInfoSizeType
Spinnaker::IDataStream, 870
GetBufferInfoUInt64Type
Spinnaker::IDataStream, 870
GetBufferOwnership
Spinnaker::CameraBase, 630
Spinnaker::ICameraBase, 849
GetBufferSize
Spinnaker::IImage, 888
Spinnaker::Image, 928
GetBuildDate
Spinnaker::Exception, 805
GetBuildTime
Spinnaker::Exception, 805
GetByDeviceID
Spinnaker::CameraList, 642
Spinnaker::ICameraList, 857
GetByIndex
Spinnaker::CameraList, 643
Spinnaker::ICameraList, 857
Spinnaker::IInterfaceList, 914
Spinnaker::InterfaceList, 1004
GetBySerial
Spinnaker::CameraList, 643
Spinnaker::ICameraList, 857
GetCRC
Spinnaker::ChunkData, 701
Spinnaker::IChunkData, 861
GetCachingMode
INode Interface, 313
Spinnaker::GenApi::Node, 1045
GetCallbackType
Spinnaker::GenApi::CNodeCallback, 719
getCameraCategory
GigEVisionPerformance.cpp, 1410
GetCameras
Spinnaker::IInterface, 905
Spinnaker::ISystem, 1016
Spinnaker::Interface, 988
Spinnaker::System, 1102
GetCastAlias
INode Interface, 313
Spinnaker::GenApi::Node, 1045
GetCategoryName
Spinnaker::LoggingEventData, 1030
GetChannelStatus
Spinnaker::IImageStatistics, 901
Spinnaker::ImageStatistics, 955
GetChildren
INode Interface, 313
Spinnaker::GenApi::Node, 1045
GetChunkData
Spinnaker::IImage, 888
Spinnaker::Image, 928
GetChunkIDLength
Spinnaker::GenApi::CChunkPort, 665
GetChunkID
Spinnaker::GenApi::PortNode, 1071
GetChunkLayoutId
Spinnaker::IImage, 889
Spinnaker::Image, 928
GetColorProcessing
Spinnaker::IImage, 889
Spinnaker::Image, 929
GetConfigLogFileName
AdapterConfig, 389
GetCookie
IPortRecorder Interface, 334
Spinnaker::GenApi::CPortWriteList, 751
GetCounterValue
Spinnaker::ChunkData, 701
Spinnaker::IChunkData, 861
GetCpuStats
CpuUtil, 392
GetCurrentEntry

IEnumeration Interface, 299
Spinnaker::GenApi::CEnumerationTRef, 669
Spinnaker::GenApi::EnumNode, 790
GetData
Spinnaker::IImage, 889
Spinnaker::Image, 929
GetDataAbsoluteMax
Spinnaker::IImage, 889
Spinnaker::Image, 929
GetDataAbsoluteMin
Spinnaker::IImage, 889
Spinnaker::Image, 930
GetDefaultColorProcessing
Spinnaker::Image, 930
GetDescription
INode Interface, 314
Spinnaker::GenApi::Node, 1046
GetDeviceEventId
Spinnaker::DeviceEventHandler, 769
Spinnaker::IDeviceEventHandler, 882
GetDeviceEventName
Spinnaker::DeviceEventHandler, 769
Spinnaker::IDeviceEventHandler, 882
GetDeviceName
INodeMap Interface, 322
Spinnaker::GenApi::Node, 1046
Spinnaker::GenApi::NodeMap, 1056
GetDeviceNodeMap
Spinnaker::IDataStream, 871
GetDeviceSerial
AcquisitionMultipleCameraRecovery.cpp, 1383
GetDeviceVersion
IDeviceInfo Interface, 294
Spinnaker::GenApi::NodeMap, 1056
GetDisplayName
INode Interface, 314
Spinnaker::GenApi::Node, 1046
GetDisplayNotation
IFloat Interface, 305
Spinnaker::GenApi::FloatNode, 815
GetDisplayPrecision
IFloat Interface, 305
Spinnaker::GenApi::FloatNode, 815
GetDocuURL
INode Interface, 314
Spinnaker::GenApi::Node, 1046
GetEncoderValue
Spinnaker::ChunkData, 701
Spinnaker::IChunkData, 861
GetEntries
IEnumeration Interface, 299
Spinnaker::GenApi::EnumNode, 790
GetEntry
IEnumeration Interface, 300
IEnumerationT Interface, 302
Spinnaker::GenApi::CEnumerationTRef, 670
Spinnaker::GenApi::EnumNode, 791
GetEntryByName
IEnumeration Interface, 300
Spinnaker::GenApi::EnumNode, 791
GetEnumAlias
Spinnaker::GenApi::CFloatPtr, 692
Spinnaker::GenApi::FloatNode, 815
GetEnumerationLogFileName
AdapterConfig, 389
GetError
Spinnaker::Exception, 805
GetErrorMessage
SpinUpdate.h, 1328
Spinnaker::Exception, 805
GetEventIDLength
Spinnaker::GenApi::CEventPort, 686
GetEventID
INode Interface, 314
Spinnaker::GenApi::Node, 1046
GetEventPayloadData
Spinnaker::EventHandler, 799
GetEventPayloadDataSize
Spinnaker::EventHandler, 799
GetEventType
Spinnaker::EventHandler, 799
GetExposureEndLineStatusAll
Spinnaker::ChunkData, 702
Spinnaker::IChunkData, 861
GetExposureTime
Spinnaker::ChunkData, 702
Spinnaker::IChunkData, 861
GetFeatureBagHandle
Spinnaker::GenApi::CFeatureBag, 689
GetFeatures
Spinnaker::GenApi::CategoryNode, 648
GetFileName
Spinnaker::Exception, 806
GetFiles
GCUtilities Utility, 281
GetFloatAlias
Spinnaker::GenApi::IntegerNode, 983
GetFrameID
Spinnaker::ChunkData, 702
Spinnaker::IChunkData, 862
Spinnaker::IImage, 889
Spinnaker::Image, 930
GetFullErrorMessage
Spinnaker::Exception, 806
GetFunctionName
Spinnaker::Exception, 806
GetGain
Spinnaker::ChunkData, 702
Spinnaker::IChunkData, 862
GetGenApiVersion
IDeviceInfo Interface, 294
Spinnaker::GenApi::NodeMap, 1056
GetGenICamCLProtocolFolder
GCUtilities Utility, 282
GetGenICamCacheFolder
GCUtilities Utility, 281

GetGenICamLogConfig
GCUtilities Utility, 282
GetGuiXml
Spinnaker::CameraBase, 630
Spinnaker::ICameraBase, 849
GetHeatmapColorGradient
Spinnaker::ImageUtilityHeatmap, 966
GetHeatmapRange
Spinnaker::ImageUtilityHeatmap, 967
GetHeight
Spinnaker::ChunkData, 703
Spinnaker::IChunkData, 862
Spinnaker::IImage, 890
Spinnaker::Image, 931
GetHistogram
Spinnaker::IImageStatistics, 901
Spinnaker::ImageStatistics, 955
GetID
Spinnaker::IImage, 890
Spinnaker::Image, 931
GetImage
Spinnaker::ChunkData, 703
Spinnaker::IChunkData, 862
getImageCount
ImageEventHandlerImpl, 948
GetImageData
Spinnaker::IImage, 890
Spinnaker::Image, 931
GetImageSize
Spinnaker::IImage, 890
Spinnaker::Image, 931
GetImageStatus
Spinnaker::IImage, 890
Spinnaker::Image, 932
GetImageStatusDescription
Spinnaker::Image, 932
GetInc
IFloat Interface, 305
Spinnaker::GenApi::FloatNode, 815
Spinnaker::GenApi::IntegerNode, 984
GetIncMode
IFloat Interface, 305
Spinnaker::GenApi::FloatNode, 816
Spinnaker::GenApi::IntegerNode, 984
GetInferenceBoundingBoxResult
Spinnaker::ChunkData, 703
Spinnaker::IChunkData, 862
GetInferenceConfidence
Spinnaker::ChunkData, 703
Spinnaker::IChunkData, 862
GetInferenceFrameId
Spinnaker::ChunkData, 704
Spinnaker::IChunkData, 863
GetInferenceResult
Spinnaker::ChunkData, 704
Spinnaker::IChunkData, 863
GetInstance
Spinnaker::System, 1103
GetIntAlias
Spinnaker::GenApi::CFloatPtr, 692
Spinnaker::GenApi::FloatNode, 816
GetIntValue
IEnumeration Interface, 300
Spinnaker::GenApi::EnumNode, 791
GetInterfaceId
InterfaceEventHandlerImpl, 1000
GetInterfaceName
Pointer Class, 359
GetInterfaces
Spinnaker::ISystem, 1016
Spinnaker::System, 1103
GetLength
IRegister Interfaces, 337
Spinnaker::GenApi::RegisterNode, 1086
GetLibraryVersion
Spinnaker::ISystem, 1016
Spinnaker::System, 1103
GetLineNumber
Spinnaker::Exception, 806
GetLinePitch
Spinnaker::ChunkData, 704
Spinnaker::IChunkData, 863
GetLineStatusAll
Spinnaker::ChunkData, 704
Spinnaker::IChunkData, 863
GetListOfValidValues
IFloat Interface, 305
Spinnaker::GenApi::FloatNode, 816
Spinnaker::GenApi::IntegerNode, 984
GetLock
INodeMap Interface, 322
Spinnaker::GenApi::NodeMap, 1056
Spinnaker::GenICam::LockableObject, 1028
GetLogMessage
Spinnaker::LoggingEventData, 1030
GetLoggingEventPriorityLevel
Spinnaker::ISystem, 1016
Spinnaker::System, 1104
GetMax
IFloat Interface, 305
Spinnaker::GenApi::FloatNode, 816
Spinnaker::GenApi::IntegerNode, 984
getMaxImages
ImageEventHandlerImpl, 949
GetMaxIpAddress
AdapterConfig, 390
GetMaxLength
IString Class, 342
Spinnaker::GenApi::StringNode, 1095
GetMean
Spinnaker::IImageStatistics, 901
Spinnaker::ImageStatistics, 956
GetMin
IFloat Interface, 306
Spinnaker::GenApi::FloatNode, 816
Spinnaker::GenApi::IntegerNode, 984

GetMinIpAddress
AdapterConfig, 390
GetModelName
Spinnaker::GenApi::NodeMap, 1056
GetModulePathFromFunction
GCUtilities Utility, 282
GetNDC
Spinnaker::LoggingEventData, 1030
GetName
Spinnaker::GenApi::Node, 1046
GetNameSpace
INode Interface, 314
Spinnaker::GenApi::Node, 1046
GetNextImage
Spinnaker::CameraBase, 630
Spinnaker::ICameraBase, 850
Spinnaker::IDataStream, 871
GetNextImageInternal
Spinnaker::IDataStream, 871
GetNode
INodeMap Interface, 322
Spinnaker::GenApi::CNodeCallback, 719
Spinnaker::GenApi::NodeMap, 1056
Spinnaker::GenApi::ValueNode, 1162
GetNodeHandle
Spinnaker::GenApi::Node, 1047
GetNodeMap
INode Interface, 314
Spinnaker::CameraBase, 631
Spinnaker::GenApi::Node, 1047
Spinnaker::ICameraBase, 850
Spinnaker::IDataStream, 871
GetNodeMapHandle
Spinnaker::GenApi::NodeMap, 1057
GetNodeStatistics
Spinnaker::GenApi::CNodeMapFactory, 726
GetNodes
Spinnaker::GenApi::NodeMap, 1057
GetNumChannels
Spinnaker::IImage, 890
Spinnaker::Image, 932
GetNumDataStreams
Spinnaker::CameraBase, 631
Spinnaker::ICameraBase, 850
GetNumImagesInUse
Spinnaker::CameraBase, 632
Spinnaker::ICameraBase, 850
Spinnaker::IDataStream, 871
GetNumNodes
INodeMap Interface, 322
Spinnaker::GenApi::NodeMap, 1057
GetNumPixelValues
Spinnaker::IImageStatistics, 901
Spinnaker::ImageStatistics, 956
GetNumReads
Spinnaker::GenApi::CTestPortStruct, 761
GetNumWrites
Spinnaker::GenApi::CTestPortStruct, 762
GetNumericValue
IEnumEntry Interface, 297
Spinnaker::GenApi::EnumEntryNode, 786
GetOffsetX
Spinnaker::ChunkData, 705
Spinnaker::IChunkData, 863
GetOffsetY
Spinnaker::ChunkData, 705
Spinnaker::IChunkData, 863
GetParamStr
AdapterConfigException, 457
GetParents
INode Interface, 314
Spinnaker::GenApi::Node, 1047
GetPartSelector
Spinnaker::ChunkData, 705
Spinnaker::IChunkData, 864
GetPayloadType
Spinnaker::IImage, 891
Spinnaker::Image, 932
GetPerformanceCounter
PerformanceCounter, 393
GetPixelDynamicRangeMax
Spinnaker::ChunkData, 705
Spinnaker::IChunkData, 864
GetPixelDynamicRangeMin
Spinnaker::ChunkData, 706
Spinnaker::IChunkData, 864
GetPixelFormat
Spinnaker::IImage, 891
Spinnaker::Image, 933
GetPixelFormatIntType
Spinnaker::IImage, 891
Spinnaker::Image, 933
GetPixelFormatName
Spinnaker::IImage, 891
Spinnaker::Image, 933
GetPixelValueRange
Spinnaker::IImageStatistics, 902
Spinnaker::ImageStatistics, 956
GetPollingTime
INode Interface, 315
Spinnaker::GenApi::Node, 1047
GetPort
Spinnaker::IDataStream, 871
GetPortHandle
Spinnaker::GenApi::PortNode, 1072
GetPortReplayHandle
Spinnaker::GenApi::PortReplay, 1080
GetPortWriteListHandle
Spinnaker::GenApi::CPortWriteList, 752
GetPrincipalInterfaceType
INode Interface, 315
Spinnaker::GenApi::CChunkPort, 665
Spinnaker::GenApi::CEventPort, 686
Spinnaker::GenApi::CTestPortStruct, 762
Spinnaker::GenApi::Node, 1047
GetPriority

Spinnaker::LoggingEventData, 1030
GetPriorityName
Spinnaker::LoggingEventData, 1031
GetPrivateData
Spinnaker::IImage, 891
Spinnaker::Image, 934
GetProductGuid
IDeviceInfo Interface, 295
Spinnaker::GenApi::NodeMap, 1057
GetProperty
INode Interface, 315
Spinnaker::GenApi::Node, 1048
GetPropertyNames
INode Interface, 315
Spinnaker::GenApi::Node, 1048
GetQuadFileNameAppendage
Polarization.cpp, 1441
GetRange
Spinnaker::IImageStatistics, 902
Spinnaker::ImageStatistics, 957
GetRepresentation
IFloat Interface, 306
Spinnaker::GenApi::FloatNode, 816
Spinnaker::GenApi::IntegerNode, 984
GetScan3dAxisMax
Spinnaker::ChunkData, 706
Spinnaker::IChunkData, 864
GetScan3dAxisMin
Spinnaker::ChunkData, 706
Spinnaker::IChunkData, 864
GetScan3dCoordinateOffset
Spinnaker::ChunkData, 706
Spinnaker::IChunkData, 864
GetScan3dCoordinateReferenceValue
Spinnaker::ChunkData, 707
Spinnaker::IChunkData, 865
GetScan3dCoordinateScale
Spinnaker::ChunkData, 707
Spinnaker::IChunkData, 865
GetScan3dInvalidDataValue
Spinnaker::ChunkData, 707
Spinnaker::IChunkData, 865
GetScan3dTransformValue
Spinnaker::ChunkData, 707
Spinnaker::IChunkData, 865
GetScanLineSelector
Spinnaker::ChunkData, 708
Spinnaker::IChunkData, 865
GetSchemaVersion
IDeviceInfo Interface, 295
Spinnaker::GenApi::NodeMap, 1057
GetSecondsCounter
SecondsCounter, 394
GetSelectedFeatures
ISelector Interface, 338
Spinnaker::GenApi::Node, 1048
GetSelectingFeatures
ISelector Interface, 338
Spinnaker::GenApi::Node, 1048
GetSelectorList
ISelectorDigit Interface, 339
Spinnaker::GenApi::CSelectorSet, 758
GetSequencerSetActive
Spinnaker::ChunkData, 708
Spinnaker::IChunkData, 865
GetSerialDataLength
Spinnaker::ChunkData, 708
Spinnaker::IChunkData, 866
GetSize
Spinnaker::CameraList, 644
Spinnaker::ICameraList, 857
Spinnaker::IInterfaceList, 915
Spinnaker::InterfaceList, 1005
GetStandardNameSpace
IDeviceInfo Interface, 295
Spinnaker::GenApi::NodeMap, 1057
GetStatistics
Spinnaker::IImageStatistics, 902
Spinnaker::ImageStatistics, 957
GetStreamChannelID
Spinnaker::ChunkData, 708
Spinnaker::IChunkData, 866
GetStreamInfoBool8Type
Spinnaker::IDataStream, 871
GetStreamInfoSizeType
Spinnaker::IDataStream, 872
GetStreamType
Spinnaker::IDataStream, 872
GetStride
Spinnaker::IImage, 891
Spinnaker::Image, 934
GetSubnetMaskLength
AdapterConfig, 390
GetSupportedSchemaVersions
INodeMapDyn Interface, 325
Spinnaker::GenApi::CNodeMapFactory, 726
Spinnaker::GenApi::NodeMap, 1058
GetSwapEndianess
IPortConstruct Interface, 333
Spinnaker::GenApi::CChunkPort, 665
Spinnaker::GenApi::CEventPort, 686
Spinnaker::GenApi::CPortImpl, 748
Spinnaker::GenApi::PortNode, 1072
GetSymbolic
IEnumEntry Interface, 297
Spinnaker::GenApi::EnumEntryNode, 787
GetSymbolics
Spinnaker::GenApi::EnumNode, 791
GetTLDeviceNodeMap
Spinnaker::CameraBase, 632
Spinnaker::ICameraBase, 850
GetTLNodeMap
Spinnaker::IInterface, 905
Spinnaker::ISystem, 1016
Spinnaker::Interface, 989
Spinnaker::System, 1104

GetTLPayloadType
Spinnaker::IImage, 892
Spinnaker::Image, 935
GetTLPixelFormat
Spinnaker::IImage, 892
Spinnaker::Image, 935
GetTLPixelFormatNamespace
Spinnaker::IImage, 892
Spinnaker::Image, 935
GetTLStreamNodeMap
Spinnaker::CameraBase, 632
Spinnaker::ICameraBase, 850
GetThreadName
Spinnaker::LoggingEventData, 1031
GetTimeStamp
Spinnaker::IImage, 892
Spinnaker::Image, 934
GetTimerValue
Spinnaker::ChunkData, 709
Spinnaker::IChunkData, 866
GetTimestamp
Spinnaker::ChunkData, 709
Spinnaker::IChunkData, 866
Spinnaker::LoggingEventData, 1031
GetTimestampLatchValue
Spinnaker::ChunkData, 709
Spinnaker::IChunkData, 866
GetToolTip
IDeviceInfo Interface, 295
Spinnaker::GenApi::Node, 1048
Spinnaker::GenApi::NodeMap, 1058
GetTransferBlockID
Spinnaker::ChunkData, 709
Spinnaker::IChunkData, 866
GetTransferQueueCurrentBlockCount
Spinnaker::ChunkData, 710
Spinnaker::IChunkData, 867
GetUniqueID
Spinnaker::CameraBase, 633
Spinnaker::ICameraBase, 851
GetUnit
IFloat Interface, 306
Spinnaker::GenApi::FloatNode, 817
Spinnaker::GenApi::IntegerNode, 985
GetUserBufferCount
Spinnaker::CameraBase, 633
Spinnaker::ICameraBase, 851
GetUserBufferSize
Spinnaker::CameraBase, 633
Spinnaker::ICameraBase, 851
GetUserBufferTotalSize
Spinnaker::CameraBase, 634
Spinnaker::ICameraBase, 851
GetValidPayloadSize
Spinnaker::IImage, 892
Spinnaker::Image, 936
GetValue
IBoolean Interface, 286
Spinnaker::GenApi::BooleanNode, 471
Spinnaker::GenApi::CEnumerationTRef, 670
Spinnaker::GenApi::Counter, 740
Spinnaker::GenApi::EnumEntryNode, 787
Spinnaker::GenApi::FloatNode, 817
Spinnaker::GenApi::IntegerNode, 985
Spinnaker::GenApi::StringNode, 1095
GetValueOfEnvironmentVariable
GCUtilities Utility, 282, 283
GetVendorName
IDeviceInfo Interface, 295
Spinnaker::GenApi::NodeMap, 1058
GetVersion
Chunk Data Inference Class, 170
GetVersionGuid
IDeviceInfo Interface, 295
Spinnaker::GenApi::NodeMap, 1058
GetVisibility
INode Interface, 315
Spinnaker::GenApi::Node, 1049
GetWidth
Spinnaker::ChunkData, 710
Spinnaker::IChunkData, 867
Spinnaker::IImage, 892
Spinnaker::Image, 936
GetXOffset
Spinnaker::IImage, 893
Spinnaker::Image, 936
GetXPadding
Spinnaker::IImage, 893
Spinnaker::Image, 937
GetYOffset
Spinnaker::IImage, 893
Spinnaker::Image, 937
GetYPadding
Spinnaker::IImage, 893
Spinnaker::Image, 937
getline
Spinnaker::GenICam, 452
GevActionDeviceKey
Spinnaker::TransportLayerInterface, 1136
GevActionGroupKey
Spinnaker::TransportLayerInterface, 1136
GevActionGroupMask
Spinnaker::TransportLayerInterface, 1136
GevActionTime
Spinnaker::TransportLayerInterface, 1137
GevActiveLinkCount
Spinnaker::Camera, 581
GevCCPEnum
TransportLayerDefs Class, 222
GevCCPEnums
CameraDefs Class, 122
GevCCP
Spinnaker::Camera, 581
Spinnaker::TransportLayerDevice, 1127
GevCurrentDefaultGateway
Spinnaker::Camera, 581

GevCurrentIPAddress
Spinnaker::Camera, 582
GevCurrentIPConfigurationDHCP
Spinnaker::Camera, 582
GevCurrentIPConfigurationLLA
Spinnaker::Camera, 582
GevCurrentIPConfigurationPersistentIP
Spinnaker::Camera, 582
GevCurrentPhysicalLinkConfiguration
Spinnaker::Camera, 582
GevCurrentPhysicalLinkConfigurationEnums
CameraDefs Class, 122
GevCurrentSubnetMask
Spinnaker::Camera, 582
GevDeviceAutoForceIP
Spinnaker::TransportLayerDevice, 1127
Spinnaker::TransportLayerInterface, 1137
GevDeviceDiscoverMaximumPacketSize
Spinnaker::TransportLayerDevice, 1127
GevDeviceForceGateway
Spinnaker::TransportLayerDevice, 1128
Spinnaker::TransportLayerInterface, 1137
GevDeviceForceIPAddress
Spinnaker::TransportLayerDevice, 1128
Spinnaker::TransportLayerInterface, 1137
GevDeviceForceIP
Spinnaker::TransportLayerDevice, 1128
Spinnaker::TransportLayerInterface, 1137
GevDeviceForceSubnetMask
Spinnaker::TransportLayerDevice, 1128
Spinnaker::TransportLayerInterface, 1137
GevDeviceGateway
Spinnaker::TransportLayerDevice, 1128
Spinnaker::TransportLayerInterface, 1138
GevDeviceIPAddress
Spinnaker::TransportLayerDevice, 1128
Spinnaker::TransportLayerInterface, 1138
GevDeviceIsWrongSubnet
Spinnaker::TransportLayerDevice, 1129
GevDeviceMACAddress
Spinnaker::TransportLayerDevice, 1129
Spinnaker::TransportLayerInterface, 1138
GevDeviceMaximumPacketSize
Spinnaker::TransportLayerDevice, 1129
GevDeviceMaximumRetryCount
Spinnaker::TransportLayerDevice, 1129
GevDeviceModeIsBigEndian
Spinnaker::TransportLayerDevice, 1129
GevDevicePort
Spinnaker::TransportLayerDevice, 1129
GevDeviceReadAndWriteTimeout
Spinnaker::TransportLayerDevice, 1130
GevDeviceSubnetMask
Spinnaker::TransportLayerDevice, 1130
Spinnaker::TransportLayerInterface, 1138
GevDiscoveryAckDelay
Spinnaker::Camera, 583
GevFailedPacketCount
Spinnaker::TransportLayerStream, 1145
GevFirstURL
Spinnaker::Camera, 583
GevGVCPExtendedStatusCodes
Spinnaker::Camera, 583
GevGVCPExtendedStatusCodesSelector
Spinnaker::Camera, 583
GevGVCPExtendedStatusCodesSelectorEnums
CameraDefs Class, 122
GevGVCPHeartbeatDisable
Spinnaker::Camera, 583
GevGVCPPendingAck
Spinnaker::Camera, 583
GevGVCPPendingTimeout
Spinnaker::Camera, 584
GevGVSPExtendedIDMode
Spinnaker::Camera, 584
GevGVSPExtendedIDModeEnums
CameraDefs Class, 123
GevHeartbeatTimeout
Spinnaker::Camera, 584
GevIEEE1588
Spinnaker::Camera, 584
GevIEEE1588ClockAccuracy
Spinnaker::Camera, 584
GevIEEE1588ClockAccuracyEnums
CameraDefs Class, 123
GevIEEE1588Mode
Spinnaker::Camera, 584
GevIEEE1588ModeEnums
CameraDefs Class, 123
GevIEEE1588Status
Spinnaker::Camera, 585
GevIEEE1588StatusEnums
CameraDefs Class, 124
GevIPConfigurationStatus
Spinnaker::Camera, 585
GevIPConfigurationStatusEnums
CameraDefs Class, 124
GevInterfaceDefaultGateway
Spinnaker::TransportLayerSystem, 1153
GevInterfaceDefaultIPAddress
Spinnaker::TransportLayerSystem, 1154
GevInterfaceDefaultSubnetMask
Spinnaker::TransportLayerSystem, 1154
GevInterfaceGateway
Spinnaker::TransportLayerInterface, 1138
GevInterfaceGatewaySelector
Spinnaker::TransportLayerInterface, 1138
GevInterfaceMACAddress
Spinnaker::TransportLayerInterface, 1139
Spinnaker::TransportLayerSystem, 1154
GevInterfaceMTU
Spinnaker::TransportLayerInterface, 1139
GevInterfaceReceiveLinkSpeed
Spinnaker::TransportLayerInterface, 1139
GevInterfaceSelector
Spinnaker::Camera, 585

GevInterfaceSubnetIPAddress
Spinnaker::TransportLayerInterface, 1139
GevInterfaceSubnetMask
Spinnaker::TransportLayerInterface, 1139
GevInterfaceSubnetSelector
Spinnaker::TransportLayerInterface, 1139
GevInterfaceTransmitLinkSpeed
Spinnaker::TransportLayerInterface, 1140
GevMACAddress
Spinnaker::Camera, 585
GevMCDA
Spinnaker::Camera, 585
GevMCPHostPort
Spinnaker::Camera, 585
GevMCRC
Spinnaker::Camera, 586
GevMCSP
Spinnaker::Camera, 586
GevMCTT
Spinnaker::Camera, 586
GevMaximumNumberResendRequests
Spinnaker::TransportLayerStream, 1145
GevNumberOfInterfaces
Spinnaker::Camera, 586
GevPAUSEFrameReception
Spinnaker::Camera, 586
GevPAUSEFrameTransmission
Spinnaker::Camera, 586
GevPacketResendMode
Spinnaker::TransportLayerStream, 1145
GevPacketResendTimeout
Spinnaker::TransportLayerStream, 1145
GevPersistentDefaultGateway
Spinnaker::Camera, 587
GevPersistentIPAddress
Spinnaker::Camera, 587
GevPersistentSubnetMask
Spinnaker::Camera, 587
GevPhysicalLinkConfiguration
Spinnaker::Camera, 587
GevPhysicalLinkConfigurationEnums
CameraDefs Class, 124
GevPrimaryApplicationIPAddress
Spinnaker::Camera, 587
GevPrimaryApplicationSocket
Spinnaker::Camera, 587
GevPrimaryApplicationSwitchoverKey
Spinnaker::Camera, 588
GevResendPacketCount
Spinnaker::TransportLayerStream, 1146
GevResendRequestCount
Spinnaker::TransportLayerStream, 1146
GevSCCFGAllInTransmission
Spinnaker::Camera, 588
GevSCCFGExtendedChunkData
Spinnaker::Camera, 588
GevSCCFGPacketResendDestination
Spinnaker::Camera, 588
GevSCCFGUnconditionalStreaming
Spinnaker::Camera, 588
GevSCDA
Spinnaker::Camera, 588
GevSCPDirection
Spinnaker::Camera, 589
GevSCPHostPort
Spinnaker::Camera, 589
GevSCPInterfaceIndex
Spinnaker::Camera, 589
GevSCPSBigEndian
Spinnaker::Camera, 589
GevSCPSDoNotFragment
Spinnaker::Camera, 589
GevSCPSFireTestPacket
Spinnaker::Camera, 590
GevSCPSPacketSize
Spinnaker::Camera, 590
GevSCPD
Spinnaker::Camera, 589
GevSCSP
Spinnaker::Camera, 590
GevSCZoneConfigurationLock
Spinnaker::Camera, 590
GevSCZoneCount
Spinnaker::Camera, 590
GevSCZoneDirectionAll
Spinnaker::Camera, 590
GevSecondURL
Spinnaker::Camera, 591
GevStreamChannelSelector
Spinnaker::Camera, 591
GevSupportedOption
Spinnaker::Camera, 591
GevSupportedOptionSelector
Spinnaker::Camera, 591
GevSupportedOptionSelectorEnums
CameraDefs Class, 125
GevTimestampTickFrequency
Spinnaker::Camera, 591
GevTotalPacketCount
Spinnaker::TransportLayerStream, 1146
GevVersionMajor
Spinnaker::TransportLayerDevice, 1130
Spinnaker::TransportLayerSystem, 1154
GevVersionMinor
Spinnaker::TransportLayerDevice, 1130
Spinnaker::TransportLayerSystem, 1154
GigEVisionPerformance.cpp
AcquireImages, 1410
argBayerRG, 1412
argDuration, 1412
argMaxFrames, 1412
argNumImages, 1412
argPacketDelay, 1412
argPacketSize, 1412
argPrintUsage, 1413
argRelease, 1413

argUserSetFrames, 1413
cpuUsageInfo, 1413
EnableManualFramerate, 1410
getCameraCategory, 1410
IsRelease, 1413
main, 1410
NumImagesToGrab, 1413
PacketDelayToSet, 1413
PacketSizeToSet, 1413
ParseArguments, 1411
PixelFormatToSet, 1414
PrintAllNodes, 1411
PrintCPUUsage, 1411
PrintDataStreamInfo, 1411
PrintDeviceInfo, 1411
PrintUsage, 1411
RunSingleCamera, 1411
SetFrameRate, 1412
TestDuration, 1414
UseDuration, 1414
UseMaxFramerate, 1414
UserSetFramerate, 1414
globalCamList
AcquisitionMultipleCameraRecovery.cpp, 1384
GrabInfo, 833
GrabInfo, 833
imageEventHandler, 833
numImagesGrabbed, 833
numIncompleteImages, 834
numRemovals, 834
GrabNextImageByTrigger
BufferHandling.cpp, 1389
Trigger.cpp, 1451
Trigger_QuickSpin.cpp, 1453
GrabTwoImages
LogicBlock.cpp, 1431
GuiXmlManifestAddress
Spinnaker::Camera, 591
Guru
Types Enums, 384

H264Option, 844
Spinnaker::Video::H264Option, 845
HasCRC
Spinnaker::GenApi::CChunkAdapterDcam, 654
Spinnaker::IImage, 893
Spinnaker::Image, 938
HasInc
IFloat Interface, 306
Spinnaker::GenApi::FloatNode, 817
Header
GVCP_EVENT_REQUEST_EXTENDED_ID, 840
GVCP_EVENT_REQUEST, 839
GVCP_EVENTDATA_REQUEST_EXTENDED_↩ ID, 843
GVCP_EVENTDATA_REQUEST, 841
HeatmapColor
Spinnaker::ImageUtilityHeatmap, 965
Height
Spinnaker::Camera, 592
height
Spinnaker::Video::H264Option, 845
HeightMax
Spinnaker::Camera, 592
HighDynamicRange.cpp
CheckNodeAccessibility, 1415
InitializeHDRImages, 1415
k_HDRGain1, 1416
k_HDRGain2, 1416
k_HDRGain3, 1417
k_HDRGain4, 1417
k_HDRShutter1, 1417
k_HDRShutter2, 1417
k_HDRShutter3, 1417
k_HDRShutter4, 1417
main, 1415
PrintBuildInfo, 1416
PrintDeviceInfo, 1416
RunSingleCamera, 1416
ToggleHDRMode, 1416
HostAdapterDriverVersion
Spinnaker::TransportLayerInterface, 1140
HostAdapterName
Spinnaker::TransportLayerInterface, 1140
HostAdapterVendor
Spinnaker::TransportLayerInterface, 1140

IBase
IBase Interface, 251
IBase Interface, 251
IBase, 251
IBoolean
IBoolean Interface, 287
IBoolean Interface, 286
GetValue, 286
IBoolean, 287
operator(), 287
operator=, 287
Verify, 287
ICameraBase, 846
Spinnaker::ICameraBase, 848
Spinnaker::TransportLayerDevice, 1124
Spinnaker::TransportLayerStream, 1145
ICameraList, 855
Spinnaker::ICameraList, 856
ICategory
ICategory Interfaces, 288
ICategory Interfaces, 288
ICategory, 288
IChunkData, 859
Spinnaker::IChunkData, 860
IChunkData Class, 231
IChunkPort
IChunkPort Interface, 290
IChunkPort Interface, 289
CHUNK_BASE_ADDRESS_REGISTER_LEN, 289
CHUNK_BASE_ADDRESS_REGISTER, 289
CHUNK_LENGTH_REGISTER_LEN, 290

CHUNK_LENGTH_REGISTER, 290
CacheChunkData, 290
IChunkPort, 290
ICommand
ICommand Interface, 291
ICommand Interface, 291
ICommand, 291
IsDone, 291
IDataStream, 868
Spinnaker::EventHandler, 800
Spinnaker::IDataStream, 869
Spinnaker::Image, 943
IDestroy
IDestroy Interface, 293
IDestroy Interface, 293
IDestroy, 293
IDevFileStream
Spinnaker::GenApi, 448
IDevFileStreamBase< CharType, Traits >, 874
IDevFileStreamBuf
Spinnaker::GenApi::IDevFileStreamBuf, 878
IDevFileStreamBuf< CharType, Traits >, 877
IDeviceArrivalEventHandler, 879
Spinnaker::IDeviceArrivalEventHandler, 880
IDeviceEventHandler, 881
Spinnaker::IDeviceEventHandler, 882
IDeviceInfo
IDeviceInfo Interface, 296
IDeviceInfo Interface, 294
GetDeviceVersion, 294
GetGenApiVersion, 294
GetProductGuid, 295
GetSchemaVersion, 295
GetStandardNameSpace, 295
GetToolTip, 295
GetVendorName, 295
GetVersionGuid, 295
IDeviceInfo, 296
IDeviceRemovalEventHandler, 883
Spinnaker::IDeviceRemovalEventHandler, 884
IEnumEntry
IEnumEntry Interface, 298
IEnumEntry Interface, 297
GetNumericValue, 297
GetSymbolic, 297
IEnumEntry, 298
IsSelfClearing, 297
IEnumReference
IEnumerationT Interface, 303
IEnumeration
IEnumeration Interface, 301
IEnumeration Interface, 299
GetCurrentEntry, 299
GetEntries, 299
GetEntry, 300
GetEntryByName, 300
GetIntValue, 300
IEnumeration, 301
operator∗, 300
SetIntValue, 301
IEnumerationT Interface, 302
GetEntry, 302
IEnumReference, 303
IEnumerationT, 303
operator=, 302, 303
IEnumerationT
IEnumerationT Interface, 303
IFloat
IFloat Interface, 307
IFloat Interface, 304
GetDisplayNotation, 305
GetDisplayPrecision, 305
GetInc, 305
GetIncMode, 305
GetListOfValidValues, 305
GetMax, 305
GetMin, 306
GetRepresentation, 306
GetUnit, 306
HasInc, 306
IFloat, 307
ImposeMax, 306
ImposeMin, 306
operator=, 307
IImage, 885
Spinnaker::IImage, 887
IImage Class, 232
IImageEventHandler, 897
Spinnaker::IImageEventHandler, 898
IImageStatistics, 899
Spinnaker::IImageStatistics, 900
IImageStatistics Class, 233
IInteger
IInteger Interface, 309
IInteger Interface, 308
IInteger, 309
ImposeMax, 308
ImposeMin, 308
operator=, 308
IInterface, 903
Spinnaker::IInterface, 904, 905
Spinnaker::TransportLayerInterface, 1134
IInterface Class, 234
IInterfaceArrivalEventHandler, 908
Spinnaker::IInterfaceArrivalEventHandler, 909
IInterfaceEventHandler, 910
Spinnaker::IInterfaceEventHandler, 911
IInterfaceList, 913
Spinnaker::IInterfaceList, 914
IInterfaceList Class, 235
IInterfaceRemovalEventHandler, 916
Spinnaker::IInterfaceRemovalEventHandler, 917
ILoggingEventHandler, 918
Spinnaker::ILoggingEventHandler, 919
INTEGRAL_CAST2
GCUtilities Utility, 283

INTEGRAL_CAST
GCUtilities Utility, 283
INode
INode Interface, 320
INode Interface, 310
Combine, 312
DeregisterCallback, 312
GetAlias, 313
GetCachingMode, 313
GetCastAlias, 313
GetChildren, 313
GetDescription, 314
GetDisplayName, 314
GetDocuURL, 314
GetEventID, 314
GetNameSpace, 314
GetNodeMap, 314
GetParents, 314
GetPollingTime, 315
GetPrincipalInterfaceType, 315
GetProperty, 315
GetPropertyNames, 315
GetVisibility, 315
INode, 320
IReference, 320
ImposeAccessMode, 315
ImposeVisibility, 316
InvalidateNode, 316
IsAccessModeCacheable, 316
IsAvailable, 316
IsCachable, 317
IsCacheable, 317
IsDeprecated, 317
IsFeature, 317
IsImplemented, 317, 318
IsReadable, 318
IsStreamable, 318
IsVisible, 318
IsWritable, 319
operator!=, 319
operator==, 319
RegisterCallback, 319
INodeMap
INodeMap Interface, 323
INodeMap Interface, 321
Connect, 321, 322
GetDeviceName, 322
GetLock, 322
GetNode, 322
GetNumNodes, 322
INodeMap, 323
InvalidateNodes, 323
Poll, 323
INodeMapDyn
INodeMapDyn Interface, 328
INodeMapDyn Interface, 324
ExtractIndependentSubtree, 325
GetSupportedSchemaVersions, 325
INodeMapDyn, 328
LoadXMLFromFile, 325
LoadXMLFromFileInject, 325
LoadXMLFromString, 326
LoadXMLFromStringInject, 326
LoadXMLFromZIPData, 326
LoadXMLFromZIPFile, 326
MergeXMLFiles, 326
PreprocessXMLFromFile, 327
PreprocessXMLFromZIPFile, 327
IPersistScript
Spinnaker::GenApi, 450
IPort
IPort Interface, 332
IPort Interface, 331
Address, 331
IPort, 332
Length, 332
Write, 331
IPortConstruct
IPortConstruct Interface, 333
IPortConstruct Interface, 333
GetSwapEndianess, 333
IPortConstruct, 333
IPortRecorder
IPortRecorder Interface, 335
IPortRecorder Interface, 334
GetCookie, 334
IPortRecorder, 335
IPortReplay, 335
IPortWriteList, 335
Invalidate, 335
Replay, 334
SetCookie, 335
StopRecording, 335
IPortReplay
IPortRecorder Interface, 335
IPortWriteList
IPortRecorder Interface, 335
IReference
INode Interface, 320
IRegister
IRegister Interfaces, 337
IRegister Interfaces, 336
Get, 336
GetAddress, 337
GetLength, 337
IRegister, 337
ISelector
ISelector Interface, 338
ISelector Interface, 338
GetSelectedFeatures, 338
GetSelectingFeatures, 338
ISelector, 338
ISelectorDigit
ISelectorDigit Interface, 340
ISelectorDigit Interface, 339
GetSelectorList, 339

ISelectorDigit, 340
Restore, 340
SetNext, 340
ToString, 340
IString
IString Class, 342
IString Class, 342
GetMaxLength, 342
IString, 342
ISystem, 1014
Spinnaker::ISystem, 1015
Spinnaker::TransportLayerSystem, 1152
ISystem Class, 236
ISystemEventHandler, 1020
Spinnaker::ISystemEventHandler, 1021
IValue
IValue Class, 344
IValue Class, 343
FromString, 343
IValue, 344
IsValueCacheValid, 344
ToString, 344
Image, 920
Spinnaker::Image, 924
Image Class, 182
Image Utility Class, 186
Image Utility Heatmap Class, 187
Image Utility Polarization Class, 188
ImageComponentEnable
Spinnaker::Camera, 592
ImageComponentSelector
Spinnaker::Camera, 592
ImageComponentSelectorEnums
CameraDefs Class, 126
ImageCompressionBitrate
Spinnaker::Camera, 592
ImageCompressionJPEGFormatOption
Spinnaker::Camera, 592
ImageCompressionJPEGFormatOptionEnums
CameraDefs Class, 126
ImageCompressionMode
Spinnaker::Camera, 593
ImageCompressionModeEnums
CameraDefs Class, 127
ImageCompressionQuality
Spinnaker::Camera, 593
ImageCompressionRateOption
Spinnaker::Camera, 593
ImageCompressionRateOptionEnums
CameraDefs Class, 127
ImageConverter
Spinnaker::Image, 943
ImageEventHandler, 944
Spinnaker::ImageEventHandler, 945
imageEventHandler
GrabInfo, 833
ImageEventHandler Class, 183
ImageEventHandlerImpl, 947
~ImageEventHandlerImpl, 948
getImageCount, 948
getMaxImages, 949
ImageEventHandlerImpl, 948
OnImageEvent, 949
ImageEvents.cpp
AcquireImages, 1418
ConfigureImageEvents, 1418
main, 1418
PrintDeviceInfo, 1419
ResetImageEvents, 1419
RunSingleCamera, 1419
SleepyWrapper, 1419
WaitForImages, 1419
ImageFileFormat
Spinnaker Definitions, 207
ImageFiler
Spinnaker::Image, 943
ImageFormatControl.cpp
AcquireImages, 1420
ConfigureCustomImageSettings, 1420
main, 1420
PrintDeviceInfo, 1420
RunSingleCamera, 1421
ImageFormatControl_QuickSpin.cpp
AcquireImages, 1421
ConfigureCustomImageSettings, 1421
main, 1422
PrintDeviceInfo, 1422
RunSingleCamera, 1422
ImagePtr, 950
Spinnaker::ImagePtr, 951
ImagePtr Class, 184
ImageScalingAlgorithm
Spinnaker::ImageUtility, 960
ImageStatistics, 952
Spinnaker::ImageStatistics, 954
ImageStatistics Class, 185
ImageStatsCalculator
Spinnaker::Image, 943
Spinnaker::ImageStatistics, 959
ImageStatus
Spinnaker Definitions, 208
ImageUtility, 959
ImageUtilityHeatmap, 964
ImageUtilityImpl
Spinnaker::Image, 944
ImageUtilityPolarization, 968
ImageUtilityPolarizationImpl
Spinnaker::Image, 944
ImposeAccessMode
INode Interface, 315
Spinnaker::GenApi::Node, 1049
ImposeMax
IFloat Interface, 306
IInteger Interface, 308
Spinnaker::GenApi::FloatNode, 817
Spinnaker::GenApi::IntegerNode, 985

ImposeMin
IFloat Interface, 306
IInteger Interface, 308
Spinnaker::GenApi::FloatNode, 817
Spinnaker::GenApi::IntegerNode, 985
ImposeVisibility
INode Interface, 316
Spinnaker::GenApi::Node, 1049
include/AVIRecorder.h, 1167
include/AdapterConfig.h, 1165
include/BasePtr.h, 1168
include/Camera.h, 1169
include/CameraBase.h, 1169
include/CameraDefs.h, 1170
include/CameraList.h, 1202
include/CameraPtr.h, 1203
include/ChunkData.h, 1203
include/ChunkDataInference.h, 1204
include/DeviceArrivalEventHandler.h, 1205
include/DeviceEventHandler.h, 1206
include/DeviceRemovalEventHandler.h, 1207
include/EventHandler.h, 1208
include/Exception.h, 1208
include/Image.h, 1209
include/ImageEventHandler.h, 1210
include/ImagePtr.h, 1210
include/ImageStatistics.h, 1211
include/ImageUtility.h, 1212
include/ImageUtilityHeatmap.h, 1213
include/ImageUtilityPolarization.h, 1214
include/Interface.h, 1215
include/Interface/ICameraBase.h, 1215
include/Interface/ICameraList.h, 1216
include/Interface/IChunkData.h, 1217
include/Interface/IDeviceArrivalEventHandler.h, 1218
include/Interface/IDeviceEventHandler.h, 1219
include/Interface/IDeviceRemovalEventHandler.h, 1220
include/Interface/IImage.h, 1221
include/Interface/IImageEventHandler.h, 1221
include/Interface/IImageStatistics.h, 1222
include/Interface/IInterface.h, 1223
include/Interface/IInterfaceArrivalEventHandler.h, 1224
include/Interface/IInterfaceEventHandler.h, 1225
include/Interface/IInterfaceList.h, 1225
include/Interface/IInterfaceRemovalEventHandler.↩
h, 1226
include/Interface/ILoggingEventHandler.h, 1227
include/Interface/IStream.h, 1227
include/Interface/ISystem.h, 1228
include/Interface/ISystemEventHandler.h, 1229
include/InterfaceArrivalEventHandler.h, 1230
include/InterfaceEventHandler.h, 1231
include/InterfaceList.h, 1232
include/InterfacePtr.h, 1232
include/InterfaceRemovalEventHandler.h, 1233
include/LoggingEventData.h, 1234
include/LoggingEventDataPtr.h, 1234
include/LoggingEventHandler.h, 1235
include/SpinGenApi/Autovector.h, 1236
include/SpinGenApi/Base.h, 1237
include/SpinGenApi/BooleanNode.h, 1238
include/SpinGenApi/CategoryNode.h, 1239
include/SpinGenApi/ChunkAdapter.h, 1240
include/SpinGenApi/ChunkAdapterDcam.h, 1241
include/SpinGenApi/ChunkAdapterGEV.h, 1243
include/SpinGenApi/ChunkAdapterGeneric.h, 1242
include/SpinGenApi/ChunkAdapterU3V.h, 1244
include/SpinGenApi/ChunkPort.h, 1245
include/SpinGenApi/CommandNode.h, 1245
include/SpinGenApi/Compatibility.h, 1246
include/SpinGenApi/Container.h, 1247
include/SpinGenApi/Counter.h, 1247
include/SpinGenApi/EnumClasses.h, 1248
include/SpinGenApi/EnumEntryNode.h, 1249
include/SpinGenApi/EnumNode.h, 1250
include/SpinGenApi/EnumNodeT.h, 1251
include/SpinGenApi/EventAdapter.h, 1251
include/SpinGenApi/EventAdapter1394.h, 1252
include/SpinGenApi/EventAdapterGEV.h, 1253
include/SpinGenApi/EventAdapterGeneric.h, 1253
include/SpinGenApi/EventAdapterU3V.h, 1254
include/SpinGenApi/EventPort.h, 1255
include/SpinGenApi/Filestream.h, 1256
include/SpinGenApi/FloatNode.h, 1257
include/SpinGenApi/FloatRegNode.h, 1258
include/SpinGenApi/GCBase.h, 1259
include/SpinGenApi/GCString.h, 1259
include/SpinGenApi/GCStringVector.h, 1261
include/SpinGenApi/GCSynch.h, 1262
include/SpinGenApi/GCTypes.h, 1263
include/SpinGenApi/GCUtilities.h, 1266
include/SpinGenApi/IBoolean.h, 1270
include/SpinGenApi/ICategory.h, 1271
include/SpinGenApi/IChunkPort.h, 1272
include/SpinGenApi/ICommand.h, 1273
include/SpinGenApi/IDestroy.h, 1274
include/SpinGenApi/IDeviceInfo.h, 1275
include/SpinGenApi/IEnumEntry.h, 1276
include/SpinGenApi/IEnumeration.h, 1277
include/SpinGenApi/IEnumerationT.h, 1278
include/SpinGenApi/IFloat.h, 1279
include/SpinGenApi/IInteger.h, 1281
include/SpinGenApi/INode.h, 1282
include/SpinGenApi/INodeMap.h, 1285
include/SpinGenApi/INodeMapDyn.h, 1287
include/SpinGenApi/IPort.h, 1290
include/SpinGenApi/IPortConstruct.h, 1291
include/SpinGenApi/IPortRecorder.h, 1292
include/SpinGenApi/IRegister.h, 1293
include/SpinGenApi/ISelector.h, 1294
include/SpinGenApi/ISelectorDigit.h, 1295
include/SpinGenApi/IString.h, 1296
include/SpinGenApi/IValue.h, 1297
include/SpinGenApi/IntRegNode.h, 1289
include/SpinGenApi/IntegerNode.h, 1288
include/SpinGenApi/Node.h, 1298

include/SpinGenApi/NodeCallback.h, 1299
include/SpinGenApi/NodeCallbackImpl.h, 1301
include/SpinGenApi/NodeMap.h, 1301
include/SpinGenApi/NodeMapFactory.h, 1302
include/SpinGenApi/NodeMapRef.h, 1303
include/SpinGenApi/Persistence.h, 1304
include/SpinGenApi/Pointer.h, 1305
include/SpinGenApi/PortImpl.h, 1307
include/SpinGenApi/PortNode.h, 1308
include/SpinGenApi/PortRecorder.h, 1309
include/SpinGenApi/PortReplay.h, 1309
include/SpinGenApi/PortWriteList.h, 1310
include/SpinGenApi/Reference.h, 1311
include/SpinGenApi/RegisterNode.h, 1312
include/SpinGenApi/RegisterPortImpl.h, 1313
include/SpinGenApi/SelectorSet.h, 1313
include/SpinGenApi/SpinTestCamera.h, 1314
include/SpinGenApi/SpinnakerGenApi.h, 1314
include/SpinGenApi/StringNode.h, 1315
include/SpinGenApi/StringRegNode.h, 1316
include/SpinGenApi/StructPort.h, 1316
include/SpinGenApi/Synch.h, 1317
include/SpinGenApi/Types.h, 1318
include/SpinGenApi/ValueNode.h, 1321
include/SpinUpdate.h, 1327
include/SpinVideo.h, 1329
include/SpinVideoDefs.h, 1330
include/Spinnaker.h, 1322
include/SpinnakerDefs.h, 1322
include/SpinnakerPlatform.h, 1326
include/System.h, 1331
include/SystemEventHandler.h, 1333
include/SystemPtr.h, 1334
include/TransportLayerDefs.h, 1334
include/TransportLayerDevice.h, 1336
include/TransportLayerInterface.h, 1337
include/TransportLayerStream.h, 1337
include/TransportLayerSystem.h, 1338
IncompatibleDeviceCount
    Spinnaker::TransportLayerInterface, 1140
IncompatibleDeviceID
    Spinnaker::TransportLayerInterface, 1140
IncompatibleDeviceModelName
    Spinnaker::TransportLayerInterface, 1141
IncompatibleDeviceSelector
    Spinnaker::TransportLayerInterface, 1141
IncompatibleDeviceVendorName
    Spinnaker::TransportLayerInterface, 1141
IncompatibleGevDeviceIPAddress
    Spinnaker::TransportLayerInterface, 1141
IncompatibleGevDeviceMACAddress
    Spinnaker::TransportLayerInterface, 1141
IncompatibleGevDeviceSubnetMask
    Spinnaker::TransportLayerInterface, 1141
Increasing
    Types Enums, 383
Indent
    NodeMapInfo.cpp, 1436
indexedColor_8bit
    Spinnaker::BMPOption, 469
Inference.cpp
    AcquireImages, 1424
    arrayLabelClassification, 1427
    arrayLabelDetection, 1427
    CameraCloseFile, 1424
    CameraDeleteFile, 1424
    CameraOpenFile, 1424
    CameraWriteToFile, 1424
    chosenFileUploadPersistence, 1428
    chosenInferenceNetworkType, 1428
    ConfigureChunkData, 1425
    ConfigureInference, 1425
    ConfigureTestPattern, 1425
    ConfigureTrigger, 1425
    DeleteFileOnCamera, 1425
    DisableChunkData, 1425
    DisableTrigger, 1426
    DisplayChunkData, 1426
    FileUploadPersistence, 1423
    InferenceNetworkType, 1423
    injectedImageFilePath, 1428
    injectedImageHeight, 1428
    injectedImageWidth, 1428
    labelClassification, 1426
    labelDetection, 1426
    LoadFileIntoMemory, 1426
    main, 1426
    networkFilePath, 1429
    PrintDeviceInfo, 1427
    RunSingleCamera, 1427
    SetChunkEnable, 1427
    UploadFileToCamera, 1427
InferenceBoundingBox, 976
InferenceBoundingBoxResult, 976
    Chunk Data Inference Class, 171
InferenceBoxCircle, 977
InferenceBoxRect, 977
InferenceBoxRotatedRect, 978
InferenceBoxType
    Spinnaker Classes, 43
InferenceNetworkType
    Inference.cpp, 1423
Init
    Spinnaker::Camera, 503
    Spinnaker::CameraBase, 634
    Spinnaker::ICameraBase, 851
InitChunkAdapter
    Spinnaker::IDataStream, 872
InitializeHDRImages
    HighDynamicRange.cpp, 1415
InitializeSystem
    FileAccess_QuickSpin.cpp, 1404
injectedImageFilePath
    Inference.cpp, 1428
injectedImageHeight
    Inference.cpp, 1428

injectedImageWidth
Inference.cpp, 1428
int64_autovector_t, 978
Spinnaker::GenApi::int64_autovector_t, 978, 979
IntRegNode, 1010
Spinnaker::GenApi::IntRegNode, 1011, 1012
IntRegNode Class, 330
IntegerNode, 981
Spinnaker::GenApi::IntegerNode, 983
IntegerNode Class, 329
CIntegerRef, 329
Interface, 987
Spinnaker::TransportLayerInterface, 1134
interface
Types.h, 1321
Interface Class, 189
InterfaceArrivalEventHandler, 992
Spinnaker::InterfaceArrivalEventHandler, 993
InterfaceArrivalEventHandler Class, 190
InterfaceDisplayName
Spinnaker::TransportLayerInterface, 1142
Spinnaker::TransportLayerSystem, 1154
InterfaceEventHandler, 995
Spinnaker::InterfaceEventHandler, 996
InterfaceEventHandler Class, 191
InterfaceEventHandlerImpl, 998
~InterfaceEventHandlerImpl, 1000
GetInterfaceId, 1000
InterfaceEventHandlerImpl, 999, 1000
OnDeviceArrival, 1000, 1001
OnDeviceRemoval, 1001
PrintGenericHandlerMessage, 1002
InterfaceID
Spinnaker::TransportLayerInterface, 1142
Spinnaker::TransportLayerSystem, 1155
InterfaceImpl
Spinnaker::CameraBase, 639
Spinnaker::ICameraBase, 854
Spinnaker::ICameraList, 858
InterfaceInternal
Spinnaker::IInterface, 906
Spinnaker::Interface, 992
Spinnaker::TransportLayerInterface, 1134
InterfaceList, 1002
Spinnaker::InterfaceList, 1003
InterfaceList Class, 192
InterfaceListImpl
Spinnaker::IInterfaceList, 915
InterfacePtr, 1006
Spinnaker::InterfacePtr, 1007
InterfacePtr Class, 193
InterfaceRemovalEventHandler, 1008
Spinnaker::InterfaceRemovalEventHandler, 1009
InterfaceRemovalEventHandler Class, 194
InterfaceSelector
Spinnaker::TransportLayerSystem, 1155
InterfaceType
Spinnaker::TransportLayerInterface, 1142
InterfaceTypeEnum
TransportLayerDefs Class, 222
InterfaceUpdateList
Spinnaker::TransportLayerSystem, 1155
interlaced
Spinnaker::PNGOption, 1068
Invalidate
IPortRecorder Interface, 335
InvalidateNode
INode Interface, 316
Spinnaker::GenApi::CChunkPort, 666
Spinnaker::GenApi::CEventPort, 687
Spinnaker::GenApi::CPortImpl, 748
Spinnaker::GenApi::Node, 1049
InvalidateNodes
INodeMap Interface, 323
Spinnaker::GenApi::NodeMap, 1059
InverseChunkLength
DCAM_CHUNK_TRAILER, 764
Invisible
Types Enums, 384
ios_type
Spinnaker::GenApi::IDevFileStreamBase, 875
Spinnaker::GenApi::ODevFileStreamBase, 1063
ipAddress
AdapterConfig::IpInfo, 1013
IpInfo, 1012
AdapterConfig::IpInfo, 1013
ipInfo
AdapterConfig::AdapterInfo, 459
is_open
Spinnaker::GenApi::IDevFileStreamBase, 876
Spinnaker::GenApi::IDevFileStreamBuf, 878
Spinnaker::GenApi::ODevFileStreamBase, 1063
Spinnaker::GenApi::ODevFileStreamBuf, 1065
IsAccessModeCacheable
INode Interface, 316
Spinnaker::GenApi::Node, 1049
IsAvailable
INode Interface, 316
Pointer Class, 359
IsCRCCheckEnabled
Spinnaker::IDataStream, 872
IsCachable
INode Interface, 317
Spinnaker::GenApi::Node, 1049
IsCacheable
INode Interface, 317
IsCameraDescriptionFileDataReleased
Spinnaker::GenApi::CNodeMapFactory, 727
IsCompressed
Spinnaker::Image, 938
IsDeprecated
INode Interface, 317
Spinnaker::GenApi::Node, 1050
IsDone
ICommand Interface, 291
Spinnaker::GenApi::CommandNode, 739

IsEmpty
Spinnaker::GenApi::CNodeMapFactory, 727
Spinnaker::GenApi::CSelectorSet, 758
IsFeature
INode Interface, 317
Spinnaker::GenApi::Node, 1050
IsImageInUse
Spinnaker::IDataStream, 872
IsImplemented
INode Interface, 317, 318
Pointer Class, 359
IsInUse
Spinnaker::IImage, 894
Spinnaker::IInterface, 905
Spinnaker::ISystem, 1017
Spinnaker::Image, 938
Spinnaker::Interface, 989
Spinnaker::System, 1104
IsIncomplete
Spinnaker::IImage, 893
Spinnaker::Image, 938
IsInitialized
Spinnaker::CameraBase, 635
Spinnaker::ICameraBase, 851
IsLoaded
Spinnaker::GenApi::CNodeMapFactory, 727
IsOnSameSubnet
AdapterConfig, 390
isPixelFormatColor
Polarization.cpp, 1442
IsPreprocessed
Spinnaker::GenApi::CNodeMapFactory, 727
IsReadable
INode Interface, 318
Pointer Class, 359
IsRelease
GigEVisionPerformance.cpp, 1413
IsSelector
Spinnaker::GenApi::Node, 1050
IsSelfClearing
IEnumEntry Interface, 297
Spinnaker::GenApi::EnumEntryNode, 787
IsStreamable
INode Interface, 318
Spinnaker::GenApi::Node, 1050
IsStreaming
Spinnaker::CameraBase, 635
Spinnaker::ICameraBase, 852
Spinnaker::IDataStream, 872
IsValid
Spinnaker::BasePtr, 466
Spinnaker::CameraBase, 635
Spinnaker::GenApi::CPointer, 743
Spinnaker::GenICam::CGlobalLock, 695
Spinnaker::ICameraBase, 852
Spinnaker::IInterface, 905
Spinnaker::Interface, 989
IsValidIpAddress
AdapterConfig, 390
IsValidSubnetMask
AdapterConfig, 390
IsValueCacheValid
IValue Class, 344
Spinnaker::GenApi::ValueNode, 1162
IsVisible
INode Interface, 318
IsWritable
INode Interface, 319
Pointer Class, 359
IsZero
Spinnaker::GenApi::Counter, 740
IspEnable
Spinnaker::Camera, 593
istream_type
Spinnaker::GenApi::IDevFileStreamBase, 875
Items
GVCP_EVENT_REQUEST_EXTENDED_ID, 840
GVCP_EVENT_REQUEST, 839

JPEGOption, 1022
Spinnaker::JPEGOption, 1022
JPG2Option, 1023
Spinnaker::JPG2Option, 1024
jumboPacketValidValues
AdapterConfig::AdapterInfo, 459
jumboPackets
AdapterConfig::AdapterInfo, 459
jumboPacketsRegKey
AdapterConfig::AdapterInfo, 459

k_HDRGain1
HighDynamicRange.cpp, 1416
k_HDRGain2
HighDynamicRange.cpp, 1416
k_HDRGain3
HighDynamicRange.cpp, 1417
k_HDRGain4
HighDynamicRange.cpp, 1417
k_HDRShutter1
HighDynamicRange.cpp, 1417
k_HDRShutter2
HighDynamicRange.cpp, 1417
k_HDRShutter3
HighDynamicRange.cpp, 1417
k_HDRShutter4
HighDynamicRange.cpp, 1417
k_LoggingLevel
Logging.cpp, 1430
k_numLoops
BufferHandling.cpp, 1388
KillBufferEvent
Spinnaker::IDataStream, 872

LUTEnable
Spinnaker::Camera, 597
LUTIndex
Spinnaker::Camera, 597

LUTSelector
Spinnaker::Camera, 597
LUTSelectorEnums
CameraDefs Class, 131
LUTValue
Spinnaker::Camera, 597
LUTValueAll
Spinnaker::Camera, 598
labelClassification
Inference.cpp, 1426
labelDetection
Inference.cpp, 1426
Length
GVCP_REQUEST_HEADER, 843
IPort Interface, 332
U3V_COMMAND_HEADER, 1158
length
Spinnaker::GenICam::gcstring, 829
LibraryVersion, 1025
LineFilterWidth
Spinnaker::Camera, 593
LineFormat
Spinnaker::Camera, 594
LineFormatEnums
CameraDefs Class, 127
LineInputFilterSelector
Spinnaker::Camera, 594
LineInputFilterSelectorEnums
CameraDefs Class, 128
LineInverter
Spinnaker::Camera, 594
LineMode
Spinnaker::Camera, 594
LineModeEnums
CameraDefs Class, 128
LinePitch
Spinnaker::Camera, 594
LineSelector
Spinnaker::Camera, 594
LineSelectorEnums
CameraDefs Class, 128
LineSource
Spinnaker::Camera, 595
LineSourceEnums
CameraDefs Class, 129
LineStatus
Spinnaker::Camera, 595
LineStatusAll
Spinnaker::Camera, 595
Linear
Types Enums, 382
LinkErrorCount
Spinnaker::Camera, 595
LinkUptime
Spinnaker::Camera, 595
LoadAndInject
Spinnaker::GenApi::CNodeMapFactory, 727
LoadFileIntoMemory
Inference.cpp, 1426
LoadFromBag
Spinnaker::GenApi::CFeatureBag, 689
LoadXMLFromFile
INodeMapDyn Interface, 325
Spinnaker::GenApi::NodeMap, 1059
LoadXMLFromFileInject
INodeMapDyn Interface, 325
Spinnaker::GenApi::NodeMap, 1059
LoadXMLFromString
INodeMapDyn Interface, 326
Spinnaker::GenApi::NodeMap, 1059
LoadXMLFromStringInject
INodeMapDyn Interface, 326
Spinnaker::GenApi::NodeMap, 1059
LoadXMLFromZIPData
INodeMapDyn Interface, 326
Spinnaker::GenApi::NodeMap, 1060
LoadXMLFromZIPFile
INodeMapDyn Interface, 326
Spinnaker::GenApi::NodeMap, 1060
Lock
Spinnaker::GenApi::CLock, 714
Spinnaker::GenICam::CGlobalLock, 695
Spinnaker::GenICam::CLock, 712
Spinnaker::GenICam::LockableObject, 1028
Spinnaker::GenICam::LockableObject::Lock, 1026
LockEventHandlerMutex
SystemEventHandlerImpl, 1115
LockableObject< Object >, 1027
LockableObject< Object >::Lock, 1026
Logarithmic
Types Enums, 382
Logging EventHandler Class, 195
Logging.cpp
k_LoggingLevel, 1430
main, 1429
LoggingEventData, 1028
Spinnaker::LoggingEventData, 1029
LoggingEventDataPtr, 1032
Spinnaker::LoggingEventDataPtr, 1033
LoggingEventDataPtr Class, 196
LoggingEventHandler, 1034
Spinnaker::LoggingEventHandler, 1035
LoggingEventHandler Class, 197
LoggingEventHandlerImpl, 1036
LogicBlock.cpp
AcquireImages, 1430
ConfigureLogicBlock, 1430
ConfigureTrigger, 1431
GrabTwoImages, 1431
main, 1431
PrintDeviceInfo, 1431
ResetExposure, 1431
ResetTrigger, 1431
RunSingleCamera, 1431
LogicBlockLUTInputActivation
Spinnaker::Camera, 595

LogicBlockLUTInputActivationEnums
CameraDefs Class, 129
LogicBlockLUTInputSelector
Spinnaker::Camera, 596
LogicBlockLUTInputSelectorEnums
CameraDefs Class, 130
LogicBlockLUTInputSource
Spinnaker::Camera, 596
LogicBlockLUTInputSourceEnums
CameraDefs Class, 130
LogicBlockLUTOutputValue
Spinnaker::Camera, 596
LogicBlockLUTOutputValueAll
Spinnaker::Camera, 596
LogicBlockLUTRowIndex
Spinnaker::Camera, 596
LogicBlockLUTSelector
Spinnaker::Camera, 596
LogicBlockLUTSelectorEnums
CameraDefs Class, 131
LogicBlockSelector
Spinnaker::Camera, 597
LogicBlockSelectorEnums
CameraDefs Class, 131
LookupTable.cpp
AcquireImages, 1432
ConfigureLookupTables, 1432
main, 1432
PrintDeviceInfo, 1433
PrintRetrieveNodeFailure, 1433
ResetLookupTables, 1433
RunSingleCamera, 1433

m_BaseAddress
Spinnaker::GenApi::CTestPortStruct, 763
m_CallbackType
Spinnaker::GenApi::CNodeCallback, 719
m_Callbacks
Spinnaker::GenApi::Node, 1052
m_DebugCount
Spinnaker::GenICam::CGlobalLock, 696
m_Lock
Spinnaker::GenICam::CGlobalLockUnlocker, 698
Spinnaker::GenICam::LockableObject, 1028
m_NumReads
Spinnaker::GenApi::CTestPortStruct, 763
m_NumWrites
Spinnaker::GenApi::CTestPortStruct, 763
m_bOwnLock
Spinnaker::GenApi::CLock, 714
m_enabled
Spinnaker::GenICam::CGlobalLockUnlocker, 698
m_lock
Spinnaker::GenApi::CLock, 715
m_lockEx
Spinnaker::GenApi::CLockEx, 717
m_pCameraBaseData
Spinnaker::ICameraBase, 854
m_pCameraListData
Spinnaker::ICameraList, 859
m_pChunkAdapter
Spinnaker::GenApi::CChunkAdapter, 652
m_pChunkPort
Spinnaker::GenApi::CChunkPort, 667
m_pEnumeration
Spinnaker::GenApi::EnumNode, 793
m_pEventAdapter
Spinnaker::GenApi::CEventAdapter, 674
m_pEventData
Spinnaker::EventHandler, 801
m_pEventPort
Spinnaker::GenApi::CEventPort, 687
m_pInterfaceData
Spinnaker::IInterface, 907
m_pInterfaceListData
Spinnaker::IInterfaceList, 915
m_pNode
Spinnaker::GenApi::CEventPort, 688
Spinnaker::GenApi::CNodeCallback, 719
m_pNodeData
Spinnaker::GenApi::Node, 1052
m_pNodeMap
Spinnaker::GenApi::Node, 1052
m_pPort
Spinnaker::GenApi::CChunkPort, 667
m_pPortAdapter
Spinnaker::GenApi::CChunkPort, 667
Spinnaker::GenApi::CEventPort, 688
m_pWriteList
Spinnaker::GenApi::CPortWriteList, 752
m_pT
Spinnaker::BasePtr, 468
Spinnaker::GenApi::CPointer, 746
m_ptrPort
Spinnaker::GenApi::CPortImpl, 749
MILLISECOND
SerialRxTx.cpp, 1447
MJPGOption, 1039
Spinnaker::Video::MJPGOption, 1040
Magic
GVCP_REQUEST_HEADER, 844
main
Acquisition.cpp, 1339
AcquisitionMultipleCameraRecovery.cpp, 1383
AcquisitionMultipleThread.cpp, 1384
ActionCommand.cpp, 1387
BufferHandling.cpp, 1389
ChunkData.cpp, 1391
CounterAndTimer.cpp, 1393
DeviceEvents.cpp, 1395
Enumeration.cpp, 1396
Enumeration_QuickSpin.cpp, 1397
EnumerationEvents.cpp, 1398
ExceptionHandling.cpp, 1399
Exposure.cpp, 1400
Exposure_QuickSpin.cpp, 1402
FileAccess_QuickSpin.cpp, 1404

GenTLInfo_QuickSpin.cpp, 1406
GigEVisionPerformance.cpp, 1410
HighDynamicRange.cpp, 1415
ImageEvents.cpp, 1418
ImageFormatControl.cpp, 1420
ImageFormatControl_QuickSpin.cpp, 1422
Inference.cpp, 1426
Logging.cpp, 1429
LogicBlock.cpp, 1431
LookupTable.cpp, 1432
NodeMapCallback.cpp, 1434
NodeMapInfo.cpp, 1436
Polarization.cpp, 1441
SaveToAvi.cpp, 1443
Sequencer.cpp, 1445
SerialRxTx.cpp, 1448
Trigger.cpp, 1451
Trigger_QuickSpin.cpp, 1453
Major
Spinnaker::GenICam::Version_t, 1164
major
Spinnaker::LibraryVersion, 1025
make_NodeCallback
NodeCallback Class, 347
max_size
Spinnaker::GenICam::gcstring, 829
maxChars
NodeMapInfo.cpp, 1439
MaxDeviceResetTime
Spinnaker::Camera, 598
MemSet
Spinnaker::GenApi::CTestPortStruct, 762
Member_NodeCallback
Spinnaker::GenApi::Member_NodeCallback, 1039
Member_NodeCallback< Client, Member >, 1037
MergeXMLFiles
INodeMapDyn Interface, 326
Minor
Spinnaker::GenICam::Version_t, 1164
minor
Spinnaker::LibraryVersion, 1025

NA
Types Enums, 378
networkFilePath
Inference.cpp, 1429
NI
Types Enums, 378
No
Types Enums, 384
Node, 1041
Spinnaker::GenApi::Node, 1044
Node Class, 345
NodeCallback Class, 346
Deregister, 347
ECallbackType, 347
make_NodeCallback, 347
Register, 348
NodeList_t
Spinnaker GenApi Interfaces, 250
NodeMap, 1052
Spinnaker::GenApi::CLock, 714
Spinnaker::GenApi::NodeMap, 1054
NodeMap Class, 349
NodeMapCallback.cpp
ChangeHeightAndGain, 1434
ConfigureCallbacks, 1434
main, 1434
OnGainNodeUpdate, 1434
OnHeightNodeUpdate, 1434
PrintDeviceInfo, 1435
ResetCallbacks, 1435
RunSingleCamera, 1435
NodeMapFactory Class, 350
ECacheUsage_t, 350
EContentType_t, 351
NodeMapInfo.cpp
chosenRead, 1438
Indent, 1436
main, 1436
maxChars, 1439
PrintBooleanNode, 1437
PrintCategoryNodeAndAllFeatures, 1437
PrintCommandNode, 1437
PrintEnumerationNodeAndCurrentEntry, 1437
PrintEnumerationSelector, 1437
PrintFloatNode, 1437
PrintIntegerNode, 1438
PrintNode, 1438
PrintStringNode, 1438
PrintValueNode, 1438
readType, 1436
RunSingleCamera, 1438
NodeMapRef Class, 352
None
Types Enums, 383
npos
Spinnaker::GenICam::gcstring, 833
NumAttachedChunks
AttachStatistics_t, 461
numBuffers
BufferHandling.cpp, 1388
NumChunkPorts
AttachStatistics_t, 461
NumChunks
AttachStatistics_t, 461
numImagesGrabbed
GrabInfo, 833
NumImagesToGrab
GigEVisionPerformance.cpp, 1413
numIncompleteImages
GrabInfo, 834
NumLinks
Spinnaker::GenApi::CNodeMapFactory::Node↩
Statistics_t, 1061
NumNodes

Spinnaker::GenApi::CNodeMapFactory::Node↩
Statistics_t, 1061
NumProperties
Spinnaker::GenApi::CNodeMapFactory::Node↩
Statistics_t, 1061
numRemovals
GrabInfo, 834
NumStrings
Spinnaker::GenApi::CNodeMapFactory::Node↩
Statistics_t, 1061
NumToCString
Conversion, 391

ODevFileStream
Spinnaker::GenApi, 448
ODevFileStreamBase< CharType, Traits >, 1062
ODevFileStreamBuf
Spinnaker::GenApi::ODevFileStreamBuf, 1065
ODevFileStreamBuf< CharType, Traits >, 1064
OffsetX
Spinnaker::Camera, 598
OffsetY
Spinnaker::Camera, 598
OnDeviceArrival
InterfaceEventHandlerImpl, 1000, 1001
Spinnaker::DeviceArrivalEventHandler, 766
Spinnaker::IDeviceArrivalEventHandler, 880
Spinnaker::IInterfaceEventHandler, 912
Spinnaker::InterfaceEventHandler, 996
OnDeviceEvent
DeviceEventHandlerImpl, 772
Spinnaker::DeviceEventHandler, 769
Spinnaker::IDeviceEventHandler, 883
OnDeviceRemoval
InterfaceEventHandlerImpl, 1001
Spinnaker::DeviceRemovalEventHandler, 774
Spinnaker::IDeviceRemovalEventHandler, 885
Spinnaker::IInterfaceEventHandler, 912
Spinnaker::InterfaceEventHandler, 997
OnGainNodeUpdate
NodeMapCallback.cpp, 1434
OnHeightNodeUpdate
NodeMapCallback.cpp, 1434
OnImageEvent
ImageEventHandlerImpl, 949
Spinnaker::IImageEventHandler, 898
Spinnaker::ImageEventHandler, 946
OnInterfaceArrival
Spinnaker::IInterfaceArrivalEventHandler, 909
Spinnaker::ISystemEventHandler, 1021
Spinnaker::InterfaceArrivalEventHandler, 994
Spinnaker::SystemEventHandler, 1112
SystemEventHandlerImpl, 1115
OnInterfaceRemoval
Spinnaker::IInterfaceRemovalEventHandler, 917
Spinnaker::ISystemEventHandler, 1021
Spinnaker::InterfaceRemovalEventHandler, 1009
Spinnaker::SystemEventHandler, 1112
SystemEventHandlerImpl, 1115
OnLogEvent
Spinnaker::ILoggingEventHandler, 919
Spinnaker::LoggingEventHandler, 1035
Open
Spinnaker::Video::SpinVideo, 1090, 1091
open
Spinnaker::GenApi::IDevFileStreamBase, 876
Spinnaker::GenApi::IDevFileStreamBuf, 878
Spinnaker::GenApi::ODevFileStreamBase, 1063
Spinnaker::GenApi::ODevFileStreamBuf, 1065
openFile
Spinnaker::GenApi::FileProtocolAdapter, 811
OpenFileToRead
FileAccess_QuickSpin.cpp, 1404
OpenFileToWrite
FileAccess_QuickSpin.cpp, 1405
operator bool
Spinnaker::BasePtr, 466
Spinnaker::GenApi::CPointer, 743
operator const char ∗
Spinnaker::GenICam::gcstring, 829
operator delete
Spinnaker::GenApi::double_autovector_t, 776
Spinnaker::GenApi::int64_autovector_t, 979
Spinnaker::GenICam::gcstring, 829
operator new
Spinnaker::GenApi::double_autovector_t, 776
Spinnaker::GenApi::int64_autovector_t, 979
Spinnaker::GenICam::gcstring, 829
operator T∗
Spinnaker::BasePtr, 466
Spinnaker::GenApi::CPointer, 743
operator unsigned int
Spinnaker::GenApi::Counter, 740
operator!=
INode Interface, 319
Spinnaker::Exception, 806
Spinnaker::GenApi::CPointer, 744
Spinnaker::GenApi::Node, 1050
Spinnaker::GenICam::gcstring, 830
operator<
Spinnaker::GenICam::gcstring, 831
operator<<
GCString.h, 1260
Spinnaker GenApi Classes, 246
operator>
Spinnaker::GenICam::gcstring, 831
operator>>
GCString.h, 1261
Spinnaker GenApi Classes, 247
operator∗
IEnumeration Interface, 300
Spinnaker::GenApi::CPointer, 745
Spinnaker::GenApi::EnumNode, 792
Spinnaker::GenApi::FloatNode, 818
Spinnaker::GenApi::IntegerNode, 986
Spinnaker::GenApi::StringNode, 1095
operator()

IBoolean Interface, 287
Spinnaker::GenApi::CEnumerationTRef, 670
Spinnaker::GenApi::CNodeCallback, 719
Spinnaker::GenApi::CPointer, 744
Spinnaker::GenApi::CommandNode, 739
Spinnaker::GenApi::FloatNode, 818
Spinnaker::GenApi::Function_NodeCallback, 823
Spinnaker::GenApi::IntegerNode, 985
Spinnaker::GenApi::Member_NodeCallback, 1039
Spinnaker::GenApi::StringNode, 1095
operator+
Spinnaker::GenICam::gcstring, 832
operator++
Spinnaker::GenApi::Counter, 740, 741
operator+=
Spinnaker::GenICam::gcstring, 830
operator->
Spinnaker::BasePtr, 466
Spinnaker::GenApi::CPointer, 745
operator--
Spinnaker::GenApi::Counter, 741
operator=
Chunk Data Inference Class, 171
IBoolean Interface, 287
IEnumerationT Interface, 302, 303
IFloat Interface, 307
IInteger Interface, 308
Spinnaker GenApi Classes, 247
Spinnaker::BasePtr, 467
Spinnaker::CameraBase, 636
Spinnaker::CameraList, 644
Spinnaker::DeviceArrivalEventHandler, 766
Spinnaker::DeviceEventHandler, 770
Spinnaker::DeviceRemovalEventHandler, 774
Spinnaker::EventHandler, 800
Spinnaker::Exception, 806
Spinnaker::GenApi::BooleanNode, 472
Spinnaker::GenApi::CEnumerationTRef, 671
Spinnaker::GenApi::CFloatPtr, 692
Spinnaker::GenApi::CNodeMapFactory, 727
Spinnaker::GenApi::CNodeMapRef, 731
Spinnaker::GenApi::CPointer, 745
Spinnaker::GenApi::EnumNode, 792
Spinnaker::GenApi::FloatNode, 818
Spinnaker::GenApi::IntegerNode, 986
Spinnaker::GenApi::StringNode, 1095
Spinnaker::GenApi::double_autovector_t, 776
Spinnaker::GenApi::int64_autovector_t, 979
Spinnaker::GenICam::gcstring, 831
Spinnaker::ICameraBase, 852
Spinnaker::ICameraList, 857
Spinnaker::IDeviceArrivalEventHandler, 880
Spinnaker::IDeviceEventHandler, 883
Spinnaker::IDeviceRemovalEventHandler, 885
Spinnaker::IImageEventHandler, 898
Spinnaker::IInterface, 905
Spinnaker::IInterfaceArrivalEventHandler, 909
Spinnaker::IInterfaceEventHandler, 912
Spinnaker::IInterfaceList, 915
Spinnaker::IInterfaceRemovalEventHandler, 917
Spinnaker::ILoggingEventHandler, 919
Spinnaker::ISystem, 1017
Spinnaker::ISystemEventHandler, 1022
Spinnaker::ImageEventHandler, 946
Spinnaker::ImagePtr, 952
Spinnaker::ImageStatistics, 958
Spinnaker::InterfaceArrivalEventHandler, 994
Spinnaker::InterfaceEventHandler, 997
Spinnaker::InterfaceList, 1005
Spinnaker::InterfaceRemovalEventHandler, 1010
Spinnaker::LoggingEventHandler, 1036
Spinnaker::SystemEventHandler, 1113
operator==
BasePtr Class, 47
INode Interface, 319
Spinnaker::BasePtr, 467, 468
Spinnaker::Exception, 807
Spinnaker::GenApi::CFeatureBag, 690
Spinnaker::GenApi::CPointer, 745
Spinnaker::GenApi::Node, 1050
Spinnaker::GenICam::gcstring, 831
operator[]
Spinnaker::CameraList, 644
Spinnaker::GenApi::double_autovector_t, 776
Spinnaker::GenApi::int64_autovector_t, 980
Spinnaker::ICameraList, 857
Spinnaker::IInterfaceList, 915
Spinnaker::InterfaceList, 1005
ostream_type
Spinnaker::GenApi::ODevFileStreamBase, 1063
overflow
Spinnaker::GenApi::ODevFileStreamBuf, 1066

PCFreq
PerformanceCounter, 393
PGMOption, 1066
Spinnaker::PGMOption, 1067
PMEMBERFUNC
Spinnaker::GenApi::Member_NodeCallback, 1038
PNGOption, 1067
Spinnaker::PNGOption, 1068
POEStatus
Spinnaker::TransportLayerInterface, 1142
POEStatusEnum
TransportLayerDefs Class, 223
PPMOption, 1081
Spinnaker::PPMOption, 1082
PacketDelayToSet
GigEVisionPerformance.cpp, 1413
PacketResendRequestCount
Spinnaker::Camera, 598
PacketSizeToSet
GigEVisionPerformance.cpp, 1413
ParseArguments
GigEVisionPerformance.cpp, 1411
PayloadSize
Spinnaker::Camera, 599

PayloadTypeInfoIDs
Spinnaker Definitions, 208
pbackfail
Spinnaker::GenApi::IDevFileStreamBuf, 878
PerformanceCounter, 392
CounterStart, 393
GetPerformanceCounter, 393
PCFreq, 393
StartPerformanceCounter, 393
PersistFeature
Spinnaker::GenApi, 449
Spinnaker::GenApi::CFeatureBag, 690
Persistence Class, 353
PixelColorFilter
Spinnaker::Camera, 599
PixelColorFilterEnums
CameraDefs Class, 132
PixelDynamicRangeMax
Spinnaker::Camera, 599
PixelDynamicRangeMin
Spinnaker::Camera, 599
PixelFormat
Spinnaker::Camera, 599
PixelFormatEnums
CameraDefs Class, 132
PixelFormatInfoID
Spinnaker::Camera, 600
PixelFormatInfoSelector
Spinnaker::Camera, 600
PixelFormatInfoSelectorEnums
CameraDefs Class, 138
PixelFormatIntType
Spinnaker Definitions, 209
PixelFormatNamespaceID
Spinnaker Definitions, 209
PixelFormatToSet
GigEVisionPerformance.cpp, 1414
PixelSize
Spinnaker::Camera, 600
PixelSizeEnums
CameraDefs Class, 143
Pointer Class, 354
CBasePtr, 355
CBooleanPtr, 355
CCategoryPtr, 356
CChunkPortPtr, 356
CCommandPtr, 356
CDeviceInfoPtr, 356
CEnumEntryPtr, 356
CEnumerationPtr, 356
CIntegerPtr, 357
CNodeMapDynPtr, 357
CNodeMapPtr, 357
CNodePtr, 357
CPortConstructPtr, 357
CPortPtr, 357
CPortRecorderPtr, 358
CPortReplayPtr, 358
CPortWriteListPtr, 358
CRegisterPtr, 358
CSelectorPtr, 358
CStringPtr, 358
CValuePtr, 359
GetInterfaceName, 359
IsAvailable, 359
IsImplemented, 359
IsReadable, 359
IsWritable, 359
Polarization.cpp
AcquireImages, 1439
ConfigureStream, 1440
CreateAndSaveAolpDolpImages, 1440
CreateAndSaveGlareReducedImage, 1440
CreateAndSaveStokesImages, 1440
CreateHeatmapImages, 1440
CreateNormalizedImage, 1440
ExtractAndSavePolarQuadImages, 1441
GetQuadFileNameAppendage, 1441
isPixelFormatColor, 1442
main, 1441
PrintDeviceInfo, 1441
RunSingleCamera, 1441
SaveImage, 1441
PolarizationQuadrant
Spinnaker::ImageUtilityPolarization, 969
Poll
INodeMap Interface, 323
Spinnaker::GenApi::NodeMap, 1060
PopulateAdapterIpInfo
AdapterConfig, 390
PortImpl Class, 360
PortNode, 1069
Spinnaker::GenApi::PortNode, 1071
PortNode Class, 361
CPortRef, 361
PortRecorder, 1074
Spinnaker::GenApi::PortRecorder, 1076
PortRecorder Class, 362
CPortRecorderRef, 362
PortReplay, 1078
Spinnaker::GenApi::PortReplay, 1079
PortReplay Class, 363
PortWriteList Class, 364
PowerSupplyCurrent
Spinnaker::Camera, 600
PowerSupplyVoltage
Spinnaker::Camera, 600
Prefix
U3V_COMMAND_HEADER, 1158
Preprocess
Spinnaker::GenApi::CNodeMapFactory, 728
PreprocessXMLFromFile
INodeMapDyn Interface, 327
PreprocessXMLFromZIPFile
INodeMapDyn Interface, 327
PrintAllNodes

GigEVisionPerformance.cpp, 1411
PrintApplicationLayerDeviceInfo
GenTLInfo_QuickSpin.cpp, 1406
PrintBooleanNode
NodeMapInfo.cpp, 1437
PrintBuildInfo
HighDynamicRange.cpp, 1416
PrintCPUUsage
GigEVisionPerformance.cpp, 1411
PrintCategoryNodeAndAllFeatures
NodeMapInfo.cpp, 1437
PrintCommandNode
NodeMapInfo.cpp, 1437
PrintDataStreamInfo
GigEVisionPerformance.cpp, 1411
PrintDebugMessage
FileAccess_QuickSpin.cpp, 1405
PrintDeviceInfo
Acquisition.cpp, 1339
AcquisitionMultipleThread.cpp, 1385
ActionCommand.cpp, 1387
BufferHandling.cpp, 1389
ChunkData.cpp, 1391
CounterAndTimer.cpp, 1393
DeviceEvents.cpp, 1395
Exposure.cpp, 1401
Exposure_QuickSpin.cpp, 1402
FileAccess_QuickSpin.cpp, 1405
GigEVisionPerformance.cpp, 1411
HighDynamicRange.cpp, 1416
ImageEvents.cpp, 1419
ImageFormatControl.cpp, 1420
ImageFormatControl_QuickSpin.cpp, 1422
Inference.cpp, 1427
LogicBlock.cpp, 1431
LookupTable.cpp, 1433
NodeMapCallback.cpp, 1435
Polarization.cpp, 1441
SaveToAvi.cpp, 1443
Sequencer.cpp, 1445
SerialRxTx.cpp, 1448
Trigger.cpp, 1451
Trigger_QuickSpin.cpp, 1454
PrintEnumerationNodeAndCurrentEntry
NodeMapInfo.cpp, 1437
PrintEnumerationSelector
NodeMapInfo.cpp, 1437
PrintExampleStatistics
AcquisitionMultipleCameraRecovery.cpp, 1383
PrintFloatNode
NodeMapInfo.cpp, 1437
PrintGenericHandlerMessage
InterfaceEventHandlerImpl, 1002
PrintIntegerNode
NodeMapInfo.cpp, 1438
PrintNode
NodeMapInfo.cpp, 1438
PrintResultMessage
FileAccess_QuickSpin.cpp, 1405
PrintRetrieveNodeFailure
LookupTable.cpp, 1433
Sequencer.cpp, 1445
PrintStringNode
NodeMapInfo.cpp, 1438
PrintTransportLayerDeviceInfo
GenTLInfo_QuickSpin.cpp, 1407
PrintTransportLayerInterfaceInfo
GenTLInfo_QuickSpin.cpp, 1407
PrintTransportLayerStreamInfo
GenTLInfo_QuickSpin.cpp, 1407
PrintUsage
FileAccess_QuickSpin.cpp, 1405
GigEVisionPerformance.cpp, 1411
PrintValueNode
NodeMapInfo.cpp, 1438
ProducerImpl
Spinnaker::IInterface, 907
Spinnaker::InterfaceList, 1005
progressive
Spinnaker::JPEGOption, 1023

quality
Spinnaker::JPEGOption, 1023
Spinnaker::JPG2Option, 1024
Spinnaker::Video::MJPGOption, 1040
QueryInterface
Enumeration.cpp, 1396
Enumeration_QuickSpin.cpp, 1397

radius
Chunk Data Inference Class, 173
rdbuf
Spinnaker::GenApi::IDevFileStreamBase, 876
Spinnaker::GenApi::ODevFileStreamBase, 1064
Read
Spinnaker::GenApi::CChunkPort, 666
Spinnaker::GenApi::CEventPort, 687
Spinnaker::GenApi::CPortImpl, 748
Spinnaker::GenApi::CRegisterPortImpl, 755
Spinnaker::GenApi::CTestPortStruct, 762
Spinnaker::GenApi::PortNode, 1072
Spinnaker::GenApi::PortRecorder, 1076
Spinnaker::GenApi::PortReplay, 1080
read
Spinnaker::GenApi::FileProtocolAdapter, 811
ReadPort
Spinnaker::CameraBase, 636
Spinnaker::ICameraBase, 852
ReadRegister
Spinnaker::GenApi::CRegisterPortImpl, 755
readType
NodeMapInfo.cpp, 1436
receiveBuffers
AdapterConfig::AdapterInfo, 459
receiveBuffersMax
AdapterConfig::AdapterInfo, 459
receiveBuffersMin

AdapterConfig::AdapterInfo, 460
receiveBuffersRegKey
AdapterConfig::AdapterInfo, 460
receiveBuffersStep
AdapterConfig::AdapterInfo, 460
rect
Chunk Data Inference Class, 173
Reference Interfaces, 365
SetNumEnums, 365
RefreshCameraList
AcquisitionMultipleCameraRecovery.cpp, 1383
RegionDestination
Spinnaker::Camera, 600
RegionDestinationEnums
CameraDefs Class, 144
RegionMode
Spinnaker::Camera, 601
RegionModeEnums
CameraDefs Class, 144
RegionSelector
Spinnaker::Camera, 601
RegionSelectorEnums
CameraDefs Class, 145
Register
NodeCallback Class, 348
RegisterAllInterfaceEvents
SystemEventHandlerImpl, 1115
RegisterCallback
INode Interface, 319
Spinnaker::GenApi::Node, 1051
RegisterEventHandler
Spinnaker::CameraBase, 636, 637
Spinnaker::ICameraBase, 852
Spinnaker::IInterface, 906
Spinnaker::ISystem, 1017
Spinnaker::Interface, 990
Spinnaker::System, 1105
RegisterImageEventHandler
Spinnaker::IDataStream, 873
RegisterInterfaceEventHandler
Spinnaker::ISystem, 1017
Spinnaker::System, 1105
RegisterInterfaceEventToSystem
SystemEventHandlerImpl, 1116
RegisterLoggingEventHandler
Spinnaker::ISystem, 1017
Spinnaker::System, 1106
RegisterNode, 1083
Spinnaker::GenApi::RegisterNode, 1084, 1085
RegisterNode Class, 366
CRegisterRef, 366
RegisterPortImpl Class, 367
Release
Spinnaker::IImage, 894
Spinnaker::Image, 939
ReleaseCameraDescriptionFileData
Spinnaker::GenApi::CNodeMapFactory, 728
ReleaseImage
Spinnaker::IDataStream, 873
ReleaseInstance
Spinnaker::ISystem, 1017
Spinnaker::System, 1106
RemoveByDeviceID
Spinnaker::CameraList, 644
Spinnaker::ICameraList, 858
RemoveByIndex
Spinnaker::CameraList, 645
Spinnaker::ICameraList, 858
RemoveBySerial
Spinnaker::CameraList, 645
Spinnaker::ICameraList, 858
ReplaceEnvironmentVariables
GCUtilities Utility, 283
Replay
IPortRecorder Interface, 334
Spinnaker::GenApi::CPortImpl, 749
Spinnaker::GenApi::CPortWriteList, 752
Spinnaker::GenApi::PortNode, 1072
Spinnaker::GenApi::PortRecorder, 1076
Spinnaker::GenApi::PortReplay, 1080
ReqId
GVCP_REQUEST_HEADER, 844
U3V_COMMAND_HEADER, 1158
Reserved
U3V_EVENT_DATA, 1159
reserved
Spinnaker::BMPOption, 469
Spinnaker::JPEGOption, 1023
Spinnaker::JPG2Option, 1024
Spinnaker::PGMOption, 1067
Spinnaker::PNGOption, 1068
Spinnaker::PPMOption, 1082
Spinnaker::TIFFOption, 1120
Spinnaker::Video::AVIOption, 464
Spinnaker::Video::H264Option, 845
Spinnaker::Video::MJPGOption, 1040
ReservedOrEventSize
GVCP_EVENT_ITEM_BASIC, 837
GVCP_EVENT_ITEM_EXTENDED_ID, 838
GVCP_EVENT_ITEM, 835
ResetCallbacks
NodeMapCallback.cpp, 1435
ResetCameraUserSetToDefault
AcquisitionMultipleCameraRecovery.cpp, 1383
ResetDeviceEvents
DeviceEvents.cpp, 1395
ResetExposure
Exposure.cpp, 1401
Exposure_QuickSpin.cpp, 1402
LogicBlock.cpp, 1431
ResetImage
Spinnaker::IImage, 894
Spinnaker::Image, 939
ResetImageEvents
ImageEvents.cpp, 1419
ResetLookupTables

LookupTable.cpp, 1433
ResetSequencer
Sequencer.cpp, 1445
ResetStatistics
Spinnaker::GenApi::CTestPortStruct, 762
ResetStreamHandle
Spinnaker::IDataStream, 873
ResetTrigger
BufferHandling.cpp, 1389
CounterAndTimer.cpp, 1393
LogicBlock.cpp, 1431
Trigger.cpp, 1451
Trigger_QuickSpin.cpp, 1454
resize
Spinnaker::GenICam::gcstring, 831
Restore
ISelectorDigit Interface, 340
Spinnaker::GenApi::CSelectorSet, 759
RetrieveAllAdapters
AdapterConfig, 391
ReverseX
Spinnaker::Camera, 601
ReverseY
Spinnaker::Camera, 601
RevokeImages
Spinnaker::IDataStream, 873
RgbTransformLightSource
Spinnaker::Camera, 601
RgbTransformLightSourceEnums
CameraDefs Class, 145
RO
Types Enums, 378
rotatedRect
Chunk Data Inference Class, 173
rotationAngle
Chunk Data Inference Class, 173
RunMultipleCameras
AcquisitionMultipleThread.cpp, 1385
ActionCommand.cpp, 1387
RunSingleCamera
Acquisition.cpp, 1339
BufferHandling.cpp, 1389
ChunkData.cpp, 1392
CounterAndTimer.cpp, 1393
DeviceEvents.cpp, 1395
Exposure.cpp, 1401
Exposure_QuickSpin.cpp, 1402
GigEVisionPerformance.cpp, 1411
HighDynamicRange.cpp, 1416
ImageEvents.cpp, 1419
ImageFormatControl.cpp, 1421
ImageFormatControl_QuickSpin.cpp, 1422
Inference.cpp, 1427
LogicBlock.cpp, 1431
LookupTable.cpp, 1433
NodeMapCallback.cpp, 1435
NodeMapInfo.cpp, 1438
Polarization.cpp, 1441
SaveToAvi.cpp, 1443
Sequencer.cpp, 1445
SerialRxTx.cpp, 1448
Trigger.cpp, 1452
Trigger_QuickSpin.cpp, 1454
RW
Types Enums, 378

SERIAL_PORT_BAUD_RATE
SerialRxTx.cpp, 1447
SERIAL_PORT_COMMUNICATION_TIMEOUT_MIL↩
LISECOND
SerialRxTx.cpp, 1447
SERIAL_PORT_DELAY
SerialRxTx.cpp, 1447
SERIAL_PORT_PARITY_BITS
SerialRxTx.cpp, 1447
SERIAL_PORT_STOP_BITS
SerialRxTx.cpp, 1447
SET_GUID
Spinnaker::GenApi, 449
SPINNAKER_API_ABSTRACT
Spinnaker Platform, 211
SPINNAKER_API
Spinnaker Platform, 211
SPINNAKER_LOCAL
Spinnaker Platform, 211
SPINUPDATE_API
SpinUpdate.h, 1327
Saturation
Spinnaker::Camera, 602
SaturationEnable
Spinnaker::Camera, 602
Save
Spinnaker::IImage, 894–896
Spinnaker::Image, 940–942
SaveImage
Polarization.cpp, 1441
SaveToAvi.cpp
AcquireImages, 1443
chosenVideoType, 1444
main, 1443
PrintDeviceInfo, 1443
RunSingleCamera, 1443
SaveVectorToVideo, 1443
videoType, 1442
SaveVectorToVideo
SaveToAvi.cpp, 1443
Scan3dAxisMax
Spinnaker::Camera, 602
Scan3dAxisMin
Spinnaker::Camera, 602
Scan3dCoordinateOffset
Spinnaker::Camera, 602
Scan3dCoordinateReferenceSelector
Spinnaker::Camera, 603
Scan3dCoordinateReferenceSelectorEnums
CameraDefs Class, 145
Scan3dCoordinateReferenceValue

Spinnaker::Camera, 603
Scan3dCoordinateScale
Spinnaker::Camera, 603
Scan3dCoordinateSelector
Spinnaker::Camera, 603
Scan3dCoordinateSelectorEnums
CameraDefs Class, 146
Scan3dCoordinateSystem
Spinnaker::Camera, 603
Scan3dCoordinateSystemEnums
CameraDefs Class, 146
Scan3dCoordinateSystemReference
Spinnaker::Camera, 603
Scan3dCoordinateSystemReferenceEnums
CameraDefs Class, 146
Scan3dCoordinateTransformSelector
Spinnaker::Camera, 604
Scan3dCoordinateTransformSelectorEnums
CameraDefs Class, 147
Scan3dDistanceUnit
Spinnaker::Camera, 604
Scan3dDistanceUnitEnums
CameraDefs Class, 147
Scan3dInvalidDataFlag
Spinnaker::Camera, 604
Scan3dInvalidDataValue
Spinnaker::Camera, 604
Scan3dOutputMode
Spinnaker::Camera, 604
Scan3dOutputModeEnums
CameraDefs Class, 148
Scan3dTransformValue
Spinnaker::Camera, 604
SecondsCounter, 393
endTime, 394
GetSecondsCounter, 394
StartSecondsCounter, 394
startTime, 394
timeDiff, 394
SelectorSet Class, 368
SendActionCommand
Spinnaker::IInterface, 906
Spinnaker::ISystem, 1018
Spinnaker::Interface, 990
Spinnaker::System, 1106
SensorDescription
Spinnaker::Camera, 605
SensorDigitizationTaps
Spinnaker::Camera, 605
SensorDigitizationTapsEnums
CameraDefs Class, 148
SensorHeight
Spinnaker::Camera, 605
SensorShutterMode
Spinnaker::Camera, 605
SensorShutterModeEnums
CameraDefs Class, 149
SensorTaps
Spinnaker::Camera, 605
SensorTapsEnums
CameraDefs Class, 149
SensorWidth
Spinnaker::Camera, 605
Sequencer.cpp
AcquireImages, 1444
ConfigureSequencerPartOne, 1444
ConfigureSequencerPartTwo, 1445
main, 1445
PrintDeviceInfo, 1445
PrintRetrieveNodeFailure, 1445
ResetSequencer, 1445
RunSingleCamera, 1445
SetSingleState, 1445
SequencerConfigurationMode
Spinnaker::Camera, 606
SequencerConfigurationModeEnums
CameraDefs Class, 150
SequencerConfigurationValid
Spinnaker::Camera, 606
SequencerConfigurationValidEnums
CameraDefs Class, 150
SequencerFeatureEnable
Spinnaker::Camera, 606
SequencerMode
Spinnaker::Camera, 606
SequencerModeEnums
CameraDefs Class, 150
SequencerPathSelector
Spinnaker::Camera, 606
SequencerSetActive
Spinnaker::Camera, 607
SequencerSetLoad
Spinnaker::Camera, 607
SequencerSetNext
Spinnaker::Camera, 607
SequencerSetSave
Spinnaker::Camera, 607
SequencerSetSelector
Spinnaker::Camera, 607
SequencerSetStart
Spinnaker::Camera, 608
SequencerSetValid
Spinnaker::Camera, 608
SequencerSetValidEnums
CameraDefs Class, 150
SequencerTriggerActivation
Spinnaker::Camera, 608
SequencerTriggerActivationEnums
CameraDefs Class, 151
SequencerTriggerSource
Spinnaker::Camera, 608
SequencerTriggerSourceEnums
CameraDefs Class, 151
SerialPortBaudRate
Spinnaker::Camera, 608
SerialPortBaudRateEnums

CameraDefs Class, 151
SerialPortDataBits
Spinnaker::Camera, 609
SerialPortParity
Spinnaker::Camera, 609
SerialPortParityEnums
CameraDefs Class, 152
SerialPortSelector
Spinnaker::Camera, 609
SerialPortSelectorEnums
CameraDefs Class, 152
SerialPortSource
Spinnaker::Camera, 609
SerialPortSourceEnums
CameraDefs Class, 153
SerialPortStopBits
Spinnaker::Camera, 609
SerialPortStopBitsEnums
CameraDefs Class, 153
SerialReceiveFramingErrorCount
Spinnaker::Camera, 609
SerialReceiveParityErrorCount
Spinnaker::Camera, 610
SerialReceiveQueueClear
Spinnaker::Camera, 610
SerialReceiveQueueCurrentCharacterCount
Spinnaker::Camera, 610
SerialReceiveQueueMaxCharacterCount
Spinnaker::Camera, 610
SerialRx
SerialRxTx.cpp, 1448
SerialRxTx.cpp
COM_PORT_COUNT_MAX, 1446
CleanUp, 1448
ConfigureDevice, 1448
DATA_BITS, 1447
MILLISECOND, 1447
main, 1448
PrintDeviceInfo, 1448
RunSingleCamera, 1448
SERIAL_PORT_BAUD_RATE, 1447
SERIAL_PORT_COMMUNICATION_TIMEOUT↩
_MILLISECOND, 1447
SERIAL_PORT_DELAY, 1447
SERIAL_PORT_PARITY_BITS, 1447
SERIAL_PORT_STOP_BITS, 1447
SerialRx, 1448
SerialTx, 1449
TWO_SECOND_DELAY, 1447
SerialTransmitQueueCurrentCharacterCount
Spinnaker::Camera, 610
SerialTransmitQueueMaxCharacterCount
Spinnaker::Camera, 610
SerialTx
SerialRxTx.cpp, 1449
Set
Spinnaker::GenApi::RegisterNode, 1086
SetBufferOwnership
Spinnaker::CameraBase, 637
Spinnaker::ICameraBase, 853
SetChannelStatus
Spinnaker::IImageStatistics, 902
Spinnaker::ImageStatistics, 958
SetChunkEnable
Inference.cpp, 1427
SetChunks
Spinnaker::ChunkData, 710
Spinnaker::IChunkData, 867
SetCookie
IPortRecorder Interface, 335
Spinnaker::GenApi::CPortWriteList, 752
SetDefaultColorProcessing
Spinnaker::Image, 943
SetEnumReference
Spinnaker::GenApi::CEnumerationTRef, 671
SetEventPayload
Spinnaker::EventHandler, 800
SetEventType
Spinnaker::EventHandler, 800
SetFirst
Spinnaker::GenApi::CSelectorSet, 759
SetFrameRate
GigEVisionPerformance.cpp, 1412
SetGenICamCLProtocolFolder
GCUtilities Utility, 284
SetGenICamCacheFolder
GCUtilities Utility, 283
SetGenICamLogConfig
GCUtilities Utility, 284
SetHeatmapColorGradient
Spinnaker::ImageUtilityHeatmap, 967
SetHeatmapRange
Spinnaker::ImageUtilityHeatmap, 967
SetInfo
Spinnaker::GenApi::CFeatureBag, 690
SetIntValue
IEnumeration Interface, 301
Spinnaker::GenApi::EnumNode, 792
SetLoggingEventPriorityLevel
Spinnaker::ISystem, 1018
Spinnaker::System, 1107
SetMaximumFileSize
Spinnaker::Video::SpinVideo, 1092
SetMessageCallback
SpinUpdate.h, 1328
SetNext
ISelectorDigit Interface, 340
Spinnaker::GenApi::CSelectorSet, 759
SetNodeHandle
Spinnaker::GenApi::Node, 1051
SetNodeMap
Spinnaker::GenApi::Node, 1051
SetNumEnums
Reference Interfaces, 365
Spinnaker::GenApi::CEnumerationTRef, 671
SetPortImpl

Spinnaker::GenApi::CChunkPort, 666
Spinnaker::GenApi::CEventPort, 687
Spinnaker::GenApi::CPortImpl, 749
Spinnaker::GenApi::CRegisterPortImpl, 756
Spinnaker::GenApi::PortNode, 1072
SetProgressCallback
SpinUpdate.h, 1328
SetReference
Spinnaker::GenApi::BooleanNode, 472
Spinnaker::GenApi::CEnumerationTRef, 671
Spinnaker::GenApi::CategoryNode, 649
Spinnaker::GenApi::CommandNode, 739
Spinnaker::GenApi::EnumEntryNode, 787
Spinnaker::GenApi::EnumNode, 792
Spinnaker::GenApi::FloatNode, 818
Spinnaker::GenApi::FloatRegNode, 821
Spinnaker::GenApi::IntRegNode, 1012
Spinnaker::GenApi::IntegerNode, 986
Spinnaker::GenApi::Node, 1051
Spinnaker::GenApi::PortNode, 1073
Spinnaker::GenApi::PortRecorder, 1077
Spinnaker::GenApi::PortReplay, 1081
Spinnaker::GenApi::RegisterNode, 1086
Spinnaker::GenApi::StringNode, 1096
Spinnaker::GenApi::StringRegNode, 1099
Spinnaker::GenApi::ValueNode, 1163
SetSingleState
Sequencer.cpp, 1445
SetUserBuffers
Spinnaker::CameraBase, 637, 638
Spinnaker::ICameraBase, 853
SetValue
Spinnaker::GenApi::BooleanNode, 472
Spinnaker::GenApi::CEnumerationTRef, 672
Spinnaker::GenApi::FloatNode, 818
Spinnaker::GenApi::IntegerNode, 986
Spinnaker::GenApi::StringNode, 1096
SetupCounterAndTimer
CounterAndTimer.cpp, 1394
Sharpening
Spinnaker::Camera, 611
SharpeningAuto
Spinnaker::Camera, 611
SharpeningEnable
Spinnaker::Camera, 611
SharpeningThreshold
Spinnaker::Camera, 611
Signed
Types Enums, 383
SingleChunkData_t, 1086
ChunkID, 1087
ChunkLength, 1087
ChunkOffset, 1087
SingleChunkDataStr_t, 1087
ChunkID, 1087
ChunkLength, 1087
ChunkOffset, 1088
size
Spinnaker::GenApi::double_autovector_t, 776
Spinnaker::GenApi::int64_autovector_t, 980
Spinnaker::GenICam::gcstring, 831
SleepyWrapper
AcquisitionMultipleCameraRecovery.cpp, 1383
ActionCommand.cpp, 1387
BufferHandling.cpp, 1389
ImageEvents.cpp, 1419
SoftwareSignalPulse
Spinnaker::Camera, 612
SoftwareSignalSelector
Spinnaker::Camera, 612
SoftwareSignalSelectorEnums
CameraDefs Class, 153
SourceCount
Spinnaker::Camera, 612
SourceDataRange
Spinnaker::ImageUtility, 960
SourceSelector
Spinnaker::Camera, 612
SourceSelectorEnums
CameraDefs Class, 154
SpinTestCamera, 1088
SpinTestCamera Class, 369
SpinUpdate.h
GetErrorMessage, 1328
SPINUPDATE_API, 1327
SetMessageCallback, 1328
SetProgressCallback, 1328
UpdateFirmware, 1328
UpdateFirmwareConsole, 1328
UpdateFirmwareGUI, 1329
UpdatorMessageCallback, 1329
UpdatorProgressCallback, 1329
SpinVideo, 1089
Spinnaker::Video::SpinVideo, 1089
Spinnaker, 395
Spinnaker Classes, 40
InferenceBoxType, 43
Spinnaker Definitions, 201
ActionCommandStatus, 204
BufferOwnership, 205
ColorProcessingAlgorithm, 205
Error, 205
EventType, 207
ImageFileFormat, 207
ImageStatus, 208
PayloadTypeInfoIDs, 208
PixelFormatIntType, 209
PixelFormatNamespaceID, 209
SpinnakerLogLevel, 210
StatisticsChannel, 210
Spinnaker EventHandler Classes, 175
Spinnaker GenApi Classes, 237
_ClearXMLCache, 243
_Connect, 243, 244
_Destroy, 244
_GetDeviceName, 244

_GetNode, 244
_GetNodes, 244
_GetSupportedSchemaVersions, 244
_InvalidateNodes, 244
_LoadXMLFromFile, 245
_LoadXMLFromFileInject, 245
_LoadXMLFromString, 245
_LoadXMLFromStringInject, 245
_LoadXMLFromZIPData, 245
_LoadXMLFromZIPFile, 245
_Poll, 245
~CNodeMapRefT, 247
CNodeMapRef, 243
CNodeMapRefT, 246
CNodeRef, 243
CSelectorRef, 243
CastToIDestroy, 246
EatComments, 246
operator<<, 246
operator>>, 247
operator=, 247
Spinnaker GenApi Enums, 374
Spinnaker GenApi Interfaces, 249
CallbackHandleType, 250
NodeList_t, 250
Spinnaker GenApi Utilities, 279
Spinnaker Headers, 198
EVENT_TIMEOUT_INFINITE, 199
EVENT_TIMEOUT_NONE, 199
Spinnaker Platform, 211
SPINNAKER_API_ABSTRACT, 211
SPINNAKER_API, 211
SPINNAKER_LOCAL, 211
Spinnaker QuickSpin Classes, 217
Spinnaker Video Class, 212
Spinnaker Video Definitions, 213
Spinnaker.h, 200
Spinnaker::ActionCommandResult
DeviceAddress, 455
Status, 455
Spinnaker::BMPOption
BMPOption, 469
indexedColor_8bit, 469
reserved, 469
Spinnaker::BasePtr
~BasePtr, 465
BasePtr, 465
get, 466
IsValid, 466
m_pT, 468
operator bool, 466
operator T*, 466
operator->, 466
operator=, 467
operator==, 467, 468
Spinnaker::Camera
~Camera, 503
aPAUSEMACCtrlFramesReceived, 508
aPAUSEMACCtrlFramesTransmitted, 508
AasRoiEnable, 503
AasRoiHeight, 503
AasRoiOffsetX, 504
AasRoiOffsetY, 504
AasRoiWidth, 504
AcquisitionAbort, 504
AcquisitionArm, 505
AcquisitionBurstFrameCount, 505
AcquisitionFrameCount, 505
AcquisitionFrameRate, 505
AcquisitionFrameRateEnable, 505
AcquisitionLineRate, 506
AcquisitionMode, 506
AcquisitionResultingFrameRate, 506
AcquisitionStart, 506
AcquisitionStatus, 506
AcquisitionStatusSelector, 506
AcquisitionStop, 507
ActionDeviceKey, 507
ActionGroupKey, 507
ActionGroupMask, 507
ActionQueueSize, 507
ActionSelector, 507
ActionUnconditionalMode, 508
AdaptiveCompressionEnable, 508
AdcBitDepth, 508
AutoAlgorithmSelector, 509
AutoExposureControlLoopDamping, 509
AutoExposureControlPriority, 509
AutoExposureEVCompensation, 509
AutoExposureExposureTimeLowerLimit, 510
AutoExposureExposureTimeUpperLimit, 510
AutoExposureGainLowerLimit, 510
AutoExposureGainUpperLimit, 510
AutoExposureGreyValueLowerLimit, 511
AutoExposureGreyValueUpperLimit, 511
AutoExposureLightingMode, 511
AutoExposureMeteringMode, 511
AutoExposureTargetGreyValue, 512
AutoExposureTargetGreyValueAuto, 512
BalanceRatio, 512
BalanceRatioSelector, 513
BalanceWhiteAuto, 513
BalanceWhiteAutoDamping, 513
BalanceWhiteAutoLowerLimit, 513
BalanceWhiteAutoProfile, 514
BalanceWhiteAutoUpperLimit, 514
BinningHorizontal, 514
BinningHorizontalMode, 514
BinningSelector, 515
BinningVertical, 515
BinningVerticalMode, 515
BlackLevel, 515
BlackLevelAuto, 515
BlackLevelAutoBalance, 516
BlackLevelClampingEnable, 516
BlackLevelRaw, 516

BlackLevelSelector, 516
Camera, 503
ChunkBlackLevel, 516
ChunkBlackLevelSelector, 517
ChunkCRC, 517
ChunkCounterSelector, 517
ChunkCounterValue, 517
ChunkEnable, 517
ChunkEncoderSelector, 517
ChunkEncoderStatus, 518
ChunkEncoderValue, 518
ChunkExposureEndLineStatusAll, 518
ChunkExposureTime, 518
ChunkExposureTimeSelector, 518
ChunkFrameID, 518
ChunkGain, 519
ChunkGainSelector, 519
ChunkHeight, 519
ChunkImage, 519
ChunkImageComponent, 519
ChunkInferenceBoundingBoxResult, 519
ChunkInferenceConfidence, 520
ChunkInferenceFrameId, 520
ChunkInferenceResult, 520
ChunkLinePitch, 520
ChunkLineStatusAll, 520
ChunkModeActive, 520
ChunkOffsetX, 521
ChunkOffsetY, 521
ChunkPartSelector, 521
ChunkPixelDynamicRangeMax, 521
ChunkPixelDynamicRangeMin, 521
ChunkPixelFormat, 521
ChunkRegionID, 522
ChunkScan3dAxisMax, 522
ChunkScan3dAxisMin, 522
ChunkScan3dCoordinateOffset, 522
ChunkScan3dCoordinateReferenceSelector, 522
ChunkScan3dCoordinateReferenceValue, 522
ChunkScan3dCoordinateScale, 523
ChunkScan3dCoordinateSelector, 523
ChunkScan3dCoordinateSystem, 523
ChunkScan3dCoordinateSystemReference, 523
ChunkScan3dCoordinateTransformSelector, 523
ChunkScan3dDistanceUnit, 523
ChunkScan3dInvalidDataFlag, 524
ChunkScan3dInvalidDataValue, 524
ChunkScan3dOutputMode, 524
ChunkScan3dTransformValue, 524
ChunkScanLineSelector, 524
ChunkSelector, 524
ChunkSequencerSetActive, 525
ChunkSerialData, 525
ChunkSerialDataLength, 525
ChunkSerialReceiveOverflow, 525
ChunkSourceID, 525
ChunkStreamChannelID, 525
ChunkTimerSelector, 526
ChunkTimerValue, 526
ChunkTimestamp, 526
ChunkTimestampLatchValue, 526
ChunkTransferBlockID, 526
ChunkTransferQueueCurrentBlockCount, 526
ChunkTransferStreamID, 527
ChunkWidth, 527
CIConfiguration, 527
CITimeSlotsCount, 527
ColorTransformationEnable, 527
ColorTransformationSelector, 528
ColorTransformationValue, 528
ColorTransformationValueSelector, 528
CompressionRatio, 528
CounterDelay, 528
CounterDuration, 529
CounterEventActivation, 529
CounterEventSource, 529
CounterReset, 529
CounterResetActivation, 529
CounterResetSource, 529
CounterSelector, 530
CounterStatus, 530
CounterTriggerActivation, 530
CounterTriggerSource, 530
CounterValue, 530
CounterValueAtReset, 530
CxpConnectionSelector, 531
CxpConnectionTestErrorCount, 531
CxpConnectionTestMode, 531
CxpConnectionTestPacketCount, 531
CxpLinkConfiguration, 531
CxpLinkConfigurationPreferred, 531
CxpLinkConfigurationStatus, 532
CxpPoCxpAuto, 532
CxpPoCxpStatus, 532
CxpPoCxpTripReset, 532
CxpPoCxpTurnOff, 532
DecimationHorizontal, 532
DecimationHorizontalMode, 533
DecimationSelector, 533
DecimationVertical, 533
DecimationVerticalMode, 533
DefectCorrectStaticEnable, 534
DefectCorrectionMode, 534
DefectTableApply, 534
DefectTableCoordinateX, 534
DefectTableCoordinateY, 534
DefectTableFactoryRestore, 535
DefectTableIndex, 535
DefectTablePixelCount, 535
DefectTableSave, 535
Deinterlacing, 536
DeviceCharacterSet, 536
DeviceClockFrequency, 536
DeviceClockSelector, 536
DeviceConnectionSelector, 536
DeviceConnectionSpeed, 537

DeviceConnectionStatus, 537
DeviceEventChannelCount, 537
DeviceFamilyName, 537
DeviceFeaturePersistenceEnd, 537
DeviceFeaturePersistenceStart, 537
DeviceFirmwareVersion, 538
DeviceGenCPVersionMajor, 538
DeviceGenCPVersionMinor, 538
DeviceID, 538
DeviceIndicatorMode, 538
DeviceLinkBandwidthReserve, 538
DeviceLinkCommandTimeout, 539
DeviceLinkConnectionCount, 539
DeviceLinkCurrentThroughput, 539
DeviceLinkHeartbeatMode, 539
DeviceLinkHeartbeatTimeout, 539
DeviceLinkSelector, 539
DeviceLinkSpeed, 540
DeviceLinkThroughputLimit, 540
DeviceLinkThroughputLimitMode, 540
DeviceManifestEntrySelector, 540
DeviceManifestPrimaryURL, 541
DeviceManifestSchemaMajorVersion, 541
DeviceManifestSchemaMinorVersion, 541
DeviceManifestSecondaryURL, 541
DeviceManifestXMLMajorVersion, 541
DeviceManifestXMLMinorVersion, 541
DeviceManifestXMLSubMinorVersion, 542
DeviceManufacturerInfo, 542
DeviceMaxThroughput, 542
DeviceModelName, 542
DevicePowerSupplySelector, 542
DeviceRegistersCheck, 543
DeviceRegistersEndianness, 543
DeviceRegistersStreamingEnd, 543
DeviceRegistersStreamingStart, 543
DeviceRegistersValid, 543
DeviceReset, 543
DeviceSFNCVersionMajor, 544
DeviceSFNCVersionMinor, 545
DeviceSFNCVersionSubMinor, 545
DeviceScanType, 544
DeviceSerialNumber, 544
DeviceSerialPortBaudRate, 544
DeviceSerialPortSelector, 544
DeviceStreamChannelCount, 545
DeviceStreamChannelEndianness, 545
DeviceStreamChannelLink, 545
DeviceStreamChannelPacketSize, 546
DeviceStreamChannelSelector, 546
DeviceStreamChannelType, 546
DeviceTLType, 547
DeviceTLVersionMajor, 547
DeviceTLVersionMinor, 547
DeviceTLVersionSubMinor, 547
DeviceTapGeometry, 546
DeviceTemperature, 546
DeviceTemperatureSelector, 546
DeviceType, 547
DeviceUptime, 548
DeviceUserID, 548
DeviceVendorName, 548
DeviceVersion, 548
EncoderDivider, 548
EncoderMode, 548
EncoderOutputMode, 549
EncoderReset, 549
EncoderResetActivation, 549
EncoderResetSource, 549
EncoderSelector, 549
EncoderSourceA, 549
EncoderSourceB, 550
EncoderStatus, 550
EncoderTimeout, 550
EncoderValue, 550
EncoderValueAtReset, 550
EnumerationCount, 550
EventAcquisitionEnd, 551
EventAcquisitionEndFrameID, 551
EventAcquisitionEndTimestamp, 551
EventAcquisitionError, 551
EventAcquisitionErrorFrameID, 551
EventAcquisitionErrorTimestamp, 551
EventAcquisitionStart, 552
EventAcquisitionStartFrameID, 552
EventAcquisitionStartTimestamp, 552
EventAcquisitionTransferEnd, 552
EventAcquisitionTransferEndFrameID, 552
EventAcquisitionTransferEndTimestamp, 552
EventAcquisitionTransferStart, 553
EventAcquisitionTransferStartFrameID, 553
EventAcquisitionTransferStartTimestamp, 553
EventAcquisitionTrigger, 553
EventAcquisitionTriggerFrameID, 553
EventAcquisitionTriggerTimestamp, 553
EventActionLate, 554
EventActionLateFrameID, 554
EventActionLateTimestamp, 554
EventCounter0End, 554
EventCounter0EndFrameID, 554
EventCounter0EndTimestamp, 554
EventCounter0Start, 555
EventCounter0StartFrameID, 555
EventCounter0StartTimestamp, 555
EventCounter1End, 555
EventCounter1EndFrameID, 555
EventCounter1EndTimestamp, 555
EventCounter1Start, 556
EventCounter1StartFrameID, 556
EventCounter1StartTimestamp, 556
EventEncoder0Restarted, 556
EventEncoder0RestartedFrameID, 556
EventEncoder0RestartedTimestamp, 556
EventEncoder0Stopped, 557
EventEncoder0StoppedFrameID, 557
EventEncoder0StoppedTimestamp, 557

EventEncoder1Restarted, 557
EventEncoder1RestartedFrameID, 557
EventEncoder1RestartedTimestamp, 557
EventEncoder1Stopped, 558
EventEncoder1StoppedFrameID, 558
EventEncoder1StoppedTimestamp, 558
EventError, 558
EventErrorCode, 558
EventErrorFrameID, 558
EventErrorTimestamp, 559
EventExposureEnd, 559
EventExposureEndFrameID, 559
EventExposureEndTimestamp, 559
EventExposureStart, 559
EventExposureStartFrameID, 559
EventExposureStartTimestamp, 560
EventFrameBurstEnd, 560
EventFrameBurstEndFrameID, 560
EventFrameBurstEndTimestamp, 560
EventFrameBurstStart, 560
EventFrameBurstStartFrameID, 560
EventFrameBurstStartTimestamp, 561
EventFrameEnd, 561
EventFrameEndFrameID, 561
EventFrameEndTimestamp, 561
EventFrameStart, 561
EventFrameStartFrameID, 561
EventFrameStartTimestamp, 562
EventFrameTransferEnd, 562
EventFrameTransferEndFrameID, 562
EventFrameTransferEndTimestamp, 562
EventFrameTransferStart, 562
EventFrameTransferStartFrameID, 562
EventFrameTransferStartTimestamp, 563
EventFrameTrigger, 563
EventFrameTriggerFrameID, 563
EventFrameTriggerTimestamp, 563
EventLine0AnyEdge, 563
EventLine0AnyEdgeFrameID, 563
EventLine0AnyEdgeTimestamp, 564
EventLine0FallingEdge, 564
EventLine0FallingEdgeFrameID, 564
EventLine0FallingEdgeTimestamp, 564
EventLine0RisingEdge, 564
EventLine0RisingEdgeFrameID, 564
EventLine0RisingEdgeTimestamp, 565
EventLine1AnyEdge, 565
EventLine1AnyEdgeFrameID, 565
EventLine1AnyEdgeTimestamp, 565
EventLine1FallingEdge, 565
EventLine1FallingEdgeFrameID, 565
EventLine1FallingEdgeTimestamp, 566
EventLine1RisingEdge, 566
EventLine1RisingEdgeFrameID, 566
EventLine1RisingEdgeTimestamp, 566
EventLinkSpeedChange, 566
EventLinkSpeedChangeFrameID, 566
EventLinkSpeedChangeTimestamp, 567
EventLinkTrigger0, 567
EventLinkTrigger0FrameID, 567
EventLinkTrigger0Timestamp, 567
EventLinkTrigger1, 567
EventLinkTrigger1FrameID, 567
EventLinkTrigger1Timestamp, 568
EventNotification, 568
EventSelector, 568
EventSequencerSetChange, 568
EventSequencerSetChangeFrameID, 568
EventSequencerSetChangeTimestamp, 568
EventSerialData, 569
EventSerialDataLength, 569
EventSerialPortReceive, 569
EventSerialPortReceiveTimestamp, 569
EventSerialReceiveOverflow, 569
EventStream0TransferBlockEnd, 569
EventStream0TransferBlockEndFrameID, 570
EventStream0TransferBlockEndTimestamp, 570
EventStream0TransferBlockStart, 570
EventStream0TransferBlockStartFrameID, 570
EventStream0TransferBlockStartTimestamp, 570
EventStream0TransferBlockTrigger, 570
EventStream0TransferBlockTriggerFrameID, 571
EventStream0TransferBlockTriggerTimestamp, 571
EventStream0TransferBurstEnd, 571
EventStream0TransferBurstEndFrameID, 571
EventStream0TransferBurstEndTimestamp, 571
EventStream0TransferBurstStart, 571
EventStream0TransferBurstStartFrameID, 572
EventStream0TransferBurstStartTimestamp, 572
EventStream0TransferEnd, 572
EventStream0TransferEndFrameID, 572
EventStream0TransferEndTimestamp, 572
EventStream0TransferOverflow, 572
EventStream0TransferOverflowFrameID, 573
EventStream0TransferOverflowTimestamp, 573
EventStream0TransferPause, 573
EventStream0TransferPauseFrameID, 573
EventStream0TransferPauseTimestamp, 573
EventStream0TransferResume, 573
EventStream0TransferResumeFrameID, 574
EventStream0TransferResumeTimestamp, 574
EventStream0TransferStart, 574
EventStream0TransferStartFrameID, 574
EventStream0TransferStartTimestamp, 574
EventTest, 574
EventTestTimestamp, 575
EventTimer0End, 575
EventTimer0EndFrameID, 575
EventTimer0EndTimestamp, 575
EventTimer0Start, 575
EventTimer0StartFrameID, 575
EventTimer0StartTimestamp, 576
EventTimer1End, 576
EventTimer1EndFrameID, 576
EventTimer1EndTimestamp, 576
EventTimer1Start, 576

EventTimer1StartFrameID, 576
EventTimer1StartTimestamp, 577
ExposureActiveMode, 577
ExposureAuto, 577
ExposureMode, 577
ExposureTime, 577
ExposureTimeMode, 577
ExposureTimeSelector, 578
FactoryReset, 578
FileAccessBuffer, 578
FileAccessLength, 578
FileAccessOffset, 578
FileOpenMode, 578
FileOperationExecute, 579
FileOperationResult, 579
FileOperationSelector, 579
FileOperationStatus, 579
FileSelector, 579
FileSize, 580
Gain, 580
GainAuto, 580
GainAutoBalance, 580
GainSelector, 581
Gamma, 581
GammaEnable, 581
GevActiveLinkCount, 581
GevCCP, 581
GevCurrentDefaultGateway, 581
GevCurrentIPAddress, 582
GevCurrentIPConfigurationDHCP, 582
GevCurrentIPConfigurationLLA, 582
GevCurrentIPConfigurationPersistentIP, 582
GevCurrentPhysicalLinkConfiguration, 582
GevCurrentSubnetMask, 582
GevDiscoveryAckDelay, 583
GevFirstURL, 583
GevGVCPExtendedStatusCodes, 583
GevGVCPExtendedStatusCodesSelector, 583
GevGVCPHeartbeatDisable, 583
GevGVCPPendingAck, 583
GevGVCPPendingTimeout, 584
GevGVSPExtendedIDMode, 584
GevHeartbeatTimeout, 584
GevIEEE1588, 584
GevIEEE1588ClockAccuracy, 584
GevIEEE1588Mode, 584
GevIEEE1588Status, 585
GevIPConfigurationStatus, 585
GevInterfaceSelector, 585
GevMACAddress, 585
GevMCDA, 585
GevMCPHostPort, 585
GevMCRC, 586
GevMCSP, 586
GevMCTT, 586
GevNumberOfInterfaces, 586
GevPAUSEFrameReception, 586
GevPAUSEFrameTransmission, 586
GevPersistentDefaultGateway, 587
GevPersistentIPAddress, 587
GevPersistentSubnetMask, 587
GevPhysicalLinkConfiguration, 587
GevPrimaryApplicationIPAddress, 587
GevPrimaryApplicationSocket, 587
GevPrimaryApplicationSwitchoverKey, 588
GevSCCFGAllInTransmission, 588
GevSCCFGExtendedChunkData, 588
GevSCCFGPacketResendDestination, 588
GevSCCFGUnconditionalStreaming, 588
GevSCDA, 588
GevSCPDirection, 589
GevSCPHostPort, 589
GevSCPInterfaceIndex, 589
GevSCPSBigEndian, 589
GevSCPSDoNotFragment, 589
GevSCPSFireTestPacket, 590
GevSCPSPacketSize, 590
GevSCPD, 589
GevSCSP, 590
GevSCZoneConfigurationLock, 590
GevSCZoneCount, 590
GevSCZoneDirectionAll, 590
GevSecondURL, 591
GevStreamChannelSelector, 591
GevSupportedOption, 591
GevSupportedOptionSelector, 591
GevTimestampTickFrequency, 591
GuiXmlManifestAddress, 591
Height, 592
HeightMax, 592
ImageComponentEnable, 592
ImageComponentSelector, 592
ImageCompressionBitrate, 592
ImageCompressionJPEGFormatOption, 592
ImageCompressionMode, 593
ImageCompressionQuality, 593
ImageCompressionRateOption, 593
Init, 503
IspEnable, 593
LUTEnable, 597
LUTIndex, 597
LUTSelector, 597
LUTValue, 597
LUTValueAll, 598
LineFilterWidth, 593
LineFormat, 594
LineInputFilterSelector, 594
LineInverter, 594
LineMode, 594
LinePitch, 594
LineSelector, 594
LineSource, 595
LineStatus, 595
LineStatusAll, 595
LinkErrorCount, 595
LinkUptime, 595

LogicBlockLUTInputActivation, 595
LogicBlockLUTInputSelector, 596
LogicBlockLUTInputSource, 596
LogicBlockLUTOutputValue, 596
LogicBlockLUTOutputValueAll, 596
LogicBlockLUTRowIndex, 596
LogicBlockLUTSelector, 596
LogicBlockSelector, 597
MaxDeviceResetTime, 598
OffsetX, 598
OffsetY, 598
PacketResendRequestCount, 598
PayloadSize, 599
PixelColorFilter, 599
PixelDynamicRangeMax, 599
PixelDynamicRangeMin, 599
PixelFormat, 599
PixelFormatInfoID, 600
PixelFormatInfoSelector, 600
PixelSize, 600
PowerSupplyCurrent, 600
PowerSupplyVoltage, 600
RegionDestination, 600
RegionMode, 601
RegionSelector, 601
ReverseX, 601
ReverseY, 601
RgbTransformLightSource, 601
Saturation, 602
SaturationEnable, 602
Scan3dAxisMax, 602
Scan3dAxisMin, 602
Scan3dCoordinateOffset, 602
Scan3dCoordinateReferenceSelector, 603
Scan3dCoordinateReferenceValue, 603
Scan3dCoordinateScale, 603
Scan3dCoordinateSelector, 603
Scan3dCoordinateSystem, 603
Scan3dCoordinateSystemReference, 603
Scan3dCoordinateTransformSelector, 604
Scan3dDistanceUnit, 604
Scan3dInvalidDataFlag, 604
Scan3dInvalidDataValue, 604
Scan3dOutputMode, 604
Scan3dTransformValue, 604
SensorDescription, 605
SensorDigitizationTaps, 605
SensorHeight, 605
SensorShutterMode, 605
SensorTaps, 605
SensorWidth, 605
SequencerConfigurationMode, 606
SequencerConfigurationValid, 606
SequencerFeatureEnable, 606
SequencerMode, 606
SequencerPathSelector, 606
SequencerSetActive, 607
SequencerSetLoad, 607
SequencerSetNext, 607
SequencerSetSave, 607
SequencerSetSelector, 607
SequencerSetStart, 608
SequencerSetValid, 608
SequencerTriggerActivation, 608
SequencerTriggerSource, 608
SerialPortBaudRate, 608
SerialPortDataBits, 609
SerialPortParity, 609
SerialPortSelector, 609
SerialPortSource, 609
SerialPortStopBits, 609
SerialReceiveFramingErrorCount, 609
SerialReceiveParityErrorCount, 610
SerialReceiveQueueClear, 610
SerialReceiveQueueCurrentCharacterCount, 610
SerialReceiveQueueMaxCharacterCount, 610
SerialTransmitQueueCurrentCharacterCount, 610
SerialTransmitQueueMaxCharacterCount, 610
Sharpening, 611
SharpeningAuto, 611
SharpeningEnable, 611
SharpeningThreshold, 611
SoftwareSignalPulse, 612
SoftwareSignalSelector, 612
SourceCount, 612
SourceSelector, 612
TLParamsLocked, 615
Test0001, 612
TestEventGenerate, 613
TestPattern, 613
TestPatternGeneratorSelector, 613
TestPendingAck, 613
TimerDelay, 613
TimerDuration, 614
TimerReset, 614
TimerSelector, 614
TimerStatus, 614
TimerTriggerActivation, 614
TimerTriggerSource, 614
TimerValue, 615
Timestamp, 615
TimestampLatch, 615
TimestampLatchValue, 615
TimestampReset, 615
TransferAbort, 616
TransferBlockCount, 616
TransferBurstCount, 616
TransferComponentSelector, 616
TransferControlMode, 616
TransferOperationMode, 616
TransferPause, 617
TransferQueueCurrentBlockCount, 617
TransferQueueMaxBlockCount, 617
TransferQueueMode, 617
TransferQueueOverflowCount, 617
TransferResume, 617

TransferSelector, 618
TransferStart, 618
TransferStatus, 618
TransferStatusSelector, 618
TransferStop, 618
TransferStreamChannel, 618
TransferTriggerActivation, 619
TransferTriggerMode, 619
TransferTriggerSelector, 619
TransferTriggerSource, 619
TriggerActivation, 619
TriggerDelay, 619
TriggerDivider, 620
TriggerEventTest, 620
TriggerMode, 620
TriggerMultiplier, 620
TriggerOverlap, 620
TriggerSelector, 621
TriggerSoftware, 621
TriggerSource, 621
UserOutputSelector, 621
UserOutputValue, 621
UserOutputValueAll, 622
UserOutputValueAllMask, 622
UserSetDefault, 622
UserSetFeatureEnable, 622
UserSetLoad, 622
UserSetSave, 623
UserSetSelector, 623
V3_3Enable, 623
WhiteClip, 623
WhiteClipSelector, 623
Width, 624
WidthMax, 624
Spinnaker::CameraBase
~CameraBase, 627
BeginAcquisition, 628
CameraBase, 627, 628
DeInit, 628
DiscoverMaxPacketSize, 628
EndAcquisition, 629
ForceIP, 629
GetAccessMode, 629
GetBufferOwnership, 630
GetGuiXml, 630
GetNextImage, 630
GetNodeMap, 631
GetNumDataStreams, 631
GetNumImagesInUse, 632
GetTLDeviceNodeMap, 632
GetTLStreamNodeMap, 632
GetUniqueID, 633
GetUserBufferCount, 633
GetUserBufferSize, 633
GetUserBufferTotalSize, 634
Init, 634
InterfaceImpl, 639
IsInitialized, 635
IsStreaming, 635
IsValid, 635
operator=, 636
ReadPort, 636
RegisterEventHandler, 636, 637
SetBufferOwnership, 637
SetUserBuffers, 637, 638
UnregisterEventHandler, 639
WritePort, 639
Spinnaker::CameraList
~CameraList, 641
Append, 642
CameraList, 641
Clear, 642
GetByDeviceID, 642
GetByIndex, 643
GetBySerial, 643
GetSize, 644
operator=, 644
operator[], 644
RemoveByDeviceID, 644
RemoveByIndex, 645
RemoveBySerial, 645
Spinnaker::ChunkData
~ChunkData, 701
ChunkData, 700
GetBlackLevel, 701
GetCRC, 701
GetCounterValue, 701
GetEncoderValue, 701
GetExposureEndLineStatusAll, 702
GetExposureTime, 702
GetFrameID, 702
GetGain, 702
GetHeight, 703
GetImage, 703
GetInferenceBoundingBoxResult, 703
GetInferenceConfidence, 703
GetInferenceFrameId, 704
GetInferenceResult, 704
GetLinePitch, 704
GetLineStatusAll, 704
GetOffsetX, 705
GetOffsetY, 705
GetPartSelector, 705
GetPixelDynamicRangeMax, 705
GetPixelDynamicRangeMin, 706
GetScan3dAxisMax, 706
GetScan3dAxisMin, 706
GetScan3dCoordinateOffset, 706
GetScan3dCoordinateReferenceValue, 707
GetScan3dCoordinateScale, 707
GetScan3dInvalidDataValue, 707
GetScan3dTransformValue, 707
GetScanLineSelector, 708
GetSequencerSetActive, 708
GetSerialDataLength, 708
GetStreamChannelID, 708

GetTimerValue, 709
GetTimestamp, 709
GetTimestampLatchValue, 709
GetTransferBlockID, 709
GetTransferQueueCurrentBlockCount, 710
GetWidth, 710
SetChunks, 710
Spinnaker::DeviceArrivalEventHandler
~DeviceArrivalEventHandler, 766
DeviceArrivalEventHandler, 766
OnDeviceArrival, 766
operator=, 766
Spinnaker::DeviceEventHandler
~DeviceEventHandler, 769
DeviceEventHandler, 768
GetDeviceEventId, 769
GetDeviceEventName, 769
OnDeviceEvent, 769
operator=, 770
Spinnaker::DeviceRemovalEventHandler
~DeviceRemovalEventHandler, 774
DeviceRemovalEventHandler, 773
OnDeviceRemoval, 774
operator=, 774
Spinnaker::EventHandler
~EventHandler, 799
EventHandler, 799
EventProcessor, 800
GetEventPayloadData, 799
GetEventPayloadDataSize, 799
GetEventType, 799
IDataStream, 800
m_pEventData, 801
operator=, 800
SetEventPayload, 800
SetEventType, 800
Stream, 801
Spinnaker::Exception
~Exception, 805
Exception, 804, 805
GetBuildDate, 805
GetBuildTime, 805
GetError, 805
GetErrorMessage, 805
GetFileName, 806
GetFullErrorMessage, 806
GetFunctionName, 806
GetLineNumber, 806
operator!=, 806
operator=, 806
operator==, 807
what, 807
Spinnaker::GenApi, 434
COMMAND_MAGIC, 450
GENCP_COMMAND_HEADER_SIZE, 450
GENCP_EVENT_BASIC_SIZE, 450
GENCP_EVENT_CMD_ID, 450
GVCP_MESSAGE_TAGS, 449
IDevFileStream, 448
IPersistScript, 450
ODevFileStream, 448
PersistFeature, 449
SET_GUID, 449
U3V_EVENT_PREFIX, 450
Spinnaker::GenApi::AutoLock
~AutoLock, 463
AutoLock, 463
Spinnaker::GenApi::BooleanNode
~BooleanNode, 471
BooleanNode, 471
GetValue, 471
operator=, 472
SetReference, 472
SetValue, 472
Spinnaker::GenApi::CChunkAdapter
~CChunkAdapter, 650
AttachBuffer, 650
AttachNodeMap, 650
CChunkAdapter, 650
CheckBufferLayout, 651
ClearCaches, 651
DetachBuffer, 651
DetachNodeMap, 651
m_pChunkAdapter, 652
UpdateBuffer, 651
Spinnaker::GenApi::CChunkAdapterDcam
~CChunkAdapterDcam, 653
AttachBuffer, 653
CChunkAdapterDcam, 653
CheckBufferLayout, 654
CheckCRC, 654
HasCRC, 654
Spinnaker::GenApi::CChunkAdapterGEV
~CChunkAdapterGEV, 658
AttachBuffer, 658
CChunkAdapterGEV, 658
CheckBufferLayout, 659
Spinnaker::GenApi::CChunkAdapterGeneric
~CChunkAdapterGeneric, 656
AttachBuffer, 656
CChunkAdapterGeneric, 655
CheckBufferLayout, 656
Spinnaker::GenApi::CChunkAdapterU3V
~CChunkAdapterU3V, 660
AttachBuffer, 661
CChunkAdapterU3V, 660
CheckBufferLayout, 661
Spinnaker::GenApi::CChunkPort
~CChunkPort, 663
AttachChunk, 664
AttachPort, 664
CChunkPort, 663
CheckChunkID, 664
ClearCache, 664
DetachChunk, 665
DetachPort, 665

GetAccessMode, 665
GetChunkIDLength, 665
GetPrincipalInterfaceType, 665
GetSwapEndianess, 665
InvalidateNode, 666
m_pChunkPort, 667
m_pPort, 667
m_pPortAdapter, 667
Read, 666
SetPortImpl, 666
UpdateBuffer, 666
Write, 666
Spinnaker::GenApi::CEnumerationTRef
~CEnumerationTRef, 669
CEnumerationTRef, 669
GetCurrentEntry, 669
GetEntry, 670
GetValue, 670
operator(), 670
operator=, 671
SetEnumReference, 671
SetNumEnums, 671
SetReference, 671
SetValue, 672
Spinnaker::GenApi::CEventAdapter
~CEventAdapter, 673
AttachNodeMap, 673
CEventAdapter, 673
DeliverMessage, 673
DetachNodeMap, 674
m_pEventAdapter, 674
Spinnaker::GenApi::CEventAdapter1394
~CEventAdapter1394, 675
CEventAdapter1394, 675
DeliverEventMessage, 676
DeliverMessage, 676
Spinnaker::GenApi::CEventAdapterGEV
~CEventAdapterGEV, 680
CEventAdapterGEV, 680
DeliverEventMessage, 680
DeliverMessage, 680
Spinnaker::GenApi::CEventAdapterGeneric
~CEventAdapterGeneric, 677
CEventAdapterGeneric, 677
DeliverMessage, 678
Spinnaker::GenApi::CEventAdapterU3V
~CEventAdapterU3V, 682
CEventAdapterU3V, 682
DeliverEventMessage, 682
DeliverMessage, 682
Spinnaker::GenApi::CEventPort
~CEventPort, 685
AttachEvent, 685
AttachNode, 685
CEventPort, 684
CheckEventID, 685
DetachEvent, 686
DetachNode, 686
GetAccessMode, 686
GetEventIDLength, 686
GetPrincipalInterfaceType, 686
GetSwapEndianess, 686
InvalidateNode, 687
m_pEventPort, 687
m_pNode, 688
m_pPortAdapter, 688
Read, 687
SetPortImpl, 687
Write, 687
Spinnaker::GenApi::CFeatureBag
~CFeatureBag, 689
CFeatureBag, 689
GetFeatureBagHandle, 689
LoadFromBag, 689
operator==, 690
PersistFeature, 690
SetInfo, 690
StoreToBag, 690
Spinnaker::GenApi::CFloatPtr
CFloatPtr, 692
GetEnumAlias, 692
GetIntAlias, 692
operator=, 692
Spinnaker::GenApi::CGeneric_XMLLoaderParams
_Initialize, 693
Spinnaker::GenApi::CLock
~CLock, 713
CLock, 713
Lock, 714
m_bOwnLock, 714
m_lock, 715
NodeMap, 714
TryLock, 714
Unlock, 714
Spinnaker::GenApi::CLockEx
m_lockEx, 717
Spinnaker::GenApi::CNodeCallback
~CNodeCallback, 718
CNodeCallback, 718
Destroy, 718
GetCallbackType, 719
GetNode, 719
m_CallbackType, 719
m_pNode, 719
operator(), 719
Spinnaker::GenApi::CNodeMapFactory
~CNodeMapFactory, 722
AddInjectionData, 724
ApplyStyleSheet, 725
CNodeMapFactory, 722–724
ClearCache, 725
CreateEmptyNodeMap, 725
CreateNodeDataFromNodeMap, 725
CreateNodeMap, 725
ExtractSubtree, 726
GetNodeStatistics, 726

GetSupportedSchemaVersions, 726
IsCameraDescriptionFileDataReleased, 727
IsEmpty, 727
IsLoaded, 727
IsPreprocessed, 727
LoadAndInject, 727
operator=, 727
Preprocess, 728
ReleaseCameraDescriptionFileData, 728
ToString, 728
ToXml, 728
Spinnaker::GenApi::CNodeMapFactory::NodeStatistics↩
_t
NumLinks, 1061
NumNodes, 1061
NumProperties, 1061
NumStrings, 1061
Spinnaker::GenApi::CNodeMapRef
CNodeMapRef, 730
operator=, 731
Spinnaker::GenApi::CNodeMapRefT
_ClearXMLCache, 733
_Connect, 733, 734
_GetDeviceName, 734
_GetNode, 734
_GetNodes, 734
_GetSupportedSchemaVersions, 734
_InvalidateNodes, 735
_LoadXMLFromFile, 735
_LoadXMLFromFileInject, 735
_LoadXMLFromString, 735
_LoadXMLFromStringInject, 735
_LoadXMLFromZIPData, 736
_LoadXMLFromZIPFile, 736
_Poll, 736
_Ptr, 736
Spinnaker::GenApi::CPointer
~CPointer, 743
CPointer, 743
IsValid, 743
m_pT, 746
operator bool, 743
operator T*, 743
operator!=, 744
operator*, 745
operator(), 744
operator->, 745
operator=, 745
operator==, 745
Spinnaker::GenApi::CPortImpl
~CPortImpl, 748
CPortImpl, 747
GetAccessMode, 748
GetSwapEndianess, 748
InvalidateNode, 748
m_ptrPort, 749
Read, 748
Replay, 749
SetPortImpl, 749
Write, 749
Spinnaker::GenApi::CPortWriteList
~CPortWriteList, 751
CPortWriteList, 751
GetCookie, 751
GetPortWriteListHandle, 752
m_pWriteList, 752
Replay, 752
SetCookie, 752
Write, 752
Spinnaker::GenApi::CRegisterPortImpl
~CRegisterPortImpl, 755
CRegisterPortImpl, 754
GetAccessMode, 755
Read, 755
ReadRegister, 755
SetPortImpl, 756
Write, 756
WriteRegister, 756
Spinnaker::GenApi::CSelectorSet
~CSelectorSet, 758
CSelectorSet, 758
GetSelectorList, 758
IsEmpty, 758
Restore, 759
SetFirst, 759
SetNext, 759
ToString, 759
Spinnaker::GenApi::CTestPortStruct
CTestPortStruct, 761
GetAccessMode, 761
GetNumReads, 761
GetNumWrites, 762
GetPrincipalInterfaceType, 762
m_BaseAddress, 763
m_NumReads, 763
m_NumWrites, 763
MemSet, 762
Read, 762
ResetStatistics, 762
Write, 762
Spinnaker::GenApi::CategoryNode
~CategoryNode, 648
CategoryNode, 648
GetFeatures, 648
SetReference, 649
Spinnaker::GenApi::CommandNode
~CommandNode, 738
CommandNode, 738
Execute, 738
IsDone, 739
operator(), 739
SetReference, 739
Spinnaker::GenApi::Counter
Counter, 740
GetValue, 740
IsZero, 740

operator unsigned int, 740
operator++, 740, 741
operator--, 741
Spinnaker::GenApi::EAccessModeClass
FromString, 777
ToString, 778
Spinnaker::GenApi::ECachingModeClass
FromString, 779
ToString, 779
Spinnaker::GenApi::EDisplayNotationClass
FromString, 780
ToString, 780
Spinnaker::GenApi::EEndianessClass
FromString, 781
ToString, 781
Spinnaker::GenApi::EGenApiSchemaVersionClass
FromString, 782
ToString, 782
Spinnaker::GenApi::EInputDirectionClass
FromString, 783
ToString, 783
Spinnaker::GenApi::ENameSpaceClass
FromString, 784
ToString, 784
Spinnaker::GenApi::ERepresentationClass
FromString, 793
ToString, 794
Spinnaker::GenApi::ESignClass
FromString, 795
ToString, 795
Spinnaker::GenApi::ESlopeClass
FromString, 796
ToString, 796
Spinnaker::GenApi::EStandardNameSpaceClass
FromString, 797
ToString, 797
Spinnaker::GenApi::EVisibilityClass
FromString, 801
ToString, 802
Spinnaker::GenApi::EYesNoClass
FromString, 808
ToString, 808
Spinnaker::GenApi::EnumEntryNode
~EnumEntryNode, 786
EnumEntryNode, 786
GetNumericValue, 786
GetSymbolic, 787
GetValue, 787
IsSelfClearing, 787
SetReference, 787
Spinnaker::GenApi::EnumNode
~EnumNode, 790
EnumNode, 790
GetCurrentEntry, 790
GetEntries, 790
GetEntry, 791
GetEntryByName, 791
GetIntValue, 791
GetSymbolics, 791
m_pEnumeration, 793
operator*, 792
operator=, 792
SetIntValue, 792
SetReference, 792
Spinnaker::GenApi::FileProtocolAdapter
~FileProtocolAdapter, 809
attach, 809
closeFile, 810
deleteFile, 810
FileProtocolAdapter, 809
getBufSize, 810
openFile, 811
read, 811
write, 812
Spinnaker::GenApi::FloatNode
~FloatNode, 815
FloatNode, 815
GetDisplayNotation, 815
GetDisplayPrecision, 815
GetEnumAlias, 815
GetInc, 815
GetIncMode, 816
GetIntAlias, 816
GetListOfValidValues, 816
GetMax, 816
GetMin, 816
GetRepresentation, 816
GetUnit, 817
GetValue, 817
HasInc, 817
ImposeMax, 817
ImposeMin, 817
operator*, 818
operator(), 818
operator=, 818
SetReference, 818
SetValue, 818
Spinnaker::GenApi::FloatRegNode
~FloatRegNode, 821
FloatRegNode, 820, 821
SetReference, 821
Spinnaker::GenApi::Function_NodeCallback
Destroy, 823
Function_NodeCallback, 823
operator(), 823
Spinnaker::GenApi::IDevFileStreamBase
close, 876
filebuf_type, 875
ios_type, 875
is_open, 876
istream_type, 875
open, 876
rdbuf, 876
Spinnaker::GenApi::IDevFileStreamBuf
~IDevFileStreamBuf, 878
close, 878

IDevFileStreamBuf, 878
is_open, 878
open, 878
pbackfail, 878
underflow, 878
Spinnaker::GenApi::IntRegNode
~IntRegNode, 1012
IntRegNode, 1011, 1012
SetReference, 1012
Spinnaker::GenApi::IntegerNode
~IntegerNode, 983
GetFloatAlias, 983
GetInc, 984
GetIncMode, 984
GetListOfValidValues, 984
GetMax, 984
GetMin, 984
GetRepresentation, 984
GetUnit, 985
GetValue, 985
ImposeMax, 985
ImposeMin, 985
IntegerNode, 983
operator*, 986
operator(), 985
operator=, 986
SetReference, 986
SetValue, 986
Spinnaker::GenApi::Member_NodeCallback
Destroy, 1039
Member_NodeCallback, 1039
operator(), 1039
PMEMBERFUNC, 1038
Spinnaker::GenApi::Node
~Node, 1044
DeregisterCallback, 1044
GetAccessMode, 1044
GetAlias, 1045
GetCachingMode, 1045
GetCastAlias, 1045
GetChildren, 1045
GetDescription, 1046
GetDeviceName, 1046
GetDisplayName, 1046
GetDocuURL, 1046
GetEventID, 1046
GetName, 1046
GetNameSpace, 1046
GetNodeHandle, 1047
GetNodeMap, 1047
GetParents, 1047
GetPollingTime, 1047
GetPrincipalInterfaceType, 1047
GetProperty, 1048
GetPropertyNames, 1048
GetSelectedFeatures, 1048
GetSelectingFeatures, 1048
GetToolTip, 1048
GetVisibility, 1049
ImposeAccessMode, 1049
ImposeVisibility, 1049
InvalidateNode, 1049
IsAccessModeCacheable, 1049
IsCachable, 1049
IsDeprecated, 1050
IsFeature, 1050
IsSelector, 1050
IsStreamable, 1050
m_Callbacks, 1052
m_pNodeData, 1052
m_pNodeMap, 1052
Node, 1044
operator!=, 1050
operator==, 1050
RegisterCallback, 1051
SetNodeHandle, 1051
SetNodeMap, 1051
SetReference, 1051
Spinnaker::GenApi::NodeMap
_Ptr, 1060
~NodeMap, 1055
ClearXMLCache, 1055
Connect, 1055
Destroy, 1055
GetDeviceName, 1056
GetDeviceVersion, 1056
GetGenApiVersion, 1056
GetLock, 1056
GetModelName, 1056
GetNode, 1056
GetNodeMapHandle, 1057
GetNodes, 1057
GetNumNodes, 1057
GetProductGuid, 1057
GetSchemaVersion, 1057
GetStandardNameSpace, 1057
GetSupportedSchemaVersions, 1058
GetToolTip, 1058
GetVendorName, 1058
GetVersionGuid, 1058
InvalidateNodes, 1059
LoadXMLFromFile, 1059
LoadXMLFromFileInject, 1059
LoadXMLFromString, 1059
LoadXMLFromStringInject, 1059
LoadXMLFromZIPData, 1060
LoadXMLFromZIPFile, 1060
NodeMap, 1054
Poll, 1060
Spinnaker::GenApi::ODevFileStreamBase
close, 1063
filebuf_type, 1063
ios_type, 1063
is_open, 1063
open, 1063
ostream_type, 1063

rdbuf, 1064
Spinnaker::GenApi::ODevFileStreamBuf
~ODevFileStreamBuf, 1065
close, 1065
is_open, 1065
ODevFileStreamBuf, 1065
open, 1065
overflow, 1066
sync, 1066
xsputn, 1066
Spinnaker::GenApi::PortNode
~PortNode, 1071
CacheChunkData, 1071
GetChunkID, 1071
GetPortHandle, 1072
GetSwapEndianess, 1072
PortNode, 1071
Read, 1072
Replay, 1072
SetPortImpl, 1072
SetReference, 1073
StartRecording, 1073
StopRecording, 1073
Write, 1074
Spinnaker::GenApi::PortRecorder
~PortRecorder, 1076
GetAccessMode, 1076
PortRecorder, 1076
Read, 1076
Replay, 1076
SetReference, 1077
StartRecording, 1077
StopRecording, 1077
Write, 1077
Spinnaker::GenApi::PortReplay
~PortReplay, 1080
GetAccessMode, 1080
GetPortReplayHandle, 1080
PortReplay, 1079
Read, 1080
Replay, 1080
SetReference, 1081
Write, 1081
Spinnaker::GenApi::RegisterNode
~RegisterNode, 1085
Get, 1085
GetAddress, 1085
GetLength, 1086
RegisterNode, 1084, 1085
Set, 1086
SetReference, 1086
Spinnaker::GenApi::StringNode
~StringNode, 1094
GetMaxLength, 1095
GetValue, 1095
operator*, 1095
operator(), 1095
operator=, 1095
SetReference, 1096
SetValue, 1096
StringNode, 1094
Spinnaker::GenApi::StringRegNode
~StringRegNode, 1099
SetReference, 1099
StringRegNode, 1098, 1099
Spinnaker::GenApi::ValueNode
~ValueNode, 1162
FromString, 1162
GetNode, 1162
IsValueCacheValid, 1162
SetReference, 1163
ToString, 1163
ValueNode, 1161, 1162
Spinnaker::GenApi::double_autovector_t
_pCount, 777
_pv, 777
~double_autovector_t, 775
double_autovector_t, 775
operator delete, 776
operator new, 776
operator=, 776
operator[], 776
size, 776
Spinnaker::GenApi::int64_autovector_t
_pCount, 980
_pv, 980
~int64_autovector_t, 979
int64_autovector_t, 978, 979
operator delete, 979
operator new, 979
operator=, 979
operator[], 980
size, 980
Spinnaker::GenICam, 451
getline, 452
ThrowBadAlloc, 452
Spinnaker::GenICam::AutoLock
~AutoLock, 462
AutoLock, 462
Spinnaker::GenICam::CGlobalLock
~CGlobalLock, 695
CGlobalLock, 694, 695
IsValid, 695
Lock, 695
m_DebugCount, 696
TryLock, 695
Unlock, 696
Spinnaker::GenICam::CGlobalLockUnlocker
~CGlobalLockUnlocker, 697
CGlobalLockUnlocker, 697
m_Lock, 698
m_enabled, 698
UnlockEarly, 697
Spinnaker::GenICam::CLock
~CLock, 711
CLock, 711

Lock, 712
TryLock, 712
Unlock, 712
Spinnaker::GenICam::LockableObject
GetLock, 1028
Lock, 1028
m_Lock, 1028
Spinnaker::GenICam::LockableObject::Lock
~Lock, 1026
Lock, 1026
Spinnaker::GenICam::Version_t
Major, 1164
Minor, 1164
SubMinor, 1164
Spinnaker::GenICam::gcstring
_npos, 826
~gcstring, 826
append, 826
assign, 826, 827
c_str, 827
compare, 827
empty, 827
find, 827, 828
find_first_not_of, 828
find_first_of, 828
gcstring, 825
length, 829
max_size, 829
npos, 833
operator const char *, 829
operator delete, 829
operator new, 829
operator!=, 830
operator<, 831
operator>, 831
operator+, 832
operator+=, 830
operator=, 831
operator==, 831
resize, 831
size, 831
substr, 832
swap, 832
Spinnaker::ICameraBase
~ICameraBase, 848
BeginAcquisition, 848
CameraInternal, 854
DeInit, 848
DiscoverMaxPacketSize, 849
EndAcquisition, 849
ForceIP, 849
GetAccessMode, 849
GetBufferOwnership, 849
GetGuiXml, 849
GetNextImage, 850
GetNodeMap, 850
GetNumDataStreams, 850
GetNumImagesInUse, 850
GetTLDeviceNodeMap, 850
GetTLStreamNodeMap, 850
GetUniqueID, 851
GetUserBufferCount, 851
GetUserBufferSize, 851
GetUserBufferTotalSize, 851
ICameraBase, 848
Init, 851
InterfaceImpl, 854
IsInitialized, 851
IsStreaming, 852
IsValid, 852
m_pCameraBaseData, 854
operator=, 852
ReadPort, 852
RegisterEventHandler, 852
SetBufferOwnership, 853
SetUserBuffers, 853
TLDevice, 854
TLStream, 854
UnregisterEventHandler, 853
WritePort, 853
Spinnaker::ICameraList
~ICameraList, 856
Append, 856
CameraListImpl, 858
Clear, 856
GetByDeviceID, 857
GetByIndex, 857
GetBySerial, 857
GetSize, 857
ICameraList, 856
InterfaceImpl, 858
m_pCameraListData, 859
operator=, 857
operator[], 857
RemoveByDeviceID, 858
RemoveByIndex, 858
RemoveBySerial, 858
Spinnaker::IChunkData
~IChunkData, 860
GetBlackLevel, 861
GetCRC, 861
GetCounterValue, 861
GetEncoderValue, 861
GetExposureEndLineStatusAll, 861
GetExposureTime, 861
GetFrameID, 862
GetGain, 862
GetHeight, 862
GetImage, 862
GetInferenceBoundingBoxResult, 862
GetInferenceConfidence, 862
GetInferenceFrameId, 863
GetInferenceResult, 863
GetLinePitch, 863
GetLineStatusAll, 863
GetOffsetX, 863

GetOffsetY, 863
GetPartSelector, 864
GetPixelDynamicRangeMax, 864
GetPixelDynamicRangeMin, 864
GetScan3dAxisMax, 864
GetScan3dAxisMin, 864
GetScan3dCoordinateOffset, 864
GetScan3dCoordinateReferenceValue, 865
GetScan3dCoordinateScale, 865
GetScan3dInvalidDataValue, 865
GetScan3dTransformValue, 865
GetScanLineSelector, 865
GetSequencerSetActive, 865
GetSerialDataLength, 866
GetStreamChannelID, 866
GetTimerValue, 866
GetTimestamp, 866
GetTimestampLatchValue, 866
GetTransferBlockID, 866
GetTransferQueueCurrentBlockCount, 867
GetWidth, 867
IChunkData, 860
SetChunks, 867
Spinnaker::IDataStream
~IDataStream, 868
AnnounceImage, 869
AttachBuffer, 869
CleanupChunkAdapter, 869
FlushQueueAllDiscard, 870
GetBufferChunkData, 870
GetBufferInfoBool8Type, 870
GetBufferInfoPtrType, 870
GetBufferInfoSizeType, 870
GetBufferInfoUInt64Type, 870
GetDeviceNodeMap, 871
GetNextImage, 871
GetNextImageInternal, 871
GetNodeMap, 871
GetNumImagesInUse, 871
GetPort, 871
GetStreamInfoBool8Type, 871
GetStreamInfoSizeType, 872
GetStreamType, 872
IDataStream, 869
InitChunkAdapter, 872
IsCRCCheckEnabled, 872
IsImageInUse, 872
IsStreaming, 872
KillBufferEvent, 872
RegisterImageEventHandler, 873
ReleaseImage, 873
ResetStreamHandle, 873
RevokeImages, 873
StartStream, 873
StopStream, 873
TransportLayerStreamInfo, 873
UnregisterImageEventHandler, 874
WaitOnImageEvent, 874
Spinnaker::IDeviceArrivalEventHandler
~IDeviceArrivalEventHandler, 880
IDeviceArrivalEventHandler, 880
OnDeviceArrival, 880
operator=, 880
Spinnaker::IDeviceEventHandler
~IDeviceEventHandler, 882
GetDeviceEventId, 882
GetDeviceEventName, 882
IDeviceEventHandler, 882
OnDeviceEvent, 883
operator=, 883
Spinnaker::IDeviceRemovalEventHandler
~IDeviceRemovalEventHandler, 884
IDeviceRemovalEventHandler, 884
OnDeviceRemoval, 885
operator=, 885
Spinnaker::IImage
~IImage, 887
CalculateStatistics, 887
CheckCRC, 887
Convert, 887, 888
DeepCopy, 888
GetBitsPerPixel, 888
GetBufferSize, 888
GetChunkData, 888
GetChunkLayoutId, 889
GetColorProcessing, 889
GetData, 889
GetDataAbsoluteMax, 889
GetDataAbsoluteMin, 889
GetFrameID, 889
GetHeight, 890
GetID, 890
GetImageData, 890
GetImageSize, 890
GetImageStatus, 890
GetNumChannels, 890
GetPayloadType, 891
GetPixelFormat, 891
GetPixelFormatIntType, 891
GetPixelFormatName, 891
GetPrivateData, 891
GetStride, 891
GetTLPayloadType, 892
GetTLPixelFormat, 892
GetTLPixelFormatNamespace, 892
GetTimeStamp, 892
GetValidPayloadSize, 892
GetWidth, 892
GetXOffset, 893
GetXPadding, 893
GetYOffset, 893
GetYPadding, 893
HasCRC, 893
IImage, 887
IsInUse, 894
IsIncomplete, 893

Release, 894
ResetImage, 894
Save, 894–896
Stream, 896
Spinnaker::IImageEventHandler
~IImageEventHandler, 898
IImageEventHandler, 898
OnImageEvent, 898
operator=, 898
Spinnaker::IImageStatistics
~IImageStatistics, 900
DisableAll, 900
EnableAll, 900
EnableGreyOnly, 900
EnableHSLOnly, 900
EnableRGBOnly, 901
GetChannelStatus, 901
GetHistogram, 901
GetMean, 901
GetNumPixelValues, 901
GetPixelValueRange, 902
GetRange, 902
GetStatistics, 902
IImageStatistics, 900
SetChannelStatus, 902
Spinnaker::IInterface
~IInterface, 904
GetCameras, 905
GetTLNodeMap, 905
IInterface, 904, 905
InterfaceInternal, 906
IsInUse, 905
IsValid, 905
m_pInterfaceData, 907
operator=, 905
ProducerImpl, 907
RegisterEventHandler, 906
SendActionCommand, 906
TLInterface, 907
UnregisterEventHandler, 906
UpdateCameras, 906
Spinnaker::IInterfaceArrivalEventHandler
~IInterfaceArrivalEventHandler, 909
IInterfaceArrivalEventHandler, 909
OnInterfaceArrival, 909
operator=, 909
Spinnaker::IInterfaceEventHandler
~IInterfaceEventHandler, 911
IInterfaceEventHandler, 911
OnDeviceArrival, 912
OnDeviceRemoval, 912
operator=, 912
Spinnaker::IInterfaceList
~IInterfaceList, 914
Append, 914
Clear, 914
GetByIndex, 914
GetSize, 915
IInterfaceList, 914
InterfaceListImpl, 915
m_pInterfaceListData, 915
operator=, 915
operator[], 915
Spinnaker::IInterfaceRemovalEventHandler
~IInterfaceRemovalEventHandler, 917
IInterfaceRemovalEventHandler, 917
OnInterfaceRemoval, 917
operator=, 917
Spinnaker::ILoggingEventHandler
~ILoggingEventHandler, 919
ILoggingEventHandler, 919
OnLogEvent, 919
operator=, 919
Spinnaker::ISystem
~ISystem, 1015
GetCameras, 1016
GetInterfaces, 1016
GetLibraryVersion, 1016
GetLoggingEventPriorityLevel, 1016
GetTLNodeMap, 1016
ISystem, 1015
IsInUse, 1017
operator=, 1017
RegisterEventHandler, 1017
RegisterInterfaceEventHandler, 1017
RegisterLoggingEventHandler, 1017
ReleaseInstance, 1017
SendActionCommand, 1018
SetLoggingEventPriorityLevel, 1018
SystemPtrInternal, 1019
TLSystem, 1019
UnregisterAllLoggingEventHandlers, 1018
UnregisterEventHandler, 1018
UnregisterInterfaceEventHandler, 1018
UnregisterLoggingEventHandler, 1019
UpdateCameras, 1019
UpdateInterfaceList, 1019
Spinnaker::ISystemEventHandler
~ISystemEventHandler, 1021
ISystemEventHandler, 1021
OnInterfaceArrival, 1021
OnInterfaceRemoval, 1021
operator=, 1022
Spinnaker::Image
~Image, 924
CalculateStatistics, 924
CheckCRC, 925
Convert, 925, 926
Create, 926, 927
CreateShared, 927
DeepCopy, 927, 928
GetBitsPerPixel, 928
GetBufferSize, 928
GetChunkData, 928
GetChunkLayoutId, 928
GetColorProcessing, 929

GetData, 929
GetDataAbsoluteMax, 929
GetDataAbsoluteMin, 930
GetDefaultColorProcessing, 930
GetFrameID, 930
GetHeight, 931
GetID, 931
GetImageData, 931
GetImageSize, 931
GetImageStatus, 932
GetImageStatusDescription, 932
GetNumChannels, 932
GetPayloadType, 932
GetPixelFormat, 933
GetPixelFormatIntType, 933
GetPixelFormatName, 933
GetPrivateData, 934
GetStride, 934
GetTLPayloadType, 935
GetTLPixelFormat, 935
GetTLPixelFormatNamespace, 935
GetTimeStamp, 934
GetValidPayloadSize, 936
GetWidth, 936
GetXOffset, 936
GetXPadding, 937
GetYOffset, 937
GetYPadding, 937
HasCRC, 938
IDataStream, 943
Image, 924
ImageConverter, 943
ImageFiler, 943
ImageStatsCalculator, 943
ImageUtilityImpl, 944
ImageUtilityPolarizationImpl, 944
IsCompressed, 938
IsInUse, 938
IsIncomplete, 938
Release, 939
ResetImage, 939
Save, 940–942
SetDefaultColorProcessing, 943
Stream, 944
Spinnaker::ImageEventHandler
~ImageEventHandler, 946
ImageEventHandler, 945
OnImageEvent, 946
operator=, 946
Spinnaker::ImagePtr
~ImagePtr, 951
ImagePtr, 951
operator=, 952
Spinnaker::ImageStatistics
~ImageStatistics, 954
DisableAll, 954
EnableAll, 954
EnableGreyOnly, 954
EnableHSLOnly, 955
EnableRGBOnly, 955
GetChannelStatus, 955
GetHistogram, 955
GetMean, 956
GetNumPixelValues, 956
GetPixelValueRange, 956
GetRange, 957
GetStatistics, 957
ImageStatistics, 954
ImageStatsCalculator, 959
operator=, 958
SetChannelStatus, 958
Spinnaker::ImageUtility
CreateNormalized, 961–963
CreateScaled, 963
ImageScalingAlgorithm, 960
SourceDataRange, 960
Spinnaker::ImageUtilityHeatmap
CreateHeatmap, 965, 966
GetHeatmapColorGradient, 966
GetHeatmapRange, 967
HeatmapColor, 965
SetHeatmapColorGradient, 967
SetHeatmapRange, 967
Spinnaker::ImageUtilityPolarization
CreateAolp, 970
CreateDolp, 970, 971
CreateGlareReduced, 971, 972
CreateStokesS0, 972
CreateStokesS1, 973
CreateStokesS2, 974
ExtractPolarQuadrant, 975
PolarizationQuadrant, 969
Spinnaker::Interface
~Interface, 988
GetCameras, 988
GetTLNodeMap, 989
InterfaceInternal, 992
IsInUse, 989
IsValid, 989
RegisterEventHandler, 990
SendActionCommand, 990
UnregisterEventHandler, 991
UpdateCameras, 991
Spinnaker::InterfaceArrivalEventHandler
~InterfaceArrivalEventHandler, 994
InterfaceArrivalEventHandler, 993
OnInterfaceArrival, 994
operator=, 994
Spinnaker::InterfaceEventHandler
~InterfaceEventHandler, 996
InterfaceEventHandler, 996
OnDeviceArrival, 996
OnDeviceRemoval, 997
operator=, 997
Spinnaker::InterfaceList
~InterfaceList, 1003

Append, 1004
Clear, 1004
GetByIndex, 1004
GetSize, 1005
InterfaceList, 1003
operator=, 1005
operator[], 1005
ProducerImpl, 1005
SystemImpl, 1006
Spinnaker::InterfacePtr
InterfacePtr, 1007
Spinnaker::InterfaceRemovalEventHandler
~InterfaceRemovalEventHandler, 1009
InterfaceRemovalEventHandler, 1009
OnInterfaceRemoval, 1009
operator=, 1010
Spinnaker::JPEGOption
JPEGOption, 1022
progressive, 1023
quality, 1023
reserved, 1023
Spinnaker::JPG2Option
JPG2Option, 1024
quality, 1024
reserved, 1024
Spinnaker::LibraryVersion
build, 1025
major, 1025
minor, 1025
type, 1025
Spinnaker::LoggingEventData
~LoggingEventData, 1029
GetCategoryName, 1030
GetLogMessage, 1030
GetNDC, 1030
GetPriority, 1030
GetPriorityName, 1031
GetThreadName, 1031
GetTimestamp, 1031
LoggingEventData, 1029
SystemImpl, 1031
Spinnaker::LoggingEventDataPtr
LoggingEventDataPtr, 1033
Spinnaker::LoggingEventHandler
~LoggingEventHandler, 1035
LoggingEventHandler, 1035
OnLogEvent, 1035
operator=, 1036
Spinnaker::PGMOption
binaryFile, 1067
PGMOption, 1067
reserved, 1067
Spinnaker::PNGOption
compressionLevel, 1068
interlaced, 1068
PNGOption, 1068
reserved, 1068
Spinnaker::PPMOption
binaryFile, 1082
PPMOption, 1082
reserved, 1082
Spinnaker::System
~System, 1102
GetCameras, 1102
GetInstance, 1103
GetInterfaces, 1103
GetLibraryVersion, 1103
GetLoggingEventPriorityLevel, 1104
GetTLNodeMap, 1104
IsInUse, 1104
RegisterEventHandler, 1105
RegisterInterfaceEventHandler, 1105
RegisterLoggingEventHandler, 1106
ReleaseInstance, 1106
SendActionCommand, 1106
SetLoggingEventPriorityLevel, 1107
System, 1102
UnregisterAllLoggingEventHandlers, 1108
UnregisterEventHandler, 1108
UnregisterInterfaceEventHandler, 1108
UnregisterLoggingEventHandler, 1108
UpdateCameras, 1109
UpdateInterfaceList, 1109
Spinnaker::SystemEventHandler
~SystemEventHandler, 1112
OnInterfaceArrival, 1112
OnInterfaceRemoval, 1112
operator=, 1113
SystemEventHandler, 1111
Spinnaker::SystemPtr
~SystemPtr, 1118
SystemPtr, 1118
Spinnaker::TIFFOption
compression, 1120
CompressionMethod, 1119
reserved, 1120
TIFFOption, 1120
Spinnaker::TransportLayerDevice
~TransportLayerDevice, 1123
CameraBase, 1123
CameraInternal, 1123
DeviceAccessStatus, 1124
DeviceCurrentSpeed, 1124
DeviceDisplayName, 1124
DeviceDriverVersion, 1124
DeviceEndianessMechanism, 1124
DeviceID, 1125
DeviceInstanceId, 1125
DeviceIsUpdater, 1125
DeviceLinkSpeed, 1125
DeviceLocation, 1125
DeviceModelName, 1125
DeviceMulticastMonitorMode, 1126
DeviceSerialNumber, 1126
DeviceType, 1126
DeviceU3VProtocol, 1126

DeviceUserID, 1126
DeviceVendorName, 1126
DeviceVersion, 1127
GUIXMLLocation, 1130
GUIXMLPath, 1130
GenICamXMLLocation, 1127
GenICamXMLPath, 1127
GevCCP, 1127
GevDeviceAutoForceIP, 1127
GevDeviceDiscoverMaximumPacketSize, 1127
GevDeviceForceGateway, 1128
GevDeviceForceIPAddress, 1128
GevDeviceForceIP, 1128
GevDeviceForceSubnetMask, 1128
GevDeviceGateway, 1128
GevDeviceIPAddress, 1128
GevDeviceIsWrongSubnet, 1129
GevDeviceMACAddress, 1129
GevDeviceMaximumPacketSize, 1129
GevDeviceMaximumRetryCount, 1129
GevDeviceModeIsBigEndian, 1129
GevDevicePort, 1129
GevDeviceReadAndWriteTimeout, 1130
GevDeviceSubnetMask, 1130
GevVersionMajor, 1130
GevVersionMinor, 1130
ICameraBase, 1124
TransportLayerDevice, 1123
Spinnaker::TransportLayerInterface
~TransportLayerInterface, 1133
ActionCommand, 1134
DeviceAccessStatus, 1134
DeviceCount, 1135
DeviceID, 1135
DeviceModelName, 1135
DeviceSelector, 1135
DeviceSerialNumber, 1135
DeviceUnlock, 1135
DeviceUpdateList, 1136
DeviceVendorName, 1136
FilterDriverStatus, 1136
GevActionDeviceKey, 1136
GevActionGroupKey, 1136
GevActionGroupMask, 1136
GevActionTime, 1137
GevDeviceAutoForceIP, 1137
GevDeviceForceGateway, 1137
GevDeviceForceIPAddress, 1137
GevDeviceForceIP, 1137
GevDeviceForceSubnetMask, 1137
GevDeviceGateway, 1138
GevDeviceIPAddress, 1138
GevDeviceMACAddress, 1138
GevDeviceSubnetMask, 1138
GevInterfaceGateway, 1138
GevInterfaceGatewaySelector, 1138
GevInterfaceMACAddress, 1139
GevInterfaceMTU, 1139
GevInterfaceReceiveLinkSpeed, 1139
GevInterfaceSubnetIPAddress, 1139
GevInterfaceSubnetMask, 1139
GevInterfaceSubnetSelector, 1139
GevInterfaceTransmitLinkSpeed, 1140
HostAdapterDriverVersion, 1140
HostAdapterName, 1140
HostAdapterVendor, 1140
IInterface, 1134
IncompatibleDeviceCount, 1140
IncompatibleDeviceID, 1140
IncompatibleDeviceModelName, 1141
IncompatibleDeviceSelector, 1141
IncompatibleDeviceVendorName, 1141
IncompatibleGevDeviceIPAddress, 1141
IncompatibleGevDeviceMACAddress, 1141
IncompatibleGevDeviceSubnetMask, 1141
Interface, 1134
InterfaceDisplayName, 1142
InterfaceID, 1142
InterfaceInternal, 1134
InterfaceType, 1142
POEStatus, 1142
TransportLayerInterface, 1133, 1134
Spinnaker::TransportLayerStream
~TransportLayerStream, 1144
CameraBase, 1145
CameraInternal, 1145
GevFailedPacketCount, 1145
GevMaximumNumberResendRequests, 1145
GevPacketResendMode, 1145
GevPacketResendTimeout, 1145
GevResendPacketCount, 1146
GevResendRequestCount, 1146
GevTotalPacketCount, 1146
ICameraBase, 1145
StreamAnnounceBufferMinimum, 1146
StreamAnnouncedBufferCount, 1146
StreamBlockTransferSize, 1146
StreamBufferAlignment, 1147
StreamBufferCountManual, 1147
StreamBufferCountMax, 1147
StreamBufferCountMode, 1147
StreamBufferCountResult, 1147
StreamBufferHandlingMode, 1147
StreamCRCCheckEnable, 1148
StreamChunkCountMaximum, 1148
StreamDeliveredFrameCount, 1148
StreamFailedBufferCount, 1148
StreamID, 1148
StreamInputBufferCount, 1148
StreamIsGrabbing, 1149
StreamLostFrameCount, 1149
StreamOutputBufferCount, 1149
StreamStartedFrameCount, 1149
StreamType, 1149
TransportLayerStream, 1144
Spinnaker::TransportLayerSystem

~TransportLayerSystem, 1152
EnumerateGEVInterfaces, 1152
GenTLSFNCVersionMajor, 1153
GenTLSFNCVersionMinor, 1153
GenTLSFNCVersionSubMinor, 1153
GenTLVersionMajor, 1153
GenTLVersionMinor, 1153
GevInterfaceDefaultGateway, 1153
GevInterfaceDefaultIPAddress, 1154
GevInterfaceDefaultSubnetMask, 1154
GevInterfaceMACAddress, 1154
GevVersionMajor, 1154
GevVersionMinor, 1154
ISystem, 1152
InterfaceDisplayName, 1154
InterfaceID, 1155
InterfaceSelector, 1155
InterfaceUpdateList, 1155
System, 1152
SystemPtrInternal, 1152
TLDisplayName, 1155
TLFileName, 1155
TLID, 1155
TLModelName, 1156
TLPath, 1156
TLType, 1156
TLVendorName, 1156
TLVersion, 1156
TransportLayerSystem, 1151, 1152
Spinnaker::Video, 453
Spinnaker::Video::AVIOption
AVIOption, 463
frameRate, 464
reserved, 464
Spinnaker::Video::H264Option
bitrate, 845
frameRate, 845
H264Option, 845
height, 845
reserved, 845
width, 845
Spinnaker::Video::MJPGOption
frameRate, 1040
MJPGOption, 1040
quality, 1040
reserved, 1040
Spinnaker::Video::SpinVideo
~SpinVideo, 1089
Append, 1090
Close, 1090
Open, 1090, 1091
SetMaximumFileSize, 1092
SpinVideo, 1089
SpinnakerLogLevel
Spinnaker Definitions, 210
src/Acquisition/Acquisition.cpp, 1338
src/Acquisition/resource.h, 1340
src/Acquisition/stdafx.cpp, 1342
src/Acquisition/stdafx.h, 1351
src/Acquisition/targetver.h, 1367
src/AcquisitionMultipleCameraRecovery/Acquisition↩ MultipleCameraRecovery.cpp, 1382
src/AcquisitionMultipleCameraRecovery/resource.↩ h, 1340
src/AcquisitionMultipleThread/AcquisitionMultiple↩ Thread.cpp, 1384
src/AcquisitionMultipleThread/resource.h, 1340
src/ActionCommand/ActionCommand.cpp, 1385
src/ActionCommand/resource.h, 1340
src/ActionCommand/stdafx.cpp, 1342
src/ActionCommand/stdafx.h, 1352
src/ActionCommand/targetver.h, 1368
src/BufferHandling/BufferHandling.cpp, 1387
src/BufferHandling/resource.h, 1340
src/BufferHandling/stdafx.cpp, 1343
src/BufferHandling/stdafx.h, 1353
src/BufferHandling/targetver.h, 1369
src/ChunkData/ChunkData.cpp, 1390
src/ChunkData/resource.h, 1340
src/CounterAndTimer/CounterAndTimer.cpp, 1392
src/CounterAndTimer/resource.h, 1340
src/CounterAndTimer/stdafx.cpp, 1344
src/CounterAndTimer/stdafx.h, 1354
src/CounterAndTimer/targetver.h, 1370
src/DeviceEvents/DeviceEvents.cpp, 1394
src/DeviceEvents/resource.h, 1340
src/DeviceEvents/stdafx.cpp, 1344
src/DeviceEvents/stdafx.h, 1355
src/DeviceEvents/targetver.h, 1371
src/Enumeration/Enumeration.cpp, 1396
src/Enumeration/resource.h, 1340
src/Enumeration/stdafx.cpp, 1345
src/Enumeration/stdafx.h, 1356
src/Enumeration/targetver.h, 1372
src/Enumeration_QuickSpin/Enumeration_QuickSpin.↩ cpp, 1397
src/Enumeration_QuickSpin/resource.h, 1340
src/Enumeration_QuickSpin/stdafx.cpp, 1345
src/Enumeration_QuickSpin/stdafx.h, 1357
src/Enumeration_QuickSpin/targetver.h, 1373
src/EnumerationEvents/EnumerationEvents.cpp, 1397
src/EnumerationEvents/resource.h, 1340
src/ExceptionHandling/ExceptionHandling.cpp, 1398
src/ExceptionHandling/resource.h, 1340
src/ExceptionHandling/stdafx.cpp, 1346
src/ExceptionHandling/stdafx.h, 1358
src/ExceptionHandling/targetver.h, 1374
src/Exposure/Exposure.cpp, 1400
src/Exposure/resource.h, 1340
src/Exposure/stdafx.cpp, 1346
src/Exposure/stdafx.h, 1359
src/Exposure/targetver.h, 1375
src/Exposure_QuickSpin/Exposure_QuickSpin.cpp, 1401
src/Exposure_QuickSpin/resource.h, 1340
src/Exposure_QuickSpin/stdafx.cpp, 1347

src/Exposure_QuickSpin/stdafx.h, 1360
src/Exposure_QuickSpin/targetver.h, 1376
src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp, 1403
src/FileAccess_QuickSpin/resource.h, 1340
src/FileAccess_QuickSpin/stdafx.cpp, 1348
src/FileAccess_QuickSpin/stdafx.h, 1361
src/FileAccess_QuickSpin/targetver.h, 1377
src/GenTLInfo_QuickSpin/GenTLInfo_QuickSpin.cpp, 1406
src/GenTLInfo_QuickSpin/resource.h, 1341
src/GenTLInfo_QuickSpin/targetver.h, 1379
src/GigEVisionPerformance/CpuUtil.cpp, 1407
src/GigEVisionPerformance/CpuUtil.h, 1408
src/GigEVisionPerformance/GigEVisionPerformance.↩cpp, 1409
src/GigEVisionPerformance/GigEVisionPerformance.h, 1414
src/GigEVisionPerformance/resource.h, 1340
src/GigEVisionPerformance/stdafx.cpp, 1349
src/GigEVisionPerformance/stdafx.h, 1362
src/GigEVisionPerformance/targetver.h, 1378
src/HighDynamicRange/HighDynamicRange.cpp, 1415
src/HighDynamicRange/resource.h, 1340
src/ImageEvents/ImageEvents.cpp, 1418
src/ImageEvents/resource.h, 1341
src/ImageFormatControl/ImageFormatControl.cpp, 1420
src/ImageFormatControl/resource.h, 1341
src/ImageFormatControl/stdafx.h, 1363
src/ImageFormatControl_QuickSpin/ImageFormat↩Control_QuickSpin.cpp, 1421
src/ImageFormatControl_QuickSpin/resource.h, 1341
src/ImageFormatControl_QuickSpin/stdafx.h, 1363
src/Inference/Inference.cpp, 1422
src/Inference/resource.h, 1341
src/Logging/Logging.cpp, 1429
src/Logging/resource.h, 1341
src/LogicBlock/LogicBlock.cpp, 1430
src/LogicBlock/resource.h, 1341
src/LookupTable/LookupTable.cpp, 1432
src/LookupTable/resource.h, 1341
src/NodeMapCallback/NodeMapCallback.cpp, 1433
src/NodeMapCallback/resource.h, 1341
src/NodeMapInfo/NodeMapInfo.cpp, 1435
src/NodeMapInfo/resource.h, 1341
src/NodeMapInfo/stdafx.cpp, 1349
src/NodeMapInfo/stdafx.h, 1364
src/NodeMapInfo/targetver.h, 1379
src/Polarization/Polarization.cpp, 1439
src/Polarization/resource.h, 1341
src/Polarization/stdafx.h, 1365
src/SaveToAvi/SaveToAvi.cpp, 1442
src/SaveToAvi/resource.h, 1341
src/Sequencer/Sequencer.cpp, 1444
src/Sequencer/resource.h, 1341
src/Sequencer/stdafx.cpp, 1350
src/Sequencer/stdafx.h, 1365
src/Sequencer/targetver.h, 1380
src/SerialRxTx/SerialRxTx.cpp, 1446
src/SerialRxTx/resource.h, 1341
src/SerialRxTx/stdafx.cpp, 1350
src/SerialRxTx/stdafx.h, 1366
src/SerialRxTx/targetver.h, 1381
src/Trigger/Trigger.cpp, 1449
src/Trigger/resource.h, 1341
src/Trigger_QuickSpin/Trigger_QuickSpin.cpp, 1452
src/Trigger_QuickSpin/resource.h, 1341
Standard
    Types Enums, 382
StartCpuTracing
    CpuUtil, 392
StartPerformanceCounter
    PerformanceCounter, 393
StartRecording
    Spinnaker::GenApi::PortNode, 1073
    Spinnaker::GenApi::PortRecorder, 1077
StartSecondsCounter
    SecondsCounter, 394
StartStream
    Spinnaker::IDataStream, 873
startTime
    SecondsCounter, 394
StatisticsChannel
    Spinnaker Definitions, 210
Status
    Spinnaker::ActionCommandResult, 455
StopCpuTracing
    CpuUtil, 392
StopRecording
    IPortRecorder Interface, 335
    Spinnaker::GenApi::PortNode, 1073
    Spinnaker::GenApi::PortRecorder, 1077
StopStream
    Spinnaker::IDataStream, 873
StoreToBag
    Spinnaker::GenApi::CFeatureBag, 690
Stream
    Spinnaker::EventHandler, 801
    Spinnaker::IImage, 896
    Spinnaker::Image, 944
StreamAnnounceBufferMinimum
    Spinnaker::TransportLayerStream, 1146
StreamAnnouncedBufferCount
    Spinnaker::TransportLayerStream, 1146
StreamBlockTransferSize
    Spinnaker::TransportLayerStream, 1146
StreamBufferAlignment
    Spinnaker::TransportLayerStream, 1147
StreamBufferCountManual
    Spinnaker::TransportLayerStream, 1147
StreamBufferCountMax
    Spinnaker::TransportLayerStream, 1147
StreamBufferCountMode
    Spinnaker::TransportLayerStream, 1147
StreamBufferCountModeEnum

TransportLayerDefs Class, 223
StreamBufferCountResult
Spinnaker::TransportLayerStream, 1147
StreamBufferHandlingMode
Spinnaker::TransportLayerStream, 1147
StreamBufferHandlingModeEnum
TransportLayerDefs Class, 223
StreamCRCCheckEnable
Spinnaker::TransportLayerStream, 1148
StreamChannelId
GVCP_EVENT_ITEM_EXTENDED_ID, 838
GVCP_EVENT_ITEM, 836
StreamChunkCountMaximum
Spinnaker::TransportLayerStream, 1148
StreamDeliveredFrameCount
Spinnaker::TransportLayerStream, 1148
StreamFailedBufferCount
Spinnaker::TransportLayerStream, 1148
StreamID
Spinnaker::TransportLayerStream, 1148
StreamInputBufferCount
Spinnaker::TransportLayerStream, 1148
StreamIsGrabbing
Spinnaker::TransportLayerStream, 1149
StreamLostFrameCount
Spinnaker::TransportLayerStream, 1149
StreamOutputBufferCount
Spinnaker::TransportLayerStream, 1149
StreamStartedFrameCount
Spinnaker::TransportLayerStream, 1149
StreamType
Spinnaker::TransportLayerStream, 1149
StreamTypeEnum
TransportLayerDefs Class, 224
StringList_t
Types Enums, 377
StringNode, 1092
Spinnaker::GenApi::StringNode, 1094
StringNode Class, 370
CStringRef, 370
StringRegNode, 1096
Spinnaker::GenApi::StringRegNode, 1098, 1099
StringRegNode Class, 371
StructPort Class, 372
SubMinor
Spinnaker::GenICam::Version_t, 1164
subnetLength
AdapterConfig::IpInfo, 1013
subnetMask
AdapterConfig::IpInfo, 1013
substr
Spinnaker::GenICam::gcstring, 832
swap
Spinnaker::GenICam::gcstring, 832
sync
Spinnaker::GenApi::ODevFileStreamBuf, 1066
Synch Class, 373
System, 1100
Spinnaker::System, 1102
Spinnaker::TransportLayerSystem, 1152
System Class, 214
System.h
FLIR_SPINNAKER_VERSION_BUILD, 1332
FLIR_SPINNAKER_VERSION_MAJOR, 1332
FLIR_SPINNAKER_VERSION_MINOR, 1332
FLIR_SPINNAKER_VERSION_TYPE, 1332
SystemEventHandler, 1110
Spinnaker::SystemEventHandler, 1111
SystemEventHandler Class, 215
SystemEventHandlerImpl, 1113
~SystemEventHandlerImpl, 1114
LockEventHandlerMutex, 1115
OnInterfaceArrival, 1115
OnInterfaceRemoval, 1115
RegisterAllInterfaceEvents, 1115
RegisterInterfaceEventToSystem, 1116
SystemEventHandlerImpl, 1114
UnlockEventHandlerMutex, 1116
UnregisterAllInterfaceEvents, 1116
UnregisterInterfaceEventFromSystem, 1116
SystemImpl
Spinnaker::InterfaceList, 1006
Spinnaker::LoggingEventData, 1031
SystemPtr, 1117
Spinnaker::SystemPtr, 1118
SystemPtr Class, 216
SystemPtrInternal
Spinnaker::ISystem, 1019
Spinnaker::TransportLayerSystem, 1152

TIFFOption, 1119
Spinnaker::TIFFOption, 1120
TLDevice
Spinnaker::ICameraBase, 854
TLDisplayName
Spinnaker::TransportLayerSystem, 1155
TLFileName
Spinnaker::TransportLayerSystem, 1155
TLID
Spinnaker::TransportLayerSystem, 1155
TLInterface
Spinnaker::IInterface, 907
TLModelName
Spinnaker::TransportLayerSystem, 1156
TLParamsLocked
Spinnaker::Camera, 615
TLPath
Spinnaker::TransportLayerSystem, 1156
TLStream
Spinnaker::ICameraBase, 854
TLSystem
Spinnaker::ISystem, 1019
TLType
Spinnaker::TransportLayerSystem, 1156
TLTypeEnum
TransportLayerDefs Class, 224
TLVendorName

Spinnaker::TransportLayerSystem, 1156
TLVersion
Spinnaker::TransportLayerSystem, 1156
TWO_SECOND_DELAY
SerialRxTx.cpp, 1447
Test0001
Spinnaker::Camera, 612
TestDuration
GigEVisionPerformance.cpp, 1414
TestEventGenerate
Spinnaker::Camera, 613
TestPattern
Spinnaker::Camera, 613
TestPatternEnums
CameraDefs Class, 154
TestPatternGeneratorSelector
Spinnaker::Camera, 613
TestPatternGeneratorSelectorEnums
CameraDefs Class, 154
TestPendingAck
Spinnaker::Camera, 613
ThrowBadAlloc
Spinnaker::GenICam, 452
timeDiff
SecondsCounter, 394
TimerDelay
Spinnaker::Camera, 613
TimerDuration
Spinnaker::Camera, 614
TimerReset
Spinnaker::Camera, 614
TimerSelector
Spinnaker::Camera, 614
TimerSelectorEnums
CameraDefs Class, 155
TimerStatus
Spinnaker::Camera, 614
TimerStatusEnums
CameraDefs Class, 155
TimerTriggerActivation
Spinnaker::Camera, 614
TimerTriggerActivationEnums
CameraDefs Class, 155
TimerTriggerSource
Spinnaker::Camera, 614
TimerTriggerSourceEnums
CameraDefs Class, 156
TimerValue
Spinnaker::Camera, 615
Timestamp
Spinnaker::Camera, 615
U3V_EVENT_DATA, 1159
TimestampHigh
GVCP_EVENT_ITEM_EXTENDED_ID, 838
GVCP_EVENT_ITEM, 836
TimestampLatch
Spinnaker::Camera, 615
TimestampLatchValue
Spinnaker::Camera, 615
TimestampLow
GVCP_EVENT_ITEM_EXTENDED_ID, 838
GVCP_EVENT_ITEM, 836
TimestampReset
Spinnaker::Camera, 615
ToString
ISelectorDigit Interface, 340
IValue Class, 344
Spinnaker::GenApi::CNodeMapFactory, 728
Spinnaker::GenApi::CSelectorSet, 759
Spinnaker::GenApi::EAccessModeClass, 778
Spinnaker::GenApi::ECachingModeClass, 779
Spinnaker::GenApi::EDisplayNotationClass, 780
Spinnaker::GenApi::EEndianessClass, 781
Spinnaker::GenApi::EGenApiSchemaVersion↩
Class, 782
Spinnaker::GenApi::EInputDirectionClass, 783
Spinnaker::GenApi::ENameSpaceClass, 784
Spinnaker::GenApi::ERepresentationClass, 794
Spinnaker::GenApi::ESignClass, 795
Spinnaker::GenApi::ESlopeClass, 796
Spinnaker::GenApi::EStandardNameSpaceClass,
797
Spinnaker::GenApi::EVisibilityClass, 802
Spinnaker::GenApi::EYesNoClass, 808
Spinnaker::GenApi::ValueNode, 1163
ToXml
Spinnaker::GenApi::CNodeMapFactory, 728
ToggleHDRMode
HighDynamicRange.cpp, 1416
Tokenize
GCUtilities Utility, 284
topLeftXCoord
Chunk Data Inference Class, 173, 174
topLeftYCoord
Chunk Data Inference Class, 174
TransferAbort
Spinnaker::Camera, 616
TransferBlockCount
Spinnaker::Camera, 616
TransferBurstCount
Spinnaker::Camera, 616
TransferComponentSelector
Spinnaker::Camera, 616
TransferComponentSelectorEnums
CameraDefs Class, 157
TransferControlMode
Spinnaker::Camera, 616
TransferControlModeEnums
CameraDefs Class, 157
TransferOperationMode
Spinnaker::Camera, 616
TransferOperationModeEnums
CameraDefs Class, 158
TransferPause
Spinnaker::Camera, 617
TransferQueueCurrentBlockCount

Spinnaker::Camera, 617
TransferQueueMaxBlockCount
Spinnaker::Camera, 617
TransferQueueMode
Spinnaker::Camera, 617
TransferQueueModeEnums
CameraDefs Class, 158
TransferQueueOverflowCount
Spinnaker::Camera, 617
TransferResume
Spinnaker::Camera, 617
TransferSelector
Spinnaker::Camera, 618
TransferSelectorEnums
CameraDefs Class, 158
TransferStart
Spinnaker::Camera, 618
TransferStatus
Spinnaker::Camera, 618
TransferStatusSelector
Spinnaker::Camera, 618
TransferStatusSelectorEnums
CameraDefs Class, 159
TransferStop
Spinnaker::Camera, 618
TransferStreamChannel
Spinnaker::Camera, 618
TransferTriggerActivation
Spinnaker::Camera, 619
TransferTriggerActivationEnums
CameraDefs Class, 159
TransferTriggerMode
Spinnaker::Camera, 619
TransferTriggerModeEnums
CameraDefs Class, 159
TransferTriggerSelector
Spinnaker::Camera, 619
TransferTriggerSelectorEnums
CameraDefs Class, 160
TransferTriggerSource
Spinnaker::Camera, 619
TransferTriggerSourceEnums
CameraDefs Class, 160
transmitBuffers
AdapterConfig::AdapterInfo, 460
transmitBuffersMax
AdapterConfig::AdapterInfo, 460
transmitBuffersMin
AdapterConfig::AdapterInfo, 460
transmitBuffersRegKey
AdapterConfig::AdapterInfo, 460
transmitBuffersStep
AdapterConfig::AdapterInfo, 460
TransportLayerDefs Class, 218
DeviceAccessStatusEnum, 219
DeviceCurrentSpeedEnum, 220
DeviceEndianessMechanismEnum, 220
DeviceTypeEnum, 221
FilterDriverStatusEnum, 221
GUIXMLLocationEnum, 222
GenICamXMLLocationEnum, 221
GevCCPEnum, 222
InterfaceTypeEnum, 222
POEStatusEnum, 223
StreamBufferCountModeEnum, 223
StreamBufferHandlingModeEnum, 223
StreamTypeEnum, 224
TLTypeEnum, 224
TransportLayerDevice, 1121
Spinnaker::TransportLayerDevice, 1123
TransportLayerDevice Class, 226
TransportLayerInterface, 1131
Spinnaker::TransportLayerInterface, 1133, 1134
TransportLayerInterface Class, 227
TransportLayerStream, 1142
Spinnaker::TransportLayerStream, 1144
TransportLayerStream Class, 228
TransportLayerStreamInfo
Spinnaker::IDataStream, 873
TransportLayerSystem, 1150
Spinnaker::TransportLayerSystem, 1151, 1152
TransportLayerSystem Class, 229
Trigger.cpp
AcquireImages, 1451
chosenTrigger, 1452
ConfigureTrigger, 1451
GrabNextImageByTrigger, 1451
main, 1451
PrintDeviceInfo, 1451
ResetTrigger, 1451
RunSingleCamera, 1452
triggerType, 1450
Trigger_QuickSpin.cpp
AcquireImages, 1453
chosenTrigger, 1454
ConfigureTrigger, 1453
GrabNextImageByTrigger, 1453
main, 1453
PrintDeviceInfo, 1454
ResetTrigger, 1454
RunSingleCamera, 1454
triggerType, 1453
TriggerActivation
Spinnaker::Camera, 619
TriggerActivationEnums
CameraDefs Class, 161
TriggerDelay
Spinnaker::Camera, 619
TriggerDivider
Spinnaker::Camera, 620
TriggerEventTest
Spinnaker::Camera, 620
TriggerMode
Spinnaker::Camera, 620
TriggerModeEnums
CameraDefs Class, 162

TriggerMultiplier
Spinnaker::Camera, 620
TriggerOverlap
Spinnaker::Camera, 620
TriggerOverlapEnums
CameraDefs Class, 162
TriggerSelector
Spinnaker::Camera, 621
TriggerSelectorEnums
CameraDefs Class, 162
TriggerSoftware
Spinnaker::Camera, 621
TriggerSource
Spinnaker::Camera, 621
TriggerSourceEnums
CameraDefs Class, 162
triggerType
Trigger.cpp, 1450
Trigger_QuickSpin.cpp, 1453
TryLock
Spinnaker::GenApi::CLock, 714
Spinnaker::GenICam::CGlobalLock, 695
Spinnaker::GenICam::CLock, 712
type
Spinnaker::LibraryVersion, 1025
Types Enums, 375
_UndefindedRepresentation, 377
Automatic, 383
Beginner, 384
Boolean, 382
CL, 383
Custom, 382
Decreasing, 383
EAccessMode, 378
ECachingMode, 378
EDisplayNotation, 378
EEndianess, 380
EGenApiSchemaVersion, 380
EIncMode, 380
EInputDirection, 381
EInterfaceType, 381
ELinkType, 381
ENameSpace, 382
ERepresentation, 382
ESign, 382
ESlope, 383
EStandardNameSpace, 383
EVisibility, 383
EXMLValidation, 384
EYesNo, 384
Expert, 384
Guru, 384
Increasing, 383
Invisible, 384
Linear, 382
Logarithmic, 382
NA, 378
NI, 378
No, 384
None, 383
RO, 378
RW, 378
Signed, 383
Standard, 382
StringList_t, 377
Unsigned, 383
Varying, 383
WO, 378
Yes, 384
Types.h
interface, 1321

U3V_CHUNK_TRAILER, 1157
ChunkID, 1157
ChunkLength, 1157
U3V_COMMAND_HEADER, 1157
CommandId, 1158
Flags, 1158
Length, 1158
Prefix, 1158
ReqId, 1158
U3V_EVENT_DATA, 1158
EventId, 1159
Reserved, 1159
Timestamp, 1159
U3V_EVENT_MESSAGE, 1159
CommandHeader, 1160
EventData, 1160
U3V_EVENT_PREFIX
Spinnaker::GenApi, 450
USE_TEMP_CACHE_FILE
GCUtilities.h, 1269
underflow
Spinnaker::GenApi::IDevFileStreamBuf, 878
Unlock
Spinnaker::GenApi::CLock, 714
Spinnaker::GenICam::CGlobalLock, 696
Spinnaker::GenICam::CLock, 712
UnlockEarly
Spinnaker::GenICam::CGlobalLockUnlocker, 697
UnlockEventHandlerMutex
SystemEventHandlerImpl, 1116
UnregisterAllInterfaceEvents
SystemEventHandlerImpl, 1116
UnregisterAllLoggingEventHandlers
Spinnaker::ISystem, 1018
Spinnaker::System, 1108
UnregisterEventHandler
Spinnaker::CameraBase, 639
Spinnaker::ICameraBase, 853
Spinnaker::IInterface, 906
Spinnaker::ISystem, 1018
Spinnaker::Interface, 991
Spinnaker::System, 1108
UnregisterImageEventHandler
Spinnaker::IDataStream, 874
UnregisterInterfaceEventFromSystem

SystemEventHandlerImpl, 1116
UnregisterInterfaceEventHandler
Spinnaker::ISystem, 1018
Spinnaker::System, 1108
UnregisterLoggingEventHandler
Spinnaker::ISystem, 1019
Spinnaker::System, 1108
Unsigned
Types Enums, 383
UpdateBuffer
Spinnaker::GenApi::CChunkAdapter, 651
Spinnaker::GenApi::CChunkPort, 666
UpdateCameras
Spinnaker::IInterface, 906
Spinnaker::ISystem, 1019
Spinnaker::Interface, 991
Spinnaker::System, 1109
UpdateFirmware
SpinUpdate.h, 1328
UpdateFirmwareConsole
SpinUpdate.h, 1328
UpdateFirmwareGUI
SpinUpdate.h, 1329
UpdateInterfaceList
Spinnaker::ISystem, 1019
Spinnaker::System, 1109
UpdatorMessageCallback
SpinUpdate.h, 1329
UpdatorProgressCallback
SpinUpdate.h, 1329
UploadFileToCamera
Inference.cpp, 1427
UploadImage
FileAccess_QuickSpin.cpp, 1405
UrlDecode
GCUtilities Utility, 284
UrlEncode
GCUtilities Utility, 285
UseDuration
GigEVisionPerformance.cpp, 1414
UseMaxFramerate
GigEVisionPerformance.cpp, 1414
UserOutputSelector
Spinnaker::Camera, 621
UserOutputSelectorEnums
CameraDefs Class, 163
UserOutputValue
Spinnaker::Camera, 621
UserOutputValueAll
Spinnaker::Camera, 622
UserOutputValueAllMask
Spinnaker::Camera, 622
UserSetDefault
Spinnaker::Camera, 622
UserSetDefaultEnums
CameraDefs Class, 163
UserSetFeatureEnable
Spinnaker::Camera, 622
UserSetFramerate
GigEVisionPerformance.cpp, 1414
UserSetLoad
Spinnaker::Camera, 622
UserSetSave
Spinnaker::Camera, 623
UserSetSelector
Spinnaker::Camera, 623
UserSetSelectorEnums
CameraDefs Class, 164

V3_3Enable
Spinnaker::Camera, 623
ValidateIpAddress
AdapterConfig, 391
ValueNode, 1160
Spinnaker::GenApi::ValueNode, 1161, 1162
ValueNode Class, 385
CValueRef, 385
Varying
Types Enums, 383
Verify
IBoolean Interface, 287
Version_t, 1163
videoType
SaveToAvi.cpp, 1442

WaitForImages
ImageEvents.cpp, 1419
WaitOnImageEvent
Spinnaker::IDataStream, 874
what
Spinnaker::Exception, 807
WhiteClip
Spinnaker::Camera, 623
WhiteClipSelector
Spinnaker::Camera, 623
WhiteClipSelectorEnums
CameraDefs Class, 164
Width
Spinnaker::Camera, 624
width
Spinnaker::Video::H264Option, 845
WidthMax
Spinnaker::Camera, 624
WO
Types Enums, 378
Write
IPort Interface, 331
Spinnaker::GenApi::CChunkPort, 666
Spinnaker::GenApi::CEventPort, 687
Spinnaker::GenApi::CPortImpl, 749
Spinnaker::GenApi::CPortWriteList, 752
Spinnaker::GenApi::CRegisterPortImpl, 756
Spinnaker::GenApi::CTestPortStruct, 762
Spinnaker::GenApi::PortNode, 1074
Spinnaker::GenApi::PortRecorder, 1077
Spinnaker::GenApi::PortReplay, 1081
write

Spinnaker::GenApi::FileProtocolAdapter, 812
WritePort
Spinnaker::CameraBase, 639
Spinnaker::ICameraBase, 853
WriteRegister
Spinnaker::GenApi::CRegisterPortImpl, 756

xsputn
Spinnaker::GenApi::ODevFileStreamBuf, 1066

Yes
Types Enums, 384

z_numTriggers
BufferHandling.cpp, 1388