
PySpin API Reference

Release 4.0

Teledyne

Dec 17, 2024

CONTENTS:

1	Introduction	1
2	Software Licensing Information	3
3	Event Classes	5
3.1	PySpin.DeviceArrivalEventHandler	5
3.2	PySpin.DeviceEventHandler	5
3.3	PySpin.DeviceRemovalEventHandler	6
3.4	PySpin.EventHandler	6
3.5	PySpin.ImageEventHandler	6
3.6	PySpin.ImageListEventHandler	7
3.7	PySpin.InterfaceArrivalEventHandler	7
3.8	PySpin.InterfaceEventHandler	7
3.9	PySpin.InterfaceRemovalEventHandler	7
3.10	PySpin.LoggingEventHandler	8
3.11	PySpin.LoggingEventDataPtr	8
3.12	PySpin.SystemEventHandler	8
4	PySpin Classes	9
4.1	PySpin.CBasePtr	10
4.2	PySpin.Camera	10
4.3	PySpin.CameraBase	35
4.4	PySpin.CameraList	39
4.5	PySpin.CameraPtr	41
4.6	PySpin.ChannelStatistics	42
4.7	PySpin.ChunkData	42
4.8	PySpin.Image	46
4.9	PySpin.ImageList	55
4.10	PySpin.ImageProcessor	56
4.11	PySpin.ImagePtr	58
4.12	PySpin.ImageUtility	58
4.13	PySpin.ImageUtilityCCM	59
4.14	PySpin.ImageUtilityHeatmap	60
4.15	PySpin.ImageUtilityPolarization	62
4.16	PySpin.ImageUtilityStereo	64
4.17	PySpin.IInterface	66
4.18	PySpin.InterfaceList	67
4.19	PySpin.InterfacePtr	68
4.20	PySpin.PointCloud	68
4.21	PySpin.SpinnakerException	69

4.22	PySpin.SpinVideo	69
4.23	PySpin.System	71
4.24	PySpin.SystemPtr	75
5	QuickSpin classes	77
5.1	PySpin.TransportLayerDevice	77
5.2	PySpin.TransportLayerInterface	79
5.3	PySpin.TransportLayerStream	81
6	PySpin Module	83
6.1	Parameters:	109
6.2	Parameters:	111
6.3	Parameters:	341
6.4	Parameters:	341
6.5	Parameters:	389
6.6	Parameters:	392
6.7	Parameters:	392
	Python Module Index	427
	Index	429

INTRODUCTION

PySpin is a wrapper for Teledyne Spinnaker library.

Teledyne Machine Vision website is located at <https://www.flir.com/iis/machine-vision>.

The PySpin Python extension provides a common software interface to control and acquire images from Teledyne USB 3.0, GigE, and USB 2.0 cameras using the same API.

SOFTWARE LICENSING INFORMATION

Component	License
PySpin	Copyright (c) 2001-2023 FLIR Systems, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FLIR SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
GenICam	GenICam License http://www.emva.org/wp-content/uploads/GenICam_License_20140921.pdf
AdapterList	The Code Project Open License (CPOL) http://www.codeproject.com/info/cpol10.aspx
Boost	Boost Software License http://www.boost.org/users/license.html
FFMPEG	LGPLv2.1 License https://www.ffmpeg.org/legal.html
FreeImage	FreeImage public license http://freeimage.sourceforge.net/freeimage-license.txt
Libusb	LGPLv2. License http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt
Libraw394	LGPLv2.0 License http://www.gnu.org/licenses/old-licenses/lgpl-2.0.txt
log4Net	Apache license 2.0 https://logging.apache.org/log4net/license.html
log4Cpp	LGPL License http://log4cpp.sourceforge.net/#license
Work with Bitmaps Faster in C#	The Code Project Open License (CPOL) 1.02 http://www.codeproject.com/info/cpol10.aspx
GUI ListView Improvements	WP:CC_BY-SA License https://goo.gl/a9I9yA

EVENT CLASSES

- *PySpin.DeviceArrivalEventHandler*
- *PySpin.DeviceEventHandler*
- *PySpin.DeviceRemovalEventHandler*
- *PySpin.EventHandler*
- *PySpin.ImageEventHandler*
- *PySpin.ImageListEventHandler*
- *PySpin.InterfaceArrivalEventHandler*
- *PySpin.InterfaceEventHandler*
- *PySpin.InterfaceRemovalEventHandler*
- *PySpin.LoggingEventHandler*
- *PySpin.LoggingEventDataPtr*
- *PySpin.SystemEventHandler*

3.1 PySpin.DeviceArrivalEventHandler

class `PySpin.DeviceArrivalEventHandler`

Proxy of C++ Spinnaker::DeviceArrivalEventHandler class.

OnDeviceArrival (*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

3.2 PySpin.DeviceEventHandler

class `PySpin.DeviceEventHandler`

Proxy of C++ Spinnaker::DeviceEventHandler class.

GetDeviceEventId(*self*) → uint64_t

GetDeviceEventName(*self*) → gcstring

OnDeviceEvent(*self*, *eventName*)

Parameters

eventName (*Spinnaker::GenICam::gcstring*)

property thisown

The membership flag

3.3 PySpin.DeviceRemovalEventHandler

class PySpin.DeviceRemovalEventHandler

Proxy of C++ Spinnaker::DeviceRemovalEventHandler class.

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

3.4 PySpin.EventHandler

class PySpin.EventHandler(*args, **kwargs)

Proxy of C++ Spinnaker::EventHandler class.

GetEventPayloadData(*self*) → PyObject *

GetEventPayloadDataSize(*self*) → size_t const

GetEventType(*self*) → Spinnaker::EventType

SetEventType(*self*, *eventType*)

Parameters

eventType (*enum Spinnaker::EventType*)

property thisown

The membership flag

3.5 PySpin.ImageEventHandler

class PySpin.ImageEventHandler

Proxy of C++ Spinnaker::ImageEventHandler class.

OnImageEvent(*self*, *image*)

Parameters

image (*Spinnaker::ImagePtr*)

property thisown

The membership flag

3.6 PySpin.ImageListEventHandler

class PySpin.ImageListEventHandler

Proxy of C++ Spinnaker::ImageListEventHandler class.

OnImageListEvent(*self*, *imageList*)

Parameters

imageList (*Spinnaker::ImageList*)

property thisown

The membership flag

3.7 PySpin.InterfaceArrivalEventHandler

class PySpin.InterfaceArrivalEventHandler

Proxy of C++ Spinnaker::InterfaceArrivalEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

3.8 PySpin.InterfaceEventHandler

class PySpin.InterfaceEventHandler

Proxy of C++ Spinnaker::InterfaceEventHandler class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

3.9 PySpin.InterfaceRemovalEventHandler

class PySpin.InterfaceRemovalEventHandler

Proxy of C++ Spinnaker::InterfaceRemovalEventHandler class.

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

3.10 PySpin.LoggingEventHandler

class PySpin.LoggingEventHandler

Proxy of C++ Spinnaker::LoggingEventHandler class.

OnLogEvent(*self*, *eventPtr*)

Parameters

eventPtr (*Spinnaker::LoggingEventDataPtr*)

property thisown

The membership flag

3.11 PySpin.LoggingEventDataPtr

class PySpin.LoggingEventDataPtr(*args)

A reference tracked pointer to the LoggingEvent object.

C++ includes: LoggingEventDataPtr.h

property thisown

The membership flag

3.12 PySpin.SystemEventHandler

class PySpin.SystemEventHandler

Proxy of C++ Spinnaker::SystemEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

PYSPIN CLASSES

- *PySpin.CBasePtr*
- *PySpin.Camera*
- *PySpin.CameraBase*
- *PySpin.CameraList*
- *PySpin.CameraPtr*
- *PySpin.ChannelStatistics*
- *PySpin.ChunkData*
- *PySpin.Image*
- *PySpin.ImageList*
- *PySpin.ImageProcessor*
- *PySpin.ImagePtr*
- *PySpin.ImageUtility*
- *PySpin.ImageUtilityCCM*
- *PySpin.ImageUtilityHeatmap*
- *PySpin.ImageUtilityPolarization*
- *PySpin.ImageUtilityStereo*
- *PySpin.IInterface*
- *PySpin.InterfaceList*
- *PySpin.InterfacePtr*
- *PySpin.PointCloud*
- *PySpin.SpinnakerException*
- *PySpin.SpinVideo*
- *PySpin.System*
- *PySpin.SystemPtr*

4.1 PySpin.CBasePtr

class PySpin.CBasePtr(*args)

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

IsValid(self) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

4.2 PySpin.Camera

class PySpin.Camera(*args, **kwargs)

The camera object class.

C++ includes: Camera.h

property AasRoiEnable

property AasRoiHeight

property AasRoiOffsetX

property AasRoiOffsetY

property AasRoiWidth

property AcquisitionAbort

property AcquisitionArm

property AcquisitionBurstFrameCount

property AcquisitionFrameCount

property AcquisitionFrameRate

property AcquisitionFrameRateEnable

property AcquisitionFrameRatePersistence

property AcquisitionLineRate

property AcquisitionMode

property AcquisitionResultingFrameRate

property AcquisitionStart

property AcquisitionStatus

property AcquisitionStatusSelector

property AcquisitionStop

property AcquisitionTransferFrameRate

property ActionDeviceKey

property ActionGroupKey

property ActionGroupMask

property ActionQueueEmpty

property ActionQueueSize

property ActionSelector

property ActionSignalSize

property ActionUnconditionalMode

property AdaptiveCompressionEnable

property AdcBitDepth

property AutoAlgorithmSelector

property AutoExposureControlLoopDamping

property AutoExposureControlPriority

property AutoExposureEVCompensation

property AutoExposureExposureTimeLowerLimit

property AutoExposureExposureTimeUpperLimit

property AutoExposureGainLowerLimit

property AutoExposureGainUpperLimit

property AutoExposureGreyValueLowerLimit

property AutoExposureGreyValueUpperLimit

property AutoExposureLightingMode

property AutoExposureMeteringMode

property AutoExposureTargetGreyValue

property AutoExposureTargetGreyValueAuto

property BalanceRatio

property BalanceRatioSelector

property BalanceWhiteAuto

property BalanceWhiteAutoDamping

property BalanceWhiteAutoLowerLimit

property BalanceWhiteAutoProfile
property BalanceWhiteAutoUpperLimit
property BinningHorizontal
property BinningHorizontalMode
property BinningSelector
property BinningVertical
property BinningVerticalMode
property BlackLevel
property BlackLevelAuto
property BlackLevelAutoBalance
property BlackLevelClampingEnable
property BlackLevelRaw
property BlackLevelSelector
property BsiFlatFieldCorrectionAuto
property BsiFlatFieldCorrectionAutoDamping
property BsiFlatFieldCorrectionEnable
property BsiFlatFieldCorrectionGain
property BsiFlatFieldCorrectionGainSelector
property BufferedBurstFrameCountMax
property BufferedBurstMode
property ChunkBlackLevel
property ChunkBlackLevelSelector
property ChunkCRC
property ChunkCompressionMode
property ChunkCompressionRatio
property ChunkCounterSelector
property ChunkCounterValue
property ChunkCurrentDataRate
property ChunkEnable
property ChunkEncoderSelector
property ChunkEncoderStatus

property ChunkEncoderValue
property ChunkExposureEndLineStatusAll
property ChunkExposureTime
property ChunkExposureTimeSelector
property ChunkFrameID
property ChunkGain
property ChunkGainSelector
property ChunkHeight
property ChunkImage
property ChunkImageComponent
property ChunkInferenceBoundingBoxResult
property ChunkInferenceConfidence
property ChunkInferenceFrameId
property ChunkInferenceResult
property ChunkLinePitch
property ChunkLineStatusAll
property ChunkModeActive
property ChunkOffsetX
property ChunkOffsetY
property ChunkPartSelector
property ChunkPixelDynamicRangeMax
property ChunkPixelDynamicRangeMin
property ChunkPixelFormat
property ChunkRegionID
property ChunkScan3dAxisMax
property ChunkScan3dAxisMin
property ChunkScan3dCoordinateOffset
property ChunkScan3dCoordinateReferenceSelector
property ChunkScan3dCoordinateReferenceValue
property ChunkScan3dCoordinateScale
property ChunkScan3dCoordinateSelector

property ChunkScan3dCoordinateSystem
property ChunkScan3dCoordinateSystemReference
property ChunkScan3dCoordinateTransformSelector
property ChunkScan3dDistanceUnit
property ChunkScan3dInvalidDataFlag
property ChunkScan3dInvalidDataValue
property ChunkScan3dOutputMode
property ChunkScan3dTransformValue
property ChunkScanLineSelector
property ChunkSelector
property ChunkSequencerSetActive
property ChunkSerialData
property ChunkSerialDataLength
property ChunkSerialReceiveOverflow
property ChunkSourceID
property ChunkStreamChannelID
property ChunkTimerSelector
property ChunkTimerValue
property ChunkTimestamp
property ChunkTimestampLatchValue
property ChunkTransferBlockID
property ChunkTransferQueueCurrentBlockCount
property ChunkTransferStreamID
property ChunkWidth
property ClConfiguration
property ClTimeSlotsCount
property ColorTransformationEnable
property ColorTransformationSelector
property ColorTransformationValue
property ColorTransformationValueSelector
property ComponentActiveCount

property ComponentDestination
property ComponentEnable
property ComponentSelector
property CompressedFrameDropCount
property CompressionSaturationPriority
property ControlPacketsReservedBandwidth
property CounterDelay
property CounterDuration
property CounterEventActivation
property CounterEventSource
property CounterReset
property CounterResetActivation
property CounterResetSource
property CounterSelector
property CounterStatus
property CounterTriggerActivation
property CounterTriggerSource
property CounterValue
property CounterValueAtReset
property CxpConnectionSelector
property CxpConnectionTestErrorCount
property CxpConnectionTestMode
property CxpConnectionTestPacketCount
property CxpLinkConfiguration
property CxpLinkConfigurationPreferred
property CxpLinkConfigurationStatus
property CxpPoCxpAuto
property CxpPoCxpStatus
property CxpPoCxpTripReset
property CxpPoCxpTurnOff
property DecimationHorizontal

property DecimationHorizontalMode
property DecimationSelector
property DecimationVertical
property DecimationVerticalMode
property DefectCorrectStaticEnable
property DefectCorrectionMode
property DefectTableApply
property DefectTableCoordinateX
property DefectTableCoordinateY
property DefectTableFactoryRestore
property DefectTableIndex
property DefectTablePixelCount
property DefectTableSave
property DefectTableSensor
property Deinterlacing
property DeviceCharacterSet
property DeviceClockFrequency
property DeviceClockSelector
property DeviceConnectionSelector
property DeviceConnectionSpeed
property DeviceConnectionStatus
property DeviceEventChannelCount
property DeviceFamilyName
property DeviceFeaturePersistenceEnd
property DeviceFeaturePersistenceStart
property DeviceFirmwareVersion
property DeviceGenCPVersionMajor
property DeviceGenCPVersionMinor
property DeviceID
property DeviceIndicatorMode
property DeviceLinkBandwidthReserve

property DeviceLinkCommandTimeout
property DeviceLinkConnectionCount
property DeviceLinkCurrentThroughput
property DeviceLinkHeartbeatMode
property DeviceLinkHeartbeatTimeout
property DeviceLinkSelector
property DeviceLinkSpeed
property DeviceLinkThroughputLimit
property DeviceLinkThroughputLimitMode
property DeviceManifestEntrySelector
property DeviceManifestPrimaryURL
property DeviceManifestSchemaMajorVersion
property DeviceManifestSchemaMinorVersion
property DeviceManifestSecondaryURL
property DeviceManifestXMLMajorVersion
property DeviceManifestXMLMinorVersion
property DeviceManifestXMLSubMinorVersion
property DeviceManufacturerInfo
property DeviceMaxThroughput
property DeviceModelName
property DevicePowerSupplySelector
property DeviceRegistersCheck
property DeviceRegistersEndianness
property DeviceRegistersStreamingEnd
property DeviceRegistersStreamingStart
property DeviceRegistersValid
property DeviceReset
property DeviceSFNCVersionMajor
property DeviceSFNCVersionMinor
property DeviceSFNCVersionSubMinor
property DeviceScanType

property DeviceSensorChroma
property DeviceSerialNumber
property DeviceSerialPortBaudRate
property DeviceSerialPortSelector
property DeviceStreamChannelCount
property DeviceStreamChannelEndianness
property DeviceStreamChannelLink
property DeviceStreamChannelPacketSize
property DeviceStreamChannelSelector
property DeviceStreamChannelType
property DeviceTLType
property DeviceTLVersionMajor
property DeviceTLVersionMinor
property DeviceTLVersionSubMinor
property DeviceTapGeometry
property DeviceTemperature
property DeviceTemperatureSelector
property DeviceType
property DeviceUptime
property DeviceUserID
property DeviceVendorName
property DeviceVersion
property EncoderDivider
property EncoderMode
property EncoderOutputMode
property EncoderReset
property EncoderResetActivation
property EncoderResetSource
property EncoderSelector
property EncoderSourceA
property EncoderSourceB

property EncoderStatus
property EncoderTimeout
property EncoderValue
property EncoderValueAtReset
property EnumerationCount
property EventAcquisitionEnd
property EventAcquisitionEndFrameID
property EventAcquisitionEndTimestamp
property EventAcquisitionError
property EventAcquisitionErrorFrameID
property EventAcquisitionErrorTimestamp
property EventAcquisitionStart
property EventAcquisitionStartFrameID
property EventAcquisitionStartTimestamp
property EventAcquisitionTransferEnd
property EventAcquisitionTransferEndFrameID
property EventAcquisitionTransferEndTimestamp
property EventAcquisitionTransferStart
property EventAcquisitionTransferStartFrameID
property EventAcquisitionTransferStartTimestamp
property EventAcquisitionTrigger
property EventAcquisitionTriggerFrameID
property EventAcquisitionTriggerTimestamp
property EventActionLate
property EventActionLateFrameID
property EventActionLateTimestamp
property EventCounter0End
property EventCounter0EndFrameID
property EventCounter0EndTimestamp
property EventCounter0Start
property EventCounter0StartFrameID

property EventCounter0StartTimestamp
property EventCounter1End
property EventCounter1EndFrameID
property EventCounter1EndTimestamp
property EventCounter1Start
property EventCounter1StartFrameID
property EventCounter1StartTimestamp
property EventEncoder0Restarted
property EventEncoder0RestartedFrameID
property EventEncoder0RestartedTimestamp
property EventEncoder0Stopped
property EventEncoder0StoppedFrameID
property EventEncoder0StoppedTimestamp
property EventEncoder1Restarted
property EventEncoder1RestartedFrameID
property EventEncoder1RestartedTimestamp
property EventEncoder1Stopped
property EventEncoder1StoppedFrameID
property EventEncoder1StoppedTimestamp
property EventError
property EventErrorCode
property EventErrorFrameID
property EventErrorTimestamp
property EventExposureEnd
property EventExposureEndFrameID
property EventExposureEndTimestamp
property EventExposureStart
property EventExposureStartFrameID
property EventExposureStartTimestamp
property EventFrameBurstEnd
property EventFrameBurstEndFrameID

property EventFrameBurstEndTimestamp
property EventFrameBurstStart
property EventFrameBurstStartFrameID
property EventFrameBurstStartTimestamp
property EventFrameEnd
property EventFrameEndFrameID
property EventFrameEndTimestamp
property EventFrameStart
property EventFrameStartFrameID
property EventFrameStartTimestamp
property EventFrameTransferEnd
property EventFrameTransferEndFrameID
property EventFrameTransferEndTimestamp
property EventFrameTransferStart
property EventFrameTransferStartFrameID
property EventFrameTransferStartTimestamp
property EventFrameTrigger
property EventFrameTriggerFrameID
property EventFrameTriggerTimestamp
property EventLine0AnyEdge
property EventLine0AnyEdgeFrameID
property EventLine0AnyEdgeTimestamp
property EventLine0FallingEdge
property EventLine0FallingEdgeFrameID
property EventLine0FallingEdgeTimestamp
property EventLine0RisingEdge
property EventLine0RisingEdgeFrameID
property EventLine0RisingEdgeTimestamp
property EventLine1AnyEdge
property EventLine1AnyEdgeFrameID
property EventLine1AnyEdgeTimestamp

property EventLine1FallingEdge
property EventLine1FallingEdgeFrameID
property EventLine1FallingEdgeTimestamp
property EventLine1RisingEdge
property EventLine1RisingEdgeFrameID
property EventLine1RisingEdgeTimestamp
property EventLinkSpeedChange
property EventLinkSpeedChangeFrameID
property EventLinkSpeedChangeTimestamp
property EventLinkTrigger0
property EventLinkTrigger0FrameID
property EventLinkTrigger0Timestamp
property EventLinkTrigger1
property EventLinkTrigger1FrameID
property EventLinkTrigger1Timestamp
property EventNotification
property EventSelector
property EventSequencerSetChange
property EventSequencerSetChangeFrameID
property EventSequencerSetChangeTimestamp
property EventSerialData
property EventSerialDataLength
property EventSerialPortReceive
property EventSerialPortReceiveTimestamp
property EventSerialReceiveOverflow
property EventStream0TransferBlockEnd
property EventStream0TransferBlockEndFrameID
property EventStream0TransferBlockEndTimestamp
property EventStream0TransferBlockStart
property EventStream0TransferBlockStartFrameID
property EventStream0TransferBlockStartTimestamp

property EventStream@TransferBlockTrigger
property EventStream@TransferBlockTriggerFrameID
property EventStream@TransferBlockTriggerTimestamp
property EventStream@TransferBurstEnd
property EventStream@TransferBurstEndFrameID
property EventStream@TransferBurstEndTimestamp
property EventStream@TransferBurstStart
property EventStream@TransferBurstStartFrameID
property EventStream@TransferBurstStartTimestamp
property EventStream@TransferEnd
property EventStream@TransferEndFrameID
property EventStream@TransferEndTimestamp
property EventStream@TransferOverflow
property EventStream@TransferOverflowFrameID
property EventStream@TransferOverflowTimestamp
property EventStream@TransferPause
property EventStream@TransferPauseFrameID
property EventStream@TransferPauseTimestamp
property EventStream@TransferResume
property EventStream@TransferResumeFrameID
property EventStream@TransferResumeTimestamp
property EventStream@TransferStart
property EventStream@TransferStartFrameID
property EventStream@TransferStartTimestamp
property EventTest
property EventTestTimestamp
property EventTimer@End
property EventTimer@EndFrameID
property EventTimer@EndTimestamp
property EventTimer@Start
property EventTimer@StartFrameID

property EventTimer0StartTimestamp
property EventTimer1End
property EventTimer1EndFrameID
property EventTimer1EndTimestamp
property EventTimer1Start
property EventTimer1StartFrameID
property EventTimer1StartTimestamp
property ExposureActiveMode
property ExposureAuto
property ExposureMode
property ExposureTime
property ExposureTimeMode
property ExposureTimeSelector
property ExternalVoltageEnable
property ExternalVoltageSelector
property ExternalVoltageValue
property FactoryReset
property FfcEnable
property FfcMode
property FfcUserGain
property FfcUserOffset
property FfcUserTableReset
property FfcUserTableSave
property FfcUserTableXCoordinate
property FileAccessBuffer
property FileAccessLength
property FileAccessOffset
property FileOpenMode
property FileOperationExecute
property FileOperationResult
property FileOperationSelector

property FileOperationStatus

property FileSelector

property FileSize

property Gain

property GainAuto

property GainAutoBalance

property GainConversion

property GainSelector

property Gamma

property GammaEnable

property GevActiveLinkCount

property GevCCP

property GevCurrentDefaultGateway

property GevCurrentIPAddress

property GevCurrentIPConfigurationDHCP

property GevCurrentIPConfigurationLLA

property GevCurrentIPConfigurationPersistentIP

property GevCurrentPhysicalLinkConfiguration

property GevCurrentSubnetMask

property GevDiscoveryAckDelay

property GevFirstURL

property GevGVCPExtendedStatusCodes

property GevGVCPExtendedStatusCodesSelector

property GevGVCPHeartbeatDisable

property GevGVCPPendingAck

property GevGVCPPendingTimeout

property GevGVSPExtendedIDMode

property GevHeartbeatTimeout

property GevIEEE1588

property GevIEEE1588ClockAccuracy

property GevIEEE1588ClockId

property `GevIEEE1588DataSetLatch`
property `GevIEEE1588Mode`
property `GevIEEE1588OffsetFromMasterLatched`
property `GevIEEE1588ParentClockIdLatched`
property `GevIEEE1588Status`
property `GevIEEE1588StatusLatched`
property `GevIPConfigurationStatus`
property `GevInterfaceSelector`
property `GevMACAddress`
property `GevMCDA`
property `GevMCPHostPort`
property `GevMCRC`
property `GevMCSP`
property `GevMCTT`
property `GevNumberOfActiveLinks`
property `GevNumberOfInterfaces`
property `GevPAUSEFrameReception`
property `GevPAUSEFrameTransmission`
property `GevPersistentDefaultGateway`
property `GevPersistentIPAddress`
property `GevPersistentSubnetMask`
property `GevPhysicalLinkConfiguration`
property `GevPhysicalLinkConfigurationCapability`
property `GevPrimaryApplicationIPAddress`
property `GevPrimaryApplicationSocket`
property `GevPrimaryApplicationSwitchoverKey`
property `GevSCCFGAllInTransmission`
property `GevSCCFGExtendedChunkData`
property `GevSCCFGPacketResendDestination`
property `GevSCCFGUnconditionalStreaming`
property `GevSCDA`

property `GevSCPD`

property `GevSCPDDirection`

property `GevSCPHostPort`

property `GevSCPInterfaceIndex`

property `GevSCPSBigEndian`

property `GevSCPSDoNotFragment`

property `GevSCPSFireTestPacket`

property `GevSCPSPacketSize`

property `GevSCSP`

property `GevSCZoneConfigurationLock`

property `GevSCZoneCount`

property `GevSCZoneDirectionAll`

property `GevSecondURL`

property `GevStreamChannelSelector`

property `GevSupportedOption`

property `GevSupportedOptionSelector`

property `GevTimestampTickFrequency`

property `GuiXmlManifestAddress`

property `Height`

property `HeightMax`

property `ImageComponentEnable`

property `ImageComponentSelector`

property `ImageCompressionBitrate`

property `ImageCompressionJPEGFormatOption`

property `ImageCompressionMode`

property `ImageCompressionQuality`

property `ImageCompressionRateOption`

`Init(self)`
void Spinnaker::Camera::Init()

property `IspEnable`

property `LUTEnable`

property LUTIndex
property LUTSelector
property LUTValue
property LUTValueAll
property LargePenalty
property LensShadingCoefficientActiveSet
property LensShadingCorrectionCalibration
property LensShadingCorrectionCalibrationGainLimit
property LensShadingCorrectionCalibrationSetup
property LensShadingCorrectionCalibrationStatus
property LensShadingCorrectionMode
property LensShadingCorrectionStepSize
property LensShadingCorrectionVersion
property LineFilterWidth
property LineFormat
property LineInputFilterSelector
property LineInverter
property LineMode
property LinePitch
property LineSelector
property LineSource
property LineStatus
property LineStatusAll
property LinkErrorCount
property LinkRecoveryCount
property LinkUptime
property LogicBlockLUTInputActivation
property LogicBlockLUTInputSelector
property LogicBlockLUTInputSource
property LogicBlockLUTOutputValue
property LogicBlockLUTOutputValueAll

property LogicBlockLUTRowIndex
property LogicBlockLUTSelector
property LogicBlockSelector
property MaxDataRateThreshold
property MaxDeviceResetTime
property MultiRoiConfigurationInvalidReason
property MultiRoiConfigurationInvalidReasonAll
property MultiRoiEnable
property MultiRoiFeatureEnable
property MultiRoiHeight
property MultiRoiOffsetX
property MultiRoiOffsetY
property MultiRoiSelector
property MultiRoiWidth
property MultiRoiWindows
property NumDirections
property OffsetX
property OffsetY
property PacketResendRequestCount
property PacketResendRequestsDroppedCount
property PauseFrameCount
property PayloadSize
property PixelColorFilter
property PixelDynamicRangeMax
property PixelDynamicRangeMin
property PixelFormat
property PixelFormatInfoID
property PixelFormatInfoSelector
property PixelSize
property PowerSupplyCurrent
property PowerSupplyVoltage

property RegionDestination
property RegionMode
property RegionSelector
property ReverseX
property ReverseY
property RgbTransformLightSource
property Saturation
property SaturationEnable
property Scan3dAxisMax
property Scan3dAxisMin
property Scan3dBaseline
property Scan3dCoordinateOffset
property Scan3dCoordinateReferenceSelector
property Scan3dCoordinateReferenceValue
property Scan3dCoordinateScale
property Scan3dCoordinateSelector
property Scan3dCoordinateSystem
property Scan3dCoordinateSystemReference
property Scan3dCoordinateTransformSelector
property Scan3dDistanceUnit
property Scan3dFocalLength
property Scan3dInvalidDataFlag
property Scan3dInvalidDataValue
property Scan3dOutputMode
property Scan3dPrincipalPointU
property Scan3dPrincipalPointV
property Scan3dTransformValue
property SensorDescription
property SensorDigitizationTaps
property SensorHeight
property SensorShutterMode

property SensorTaps
property SensorWidth
property SequencerConfigurationMode
property SequencerConfigurationReset
property SequencerConfigurationValid
property SequencerFeatureEnable
property SequencerMode
property SequencerPathSelector
property SequencerSetActive
property SequencerSetLoad
property SequencerSetNext
property SequencerSetSave
property SequencerSetSelector
property SequencerSetStart
property SequencerSetValid
property SequencerTriggerActivation
property SequencerTriggerSource
property SerialPortBaudRate
property SerialPortDataBits
property SerialPortParity
property SerialPortSelector
property SerialPortSource
property SerialPortStopBits
property SerialReceiveFramingErrorCount
property SerialReceiveParityErrorCount
property SerialReceiveQueueClear
property SerialReceiveQueueCurrentCharacterCount
property SerialReceiveQueueMaxCharacterCount
property SerialTransmitQueueCurrentCharacterCount
property SerialTransmitQueueMaxCharacterCount
property Sharpening

property SharpeningAuto
property SharpeningEnable
property SharpeningThreshold
property SmallPenalty
property SoftwareSignalPulse
property SoftwareSignalSelector
property SourceCount
property SourceSelector
property StereoHeight
property StereoResolution
property StereoWidth
property TLParamsLocked
property Test0001
property TestEventGenerate
property TestPattern
property TestPatternGeneratorSelector
property TestPendingAck
property TimerDelay
property TimerDuration
property TimerReset
property TimerSelector
property TimerStatus
property TimerTriggerActivation
property TimerTriggerSource
property TimerValue
property Timestamp
property TimestampIncrement
property TimestampLatch
property TimestampLatchValue
property TimestampReset
property TotalDisparity

property TransferAbort
property TransferBlockCount
property TransferBurstCount
property TransferComponentSelector
property TransferControlMode
property TransferOperationMode
property TransferPause
property TransferQueueCurrentBlockCount
property TransferQueueMaxBlockCount
property TransferQueueMode
property TransferQueueOverflowCount
property TransferResume
property TransferSelector
property TransferStart
property TransferStatus
property TransferStatusSelector
property TransferStop
property TransferStreamChannel
property TransferTriggerActivation
property TransferTriggerMode
property TransferTriggerSelector
property TransferTriggerSource
property TransmissionDelay
property TransmissionDelayAverage
property TransmissionDelayMax
property TriggerActivation
property TriggerDelay
property TriggerDivider
property TriggerEventTest
property TriggerMode
property TriggerMultiplier

property TriggerOverlap
property TriggerSelector
property TriggerSoftware
property TriggerSource
property U3VAccessPrivilege
property U3VCPCapability
property U3VCPEIRMAvailable
property U3VCPIIDC2Available
property U3VCPSIRMAvailable
property U3VCurrentSpeed
property U3VMaxAcknowledgeTransferLength
property U3VMaxCommandTransferLength
property U3VMaxDeviceResponseTime
property U3VMessageChannelID
property U3VNumberOfStreamChannels
property U3VVersionMajor
property U3VVersionMinor
property UniquenessRatio
property UserOutputSelector
property UserOutputValue
property UserOutputValueAll
property UserOutputValueAllMask
property UserSetDefault
property UserSetFeatureEnable
property UserSetLoad
property UserSetSave
property UserSetSelector
property V3_3Enable
property WhiteClip
property WhiteClipSelector
property Width

property **WidthMax**

property **WindowSizeH**

property **WindowSizeW**

property **aPAUSEMACtrlFramesReceived**

property **aPAUSEMACtrlFramesTransmitted**

property **thisown**
 The membership flag

4.3 PySpin.CameraBase

class `PySpin.CameraBase(*args, **kwargs)`

The base class for the camera object.

C++ includes: CameraBase.h

BeginAcquisition(*self*)

void Spinnaker::CameraBase::BeginAcquisition()

Starts the image acquisition engine. The camera must be initialized via a call to `Init()` before starting an acquisition.

See: `Init()`

DeInit(*self*)

void Spinnaker::CameraBase::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: `Init()`

See: `UnregisterEvent(Event & evtToUnregister)`

DiscoverMaxPacketSize(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::DiscoverMaxPacketSize()

Returns the largest packet size that can be safely used on the interface that device is connected to

The maximum packet size returned.

EndAcquisition(*self*)

void Spinnaker::CameraBase::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: `Init()`

See: `BeginAcquisition()`

See: `GetNextImage(grabTimeout)`

See: `Image::Release()`

ForceIP(*self*)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GenApi::EAccessMode Spinnaker::CameraBase::GetAccessMode() const

Returns the access mode that the software has on the Camera. The camera does not need to be initialized before calling this function.

See: Init()

An enumeration value indicating the access mode

GetActiveNumDataStreams(*self*) → unsigned int

GetBufferOwnership(*self*) → Spinnaker::BufferOwnership

GetDeviceID(*self*) → *gcstring*

GetGuiXml(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetGuiXml() const

Returns the GUI XML that can be passed into the Spinnaker GUI framework

GenICam::gcstring that represents the uncompressed GUI XML file

GetNextImage(*self*, *grabTimeout*=EVENT_TIMEOUT_INFINITE, *streamIndex*=0) → *ImagePtr*

Parameters

- **grabTimeout** (a 64bit value that represents a timeout in milliseconds)
- **streamIndex** (*uint64_t*)
- **ImagePtr**
- **Spinnaker::CameraBase::GetNextImage(*uint64_t***
- **grabTimeout**=EVENT_TIMEOUT_INFINITE
- **streamID**=0) (*uint64_t*)
- **This** (Gets the next image that was received by the transport layer.)
- **cameras** (function will block indefinitely until an image arrives. Most)
- **camera** (support one stream so the default streamID is 0 but if a)
- **select** (supports multiple streams the user can input the streamID to)
- **images** (from which stream to grab)
- **See** (EndAcquisition())
- **See**
- **See**
- **Parameters**
- -----
- **grabTimeout**
- **streamID** (The stream to grab the image.)
- **object** (pointer to an Image)

GetNextImageSync(*self*, *grabTimeout*=EVENT_TIMEOUT_INFINITE) → *ImageList*

Parameters

grabTimeout (uint64_t)

GetNodeMap(*self*) → *INodeMap*

GenApi::INodeMap& Spinnaker::CameraBase::GetNodeMap() const

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: Init()

A reference to the INodeMap.

GetNumDataStreams(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::GetNumDataStreams()

Returns the number of streams that a device supports.

The number of data streams

GetNumImagesInUse(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::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().

The number of images that needs to be cleaned up.

GetTLDeviceNodeMap(*self*) → *INodeMap*

GenApi::INodeMap& Spinnaker::CameraBase::GetTLDeviceNodeMap() const

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.

A reference to the INodeMap.

GetTLStreamNodeMap(*self*, *streamIndex*=0) → *INodeMap*

Parameters

- **streamIndex** (uint64_t)
- **const** (GenApi::INodeMap& Spinnaker::CameraBase::GetTLStreamNodeMap())
- **XML** (Gets a reference to the node map that is generated from a GenICam)
- **be** (file for the GenTL Stream module. The camera does not need to)
- **map.** (initialized before acquiring this node)
- **INodeMap.** (A reference to the)

GetUniqueID(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetUniqueID()

This returns a unique id string that identifies the camera. This is the camera serial number.

string that uniquely identifies the camera (serial number)

GetUserBufferCount(*self*) → uint64_t

GetUserBufferSize(*self*) → uint64_t

GetUserBufferTotalSize(*self*) → uint64_t

Init(*self*)

void Spinnaker::CameraBase::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: BeginAcquisition()

See: EndAcquisition()

See: GetNodeMap()

See: GetNextImage()

IsInitialized(*self*) → bool

bool Spinnaker::CameraBase::IsInitialized()

Checks if camera is initialized. This function needs to return true in order to retrieve a valid NodeMap from the GetNodeMap() call.

See: GetNodeMap()

If camera is initialized or not

IsStreaming(*self*) → bool

bool Spinnaker::CameraBase::IsStreaming() const

Returns true if the camera is currently streaming or false if it is not.

See: Init()

returns true if camera is streaming and false otherwise.

IsValid(*self*) → bool

bool Spinnaker::CameraBase::IsValid()

Checks a flag to determine if camera is still valid for use.

If camera is valid or not

Note that CameraPtr and CameraBase both define an IsValid() function. In order to determine the validity of the camera using a CameraPtr, user must first call get() to retrieve the CameraBase object.

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::ImageEventHandler &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **eventName**)
- **evtHandlerToRegister**
- **eventName** (*Spinnaker::GenICam::gcstring const &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **streamIndex**)
- **evtHandlerToRegister**

- **streamIndex** (*uint64_t*)

SetBufferOwnership(*self, mode*)

Parameters

- mode** (*enum Spinnaker::BufferOwnership const*)

SetUserBuffers(*self, pMemBuffers, totalSize*)

Parameters

- **pMemBuffers** (*void *const*)
- **totalSize** (*uint64_t*)
- **SetUserBuffers**(*self*
- **ppMemBuffers** (*void **const*)
- **bufferCount** (*uint64_t const*)
- **bufferSize**)
- **ppMemBuffers**
- **bufferCount**
- **bufferSize** (*uint64_t const*)

UnregisterEventHandler(*self, evtHandlerToUnregister*)

Parameters

- evtHandlerToUnregister** (*Spinnaker::EventHandler &*)

property thisown

The membership flag

4.4 PySpin.CameraList

class **PySpin.CameraList**(*args)

Used to hold a list of camera objects.

C++ includes: CameraList.h

Add(*self, camera*)

Parameters

- camera** (*Spinnaker::CameraPtr*)

Append(*self, list*)

Parameters

- **list** (*Spinnaker::CameraList const &*)
- **void**
- **&otherList**) (*Spinnaker::CameraList::Append(CameraList)*)
- **list.** (*Appends a camera list to the current*)
- **Parameters**
- -----

- **otherList** (*The other list to append to this list*)

Clear(*self*)

void Spinnaker::CameraList::Clear()

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: `System:ReleaseInstance()`

GetByDeviceID(*self*, *deviceID*) → *CameraPtr*

Parameters

deviceID (*std::string*)

GetByIndex(*self*, *index*) → *CameraPtr*

Parameters

- **index** (*The index at which to retrieve the camera object*)
- **CameraPtr**
- **const** (*Spinnaker::CameraList::GetByIndex(int index)*)
- **"index"**. (*Returns a pointer to a camera object at the*)
- **Parameters**
- **-----**
- **index**
- **object.** (*A pointer to an camera*)

GetBySerial(*self*, *serialNumber*) → *CameraPtr*

Parameters

- **serialNumber** (*The serial number of the camera object to retrieve*)
- **CameraPtr**
- **const** (*Spinnaker::CameraList::GetBySerial(std::string serialNumber)*)
- **number.** (*Returns a pointer to a camera object with the specified serial*)
- **Parameters**
- **-----**
- **serialNumber**
- **object.** (*A pointer to an camera*)

GetSize(*self*) → unsigned int

int Spinnaker::CameraList::GetSize() const

Returns the size of the camera list. The size is the number of Camera objects stored in the list.

An integer that represents the list size.

Remove(*self*, *camera*)

Parameters

camera (*Spinnaker::CameraPtr*)

RemoveByDeviceID(*self*, *deviceID*)

Parameters

deviceID (*std::string*)

RemoveByIndex(*self*, *index*)

Parameters

- **index** (*The index at which to remove the Camera object*)
- **void**
- **index**) (*Spinnaker::CameraList::RemoveByIndex(int)*)
- **reference** (*Removes a camera at "index" and destroys its corresponding*)
- **object.** (*counted*)
- **Parameters**
- -----
- **index**

RemoveBySerial(*self*, *serialNumber*)

Parameters

- **serialNumber** (*The serial number of the Camera object to remove*)
- **void**
- **serialNumber**) (*Spinnaker::CameraList::RemoveBySerial(std::string)*)
- **its** (*Removes a camera using its serial number and destroys*)
- **object.** (*corresponding reference counted*)
- **Parameters**
- -----
- **serialNumber**

property thisown

The membership flag

4.5 PySpin.CameraPtr

class *PySpin.CameraPtr*(*args)

A reference tracked pointer to a camera object.

C++ includes: *CameraPtr.h*

property thisown

The membership flag

4.6 PySpin.ChannelStatistics

class PySpin.ChannelStatistics(*image, channel*)

Class used to store statistics (as properties) for one channel of an image. Properties:

- **channel**: The image channel that the statistics are based on (as an int).
- **range_min**: The smallest possible pixel value.
- **range_max**: The largest possible pixel value.
- **pixel_value_min**: The smallest pixel value in the current channel.
- **pixel_value_max**: The largest pixel value in the current channel.
- **num_pixel_values**: The total number of pixel values in the current channel.
- **pixel_value_mean**: The average pixel value in the current channel.
- **histogram**: NumPy array representing the histogram of the current channel.

property channel

property histogram

property num_pixel_values

property pixel_value_max

property pixel_value_mean

property pixel_value_min

property range_max

property range_min

property thisown

The membership flag

4.7 PySpin.ChunkData

class PySpin.ChunkData(*args)

The chunk data which contains additional information about an image.

C++ includes: ChunkData.h

GetBlackLevel(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetBlackLevel() const

Description: Returns the black level used to capture the image included in the payload. Visibility: Expert

GetCRC(*self*) → int64_t

GetCompressionMode(*self*) → int64_t

GetCompressionRatio(*self*) → float64_t

GetCounterValue(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetCounterValue() const

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event. Visibility: Expert

GetCurrentDatarate(*self*) → int64_t

GetEnable(*self*) → bool

GetEncoderValue(*self*) → int64_t

int64_t Spinnaker::ChunkData::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. Visibility: Expert

GetExposureEndLineStatusAll(*self*) → int64_t

GetExposureTime(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetExposureTime() const

Description: Returns the exposure time used to capture the image. Visibility: Expert

GetFrameID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetFrameID() const

Description: Returns the unique Identifier of the frame (or image) included in the payload. Visibility: Expert

GetGain(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetGain() const

Description: Returns the gain used to capture the image. Visibility: Expert

GetHeight(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetHeight() const

Description: Returns the Height of the image included in the payload. Visibility: Expert

GetImage(*self*) → int64_t

GetInferenceBoundingBoxResult(*self*) → *InferenceBoundingBoxResult*

GetInferenceConfidence(*self*) → float64_t

GetInferenceFrameId(*self*) → int64_t

GetInferenceResult(*self*) → int64_t

GetLinePitch(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLinePitch() const

Description: Returns the LinePitch of the image included in the payload. Visibility: Expert

GetLineStatusAll(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLineStatusAll() const

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event. Visibility: Expert

GetModeActive(*self*) → bool

GetOffsetX(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetX() const

Description: Returns the OffsetX of the image included in the payload. Visibility: Expert

GetOffsetY(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetY() const

Description: Returns the OffsetY of the image included in the payload. Visibility: Expert

GetPartSelector(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPartSelector() const

Description: Selects the part to access in chunk data in a multipart transmission. Visibility: Expert

GetPixelDynamicRangeMax(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMax() const

Description: Returns the maximum value of dynamic range of the image included in the payload. Visibility: Expert

GetPixelDynamicRangeMin(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMin() const

Description: Returns the minimum value of dynamic range of the image included in the payload. Visibility: Expert

GetScan3dAxisMax(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMax() const

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dAxisMin(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMin() const

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateOffset(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateOffset() const

Description: Returns the Offset for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateReferenceValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateReferenceValue() const

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point. Visibility: Expert

GetScan3dCoordinateScale(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateScale() const

Description: Returns the Scale for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dInvalidDataFlag(*self*) → bool

GetScan3dInvalidDataValue(*self*) → float64_t
float64_t Spinnaker::ChunkData::GetScan3dInvalidDataValue() const
Description: Returns the Invalid Data Value used for the image included in the payload. Visibility: Expert

GetScan3dTransformValue(*self*) → float64_t
float64_t Spinnaker::ChunkData::GetScan3dTransformValue() const
Description: Returns the transform value. Visibility: Expert

GetScanLineSelector(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetScanLineSelector() const
Description: Index for vector representation of one chunk value per line in an image. Visibility: Expert

GetSequencerSetActive(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetSequencerSetActive() const
Description: Return the index of the active set of the running sequencer included in the payload. Visibility: Expert

GetSerialData(*self*) → uint8_t *

GetSerialDataLength(*self*) → int64_t

GetSerialReceiveOverflow(*self*) → bool

GetStreamChannelID(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetStreamChannelID() const
Description: Returns identifier of the stream channel used to carry the block. Visibility: Expert

GetTimerValue(*self*) → float64_t
float64_t Spinnaker::ChunkData::GetTimerValue() const
Description: Returns the value of the selected Timer at the time of the FrameStart internal event. Visibility: Expert

GetTimestamp(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTimestamp() const
Description: Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event. Visibility: Expert

GetTimestampLatchValue(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTimestampLatchValue() const
Description: Returns the last Timestamp latched with the TimestampLatch command. Visibility: Expert

GetTransferBlockID(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTransferBlockID() const
Description: Returns the unique identifier of the transfer block used to transport the payload. Visibility: Expert

GetTransferQueueCurrentBlockCount(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTransferQueueCurrentBlockCount() const
Description: Returns the current number of blocks in the transfer queue. Visibility: Expert

GetWidth(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetWidth() const

Description: Returns the Width of the image included in the payload. Visibility: Expert

SetChunks(*self*, *pNodeMap*)

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap &*)
- **void**
- **&pNodeMap** (*Spinnaker::ChunkData::SetChunks(GenApi::INodeMap)*)

property thisown

The membership flag

4.8 PySpin.Image

class PySpin.**Image**(*args, **kwargs)

The image object class.

C++ includes: Image.h

CheckCRC(*self*) → bool

bool Spinnaker::Image::CheckCRC() const

Checks if the computed checksum matches with chunk data's ImageCRC

Returns true if computed checksum matches with the chunk data's CRC and false otherwise.

static Create() → *ImagePtr*

static Create(*image*) → *ImagePtr*

Parameters

- **image** (*Spinnaker::ImagePtr const*)
- **Create**(width
- **height** (*size_t*)
- **offsetX** (*size_t*)
- **offsetY** (*size_t*)
- **pixelFormat** (*enum Spinnaker::PixelFormatEnums*)
- **ImagePtr** (*copied from another*)
- **width** (*or using*)
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*void **)
- **Create**(width

- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType** (*enum Spinnaker::TLPayloadType*)
- **ImagePtr**
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType**
- **dataSize** (*size_t*)
- **object** (*Creates a new Image*)
- **constructor** (*either using a default*)
- **ImagePtr**
- **width**
- **height**

:param : :param offset_x: :param offset_y: :param pixel format: :param and a NumPy array containing 8-bit unsigned ints representing the image data: :param (replaces the void* pData argument).:

DeepCopy(*self, pSrcImage*)

Parameters

- **pSrcImage** (*The Image to copy the data from.*)
- **void**
- **pSrcImage** (*Spinnaker::Image::DeepCopy(const ImagePtr)*)
- **operation** (*Performs a deep copy of the Image. After this*)
- **image** (*the*)
- **not** (*contents and member variables will be the same. The Images will*)
- **released.** (*share a buffer. The Image's current buffer will not be*)
- **Parameters**
- -----
- **pSrcImage**

GetBitsPerPixel(*self*) → size_t

size_t Spinnaker::Image::GetBitsPerPixel() const

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.

The number of bits used per pixel.

GetBufferSize(*self*) → size_t

size_t Spinnaker::Image::GetBufferSize() const

Gets the size of the buffer associated with the image in bytes.

The size of the buffer, in bytes.

GetChunkData(*self*) → [*ChunkData*](#)

const ChunkData& Spinnaker::Image::GetChunkData() const

Returns a pointer to a chunk data interface. No ownership is transferred, the chunk data interface reference is valid until Image::Release() is called on this image.

ChunkData interface that provides access to image chunks.

GetChunkLayoutId(*self*) → uint64_t

uint64_t Spinnaker::Image::GetChunkLayoutId() const

Returns the id of the chunk data layout.

uint64_t value representing the id of the chunk data layout.

GetColorProcessing(*self*) → Spinnaker::ColorProcessingAlgorithm

ColorProcessingAlgorithm Spinnaker::Image::GetColorProcessing() const

Gets the algorithm used to produce the image.

See: Convert()

The color processing algorithm used to produce the image.

GetDataAbsoluteMax(*self*) → float

GetDataAbsoluteMin(*self*) → float

GetFrameID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetFrameID() const

Gets the frame ID for this image.

The frame ID.

GetHeight(*self*) → size_t

size_t Spinnaker::Image::GetHeight() const

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.

The height in pixels.

GetID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetID() const

Gets a unique ID for this image. Each image in a stream will have a unique ID to help identify it.

The 64 bit unique id for this image.

GetImagePayloadType(*self*) → Spinnaker::ImagePayloadType

GetImageSize(*self*) → size_t

size_t Spinnaker::Image::GetImageSize() const

Returns the size of the image

The image size in bytes.

GetImageStatus(*self*) → Spinnaker::ImageStatus

ImageStatus Spinnaker::Image::GetImageStatus() const

Returns data integrity status of the image returned from GetNextImage()

Returns whether image has any data integrity issues.

static GetImageStatusDescription(*status*) → char const *

Parameters

status (enum Spinnaker::ImageStatus)

GetNumChannels(*self*) → size_t

GetPayloadType(*self*) → size_t

size_t Spinnaker::Image::GetPayloadType() const

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.

Device types specific payload type.

GetPixelFormat(*self*) → Spinnaker::PixelFormatEnums

Spinnaker::PixelFormatEnums Spinnaker::Image::GetPixelFormat() const

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: Convert()

enum value representing the PixelFormat.

GetPixelFormatIntType(*self*) → Spinnaker::PixelFormatIntType

GetPixelFormatName(*self*) → gcstring

GenICam::gcstring Spinnaker::Image::GetPixelFormatName() const

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.

string value representing the PixelFormat.

GetPrivateData(*self*) → void *

void* Spinnaker::Image::GetPrivateData() const

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.

A pointer to the user passed data pointer.

GetStreamIndex(*self*) → uint64_t

GetStride(*self*) → size_t

size_t Spinnaker::Image::GetStride() const

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.

The stride in bytes.

GetTLPayloadType(*self*) → Spinnaker::TLPayloadType

PayloadTypeInfoIDs Spinnaker::Image::GetTLPayloadType() const

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.

Transport Layer specific payload type.

GetTLPixelFormat(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTLPixelFormat() const

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: GetTLPixelFormatNamespace()

Transport Layer specific pixel format.

GetTLPixelFormatNamespace(*self*) → Spinnaker::TLPixelFormatNamespace

PixelFormatNamespaceID Spinnaker::Image::GetTLPixelFormatNamespace() const

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: GetTLPixelFormat()

enum value representing the PixelFormatNamespace.

GetTimeStamp(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTimeStamp() const

Gets the time stamp for the image in nanoseconds.

The time stamp of the image.

GetValidPayloadSize(*self*) → size_t

size_t Spinnaker::Image::GetValidPayloadSize() const

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: GetBufferSize()

size_t value representing valid payload.

GetWidth(*self*) → size_t

size_t Spinnaker::Image::GetWidth() const

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.

The width in pixels.

GetXOffset(*self*) → size_t

size_t Spinnaker::Image::GetXOffset() const

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.

The x offset in pixels.

GetXPadding(*self*) → size_t

size_t Spinnaker::Image::GetXPadding() const

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.

The x padding in bytes.

GetYOffset(*self*) → size_t

size_t Spinnaker::Image::GetYOffset() const

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.

The y offset in pixels.

GetYPadding(*self*) → size_t

size_t Spinnaker::Image::GetYPadding() const

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.

The y padding in bytes.

HasCRC(*self*) → bool

bool Spinnaker::Image::HasCRC() const

Checks if the image contains ImageCRC checksum from chunk data

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

HasChunkData(*self*) → bool**IsCompressed(*self*)** → bool**IsInUse(*self*)** → bool

bool Spinnaker::Image::IsInUse()

Returns true if the image is still in use by the stream

Returns true if the image is in use and false otherwise.

IsIncomplete(*self*) → bool

bool Spinnaker::Image::IsIncomplete() const

Returns a boolean value indicating if this image was incomplete. An image is marked as incomplete if the transport layer received less data than it requested.

Returns true if image is incomplete, false otherwise.

static Load(*pFilename*, *format*=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT) → *ImagePtr*

Parameters

- **pFilename** (*char const **)
- **format** (*enum Spinnaker::ImageFileFormat*)

Release(*self*)

void Spinnaker::Image::Release()

ResetImage(*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

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.*)
- **ResetImage**(*self*
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**)
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*Pointer to the image buffer.*)
- **ResetImage**(*self*
- **width**
- **height**
- **offsetX**
- **offsetY**

- pixelFormat
- pData
- dataPayloadType (enum Spinnaker::TLPayloadType)
- dataSize
- width
- height
- offsetX
- offsetY
- pixelFormat
- pData
- dataPayloadType
- dataSize (size_t)
- void
- width
- height
- size_t
- offsetX
- offsetY
- pixelFormat
- void
- *pData)
- object. (*Sets new dimensions of the image*)
- Parameters
- -----
- width
- height
- offsetX
- offsetY
- pixelFormat
- pData

Save(self, pFilename, format=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT)

Parameters

- pFilename (*Filename to save image with.*)
- format (enum Spinnaker::ImageFileFormat)
- Save(self
- pFilename

- pOption)
- pFilename
- pOption(*Options to use while saving image.*)
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)

- **pFilename**
- **pOption**
- **void**
- ***pFilename** (*Spinnaker::Image::Save(const char) –*
- **&pOption**) ([BMPOption](#))
- **specified.** (*Saves the image to the specified file name with the options*)
- **Parameters**
- **-----**
- **pFilename**
- **pOption**

property thisown
The membership flag

4.9 PySpin.ImageList

class `PySpin.ImageList(*args)`

Proxy of C++ Spinnaker::ImageList class.

Add(*self, image*)

Parameters

image (*Spinnaker::ImagePtr*)

Append(*self, list*)

Parameters

list (*Spinnaker::ImageList const &*)

Clear(*self*)

GetByIndex(*self, index*) → *ImagePtr*

Parameters

index (*unsigned int*)

GetByPayloadType(*self, payloadType*) → *ImagePtr*

Parameters

payloadType (*enum Spinnaker::ImagePayloadType const*)

GetByPixelFormat(*self, pixelFormat*) → *ImagePtr*

Parameters

pixelFormat (*enum Spinnaker::PixelFormatEnums*)

GetByStreamIndex(*self, streamIndex*) → *ImagePtr*

Parameters

streamIndex (*uint64_t const*)

GetSize(*self*) → *unsigned int*

static Load(*filename*) → *ImageList*

Parameters

filename (*char const **)

Release(*self*)

RemoveByIndex(*self*, *index*)

Parameters

index (*unsigned int*)

RemoveByPayloadType(*self*, *payloadType*)

Parameters

payloadType (*enum Spinnaker::ImagePayloadType const*)

RemoveByPixelFormat(*self*, *pixelFormat*)

Parameters

pixelFormat (*enum Spinnaker::PixelFormatEnums*)

RemoveByStreamIndex(*self*, *streamIndex*)

Parameters

streamIndex (*uint64_t const*)

Save(*self*, *filename*)

Parameters

filename (*char const **)

property thisown

The membership flag

4.10 PySpin.ImageProcessor

class `PySpin.ImageProcessor(*args)`

Proxy of C++ `Spinnaker::ImageProcessor` class.

ApplyGamma(*self*, *srcImage*, *gamma*, *applyGammaInverse=False*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **gamma** (*float*)
- **applyGammaInverse** (*bool*)
- **ApplyGamma**(*self*
 - **srcImage**
 - **destImage** (*Spinnaker::ImagePtr &*)
 - **gamma**
 - **applyGammaInverse=False**)
 - **srcImage**
 - **destImage**)

- `gamma`
- `applyGammaInverse`

Convert(*self*, *srcImage*, *destFormat*) → *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `destFormat` (*enum Spinnaker::PixelFormatEnums*)
- `Convert(self`
- `srcImage`
- `destImage` (*Spinnaker::ImagePtr &*)
- `destFormat`)
- `srcImage`
- `destImage`
- `destFormat`
- `Convert(self`
- `srcImageList` (*Spinnaker::ImageList const &*)
- `ImagePtr` (*destFormat*) →)
- `srcImageList`
- `destFormat`
- `Convert(self`
- `srcImageList`
- `destImage`
- `destFormat`)
- `srcImageList`
- `destImage`
- `destFormat`

GetColorProcessing(*self*) → *Spinnaker::ColorProcessingAlgorithm*

GetNumDecompressionThreads(*self*) → unsigned int

SetColorProcessing(*self*, *colorAlgorithm*)

Parameters

- `colorAlgorithm` (*enum Spinnaker::ColorProcessingAlgorithm*)

SetNumDecompressionThreads(*self*, *numThreads*)

Parameters

- `numThreads` (*unsigned int*)

property thisown

The membership flag

4.11 PySpin.ImagePtr

class PySpin.**ImagePtr**(*args)

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.

C++ includes: ImagePtr.h

property thisown

The membership flag

4.12 PySpin.ImageUtility

class PySpin.**ImageUtility**

Proxy of C++ Spinnaker::ImageUtility class.

static CreateNormalized(srcImage, destPixelFormat, src-
DataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)
→ *ImagePtr*

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **destPixelFormat** (enum Spinnaker::PixelFormatEnums const)
- **srcDataRange** (enum Spinnaker::SourceDataRange)
- **CreateNormalized**(srcImage
- **min** (double const)
- **max** (double const)
- **ImagePtr** (srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE
->)
- **srcImage**
- **min**
- **max**
- **srcDataRange**
- **CreateNormalized**(srcImage
- **min**
- **max**
- **destPixelFormat**
- **ImagePtr**
- **srcImage**
- **min**
- **max**
- **destPixelFormat**
- **srcDataRange**

- `CreateNormalized(srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `srcDataRange`
- `CreateNormalized(srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange`

static `CreateScaled(srcImage, scalingAlg, scalingFactor) → ImagePtr`

Parameters

- `srcImage (Spinnaker::ImagePtr const &)`
- `scalingAlg (enum Spinnaker::ImageScalingAlgorithm)`
- `scalingFactor (double)`
- `CreateScaled(srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `scalingAlg`
- `scalingFactor)`
- `srcImage`
- `destImage`
- `scalingAlg`
- `scalingFactor`

property thisown

The membership flag

4.13 PySpin.ImageUtilityCCM

class `PySpin.ImageUtilityCCM`

Proxy of C++ Spinnaker::ImageUtilityCCM class.

```
static ApplicationToString(application) → std::string

    Parameters
        application (Spinnaker::CCMApplication const &)

static ColorSpaceToString(colorSpace) → std::string

    Parameters
        colorSpace (Spinnaker::CCMColorSpace const &)

static ColorTemperatureToString(colorTemperature) → std::string

    Parameters
        colorTemperature (Spinnaker::CCMColorTemperature const &)

static CreateColorCorrected(srcImage, settings) → ImagePtr

    Parameters
        • srcImage (Spinnaker::ImagePtr const &)
        • settings (Spinnaker::CCMSettings const &)
        • CreateColorCorrected(srcImage
        • destImage (Spinnaker::ImagePtr &)
        • settings)
        • srcImage
        • destImage
        • settings

static EncryptColorCorrectionMatrix(ccmMatrixEntries) → std::string

    Parameters
        ccmMatrixEntries (std::string)

static SensorToString(sensor) → std::string

    Parameters
        sensor (Spinnaker::CCMSensor const &)

static TypeToString(type) → std::string

    Parameters
        type (Spinnaker::CCMType const &)

property thisown
    The membership flag
```

4.14 PySpin.ImageUtilityHeatmap

```
class PySpin.ImageUtilityHeatmap
    Proxy of C++ Spinnaker::ImageUtilityHeatmap class.

    static CreateHeatmap(srcImage) → ImagePtr

        Parameters
            • srcImage (Spinnaker::ImagePtr const &)
```


- **CreateHeatmap**(srcImage
- **min** (*float const*)
- **max** (*float const*)
- **lowColor** (*enum Spinnaker::HeatmapColor const*)
- **highColor** (*enum Spinnaker::HeatmapColor const*)
- **doCheckInvalidVal** (*bool const*)
- **ImagePtr** (*invalidVal*) ->)
- **srcImage**
- **min**
- **max**
- **lowColor**
- **highColor**
- **doCheckInvalidVal**
- **invalidVal** (*unsigned int const*)
- **CreateHeatmap**(srcImage
- **destImage**)
- **srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)

static GetHeatmapColorGradient(*currentLowColor, currentHighColor*)

Parameters

- **currentLowColor** (*Spinnaker::HeatmapColor &*)
- **currentHighColor** (*Spinnaker::HeatmapColor &*)

static GetHeatmapRange(*currentLowValue, currentHighValue*)

Parameters

- **currentLowValue** (*unsigned int &*)
- **currentHighValue** (*unsigned int &*)

static SetHeatmapColorGradient(*newLowColor, newHighColor*)

Parameters

- **newLowColor** (*enum Spinnaker::HeatmapColor const*)
- **newHighColor** (*enum Spinnaker::HeatmapColor const*)

static SetHeatmapRange(*newLowValue, newHighValue*)

Parameters

- **newLowValue** (*unsigned int const*)
- **newHighValue** (*unsigned int const*)

property thisown

The membership flag

4.15 PySpin.ImageUtilityPolarization

class PySpin.ImageUtilityPolarization

Proxy of C++ Spinnaker::ImageUtilityPolarization class.

static CreateAolp(*srcImage*, *colorProcessingAlg*=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateAolp**(*srcImage*
• **destAolpImg** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destAolpImg**
- **colorProcessingAlg**

static CreateDolp(*srcImage*, *colorProcessingAlg*=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateDolp**(*srcImage*
• **destDolpImage** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destDolpImage**
- **colorProcessingAlg**

static CreateGlareReduced(*srcImage*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **CreateGlareReduced**(*srcImage*
• **destGlareReducedImage**)
- **srcImage**
- **destGlareReducedImage** (*Spinnaker::ImagePtr &*)

```
static CreateStokesS0(srcImage, colorProcessin-  

                     gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  

                     → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateStokesS0**(**srcImage**
- **destStokesS0Image** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destStokesS0Image**
- **colorProcessingAlg**

```
static CreateStokesS1(srcImage, colorProcessin-  

                     gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  

                     → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateStokesS1**(**srcImage**
- **destStokesS1Image** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destStokesS1Image**
- **colorProcessingAlg**

```
static CreateStokesS2(srcImage, colorProcessin-  

                     gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  

                     → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateStokesS2**(**srcImage**
- **destStokesS2Image** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destStokesS2Image**
- **colorProcessingAlg**

static **ExtractPolarQuadrant**(*srcImage*, *desiredQuadrant*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **desiredQuadrant** (*enum Spinnaker::PolarizationQuadrant const*)
- **ExtractPolarQuadrant**(**srcImage**
- **destQuadImage** (*Spinnaker::ImagePtr &*)
- **desiredQuadrant**)
- **srcImage**
- **destQuadImage**
- **desiredQuadrant**

property thisown

The membership flag

4.16 PySpin.ImageUtilityStereo

class **PySpin.ImageUtilityStereo**

Proxy of C++ Spinnaker::ImageUtilityStereo class.

static **Compute3DPointFromPixel**(*disparity*, *stereoCameraParameters*, *stereo3DPoint*) → bool

Parameters

- **disparity** (*uint16_t const*)
- **stereoCameraParameters** (*Spinnaker::StereoCameraParameters const &*)
- **stereo3DPoint** (*Spinnaker::Stereo3DPoint &*)

static **ComputeDistanceBetweenPoints**(*disparityImage*, *stereoParam*, *imagePixel1*, *ImagePixel2*, *distance*) → bool

Parameters

- **disparityImage** (*Spinnaker::ImagePtr const &*)
- **stereoParam** (*Spinnaker::StereoCameraParameters const &*)
- **imagePixel1** (*Spinnaker::ImagePixel const &*)
- **ImagePixel2** (*Spinnaker::ImagePixel const &*)
- **distance** (*float &*)

static **ComputeDistanceToPoint**(*disparityImage*, *stereoParam*, *imagePixel*, *distance*) → bool

Parameters

- **disparityImage** (*Spinnaker::ImagePtr const &*)
- **stereoParam** (*Spinnaker::StereoCameraParameters const &*)
- **imagePixel** (*Spinnaker::ImagePixel const &*)
- **distance** (*float &*)

static ComputePointCloud(*disparityImage*, *rectifiedImage*, *pointCloudParameters*,
stereoCameraParameters) → *PointCloud*

Parameters

- **disparityImage** (*Spinnaker::ImagePtr const &*)
- **rectifiedImage** (*Spinnaker::ImagePtr const &*)
- **pointCloudParameters** (*Spinnaker::PointCloudParameters const &*)
- **stereoCameraParameters** (*Spinnaker::StereoCameraParameters const &*)
- **ComputePointCloud**(**disparityImage**
- **rectifiedImage**
- **pointCloudParameters**
- **stereoCameraParameters**
- **pointCloud**)
- **disparityImage**
- **rectifiedImage**
- **pointCloudParameters**
- **stereoCameraParameters**
- **pointCloud** (*Spinnaker::PointCloud &*)

static CreateDepthImage(*disparityImage*, *stereoCameraParameters*, *invalidDepthVal*, *minDepthVal*,
maxDepthVal) → *ImagePtr*

Parameters

- **disparityImage** (*Spinnaker::ImagePtr const &*)
- **stereoCameraParameters** (*Spinnaker::StereoCameraParameters const &*)
- **invalidDepthVal** (*uint16_t const*)
- **minDepthVal** (*float &*)
- **maxDepthVal** (*float &*)
- **CreateDepthImage**(**disparityImage**
- **stereoCameraParameters**
- **invalidDepthVal**
- **depthImage** (*Spinnaker::ImagePtr &*)
- **minDepthVal**
- **maxDepthVal**)
- **disparityImage**
- **stereoCameraParameters**
- **invalidDepthVal**
- **depthImage**
- **minDepthVal**

- **maxDepthVal**

static FilterSpeckles(*disparityImage*, *maxSpeckleSize*, *speckleThreshold*, *disparityScaleFactor*, *invalidDataValue*) → *ImagePtr*

Parameters

- **disparityImage** (*Spinnaker::ImagePtr const &*)
- **maxSpeckleSize** (*int const*)
- **speckleThreshold** (*int const*)
- **disparityScaleFactor** (*float const*)
- **invalidDataValue** (*float const*)

static FilterSpecklesFromImage(*disparityImage*, *maxSpeckleSize*, *speckleThreshold*, *disparityScaleFactor*, *invalidDataValue*)

Parameters

- **disparityImage** (*Spinnaker::ImagePtr &*)
- **maxSpeckleSize** (*int const*)
- **speckleThreshold** (*int const*)
- **disparityScaleFactor** (*float const*)
- **invalidDataValue** (*float const*)

static IsStereoCamera(*pCamera*) → bool

Parameters

pCamera (*Spinnaker::CameraPtr*)

property maxDepthThresholdInMeter

property maxDepthThresholdInMm

property thisown

The membership flag

4.17 PySpin.IInterface

class **PySpin.IInterface**(*args, **kwargs)

Proxy of C++ Spinnaker::IInterface class.

GetCameras(*self*, *updateCameras=True*) → *CameraList*

Parameters

updateCameras (*bool*)

GetTLNodeMap(*self*) → *INodeMap*

IsCameraInUse(*self*) → bool

IsValid(*self*) → bool

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

evtHandlerToRegister (*Spinnaker::EventHandler* &)

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *requestAck=False*, *pResultSize=None*, *results=0*)

Parameters

- **deviceKey** (*unsigned int*)
- **groupKey** (*unsigned int*)
- **groupMask** (*unsigned int*)
- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)

property TLInterface

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler* &)

UpdateCameras(*self*) → *bool*

property thisown

The membership flag

4.18 PySpin.InterfaceList

class *PySpin.InterfaceList*(*args)

A list of the available interfaces on the system.

C++ includes: *InterfaceList.h*

Add(*self*, *iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

Append(*self*, *list*)

Parameters

list (*Spinnaker::InterfaceList const **)

Clear(*self*)

void Spinnaker::InterfaceList::Clear()

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.

GetByIndex(*self*, *index*) → *InterfacePtr*

Parameters

- **index** (*The index at which to retrieve the Interface object*)
- **const** (*InterfacePtr Spinnaker::InterfaceList::GetByIndex(int index)*)
- **"index"**. (*Returns a pointer to an Interface object at the*)
- **Parameters**
- -----
- **index**
- **object.** (*A pointer to an Interface*)

GetByInterfaceID(*self*, *interfaceID*) → *InterfacePtr*

Parameters

interfaceID (*std::string*)

GetSize(*self*) → unsigned int

int Spinnaker::InterfaceList::GetSize() const

Returns the size of the interface list. The size is the number of Interface objects stored in the list.

An integer that represents the list size.

Remove(*self*, *iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

4.19 PySpin.InterfacePtr

class PySpin.**InterfacePtr**(*args)

A reference tracked pointer to the interface object.

C++ includes: InterfacePtr.h

property thisown

The membership flag

4.20 PySpin.PointCloud

class PySpin.**PointCloud**

Proxy of C++ Spinnaker::PointCloud class.

AddPoint(*self*, *point*)

Parameters

point (*Spinnaker::Stereo3DPoint const*)

GetNumPoints(*self*) → size_t

GetPoint(*self*, *index*) → *Stereo3DPoint*

Parameters

index (*unsigned int const*)

GetPointCloudData(*self*) → *Spinnaker::IPointCloud::PointCloudData **

LoadPointCloudFromPly(*self*, *filename*)

Parameters

filename (*std::string const &*)

PrintPoints(*self*, *numPointsToPrint*)

Parameters

numPointsToPrint (*unsigned int*)

SavePointCloudAsPly(*self*, *arg0*)

Parameters

arg0 (*std::string const &*)

property thisown

The membership flag

4.21 PySpin.SpinnakerException

class *PySpin.SpinnakerException*

Exception class for the PySpin module. This class has these attributes: *message*, *errorcode*, *fullmessage*

errorcode = 0

fullmessage = ''

message = ''

4.22 PySpin.SpinVideo

class *PySpin.SpinVideo*

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

C++ includes: *SpinVideo.h*

Append(*self*, *pImage*)

Parameters

- **pImage** (*The image to append.*)
- **virtual**
- **pImage** (*void Spinnaker::Video::SpinVideo::Append(ImagePtr)*)
- **file.** (*Append an image to the AVI/MP4*)
- **Parameters**
- -----
- **pImage**

Close(*self*)

virtual void Spinnaker::Video::SpinVideo::Close()

Close the AVI/MP4 file.

See: Open()

Open(*self*, *pFileName*, *pOption*)**Parameters**

- **pFileName** (*The filename of the MP4 file.*)
- **pOption** (*H264 options to apply to the MP4 file.*)
- **Open**(*self*
- **pFileName**
- **pOption**)
- **pFileName**
- **pOption**
- **Open**(*self*
- **pFileName**
- **pOption**)
- **pFileName**
- **pOption**
- **void** (*virtual*)
- ***pFileName** (*Spinnaker::Video::SpinVideo::Open(const char) –*
- **Video::H264Option**
- **&pOption**)
- **The** (*Open an H264 MP4 file in preparation for writing Images to disk.*
)
- **automatically** (*size of MP4 files is limited to 2GB. The filenames are*)
- **specified.** (*generated using the filename*)
- **Parameters**
- **-----**
- **pFileName**
- **pOption**
- **See** ([H264Option](#))
- **See**

SetMaximumFileSize(*self*, *size*)**Parameters**

size (*unsigned int*)

property thisown

The membership flag

4.23 PySpin.System

class PySpin.System(*args, **kwargs)

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

C++ includes: System.h

GetCameras(self, updateInterfaces=True, updateCameras=True) → *CameraList*

Parameters

- **updateInterfaces** (*Determines whether or not updateInterfaceList() is*
- **updateCameras** (*Determines whether or not UpdateCameras() is called*)
- **CameraList**
- **updateInterfaces=true** (*Spinnaker::System::GetCameras(bool)*
- **bool**
- **updateCameras=true**)
- **call** (*Returns a list of cameras that are available on the system. This*)
- **interfaces.** (*returns both GigE Vision and Usb3 Vision cameras from all*)
- **It** (*The camera list object will reference count the cameras it returns.*)
- **before** (*is important that the camera list is destroyed or is cleared*)
- **system->** (*calling system-> ReleaseInstance() or else the call to*
- **a** (*ReleaseInstance() will result in an error message thrown that*
- **held.** (*reference to the camera is still*)
- **See** (*CameraList::Clear()*)
- **See**
- **Parameters**
- **-----**
- **updateInterfaces**
- **system** (*before getting cameras from available interfaces on the*)
- **updateCameras**
- **system**
- **cameras.** (*An CameraList object that contains a list of all*)

static GetInstance() → *SystemPtr*

GetInterfaces(*self*, *updateInterface=True*) → *InterfaceList*

Parameters

- **updateInterface** (*Determines whether or not UpdateInterfaceList() is*)
- **Spinnaker::System::GetInterfaces**(*bool (InterfaceList)*)
- **updateInterface=true**)
- **call** (*Returns a list of interfaces available on the system. This*)
- **interfaces.** (*An InterfaceList object that contains a list of all*)
- **Parameters**
- **-----**
- **updateInterface**
- **interfaces** (*called before getting available*)
- **interfaces.**

GetLibraryVersion(*self*) → *LibraryVersion*

GetLoggingEventPriorityLevel(*self*) → Spinnaker::SpinnakerLogLevel

SpinnakerLogLevel Spinnaker::System::GetLoggingEventPriorityLevel()

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: SpinnakerLogLevel

Level The threshold level

GetTLNodeMap(*self*) → *INodeMap*

IsInUse(*self*) → bool

bool Spinnaker::System::IsInUse()

Checks if the system is in use by any interface or camera objects.

Returns true if the system is in use and false otherwise.

RegisterEventHandler(*self*, *evtHandlerToRegister*, *updateInterface=False*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::EventHandler &*)
- **updateInterface** (*bool*)

RegisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler* &)

ReleaseInstance(*self*)

void *Spinnaker::System::ReleaseInstance*()

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_RESOURCE_IN_USE*.

See: Error

See: *GetInstance*()

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *requestAck=False*, *pResultSize=None*, *results=0*)

Parameters

- **deviceKey** (*Spinnaker::System::SendActionCommand(unsigned int)*)
- **groupKey** (*int*)
- **groupMask** (*unsigned int*)
- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)
- **void**
- **deviceKey**
- **unsigned**
- **groupKey**
- **groupMask**
- **actionTime=0** (*unsigned long long*)

:param : :param unsigned int *pResultSize=0: :param ActionCommandResult results[]=NULL): :param Broadcast an Action Command to all devices on system: :param Parameters: :param ———: :param deviceKey: :type deviceKey: The Action Command's device key :param groupKey: :type groupKey: The Action Command's group key :param groupMask: :type groupMask: The Action Command's group mask :param actionTime: :type actionTime: (Optional) Time when to assert a future action. Zero :param means immediate action.: :param pResultSize: :type pResultSize: (Optional) The number of results in the results array. :param The value passed should be equal to the expected number of devices: :param that acknowledge the command. Returns the number of received results.: :param results: :type results: (Optional) An Array with *pResultSize elements to hold the :param action command result status. The buffer is filled starting from index: :param 0. If received results are less than expected number of devices that: :param acknowledge the command: :param remaining results are not changed. If: :param received results are more than expected number of devices that: :param acknowledge the command: :param extra results are ignored and not appended to: :param array. This parameter is ignored if pResultSize is 0. Thus this: :param parameter can be NULL if pResultSize is 0 or NULL.:

SetLoggingEventPriorityLevel(*self*, *level*)

Parameters

- **level** (*enum Spinnaker::SpinnakerLogLevel*)
- **void**
- **Spinnaker::System::SetLoggingEventPriorityLevel**(*SpinnakerLogLevel* **level**)
- **events** (*Sets a threshold priority level for logging event. Logging*)
- **callbacks.** (*below such level will not trigger*)
- **logging** (*Spinnaker uses five levels of*)
- **intervention.** (*Warning - failures that are recoverable without user*)
- **intervention.**
- **removal** (*Notice - information about events such as camera arrival and*)

:param : :param initialization and deinitialization: :param starting and stopping image: :param acquisition:
:param and feature modification.: :param Info - information about recurring events that are generated reg-
ularly: :param such as information on individual images.: :param Debug - information that can be used to
troubleshoot the system.: :param See: :type See: *SpinnakerLogLevel* :param Parameters: :param ———:
:param level: :type level: The threshold level

UnregisterAllLoggingEventHandlers(*self*)

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

UnregisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler &*)

UpdateCameras(*self*, *updateInterfaces=True*) → bool

Parameters

- **updateInterfaces** (*bool*)
- **bool**
- **updateInterfaces=true**) (*Spinnaker::System::UpdateCameras*(*bool*)
- **that** (*Updates the list of cameras on the system. Note*)
- **each** (*System::GetCameras()* internally calls *UpdateCameras()* for)
- **the** (*interface it enumerates. If the list changed between this call and*)
- **true** (*last time UpdateCameras was called then the return value will be*)

:param : :param otherwise it is false.: :param See: :type See: GetCameras() :param Parameters: :param
———: :param updateInterfaces: :type updateInterfaces: Determines whether or not UpdateInter-
faceList() is :param called before updating cameras for available interfaces on the system: :param True
if cameras changed on interface and false otherwise.:

UpdateInterfaceList(*self*)

property thisown

The membership flag

4.24 PySpin.SystemPtr

class PySpin.**SystemPtr**(*args)

A reference tracked pointer to a system object.

C++ includes: SystemPtr.h

property thisown

The membership flag

QUICKSPIN CLASSES

- *PySpin.TransportLayerDevice*
- *PySpin.TransportLayerInterface*
- *PySpin.TransportLayerStream*

5.1 PySpin.TransportLayerDevice

class `PySpin.TransportLayerDevice`(*nodeMapTLDevice*)

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

C++ includes: `TransportLayerDevice.h`

property `DeviceAccessStatus`

property `DeviceBootloaderVersion`

property `DeviceCurrentSpeed`

property `DeviceDisplayName`

property `DeviceDriverVersion`

property `DeviceEndiannessMechanism`

property `DeviceID`

property `DeviceInstanceId`

property `DeviceIsUpdater`

property `DeviceLinkSpeed`

property `DeviceLocation`

property `DeviceModelName`

property `DeviceMulticastMonitorMode`

property `DevicePortId`

property `DeviceReset`

property DeviceSerialNumber
property DeviceType
property DeviceU3VProtocol
property DeviceUserID
property DeviceVendorName
property DeviceVersion
property GUIXMLLocation
property GUIXMLPath
property GenICamXMLLocation
property GenICamXMLPath
property GevCCP
property GevDeviceAutoForceIP
property GevDeviceDiscoverMaximumPacketSize
property GevDeviceForceGateway
property GevDeviceForceIP
property GevDeviceForceIPAddress
property GevDeviceForceSubnetMask
property GevDeviceGateway
property GevDeviceIPAddress
property GevDeviceIsWrongSubnet
property GevDeviceMACAddress
property GevDeviceMaximumPacketSize
property GevDeviceMaximumRetryCount
property GevDeviceModeIsBigEndian
property GevDevicePort
property GevDeviceReadAndWriteTimeout
property GevDeviceSubnetMask
property GevVersionMajor
property GevVersionMinor
property StreamID
property StreamSelector
property thisown
 The membership flag

5.2 PySpin.TransportLayerInterface

class `PySpin.TransportLayerInterface`(*nodeMapTLDevice*)

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

C++ includes: `TransportLayerInterface.h`

property `ActionCommand`

property `DeviceAccessStatus`

property `DeviceCount`

property `DeviceID`

property `DeviceModelName`

property `DeviceSelector`

property `DeviceSerialNumber`

property `DeviceUnlock`

property `DeviceUpdateList`

property `DeviceVendorName`

property `FLIRFilterDriverStatus`

property `GevActionAckRequired`

property `GevActionDeviceKey`

property `GevActionGroupKey`

property `GevActionGroupMask`

property `GevActionTime`

property `GevDeviceAutoForceIP`

property `GevDeviceDisableDiscovery`

property `GevDeviceDiscoveryEnabled`

property `GevDeviceEnableDiscovery`

property `GevDeviceForceGateway`

property `GevDeviceForceIP`

property `GevDeviceForceIPAddress`

property `GevDeviceForceSubnetMask`

property `GevDeviceGateway`

property `GevDeviceIPAddress`

property `GevDeviceMACAddress`

property `GevDeviceSubnetMask`
property `GevInterfaceGateway`
property `GevInterfaceGatewaySelector`
property `GevInterfaceIsIPConflict`
property `GevInterfaceMACAddress`
property `GevInterfaceMTU`
property `GevInterfaceReceiveLinkSpeed`
property `GevInterfaceSubnetIPAddress`
property `GevInterfaceSubnetMask`
property `GevInterfaceSubnetSelector`
property `GevInterfaceTransmitLinkSpeed`
property `HostAdapterDriverVersion`
property `HostAdapterName`
property `HostAdapterVendor`
property `IncompatibleDeviceCount`
property `IncompatibleDeviceID`
property `IncompatibleDeviceModelName`
property `IncompatibleDeviceSelector`
property `IncompatibleDeviceVendorName`
property `IncompatibleGevDeviceIPAddress`
property `IncompatibleGevDeviceMACAddress`
property `IncompatibleGevDeviceSubnetMask`
property `InterfaceDisplayName`
property `InterfaceID`
property `InterfaceType`
property `POEStatus`
property `TeledyneGigeVisionFilterDriverStatus`
property `thisown`

The membership flag

5.3 PySpin.TransportLayerStream

class `PySpin.TransportLayerStream`(*nodeMapTLDevice*)

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

C++ includes: `TransportLayerStream.h`

property `StreamAnnounceBufferMinimum`

property `StreamAnnouncedBufferCount`

property `StreamBlockTransferSize`

property `StreamBlocksProcessingTimeLast`

property `StreamBlocksProcessingTimeMax`

property `StreamBlocksProcessingTimeMin`

property `StreamBlocksReceptionTimeLast`

property `StreamBlocksReceptionTimeMax`

property `StreamBlocksReceptionTimeMin`

property `StreamBufferAlignment`

property `StreamBufferCountManual`

property `StreamBufferCountMax`

property `StreamBufferCountMode`

property `StreamBufferCountResult`

property `StreamBufferHandlingMode`

property `StreamCRCCheckEnable`

property `StreamChunkCountMaximum`

property `StreamDeliveredFrameCount`

property `StreamDroppedFrameCount`

property `StreamID`

property `StreamIncompleteFrameCount`

property `StreamInputBufferCount`

property `StreamIsGrabbing`

property `StreamLostFrameCount`

property `StreamMissedPacketCount`

property `StreamMode`

property `StreamOutputBufferCount`

property `StreamPacketResendEnable`
property `StreamPacketResendMaxRequests`
property `StreamPacketResendReceivedPacketCount`
property `StreamPacketResendRequestCount`
property `StreamPacketResendRequestTimeoutCount`
property `StreamPacketResendRequestedPacketCount`
property `StreamPacketResendTimeout`
property `StreamPacketsDuplicatedCount`
property `StreamPacketsNotYetAvailableCount`
property `StreamPacketsPerFrameCount`
property `StreamPacketsTemporarilyUnavailableCount`
property `StreamPacketsTimeoutCount`
property `StreamPacketsUnavailableCount`
property `StreamReceivedFrameCount`
property `StreamReceivedPacketCount`
property `StreamStartedFrameCount`
property `StreamType`
property `thisown`
 The membership flag

PYSPIN MODULE

class PySpin.AVIOption

Bases: object

Options for saving AVI files.

C++ includes: SpinVideoDefs.h

property frameRate

property height

property reserved

property thisown

The membership flag

property width

class PySpin.ActionCommandResult

Bases: object

Action Command Result

C++ includes: SpinnakerDefs.h

property DeviceAddress

property Status

property thisown

The membership flag

class PySpin.BMPOption

Bases: object

Options for saving Bitmap image.

C++ includes: SpinnakerDefs.h

property indexedColor_8bit

property reserved

property thisown

The membership flag

class PySpin.**BooleanNode**(*args, **kwargs)

Bases: *IBoolean*, *ValueNode*

Interface for string properties.

C++ includes: BooleanNode.h

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → bool

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **bool**
- **Verify=false** (*Spinnaker::GenApi::BooleanNode::GetValue(bool)*)
- **bool**
- **const** (*IgnoreCache=false*)
- **value** (*Get node*)
- **Parameters**
- -----
- **Verify**
- **checked.** (*is always*)
- **IgnoreCache**
- **false).**
- **read.** (*The value*)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::BooleanNode::SetReference(INode** (*virtual void*)
- ***pBase**)
- **Value** (*overload SetReference for*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*The value to set.*)
- **Verify** (*Enables AccessMode and Range verification (default = true).*)
- **void**
- **Value**
- **Verify=true)** (*bool*)
- **value** (*Set node*)
- **Parameters**

- -----
- **Value**
- **Verify**

property thisown

The membership flag

class PySpin.CBasePtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

IsValid(self) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

class PySpin.CBooleanPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(self, hCallback) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(self, ValueStr, Verify=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

GetAlias(self) → *Inode*

GetCachingMode(self) → Spinnaker::GenApi::ECachingMode

GetCastAlias(self) → *Inode*

GetChildren(self, LinkType=ctReadingChildren)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(self) → *gcstring*

GetDeviceName(self) → *gcstring*

GetDisplayName(self) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *bool*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (enum *Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (enum *Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*bool*)
- **Verify** (*bool*)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class *PySpin.CCMSettings*

Bases: *object*

Proxy of C++ *Spinnaker::CCMSettings* class.

property Application

property ColorSpace

property ColorTemperature

property CustomCCMCode

property Sensor

property Type

property thisown

The membership flag

class PySpin.CCategoryPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(self, hCallback) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(self, ValueStr, Verify=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)

- **Verify** (bool)

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

GetAlias(self) → *INode*

GetCachingMode(self) → Spinnaker::GenApi::ECachingMode

GetCastAlias(self) → *INode*

GetChildren(self, LinkType=ctReadingChildren)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(self) → *gcstring*

GetDeviceName(self) → *gcstring*

GetDisplayName(self) → *gcstring*

GetDocuURL(self) → *gcstring*

GetEventID(self) → *gcstring*

GetFeatures(self)

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → Spinnaker::GenApi::EYesNo

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

ToString(*self*, *Verify*=False, *IgnoreCache*=False) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.CCommandPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

Execute(*self*, *Verify*=True)

Parameters

Verify (bool)

FromString(*self*, *ValueStr*, *Verify*=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetName(*self*, *FullQualified*=*False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → Spinnaker::GenApi::ENamespace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → int64_t

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (Spinnaker::GenICam::gcstring const &)
- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **AttributeStr** (Spinnaker::GenICam::gcstring &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → *bool*

IsDeprecated(*self*) → *bool*

IsDone(*self*, *Verify=True*) → *bool*

Parameters

Verify (*bool*)

IsFeature(*self*) → *bool*

IsSelector(*self*) → *bool*

IsStreamable(*self*) → *bool*

IsValid(*self*) → *bool*

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → *bool*

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class PySpin.**CDeviceInfoPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetDeviceVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t* &)

GetGenApiVersion(*self*, *Version*, *Build*)

Parameters

- **Version** (*Spinnaker::GenICam::Version_t* &)
- **Build** (*uint16_t* &)

GetModelName(*self*) → *gcstring*

GetProductGuid(*self*) → *gcstring*

GetSchemaVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t* &)

GetStandardNameSpace(*self*) → *gcstring*

GetToolTip(*self*) → *gcstring*

GetVendorName(*self*) → *gcstring*

GetVersionGuid(*self*) → *gcstring*

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

class PySpin.**CEnumEntryPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (*Spinnaker::GenApi::CallbackHandleType*)

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **Verify** (*bool*)

GetAccessMode(*self*) → *Spinnaker::GenApi::EAccessMode*

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → *Spinnaker::GenApi::ECachingMode*

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType=ctReadingChildren*)

Parameters

LinkType (*enum Spinnaker::GenApi::ELinkType*)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t &*)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetNumericValue(*self*) → double

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t &*)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const &*)
- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **AttributeStr** (*Spinnaker::GenICam::gcstring &*)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t &*)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t &*)

GetSymbolic(*self*) → *gcstring*

GetToolTip(*self*) → *gcstring*

GetValue(*self*) → int64_t

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (enum *Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (enum *Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsSelfClearing(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

ToString(*self*, *Verify*=False, *IgnoreCache*=False) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.CEnumerationPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(*self*, *ValueStr*, *Verify*=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=ctReadingChildren)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetCurrentEntry(*self*, *Verify*=False, *IgnoreCache*=False) → *IEnumEntry*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEntries(*self*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

IntValue (*int64_t* *const*)

GetEntryByName(*self*, *Symbolic*) → *IEnumEntry*

Parameters

Symbolic (*Spinnaker::GenICam::gcstring const &*)

GetEventID(*self*) → *gcstring*

GetIntValue(*self*, *Verify=False*, *IgnoreCache=False*) → *int64_t*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t &*)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t &*)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const &*)
- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **AttributeStr** (*Spinnaker::GenICam::gcstring &*)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSymbolics(*self*, *Symbolics*)

Parameters

Symbolics (*Spinnaker::GenApi::StringList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → *bool*

IsDeprecated(*self*) → *bool*

IsFeature(*self*) → *bool*

IsSelector(*self*) → *bool*

IsStreamable(*self*) → *bool*

IsValid(*self*) → *bool*

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → *bool*

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetIntValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*int64_t*)

- **Verify** (*bool*)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class **PySpin.CFeatureBag**

Bases: *IPersistScript*

Bag holding streamable features of a nodetree.

C++ includes: Persistence.h

GetFeatureBagHandle(*self*) → void *

void* Spinnaker::GenApi::CFeatureBag::GetFeatureBagHandle()

LoadFromBag(*self*, *pNodeMap*, *Verify=True*, *pErrorList=None*) → bool

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap **)
- **Verify** (*bool*)
- **pErrorList** (*Spinnaker::GenICam::gcstring_vector **)
- ***pNodeMap** (*bool Spinnaker::GenApi::CFeatureBag::LoadFromBag(INodeMap)*)

:param : :param bool Verify=true: :param GenICam::gcstring_vector *pErrorList=NULL): :param Loads the features from the bag to the node tree: :param Parameters: :param ———: :param pNodeMap: :type pNodeMap: The node map :param Verify: :type Verify: If true, all streamable features are read back :param pErrorList: :type pErrorList: If an error occurs during loading the error message is :param stored in the list and the loading continues: :param For Verify=true the list of names in the feature bag is replayed: :param again. If a node is a selector it's value is set to the value from the: :param feature bag If not the value is read from the camera and compared with: :param the value from the feature bag.:

PersistFeature(*self*, *item*)

Parameters

- **item** (*Spinnaker::GenApi::IValue &*)
- **Spinnaker::GenApi::CFeatureBag::PersistFeature(IValue (virtual void)**
- **&item)**
- **feature** (*Stores a*)

SetInfo(*self*, *Info*)

Parameters

- **Info** (*Spinnaker::GenICam::gcstring &*)

- **Spinnaker::GenApi::CFeatureBag::SetInfo**(GenICam::gcstring (virtual void)
- **&Info**)
- **map** (sets information about the node)

StoreToBag(self, pNodeMap, MaxNumPersistSkriptEntries=-1) → int64_t

Parameters

- **pNodeMap** (Spinnaker::GenApi::INodeMap *)
- **MaxNumPersistSkriptEntries** (int const)
- ***pNodeMap** (int64_t Spinnaker::GenApi::CFeatureBag::StoreToBag(INodeMap) —

:param : :param const int MaxNumPersistSkriptEntries=-1): :param Stores the streamable nodes to this feature bag.: :param Parameters: :param —: :param pNodeMap: :type pNodeMap: The node map to persist :param MaxNumPersistSkriptEntries: :type MaxNumPersistSkriptEntries: The max number of entries in the :param container; -1 means unlimited: :param number of entries in the bag:

property thisown

The membership flag

class PySpin.CFloatPtr(*args)

Bases: _SWIG_CFltPtr

SmartPointer for IFloat interface pointer

C++ includes: Pointer.h

GetEnumAlias(self) → *IEnumeration*

IEnumeration* Spinnaker::GenApi::CFloatPtr::GetEnumAlias()

gets the interface of an enum alias node.

GetIntAlias(self) → *Integer*

IInteger* Spinnaker::GenApi::CFloatPtr::GetIntAlias()

gets the interface of an integer alias node.

property thisown

The membership flag

class PySpin.CIntegerPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(self, hCallback) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(self, ValueStr, Verify=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetInc(*self*) → *int64_t*

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

GetListOfValidValues(*self*, *bounded*=*True*) → *int64_autovector_t*

Parameters

bounded (*bool*)

GetLockNodes(*self*)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetMax(*self*) → *int64_t*

GetMin(*self*) → *int64_t*

GetName(*self*, *FullQualified*=*False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → Spinnaker::GenApi::ENamespace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring* const &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetRepresentation(*self*) → *Spinnaker::GenApi::ERepresentation*

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetUnit(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *int64_t*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeMax(*self*, *Value*)

Parameters

Value (*int64_t*)

ImposeMin(*self*, *Value*)

Parameters

Value (*int64_t*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (int64_t)
- **Verify** (bool)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.CNodeMapDynPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

ClearAllNodes(*self*)

Connect(*self*, *pPort*, *PortName*) → bool

Parameters

- **pPort** (IPort *)
- **PortName** (Spinnaker::GenICam::gcstring const &)
- **Connect**(self
- **bool** (pPort) ->)
- **pPort**

ExtractIndependentSubtree(*self*, *XMLData*, *InjectXMLData*, *SubTreeRootNodeName*, *ExtractedSubtree*)

Parameters

- **XMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **SubTreeRootNodeName** (*Spinnaker::GenICam::gcstring const &*)
- **ExtractedSubtree** (*Spinnaker::GenICam::gcstring &*)

GetDeviceName(*self*) → *gcstring*

GetNode(*self*, *Name*) → *INode*

Parameters

Name (*Spinnaker::GenICam::gcstring const &*)

GetNodes(*self*)

GetNumNodes(*self*) → *uint64_t*

GetSupportedSchemaVersions(*self*)

InvalidateNodes(*self*)

IsValid(*self*) → *bool*

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* *const throw ()* true if the pointer is valid

LoadXMLFromFile(*self*, *FileName*)

Parameters

FileName (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromFileInject(*self*, *TargetFileName*, *InjectFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectFileName** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromString(*self*, *XMLData*)

Parameters

XMLData (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromStringInject(*self*, *TargetXMLData*, *InjectXMLData*)

Parameters

- **TargetXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromZIPData(*self*, *zipData*, *zipSize*)

Parameters

- **zipData** (*void const **)
- **zipSize** (*size_t*)

LoadXMLFromZIPFile(*self*, *ZipFileName*)

Parameters

ZipFileName (*Spinnaker::GenICam::gcstring const &*)

MergeXMLFiles(*self*, *TargetFileName*, *InjectedFileName*, *OutputFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectedFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)

Poll(*self*, *ElapsedTime*)

Parameters

ElapsedTime (*int64_t*)

PreprocessXMLFromFile(*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*,
XMLValidation=xvDefault)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

PreprocessXMLFromZIPFile(*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*,
XMLValidation=xvDefault)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

property thisown

The membership flag

class **PySpin.CNodeMapPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

Connect(*self*, *pPort*, *PortName*) → bool

Parameters

- **pPort** (*IPort **)
- **PortName** (*Spinnaker::GenICam::gcstring const &*)
- **Connect**(*self*
- **bool** (*pPort*) →)

- **pPort**

GetDeviceName(*self*) → *gcstring*

GetNode(*self*, *Name*) → *INode*

Parameters

Name (*Spinnaker::GenICam::gcstring* const &)

GetNodes(*self*)

GetNumNodes(*self*) → *uint64_t*

InvalidateNodes(*self*)

IsValid(*self*) → *bool*

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

Poll(*self*, *ElapsedTime*)

Parameters

ElapsedTime (*int64_t*)

property thisown

The membership flag

class PySpin.**CNodePtr**(*args)

Bases: *object*

Encapsulates a GenApi pointer dealing with the *dynamic_cast* automatically.

C++ includes: *Pointer.h*

DeregisterCallback(*self*, *hCallback*) → *bool*

Parameters

hCallback (*Spinnaker::GenApi::CallbackHandleType*)

GetAccessMode(*self*) → *Spinnaker::GenApi::EAccessMode*

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → *Spinnaker::GenApi::ECachingMode*

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (*enum Spinnaker::GenApi::ELinkType*)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)

- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

property thisown

The membership flag

class PySpin.**CRegisterPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

Get(*self*, *pBuffer*, *Verify=False*, *IgnoreCache=False*)

Parameters

- **pBuffer** (uint8_t *)
- **Verify** (bool)
- **IgnoreCache** (bool)

Gets a NumPy array representing the contents of the register, as 8-bit unsigned ints.

6.1 Parameters:

pBuffer: 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)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAddress(*self*) → int64_t

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=ctReadingChildren)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLength(*self*) → int64_t

GetLockNodes(*self*)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetName(*self*, *FullQualified*=False) → *gcstring*

Parameters

FullQualified (bool)

GetNamespace(*self*) → Spinnaker::GenApi::ENamespace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → int64_t

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring* const &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

Set(*self*, *pBuffer*, *Verify=True*)

Parameters

- **pBuffer** (*uint8_t const **)
- **Verify** (*bool*)

Set the register's contents with the contents (as 8-bit unsigned ints) of the given array.

6.2 Parameters:

pBuffer: The NumPy array containing the data to set

Verify: Enables AccessMode and Range verification (default = true)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class **PySpin.CSelectorPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t &*)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t &*)

IsSelector(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

class PySpin.CSelectorSet(*args, **kwargs)

Bases: *Node*

The set of selectors selecting a given node

C++ includes: SelectorSet.h

GetSelectorList(self, Incremental=False)

Parameters

- **Incremental** (*bool*)
- **void** (*virtual*)
- **Spinnaker::GenApi::CSelectorSet::GetSelectorList**(FeatureList_t
- **&SelectorList**
- **Incremental=false**) (*bool*)

IsEmpty(self) → *bool*

bool Spinnaker::GenApi::CSelectorSet::IsEmpty()

returns true if no selectors are present

Restore(self)

virtual void Spinnaker::GenApi::CSelectorSet::Restore()

SetFirst(self) → *bool*

virtual bool Spinnaker::GenApi::CSelectorSet::SetFirst()

SetNext(self, Tick=True) → *bool*

Parameters

- **Tick** (*bool*)
- **Tick=true**) (*virtual bool* Spinnaker::GenApi::CSelectorSet::SetNext(*bool*)

ToString(self) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::CSelectorSet::ToString()

property thisown

The membership flag

class PySpin.CStringPtr(*args)

Bases: *object*

Encapsulates a GenApi pointer dealing with the `dynamic_cast` automatically.

C++ includes: Pointer.h

DeregisterCallback(self, hCallback) → *bool*

Parameters

hCallback (*Spinnaker::GenApi::CallbackHandleType*)

FromString(self, ValueStr, Verify=True)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **Verify** (*bool*)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetMaxLength(*self*) → int64_t

GetName(*self*, *FullQualified*=*False*) → *gcstring*

Parameters

FullQualified (bool)

GetNameSpace(*self*) → Spinnaker::GenApi::ENameSpace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → int64_t

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (Spinnaker::GenICam::gcstring const &)
- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **AttributeStr** (Spinnaker::GenICam::gcstring &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → *bool*

IsDeprecated(*self*) → *bool*

IsFeature(*self*) → *bool*

IsSelector(*self*) → *bool*

IsStreamable(*self*) → *bool*

IsValid(*self*) → *bool*

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → *bool*

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*Spinnaker::GenICam::gcstring const &*)
- **Verify** (*bool*)

ToString(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class **PySpin.CValuePtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (*Spinnaker::GenApi::CallbackHandleType*)

FromString(*self*, *ValueStr*, *Verify*=*True*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **Verify** (*bool*)

GetAccessMode(*self*) → *Spinnaker::GenApi::EAccessMode*

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → *Spinnaker::GenApi::ECachingMode*

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (*enum Spinnaker::GenApi::ELinkType*)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → Spinnaker::GenApi::EYesNo

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

ToString(*self*, *Verify*=False, *IgnoreCache*=False) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.Camera(*args, **kwargs)

Bases: *CameraBase*

The camera object class.

C++ includes: Camera.h

property AasRoiEnable

property AasRoiHeight

property AasRoiOffsetX

property AasRoiOffsetY

property AasRoiWidth

property AcquisitionAbort

property AcquisitionArm

property AcquisitionBurstFrameCount

property AcquisitionFrameCount

property AcquisitionFrameRate
property AcquisitionFrameRateEnable
property AcquisitionFrameRatePersistence
property AcquisitionLineRate
property AcquisitionMode
property AcquisitionResultingFrameRate
property AcquisitionStart
property AcquisitionStatus
property AcquisitionStatusSelector
property AcquisitionStop
property AcquisitionTransferFrameRate
property ActionDeviceKey
property ActionGroupKey
property ActionGroupMask
property ActionQueueEmpty
property ActionQueueSize
property ActionSelector
property ActionSignalSize
property ActionUnconditionalMode
property AdaptiveCompressionEnable
property AdcBitDepth
property AutoAlgorithmSelector
property AutoExposureControlLoopDamping
property AutoExposureControlPriority
property AutoExposureEVCompensation
property AutoExposureExposureTimeLowerLimit
property AutoExposureExposureTimeUpperLimit
property AutoExposureGainLowerLimit
property AutoExposureGainUpperLimit
property AutoExposureGreyValueLowerLimit
property AutoExposureGreyValueUpperLimit

property AutoExposureLightingMode
property AutoExposureMeteringMode
property AutoExposureTargetGreyValue
property AutoExposureTargetGreyValueAuto
property BalanceRatio
property BalanceRatioSelector
property BalanceWhiteAuto
property BalanceWhiteAutoDamping
property BalanceWhiteAutoLowerLimit
property BalanceWhiteAutoProfile
property BalanceWhiteAutoUpperLimit
property BinningHorizontal
property BinningHorizontalMode
property BinningSelector
property BinningVertical
property BinningVerticalMode
property BlackLevel
property BlackLevelAuto
property BlackLevelAutoBalance
property BlackLevelClampingEnable
property BlackLevelRaw
property BlackLevelSelector
property BsiFlatFieldCorrectionAuto
property BsiFlatFieldCorrectionAutoDamping
property BsiFlatFieldCorrectionEnable
property BsiFlatFieldCorrectionGain
property BsiFlatFieldCorrectionGainSelector
property BufferedBurstFrameCountMax
property BufferedBurstMode
property ChunkBlackLevel
property ChunkBlackLevelSelector

property ChunkCRC

property ChunkCompressionMode

property ChunkCompressionRatio

property ChunkCounterSelector

property ChunkCounterValue

property ChunkCurrentDatarate

property ChunkEnable

property ChunkEncoderSelector

property ChunkEncoderStatus

property ChunkEncoderValue

property ChunkExposureEndLineStatusAll

property ChunkExposureTime

property ChunkExposureTimeSelector

property ChunkFrameID

property ChunkGain

property ChunkGainSelector

property ChunkHeight

property ChunkImage

property ChunkImageComponent

property ChunkInferenceBoundingBoxResult

property ChunkInferenceConfidence

property ChunkInferenceFrameId

property ChunkInferenceResult

property ChunkLinePitch

property ChunkLineStatusAll

property ChunkModeActive

property ChunkOffsetX

property ChunkOffsetY

property ChunkPartSelector

property ChunkPixelDynamicRangeMax

property ChunkPixelDynamicRangeMin

property ChunkPixelFormat
property ChunkRegionID
property ChunkScan3dAxisMax
property ChunkScan3dAxisMin
property ChunkScan3dCoordinateOffset
property ChunkScan3dCoordinateReferenceSelector
property ChunkScan3dCoordinateReferenceValue
property ChunkScan3dCoordinateScale
property ChunkScan3dCoordinateSelector
property ChunkScan3dCoordinateSystem
property ChunkScan3dCoordinateSystemReference
property ChunkScan3dCoordinateTransformSelector
property ChunkScan3dDistanceUnit
property ChunkScan3dInvalidDataFlag
property ChunkScan3dInvalidDataValue
property ChunkScan3dOutputMode
property ChunkScan3dTransformValue
property ChunkScanLineSelector
property ChunkSelector
property ChunkSequencerSetActive
property ChunkSerialData
property ChunkSerialDataLength
property ChunkSerialReceiveOverflow
property ChunkSourceID
property ChunkStreamChannelID
property ChunkTimerSelector
property ChunkTimerValue
property ChunkTimestamp
property ChunkTimestampLatchValue
property ChunkTransferBlockID
property ChunkTransferQueueCurrentBlockCount

property ChunkTransferStreamID
property ChunkWidth
property ClConfiguration
property ClTimeSlotsCount
property ColorTransformationEnable
property ColorTransformationSelector
property ColorTransformationValue
property ColorTransformationValueSelector
property ComponentActiveCount
property ComponentDestination
property ComponentEnable
property ComponentSelector
property CompressedFrameDropCount
property CompressionSaturationPriority
property ControlPacketsReservedBandwidth
property CounterDelay
property CounterDuration
property CounterEventActivation
property CounterEventSource
property CounterReset
property CounterResetActivation
property CounterResetSource
property CounterSelector
property CounterStatus
property CounterTriggerActivation
property CounterTriggerSource
property CounterValue
property CounterValueAtReset
property CxpConnectionSelector
property CxpConnectionTestErrorCount
property CxpConnectionTestMode

property CxpConnectionTestPacketCount
property CxpLinkConfiguration
property CxpLinkConfigurationPreferred
property CxpLinkConfigurationStatus
property CxpPoCxpAuto
property CxpPoCxpStatus
property CxpPoCxpTripReset
property CxpPoCxpTurnOff
property DecimationHorizontal
property DecimationHorizontalMode
property DecimationSelector
property DecimationVertical
property DecimationVerticalMode
property DefectCorrectStaticEnable
property DefectCorrectionMode
property DefectTableApply
property DefectTableCoordinateX
property DefectTableCoordinateY
property DefectTableFactoryRestore
property DefectTableIndex
property DefectTablePixelCount
property DefectTableSave
property DefectTableSensor
property Deinterlacing
property DeviceCharacterSet
property DeviceClockFrequency
property DeviceClockSelector
property DeviceConnectionSelector
property DeviceConnectionSpeed
property DeviceConnectionStatus
property DeviceEventChannelCount

property DeviceFamilyName

property DeviceFeaturePersistenceEnd

property DeviceFeaturePersistenceStart

property DeviceFirmwareVersion

property DeviceGenCPVersionMajor

property DeviceGenCPVersionMinor

property DeviceID

property DeviceIndicatorMode

property DeviceLinkBandwidthReserve

property DeviceLinkCommandTimeout

property DeviceLinkConnectionCount

property DeviceLinkCurrentThroughput

property DeviceLinkHeartbeatMode

property DeviceLinkHeartbeatTimeout

property DeviceLinkSelector

property DeviceLinkSpeed

property DeviceLinkThroughputLimit

property DeviceLinkThroughputLimitMode

property DeviceManifestEntrySelector

property DeviceManifestPrimaryURL

property DeviceManifestSchemaMajorVersion

property DeviceManifestSchemaMinorVersion

property DeviceManifestSecondaryURL

property DeviceManifestXMLMajorVersion

property DeviceManifestXMLMinorVersion

property DeviceManifestXMLSubMinorVersion

property DeviceManufacturerInfo

property DeviceMaxThroughput

property DeviceModelName

property DevicePowerSupplySelector

property DeviceRegistersCheck

property DeviceRegistersEndianness
property DeviceRegistersStreamingEnd
property DeviceRegistersStreamingStart
property DeviceRegistersValid
property DeviceReset
property DeviceSFNCVersionMajor
property DeviceSFNCVersionMinor
property DeviceSFNCVersionSubMinor
property DeviceScanType
property DeviceSensorChroma
property DeviceSerialNumber
property DeviceSerialPortBaudRate
property DeviceSerialPortSelector
property DeviceStreamChannelCount
property DeviceStreamChannelEndianness
property DeviceStreamChannelLink
property DeviceStreamChannelPacketSize
property DeviceStreamChannelSelector
property DeviceStreamChannelType
property DeviceTLType
property DeviceTLVersionMajor
property DeviceTLVersionMinor
property DeviceTLVersionSubMinor
property DeviceTapGeometry
property DeviceTemperature
property DeviceTemperatureSelector
property DeviceType
property DeviceUptime
property DeviceUserID
property DeviceVendorName
property DeviceVersion

property EncoderDivider
property EncoderMode
property EncoderOutputMode
property EncoderReset
property EncoderResetActivation
property EncoderResetSource
property EncoderSelector
property EncoderSourceA
property EncoderSourceB
property EncoderStatus
property EncoderTimeout
property EncoderValue
property EncoderValueAtReset
property EnumerationCount
property EventAcquisitionEnd
property EventAcquisitionEndFrameID
property EventAcquisitionEndTimestamp
property EventAcquisitionError
property EventAcquisitionErrorFrameID
property EventAcquisitionErrorTimestamp
property EventAcquisitionStart
property EventAcquisitionStartFrameID
property EventAcquisitionStartTimestamp
property EventAcquisitionTransferEnd
property EventAcquisitionTransferEndFrameID
property EventAcquisitionTransferEndTimestamp
property EventAcquisitionTransferStart
property EventAcquisitionTransferStartFrameID
property EventAcquisitionTransferStartTimestamp
property EventAcquisitionTrigger
property EventAcquisitionTriggerFrameID

```
property EventAcquisitionTriggerTimestamp
property EventActionLate
property EventActionLateFrameID
property EventActionLateTimestamp
property EventCounter0End
property EventCounter0EndFrameID
property EventCounter0EndTimestamp
property EventCounter0Start
property EventCounter0StartFrameID
property EventCounter0StartTimestamp
property EventCounter1End
property EventCounter1EndFrameID
property EventCounter1EndTimestamp
property EventCounter1Start
property EventCounter1StartFrameID
property EventCounter1StartTimestamp
property EventEncoder0Restarted
property EventEncoder0RestartedFrameID
property EventEncoder0RestartedTimestamp
property EventEncoder0Stopped
property EventEncoder0StoppedFrameID
property EventEncoder0StoppedTimestamp
property EventEncoder1Restarted
property EventEncoder1RestartedFrameID
property EventEncoder1RestartedTimestamp
property EventEncoder1Stopped
property EventEncoder1StoppedFrameID
property EventEncoder1StoppedTimestamp
property EventError
property EventErrorCode
property EventErrorFrameID
```

property EventErrorTimestamp
property EventExposureEnd
property EventExposureEndFrameID
property EventExposureEndTimestamp
property EventExposureStart
property EventExposureStartFrameID
property EventExposureStartTimestamp
property EventFrameBurstEnd
property EventFrameBurstEndFrameID
property EventFrameBurstEndTimestamp
property EventFrameBurstStart
property EventFrameBurstStartFrameID
property EventFrameBurstStartTimestamp
property EventFrameEnd
property EventFrameEndFrameID
property EventFrameEndTimestamp
property EventFrameStart
property EventFrameStartFrameID
property EventFrameStartTimestamp
property EventFrameTransferEnd
property EventFrameTransferEndFrameID
property EventFrameTransferEndTimestamp
property EventFrameTransferStart
property EventFrameTransferStartFrameID
property EventFrameTransferStartTimestamp
property EventFrameTrigger
property EventFrameTriggerFrameID
property EventFrameTriggerTimestamp
property EventLine0AnyEdge
property EventLine0AnyEdgeFrameID
property EventLine0AnyEdgeTimestamp

property EventLine0FallingEdge
property EventLine0FallingEdgeFrameID
property EventLine0FallingEdgeTimestamp
property EventLine0RisingEdge
property EventLine0RisingEdgeFrameID
property EventLine0RisingEdgeTimestamp
property EventLine1AnyEdge
property EventLine1AnyEdgeFrameID
property EventLine1AnyEdgeTimestamp
property EventLine1FallingEdge
property EventLine1FallingEdgeFrameID
property EventLine1FallingEdgeTimestamp
property EventLine1RisingEdge
property EventLine1RisingEdgeFrameID
property EventLine1RisingEdgeTimestamp
property EventLinkSpeedChange
property EventLinkSpeedChangeFrameID
property EventLinkSpeedChangeTimestamp
property EventLinkTrigger0
property EventLinkTrigger0FrameID
property EventLinkTrigger0Timestamp
property EventLinkTrigger1
property EventLinkTrigger1FrameID
property EventLinkTrigger1Timestamp
property EventNotification
property EventSelector
property EventSequencerSetChange
property EventSequencerSetChangeFrameID
property EventSequencerSetChangeTimestamp
property EventSerialData
property EventSerialDataLength

property EventSerialPortReceive
property EventSerialPortReceiveTimestamp
property EventSerialReceiveOverflow
property EventStream@TransferBlockEnd
property EventStream@TransferBlockEndFrameID
property EventStream@TransferBlockEndTimestamp
property EventStream@TransferBlockStart
property EventStream@TransferBlockStartFrameID
property EventStream@TransferBlockStartTimestamp
property EventStream@TransferBlockTrigger
property EventStream@TransferBlockTriggerFrameID
property EventStream@TransferBlockTriggerTimestamp
property EventStream@TransferBurstEnd
property EventStream@TransferBurstEndFrameID
property EventStream@TransferBurstEndTimestamp
property EventStream@TransferBurstStart
property EventStream@TransferBurstStartFrameID
property EventStream@TransferBurstStartTimestamp
property EventStream@TransferEnd
property EventStream@TransferEndFrameID
property EventStream@TransferEndTimestamp
property EventStream@TransferOverflow
property EventStream@TransferOverflowFrameID
property EventStream@TransferOverflowTimestamp
property EventStream@TransferPause
property EventStream@TransferPauseFrameID
property EventStream@TransferPauseTimestamp
property EventStream@TransferResume
property EventStream@TransferResumeFrameID
property EventStream@TransferResumeTimestamp
property EventStream@TransferStart

property EventStream0TransferStartFrameID
property EventStream0TransferStartTimestamp
property EventTest
property EventTestTimestamp
property EventTimer0End
property EventTimer0EndFrameID
property EventTimer0EndTimestamp
property EventTimer0Start
property EventTimer0StartFrameID
property EventTimer0StartTimestamp
property EventTimer1End
property EventTimer1EndFrameID
property EventTimer1EndTimestamp
property EventTimer1Start
property EventTimer1StartFrameID
property EventTimer1StartTimestamp
property ExposureActiveMode
property ExposureAuto
property ExposureMode
property ExposureTime
property ExposureTimeMode
property ExposureTimeSelector
property ExternalVoltageEnable
property ExternalVoltageSelector
property ExternalVoltageValue
property FactoryReset
property FfcEnable
property FfcMode
property FfcUserGain
property FfcUserOffset
property FfcUserTableReset

property FfcUserTableSave

property FfcUserTableXCoordinate

property FileAccessBuffer

property FileAccessLength

property FileAccessOffset

property FileOpenMode

property FileOperationExecute

property FileOperationResult

property FileOperationSelector

property FileOperationStatus

property FileSelector

property FileSize

property Gain

property GainAuto

property GainAutoBalance

property GainConversion

property GainSelector

property Gamma

property GammaEnable

property GevActiveLinkCount

property GevCCP

property GevCurrentDefaultGateway

property GevCurrentIPAddress

property GevCurrentIPConfigurationDHCP

property GevCurrentIPConfigurationLLA

property GevCurrentIPConfigurationPersistentIP

property GevCurrentPhysicalLinkConfiguration

property GevCurrentSubnetMask

property GevDiscoveryAckDelay

property GevFirstURL

property GevGVCPExtendedStatusCodes

property `GevGVCPExtendedStatusCodesSelector`
property `GevGVCPHeartbeatDisable`
property `GevGVCPPendingAck`
property `GevGVCPPendingTimeout`
property `GevGVSPExtendedIDMode`
property `GevHeartbeatTimeout`
property `GevIEEE1588`
property `GevIEEE1588ClockAccuracy`
property `GevIEEE1588ClockId`
property `GevIEEE1588DataSetLatch`
property `GevIEEE1588Mode`
property `GevIEEE1588OffsetFromMasterLatched`
property `GevIEEE1588ParentClockIdLatched`
property `GevIEEE1588Status`
property `GevIEEE1588StatusLatched`
property `GevIPConfigurationStatus`
property `GevInterfaceSelector`
property `GevMACAddress`
property `GevMCDA`
property `GevMCPHostPort`
property `GevMCRC`
property `GevMCSP`
property `GevMCTT`
property `GevNumberOfActiveLinks`
property `GevNumberOfInterfaces`
property `GevPAUSEFrameReception`
property `GevPAUSEFrameTransmission`
property `GevPersistentDefaultGateway`
property `GevPersistentIPAddress`
property `GevPersistentSubnetMask`
property `GevPhysicalLinkConfiguration`

property `GevPhysicalLinkConfigurationCapability`
property `GevPrimaryApplicationIPAddress`
property `GevPrimaryApplicationSocket`
property `GevPrimaryApplicationSwitchoverKey`
property `GevSCCFGAllInTransmission`
property `GevSCCFGExtendedChunkData`
property `GevSCCFGPacketResendDestination`
property `GevSCCFGUnconditionalStreaming`
property `GevSCDA`
property `GevSCPD`
property `GevSCPDDirection`
property `GevSCPHostPort`
property `GevSCPIInterfaceIndex`
property `GevSCPSBigEndian`
property `GevSCPSDoNotFragment`
property `GevSCPSFireTestPacket`
property `GevSCPSPacketSize`
property `GevSCSP`
property `GevSCZoneConfigurationLock`
property `GevSCZoneCount`
property `GevSCZoneDirectionAll`
property `GevSecondURL`
property `GevStreamChannelSelector`
property `GevSupportedOption`
property `GevSupportedOptionSelector`
property `GevTimestampTickFrequency`
property `GuiXmlManifestAddress`
property `Height`
property `HeightMax`
property `ImageComponentEnable`
property `ImageComponentSelector`

```
property ImageCompressionBitrate
property ImageCompressionJPEGFormatOption
property ImageCompressionMode
property ImageCompressionQuality
property ImageCompressionRateOption
Init(self)
    void Spinnaker::Camera::Init()
property IspEnable
property LUTEnable
property LUTIndex
property LUTSelector
property LUTValue
property LUTValueAll
property LargePenalty
property LensShadingCoefficientActiveSet
property LensShadingCorrectionCalibration
property LensShadingCorrectionCalibrationGainLimit
property LensShadingCorrectionCalibrationSetup
property LensShadingCorrectionCalibrationStatus
property LensShadingCorrectionMode
property LensShadingCorrectionStepSize
property LensShadingCorrectionVersion
property LineFilterWidth
property LineFormat
property LineInputFilterSelector
property LineInverter
property LineMode
property LinePitch
property LineSelector
property LineSource
property LineStatus
```

property LineStatusAll
property LinkErrorCount
property LinkRecoveryCount
property LinkUptime
property LogicBlockLUTInputActivation
property LogicBlockLUTInputSelector
property LogicBlockLUTInputSource
property LogicBlockLUTOutputValue
property LogicBlockLUTOutputValueAll
property LogicBlockLUTRowIndex
property LogicBlockLUTSelector
property LogicBlockSelector
property MaxDataRateThreshold
property MaxDeviceResetTime
property MultiRoiConfigurationInvalidReason
property MultiRoiConfigurationInvalidReasonAll
property MultiRoiEnable
property MultiRoiFeatureEnable
property MultiRoiHeight
property MultiRoiOffsetX
property MultiRoiOffsetY
property MultiRoiSelector
property MultiRoiWidth
property MultiRoiWindows
property NumDirections
property OffsetX
property OffsetY
property PacketResendRequestCount
property PacketResendRequestsDroppedCount
property PauseFrameCount
property PayloadSize

property PixelColorFilter

property PixelDynamicRangeMax

property PixelDynamicRangeMin

property PixelFormat

property PixelFormatInfoID

property PixelFormatInfoSelector

property PixelSize

property PowerSupplyCurrent

property PowerSupplyVoltage

property RegionDestination

property RegionMode

property RegionSelector

property ReverseX

property ReverseY

property RgbTransformLightSource

property Saturation

property SaturationEnable

property Scan3dAxisMax

property Scan3dAxisMin

property Scan3dBaseline

property Scan3dCoordinateOffset

property Scan3dCoordinateReferenceSelector

property Scan3dCoordinateReferenceValue

property Scan3dCoordinateScale

property Scan3dCoordinateSelector

property Scan3dCoordinateSystem

property Scan3dCoordinateSystemReference

property Scan3dCoordinateTransformSelector

property Scan3dDistanceUnit

property Scan3dFocalLength

property Scan3dInvalidDataFlag

property Scan3dInvalidDataValue
property Scan3dOutputMode
property Scan3dPrincipalPointU
property Scan3dPrincipalPointV
property Scan3dTransformValue
property SensorDescription
property SensorDigitizationTaps
property SensorHeight
property SensorShutterMode
property SensorTaps
property SensorWidth
property SequencerConfigurationMode
property SequencerConfigurationReset
property SequencerConfigurationValid
property SequencerFeatureEnable
property SequencerMode
property SequencerPathSelector
property SequencerSetActive
property SequencerSetLoad
property SequencerSetNext
property SequencerSetSave
property SequencerSetSelector
property SequencerSetStart
property SequencerSetValid
property SequencerTriggerActivation
property SequencerTriggerSource
property SerialPortBaudRate
property SerialPortDataBits
property SerialPortParity
property SerialPortSelector
property SerialPortSource

property SerialPortStopBits
property SerialReceiveFramingErrorCount
property SerialReceiveParityErrorCount
property SerialReceiveQueueClear
property SerialReceiveQueueCurrentCharacterCount
property SerialReceiveQueueMaxCharacterCount
property SerialTransmitQueueCurrentCharacterCount
property SerialTransmitQueueMaxCharacterCount
property Sharpening
property SharpeningAuto
property SharpeningEnable
property SharpeningThreshold
property SmallPenalty
property SoftwareSignalPulse
property SoftwareSignalSelector
property SourceCount
property SourceSelector
property StereoHeight
property StereoResolution
property StereoWidth
property TLParamsLocked
property Test0001
property TestEventGenerate
property TestPattern
property TestPatternGeneratorSelector
property TestPendingAck
property TimerDelay
property TimerDuration
property TimerReset
property TimerSelector
property TimerStatus

property TimerTriggerActivation
property TimerTriggerSource
property TimerValue
property Timestamp
property TimestampIncrement
property TimestampLatch
property TimestampLatchValue
property TimestampReset
property TotalDisparity
property TransferAbort
property TransferBlockCount
property TransferBurstCount
property TransferComponentSelector
property TransferControlMode
property TransferOperationMode
property TransferPause
property TransferQueueCurrentBlockCount
property TransferQueueMaxBlockCount
property TransferQueueMode
property TransferQueueOverflowCount
property TransferResume
property TransferSelector
property TransferStart
property TransferStatus
property TransferStatusSelector
property TransferStop
property TransferStreamChannel
property TransferTriggerActivation
property TransferTriggerMode
property TransferTriggerSelector
property TransferTriggerSource

property TransmissionDelay
property TransmissionDelayAverage
property TransmissionDelayMax
property TriggerActivation
property TriggerDelay
property TriggerDivider
property TriggerEventTest
property TriggerMode
property TriggerMultiplier
property TriggerOverlap
property TriggerSelector
property TriggerSoftware
property TriggerSource
property U3VAccessPrivilege
property U3VCPCapability
property U3VCPEIRMAvailable
property U3VCPIIDC2Available
property U3VCPSIRMAvailable
property U3VCurrentSpeed
property U3VMaxAcknowledgeTransferLength
property U3VMaxCommandTransferLength
property U3VMaxDeviceResponseTime
property U3VMessageChannelID
property U3VNumberOfStreamChannels
property U3VVersionMajor
property U3VVersionMinor
property UniquenessRatio
property UserOutputSelector
property UserOutputValue
property UserOutputValueAll
property UserOutputValueAllMask

property `UserSetDefault`

property `UserSetFeatureEnable`

property `UserSetLoad`

property `UserSetSave`

property `UserSetSelector`

property `V3_3Enable`

property `WhiteClip`

property `WhiteClipSelector`

property `Width`

property `WidthMax`

property `WindowSizeH`

property `WindowSizeW`

property `aPAUSEMACtrlFramesReceived`

property `aPAUSEMACtrlFramesTransmitted`

property `thisown`

The membership flag

class `PySpin.CameraBase(*args, **kwargs)`

Bases: [*ICameraBase*](#)

The base class for the camera object.

C++ includes: `CameraBase.h`

BeginAcquisition(*self*)

`void Spinnaker::CameraBase::BeginAcquisition()`

Starts the image acquisition engine. The camera must be initialized via a call to `Init()` before starting an acquisition.

See: `Init()`

DeInit(*self*)

`void Spinnaker::CameraBase::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: `Init()`

See: `UnregisterEvent(Event & evtToUnregister)`

DiscoverMaxPacketSize(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::DiscoverMaxPacketSize()

Returns the largest packet size that can be safely used on the interface that device is connected to

The maximum packet size returned.

EndAcquisition(*self*)

void Spinnaker::CameraBase::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: Init()

See: BeginAcquisition()

See: GetNextImage(grabTimeout)

See: Image::Release()

ForceIP(*self*)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GenApi::EAccessMode Spinnaker::CameraBase::GetAccessMode() const

Returns the access mode that the software has on the Camera. The camera does not need to be initialized before calling this function.

See: Init()

An enumeration value indicating the access mode

GetActiveNumDataStreams(*self*) → unsigned int

GetBufferOwnership(*self*) → Spinnaker::BufferOwnership

GetDeviceID(*self*) → *gcstring*

GetGuiXml(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetGuiXml() const

Returns the GUI XML that can be passed into the Spinnaker GUI framework

GenICam::gcstring that represents the uncompressed GUI XML file

GetNextImage(*self*, *grabTimeout*=EVENT_TIMEOUT_INFINITE, *streamIndex*=0) → *ImagePtr*

Parameters

- **grabTimeout** (a 64bit value that represents a timeout in milliseconds)
- **streamIndex** (uint64_t)
- **ImagePtr**
- **Spinnaker::CameraBase::GetNextImage(uint64_t**
- **grabTimeout=EVENT_TIMEOUT_INFINITE**
- **streamID=0)** (uint64_t)
- **This** (Gets the next image that was received by the transport layer.)

- **cameras** (function will block indefinitely until an image arrives. Most)
- **camera** (support one stream so the default streamID is 0 but if a)
- **select** (supports multiple streams the user can input the streamID to)
- **images** (from which stream to grab)
- **See** (`EndAcquisition()`)
- **See**
- **See**
- **Parameters**
- -----
- **grabTimeout**
- **streamID** (The stream to grab the image.)
- **object** (pointer to an Image)

GetNextImageSync(*self*, *grabTimeout*=`EVENT_TIMEOUT_INFINITE`) → *ImageList*

Parameters

grabTimeout (`uint64_t`)

GetNodeMap(*self*) → *INodeMap*

`GenApi::INodeMap& Spinnaker::CameraBase::GetNodeMap() const`

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: `Init()`

A reference to the `INodeMap`.

GetNumDataStreams(*self*) → unsigned int

`unsigned int Spinnaker::CameraBase::GetNumDataStreams()`

Returns the number of streams that a device supports.

The number of data streams

GetNumImagesInUse(*self*) → unsigned int

`unsigned int Spinnaker::CameraBase::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()`.

The number of images that needs to be cleaned up.

GetTLDeviceNodeMap(*self*) → *INodeMap*

`GenApi::INodeMap& Spinnaker::CameraBase::GetTLDeviceNodeMap() const`

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.

A reference to the `INodeMap`.

GetTLStreamNodeMap(*self*, *streamIndex*=0) → *INodeMap*

Parameters

- **streamIndex** (*uint64_t*)
- **const** (*GenApi::INodeMap& Spinnaker::CameraBase::GetTLStreamNodeMap()*)
- **XML** (Gets a reference to the node map that is generated from a *GenICam*)
- **be** (file for the *GenTL Stream* module. The camera does not need to)
- **map.** (initialized before acquiring this node)
- **INodeMap.** (A reference to the)

GetUniqueID(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetUniqueID()

This returns a unique id string that identifies the camera. This is the camera serial number.
string that uniquely identifies the camera (serial number)

GetUserBufferCount(*self*) → *uint64_t*

GetUserBufferSize(*self*) → *uint64_t*

GetUserBufferTotalSize(*self*) → *uint64_t*

Init(*self*)

void Spinnaker::CameraBase::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: *BeginAcquisition()*

See: *EndAcquisition()*

See: *GetNodeMap()*

See: *GetNextImage()*

IsInitialized(*self*) → *bool*

bool Spinnaker::CameraBase::IsInitialized()

Checks if camera is initialized. This function needs to return true in order to retrieve a valid *NodeMap* from the *GetNodeMap()* call.

See: *GetNodeMap()*

If camera is initialized or not

IsStreaming(*self*) → *bool*

bool Spinnaker::CameraBase::IsStreaming() const

Returns true if the camera is currently streaming or false if it is not.

See: *Init()*

returns true if camera is streaming and false otherwise.

IsValid(*self*) → bool

bool Spinnaker::CameraBase::IsValid()

Checks a flag to determine if camera is still valid for use.

If camera is valid or not

Note that CameraPtr and CameraBase both define an IsValid() function. In order to determine the validity of the camera using a CameraPtr, user must first call get() to retrieve the CameraBase object.

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::ImageEventHandler &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **eventName**)
- **evtHandlerToRegister**
- **eventName** (*Spinnaker::GenICam::gcstring const &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **streamIndex**)
- **evtHandlerToRegister**
- **streamIndex** (*uint64_t*)

SetBufferOwnership(*self*, *mode*)

Parameters

mode (*enum Spinnaker::BufferOwnership const*)

SetUserBuffers(*self*, *pMemBuffers*, *totalSize*)

Parameters

- **pMemBuffers** (*void *const*)
- **totalSize** (*uint64_t*)
- **SetUserBuffers**(*self*
- **ppMemBuffers** (*void **const*)
- **bufferCount** (*uint64_t const*)
- **bufferSize**)
- **ppMemBuffers**
- **bufferCount**
- **bufferSize** (*uint64_t const*)

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

property thisown

The membership flag

class PySpin.CameraList(*args)

Bases: *ICameraList*

Used to hold a list of camera objects.

C++ includes: CameraList.h

Add(self, camera)

Parameters

camera (Spinnaker::CameraPtr)

Append(self, list)

Parameters

- **list** (Spinnaker::CameraList const &)
- **void**
- **&otherList** (Spinnaker::CameraList::Append(*CameraList*)
- **list.** (Appends a camera list to the current)
- **Parameters**
- -----
- **otherList** (The other list to append to this list)

Clear(self)

void Spinnaker::CameraList::Clear()

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: System:ReleaseInstance()

GetByDeviceID(self, deviceID) → *CameraPtr*

Parameters

deviceID (std::string)

GetByIndex(self, index) → *CameraPtr*

Parameters

- **index** (The index at which to retrieve the camera object)
- **CameraPtr**
- **const** (Spinnaker::CameraList::GetByIndex(int index))
- **"index".** (Returns a pointer to a camera object at the)
- **Parameters**
- -----
- **index**
- **object.** (A pointer to an camera)

GetBySerial(*self*, *serialNumber*) → *CameraPtr*

Parameters

- **serialNumber** (*The serial number of the camera object to retrieve*)
- **CameraPtr**
- **const** (*Spinnaker::CameraList::GetBySerial(std::string serialNumber)*)
- **number.** (*Returns a pointer to a camera object with the specified serial*)
- **Parameters**
- -----
- **serialNumber**
- **object.** (*A pointer to an camera*)

GetSize(*self*) → unsigned int

int Spinnaker::CameraList::GetSize() const

Returns the size of the camera list. The size is the number of Camera objects stored in the list.

An integer that represents the list size.

Remove(*self*, *camera*)

Parameters

camera (*Spinnaker::CameraPtr*)

RemoveByDeviceID(*self*, *deviceID*)

Parameters

deviceID (*std::string*)

RemoveByIndex(*self*, *index*)

Parameters

- **index** (*The index at which to remove the Camera object*)
- **void**
- **index** (*Spinnaker::CameraList::RemoveByIndex(int)*)
- **reference** (*Removes a camera at "index" and destroys its corresponding*)
- **object.** (*counted*)
- **Parameters**
- -----
- **index**

RemoveBySerial(*self*, *serialNumber*)

Parameters

- **serialNumber** (*The serial number of the Camera object to remove*)
- **void**
- **serialNumber** (*Spinnaker::CameraList::RemoveBySerial(std::string)*)

- **its** (*Removes a camera using its serial number and destroys*)
- **object.** (*corresponding reference counted*)
- **Parameters**
- -----
- **serialNumber**

property thisown

The membership flag

class PySpin.CameraPtr(*args)

Bases: _SWIG_CamPtr

A reference tracked pointer to a camera object.

C++ includes: CameraPtr.h

property thisown

The membership flag

class PySpin.CategoryNode(*args, **kwargs)

Bases: *ICategory*, *ValueNode*

Interface for string properties.

C++ includes: CategoryNode.h

GetFeatures(self)

virtual void Spinnaker::GenApi::CategoryNode::GetFeatures(FeatureList_t &Features) const

Get all features of the category (including sub-categories)

SetReference(self, pBase)**Parameters**

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::CategoryNode::SetReference(INode** (*virtual void*)
- ***pBase)**
- **Value** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.ChannelStatistics(image, channel)

Bases: object

Class used to store statistics (as properties) for one channel of an image. Properties:

- **channel**: The image channel that the statistics are based on (as an int).
- **range_min**: The smallest possible pixel value.
- **range_max**: The largest possible pixel value.
- **pixel_value_min**: The smallest pixel value in the current channel.
- **pixel_value_max**: The largest pixel value in the current channel.
- **num_pixel_values**: The total number of pixel values in the current channel.

- `pixel_value_mean`: The average pixel value in the current channel.
- `histogram`: NumPy array representing the histogram of the current channel.

property `channel`

property `histogram`

property `num_pixel_values`

property `pixel_value_max`

property `pixel_value_mean`

property `pixel_value_min`

property `range_max`

property `range_min`

property `thisown`

The membership flag

class `PySpin.ChunkData(*args)`

Bases: [*IChunkData*](#)

The chunk data which contains additional information about an image.

C++ includes: `ChunkData.h`

GetBlackLevel(*self*) → `float64_t`

`float64_t Spinnaker::ChunkData::GetBlackLevel() const`

Description: Returns the black level used to capture the image included in the payload. Visibility: Expert

GetCRC(*self*) → `int64_t`

GetCompressionMode(*self*) → `int64_t`

GetCompressionRatio(*self*) → `float64_t`

GetCounterValue(*self*) → `int64_t`

`int64_t Spinnaker::ChunkData::GetCounterValue() const`

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event. Visibility: Expert

GetCurrentDataRate(*self*) → `int64_t`

GetEnable(*self*) → `bool`

GetEncoderValue(*self*) → `int64_t`

`int64_t Spinnaker::ChunkData::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. Visibility: Expert

GetExposureEndLineStatusAll(*self*) → `int64_t`

GetExposureTime(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetExposureTime() const

Description: Returns the exposure time used to capture the image. Visibility: Expert

GetFrameID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetFrameID() const

Description: Returns the unique Identifier of the frame (or image) included in the payload. Visibility: Expert

GetGain(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetGain() const

Description: Returns the gain used to capture the image. Visibility: Expert

GetHeight(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetHeight() const

Description: Returns the Height of the image included in the payload. Visibility: Expert

GetImage(*self*) → int64_t

GetInferenceBoundingBoxResult(*self*) → *InferenceBoundingBoxResult*

GetInferenceConfidence(*self*) → float64_t

GetInferenceFrameId(*self*) → int64_t

GetInferenceResult(*self*) → int64_t

GetLinePitch(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLinePitch() const

Description: Returns the LinePitch of the image included in the payload. Visibility: Expert

GetLineStatusAll(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLineStatusAll() const

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event. Visibility: Expert

GetModeActive(*self*) → bool

GetOffsetX(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetX() const

Description: Returns the OffsetX of the image included in the payload. Visibility: Expert

GetOffsetY(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetY() const

Description: Returns the OffsetY of the image included in the payload. Visibility: Expert

GetPartSelector(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPartSelector() const

Description: Selects the part to access in chunk data in a multipart transmission. Visibility: Expert

GetPixelDynamicRangeMax(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMax() const

Description: Returns the maximum value of dynamic range of the image included in the payload. Visibility: Expert

GetPixelDynamicRangeMin(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMin() const

Description: Returns the minimum value of dynamic range of the image included in the payload. Visibility: Expert

GetScan3dAxisMax(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMax() const

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dAxisMin(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMin() const

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateOffset(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateOffset() const

Description: Returns the Offset for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateReferenceValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateReferenceValue() const

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point. Visibility: Expert

GetScan3dCoordinateScale(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateScale() const

Description: Returns the Scale for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dInvalidDataFlag(*self*) → bool

GetScan3dInvalidDataValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dInvalidDataValue() const

Description: Returns the Invalid Data Value used for the image included in the payload. Visibility: Expert

GetScan3dTransformValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dTransformValue() const

Description: Returns the transform value. Visibility: Expert

GetScanLineSelector(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetScanLineSelector() const

Description: Index for vector representation of one chunk value per line in an image. Visibility: Expert

GetSequencerSetActive(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetSequencerSetActive() const

Description: Return the index of the active set of the running sequencer included in the payload. Visibility: Expert

GetSerialData(*self*) → uint8_t *

GetSerialDataLength(*self*) → int64_t

GetSerialReceiveOverflow(*self*) → bool

GetStreamChannelID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetStreamChannelID() const

Description: Returns identifier of the stream channel used to carry the block. Visibility: Expert

GetTimerValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetTimerValue() const

Description: Returns the value of the selected Timer at the time of the FrameStart internal event. Visibility: Expert

GetTimestamp(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTimestamp() const

Description: Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event. Visibility: Expert

GetTimestampLatchValue(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTimestampLatchValue() const

Description: Returns the last Timestamp latched with the TimestampLatch command. Visibility: Expert

GetTransferBlockID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTransferBlockID() const

Description: Returns the unique identifier of the transfer block used to transport the payload. Visibility: Expert

GetTransferQueueCurrentBlockCount(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTransferQueueCurrentBlockCount() const

Description: Returns the current number of blocks in the transfer queue. Visibility: Expert

GetWidth(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetWidth() const

Description: Returns the Width of the image included in the payload. Visibility: Expert

SetChunks(*self*, *pNodeMap*)

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap &*)
- **void**
- **&pNodeMap** (*Spinnaker::ChunkData::SetChunks(GenApi::INodeMap)*)

property thisown

The membership flag

PySpin.**Combine**(*Peter, Paul*) → Spinnaker::GenApi::EAccessMode

Parameters

- **Peter** (*Spinnaker::GenApi::Combine(ECachingMode)*)
- **Paul** (*enum Spinnaker::GenApi::ECachingMode*)
- **Combine**(**Peter**
- **Spinnaker::GenApi::EVisibility** (*Paul*) ->)
- **Peter**
- **Paul**
- **Combine**(**Peter**
- **Spinnaker::GenApi::ECachingMode** (*Paul*) ->)
- **Peter**
- **Paul**
- **ECachingMode**
- **Peter**
- **Paul**) (*ECachingMode*)
- **combination** (*Computes which CachingMode results from a*)

class PySpin.**CommandNode**(*args, **kwargs)

Bases: [ICommand](#), [ValueNode](#)

Interface for string properties.

C++ includes: CommandNode.h

Execute(*self, Verify=True*)

Parameters

- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Verify=true** (*virtual void Spinnaker::GenApi::CommandNode::Execute(bool)*)
- **command** (*Execute the*)
- **Parameters**
- -----
- **Verify**

IsDone(*self, Verify=True*) → bool

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **Verify=true** (*virtual bool Spinnaker::GenApi::CommandNode::IsDone(bool)*)
- **executed** (*Query whether the command is*)
- **Parameters**
- -----
- **Verify**

- **checked** (*is always*)
- **otherwise** (*True if the Execute command has finished; false*)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::CommandNode::SetReference**(*INode* (*virtual void*))
- ***pBase**
- **Value** (*overload SetReference for*)

property thisown

The membership flag

PySpin.DeregisterNodeCallback(*f*)

Parameters

f (*NodeCallback &*)

class PySpin.DeviceArrivalEventHandler

Bases: *IDeviceArrivalEventHandler*

Proxy of C++ Spinnaker::DeviceArrivalEventHandler class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class PySpin.DeviceEventExposureEndData

Bases: *object*

Proxy of C++ Spinnaker::DeviceEventExposureEndData class.

property frameID

property thisown

The membership flag

class PySpin.DeviceEventHandler

Bases: *IDeviceEventHandler*

Proxy of C++ Spinnaker::DeviceEventHandler class.

GetDeviceEventId(*self*) → *uint64_t*

GetDeviceEventName(*self*) → *gcstring*

OnDeviceEvent(*self*, *eventName*)

Parameters

eventName (*Spinnaker::GenICam::gcstring*)

property thisown

The membership flag

class PySpin.DeviceEventInferenceData

Bases: object

Proxy of C++ Spinnaker::DeviceEventInferenceData class.

property confidence**property** frameID**property** result**property** thisown

The membership flag

class PySpin.DeviceRemovalEventHandlerBases: *IDeviceRemovalEventHandler*

Proxy of C++ Spinnaker::DeviceRemovalEventHandler class.

OnDeviceRemoval(*self*, *pCamera*)**Parameters****pCamera** (*Spinnaker::CameraPtr*)**property** thisown

The membership flag

PySpin.DoesEnvironmentVariableExist(*VariableName*) → bool**Parameters**

- **VariableName** (*Spinnaker::GenICam::gcstring const &*)
- **bool** (*SPINNAKER_API*)
- **Spinnaker::GenICam::DoesEnvironmentVariableExist(const**
- **&VariableName)** (*Spinnaker::GenICam::gcstring*)
- **exists** (*Returns true if an environment variable*)

class PySpin.EAccessModeClass

Bases: object

Holds conversion methods for the access mode enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr*, *pValue*) → bool**Parameters**

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EAccessMode **)

static ToString(*ValueStr*, *pValue*)**Parameters**

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EAccessMode **)
- **gcstring** (*ToString(Value) ->*)

- **Value** (*enum Spinnaker::GenApi::EAccessMode*)

property thisown

The membership flag

PySpin.EAccessModeClass_FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EAccessMode **)

PySpin.EAccessModeClass_ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EAccessMode **)
- **gcstring** (*EAccessModeClass_ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EAccessMode*)

class PySpin.ECachingModeClass

Bases: object

Holds conversion methods for the caching mode enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ECachingMode **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ECachingMode **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ECachingMode*)

property thisown

The membership flag

PySpin.ECachingModeClass_FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ECachingMode **)

PySpin.ECachingModeClass_ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ECachingMode **)
- **gcstring** (*ECachingModeClass_ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ECachingMode*)

class PySpin.EDisplayNotationClass

Bases: object

Holds conversion methods for the notation type of floats.

C++ includes: EnumClasses.h

static FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EDisplayNotation **)

static ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EDisplayNotation **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EDisplayNotation*)

property thisown

The membership flag

PySpin.EDisplayNotationClass_FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EDisplayNotation **)

PySpin.EDisplayNotationClass_ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EDisplayNotation **)
- **gcstring** (*EDisplayNotationClass_ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EDisplayNotation*)

class PySpin.EEndianessClass

Bases: object

Holds conversion methods for the endianess enumeration.

C++ includes: EnumClasses.h

```
static FromString(ValueStr, pValue) → bool
```

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::EEndianness *)

```
static ToString(ValueStr, pValue)
```

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **pValue** (Spinnaker::GenApi::EEndianness *)
- **gcstring** (ToString(Value) ->)
- **Value** (enum Spinnaker::GenApi::EEndianness)

property thisown

The membership flag

```
PySpin.EEndiannessClass_FromString(ValueStr, pValue) → bool
```

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::EEndianness *)

```
PySpin.EEndiannessClass_ToString(ValueStr, pValue)
```

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **pValue** (Spinnaker::GenApi::EEndianness *)
- **gcstring** (EEndiannessClass_ToString(Value) ->)
- **Value** (enum Spinnaker::GenApi::EEndianness)

```
class PySpin.EGenApiSchemaVersionClass
```

Bases: object

helper class converting EGenApiSchemaVersion from and to string

C++ includes: EnumClasses.h

```
static FromString(ValueStr, pValue) → bool
```

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::EGenApiSchemaVersion *)

```
static ToString(ValueStr, pValue)
```

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **pValue** (Spinnaker::GenApi::EGenApiSchemaVersion *)
- **gcstring** (ToString(Value) ->)
- **Value** (enum Spinnaker::GenApi::EGenApiSchemaVersion)

property thisown

The membership flag

`PySpin.EGenApiSchemaVersionClass_FromString(ValueStr, pValue) → bool`

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EGenApiSchemaVersion **)

`PySpin.EGenApiSchemaVersionClass_ToString(ValueStr, pValue)`

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EGenApiSchemaVersion **)
- **gcstring** (*EGenApiSchemaVersionClass_ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EGenApiSchemaVersion*)

class PySpin.EInputDirectionClass

Bases: object

Holds conversion methods for the notation type of floats.

C++ includes: EnumClasses.h

static FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EInputDirection **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EInputDirection **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EInputDirection*)

property thisown

The membership flag

`PySpin.EInputDirectionClass_FromString(ValueStr, pValue) → bool`

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EInputDirection **)

`PySpin.EInputDirectionClass_ToString(ValueStr, pValue)`

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EInputDirection **)

- **gcstring** (*EInputDirectionClass_ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EInputDirection*)

class PySpin.ENamespaceClass

Bases: object

Holds conversion methods for the namespace enumeration.

C++ includes: EnumClasses.h

static FromString (*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ENamespace **)

static ToString (*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ENamespace **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ENamespace*)

property thisown

The membership flag

PySpin.ENamespaceClass_FromString (*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ENamespace **)

PySpin.ENamespaceClass_ToString (*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ENamespace **)
- **gcstring** (*ENamespaceClass_ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ENamespace*)

class PySpin.ERepresentationClass

Bases: object

Holds conversion methods for the representation enumeration.

C++ includes: EnumClasses.h

static FromString (*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ERepresentation **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ERepresentation **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ERepresentation*)

property thisown

The membership flag

PySpin.ERepresentationClass_FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ERepresentation **)

PySpin.ERepresentationClass_ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ERepresentation **)
- **gcstring** (*ERepresentationClass_ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ERepresentation*)

class PySpin.ESignClass

Bases: object

Holds conversion methods for the sign enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ESign **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ESign **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ESign*)

property thisown

The membership flag

`PySpin.ESignClass_FromString(ValueStr, pValue) → bool`

Parameters

- `ValueStr` (`Spinnaker::GenICam::gcstring const &`)
- `pValue` (`Spinnaker::GenApi::ESign *`)

`PySpin.ESignClass_ToString(ValueStr, pValue)`

Parameters

- `ValueStr` (`Spinnaker::GenICam::gcstring &`)
- `pValue` (`Spinnaker::GenApi::ESign *`)
- `gcstring` (`ESignClass_ToString(Value) ->`)
- `Value` (`enum Spinnaker::GenApi::ESign`)

class `PySpin.ESlopeClass`

Bases: `object`

Holds conversion methods for the converter formulas.

C++ includes: `EnumClasses.h`

static `FromString(ValueStr, pValue) → bool`

Parameters

- `ValueStr` (`Spinnaker::GenICam::gcstring const &`)
- `pValue` (`Spinnaker::GenApi::ESlope *`)

static `ToString(ValueStr, pValue)`

Parameters

- `ValueStr` (`Spinnaker::GenICam::gcstring &`)
- `pValue` (`Spinnaker::GenApi::ESlope *`)
- `gcstring` (`ToString(Value) ->`)
- `Value` (`enum Spinnaker::GenApi::ESlope`)

property `thisown`

The membership flag

`PySpin.ESlopeClass_FromString(ValueStr, pValue) → bool`

Parameters

- `ValueStr` (`Spinnaker::GenICam::gcstring const &`)
- `pValue` (`Spinnaker::GenApi::ESlope *`)

`PySpin.ESlopeClass_ToString(ValueStr, pValue)`

Parameters

- `ValueStr` (`Spinnaker::GenICam::gcstring &`)
- `pValue` (`Spinnaker::GenApi::ESlope *`)
- `gcstring` (`ESlopeClass_ToString(Value) ->`)
- `Value` (`enum Spinnaker::GenApi::ESlope`)

class PySpin.EStandardNameSpaceClass

Bases: object

Holds conversion methods for the standard namespace enumeration.

C++ includes: EnumClasses.h

static FromString(ValueStr, pValue) → bool

Parameters

- ValueStr (Spinnaker::GenICam::gcstring const &)
- pValue (Spinnaker::GenApi::EStandardNameSpace *)

static ToString(ValueStr, pValue)

Parameters

- ValueStr (Spinnaker::GenICam::gcstring &)
- pValue (Spinnaker::GenApi::EStandardNameSpace *)
- gcstring (ToString(Value) ->)
- Value (enum Spinnaker::GenApi::EStandardNameSpace)

property thisown

The membership flag

PySpin.EStandardNameSpaceClass_FromString(ValueStr, pValue) → bool

Parameters

- ValueStr (Spinnaker::GenICam::gcstring const &)
- pValue (Spinnaker::GenApi::EStandardNameSpace *)

PySpin.EStandardNameSpaceClass_ToString(ValueStr, pValue)

Parameters

- ValueStr (Spinnaker::GenICam::gcstring &)
- pValue (Spinnaker::GenApi::EStandardNameSpace *)
- gcstring (EStandardNameSpaceClass_ToString(Value) ->)
- Value (enum Spinnaker::GenApi::EStandardNameSpace)

class PySpin.EVisibilityClass

Bases: object

Holds conversion methods for the visibility enumeration.

C++ includes: EnumClasses.h

static FromString(ValueStr, pValue) → bool

Parameters

- ValueStr (Spinnaker::GenICam::gcstring const &)
- pValue (Spinnaker::GenApi::EVisibility *)

static ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **pValue** (*Spinnaker::GenApi::EVisibility* *)
- **gcstring** (*ToString(Value)* ->)
- **Value** (*enum Spinnaker::GenApi::EVisibility*)

property thisown

The membership flag

PySpin.EVisibilityClass_FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* const &)
- **pValue** (*Spinnaker::GenApi::EVisibility* *)

PySpin.EVisibilityClass_ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **pValue** (*Spinnaker::GenApi::EVisibility* *)
- **gcstring** (*EVisibilityClass_ToString(Value)* ->)
- **Value** (*enum Spinnaker::GenApi::EVisibility*)

class PySpin.EYesNoClass

Bases: object

Holds conversion methods for the standard namespace enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* const &)
- **pValue** (*Spinnaker::GenApi::EYesNo* *)

static ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **pValue** (*Spinnaker::GenApi::EYesNo* *)
- **gcstring** (*ToString(Value)* ->)
- **Value** (*enum Spinnaker::GenApi::EYesNo*)

property thisown

The membership flag

PySpin.EYesNoClass_FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* const &)
- **pValue** (*Spinnaker::GenApi::EYesNo* *)

PySpin.EYesNoClass_ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **pValue** (*Spinnaker::GenApi::EYesNo* *)
- **gcstring** (*EYesNoClass_ToString*(*Value*) ->)
- **Value** (enum *Spinnaker::GenApi::EYesNo*)

PySpin.EatComments(*_is*) → std::istream &

Parameters

- **is** (std::istream &)
- **SPINNAKER_API**
- **&is** (std::istream& *Spinnaker::GenApi::EatComments*(std::istream))
- **'#'**. (Helper function ignoring lines starting with comment character)

class PySpin.EnumEntryNode(**args*, ***kwargs*)

Bases: [IEnumEntry](#), [ValueNode](#)

Interface for string properties.

C++ includes: EnumEntryNode.h

GetNumericValue(*self*) → double

virtual double *Spinnaker::GenApi::EnumEntryNode::GetNumericValue*()

Get double number associated with the entry

GetSymbolic(*self*) → *gcstring*

virtual *GenICam::gcstring* *Spinnaker::GenApi::EnumEntryNode::GetSymbolic*() const

Get symbolic enum value

GetValue(*self*) → int64_t

virtual int64_t *Spinnaker::GenApi::EnumEntryNode::GetValue*()

Get numeric enum value

IsSelfClearing(*self*) → bool

virtual bool *Spinnaker::GenApi::EnumEntryNode::IsSelfClearing*()

Indicates if the corresponding EnumEntry is self clearing

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode* *)
- **Spinnaker::GenApi::EnumEntryNode::SetReference**(*INode* (virtual void))

- ***pBase)**
- **EnumEntry** (overload SetReference for)

property thisown

The membership flag

class PySpin.**EnumNode**(*args, **kwargs)

Bases: [IEnumeration](#), [ValueNode](#)

Interface for string properties.

C++ includes: EnumNode.h

GetCurrentEntry(self, Verify=False, IgnoreCache=False) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)
- **Spinnaker::GenApi::EnumNode::GetCurrentEntry**(bool [IEnumEntry*](#)) (virtual)
- **Verify=false**
- **IgnoreCache=false** (bool)
- **entry** (Get the current)

GetEntries(self)

virtual void Spinnaker::GenApi::EnumNode::GetEntries(NodeList_t &Entries)

Get list of entry nodes

GetEntry(self, IntValue) → [IEnumEntry](#)

Parameters

- **IntValue** (Get an entry node by its)
- **virtual**
- **int64_t** ([IEnumEntry*](#) **Spinnaker::GenApi::EnumNode::GetEntry**(const
- **IntValue)**
- **IntValue**

GetEntryByName(self, Symbolic) → [IEnumEntry](#)

Parameters

- **Symbolic** (**Spinnaker::GenICam::gcstring** const &)
- **Spinnaker::GenApi::EnumNode::GetEntryByName**(const [IEnumEntry*](#)) (virtual)
- **&Symbolic** (**GenICam::gcstring**)
- **name** (Get an entry node by)

GetIntValue(self, Verify=False, IgnoreCache=False) → int64_t

Parameters

- **Verify** (Enables Range verification (default = false). The AccessMode)

- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::EnumNode::GetIntValue**(bool (virtual int64_t)
- **Verify=false**
- **IgnoreCache=false**) (bool)
- **value** (Get integer node)
- **Parameters**
- -----
- **Verify**
- **checked** (is always)
- **IgnoreCache**
- **false**)
- **read** (The value)

GetSymbolics(self, Symbolics)

Parameters

- **Symbolics** (Spinnaker::GenApi::StringList_t &)
- **Spinnaker::GenApi::EnumNode::GetSymbolics**(StringList_t (virtual void)
- **&Symbolics**)
- **Values** (Get list of symbolic)

SetIntValue(self, Value, Verify=True)

Parameters

- **Value** (virtual void Spinnaker::GenApi::EnumNode::SetIntValue(int64_t)
- **Verify** (bool)
- **Value**

:param : :param bool Verify=true): :param Set integer node value: :param Parameters: :param ———: :param Value: :type Value: The value to set :param Verify: :type Verify: Enables AccessMode and Range verification (default = true)

SetReference(self, pBase)

Parameters

- **pBase** (Spinnaker::GenApi::INode *)
- ***pBase**) (virtual void Spinnaker::GenApi::EnumNode::SetReference(INode)
-
- **Enumeration** (overload SetReference for)

property thisown

The membership flag

class PySpin.**EventHandler**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::EventHandler class.

GetEventPayloadData(self) → PyObject *

GetEventPayloadDataSize(self) → size_t const

GetEventType(self) → Spinnaker::EventType

SetEventType(self, eventType)

Parameters

eventType (enum Spinnaker::EventType)

property thisown

The membership flag

class PySpin.**FloatNode**(*args, **kwargs)

Bases: [IFloat](#), [ValueNode](#)

Interface for string properties.

C++ includes: FloatNode.h

GetDisplayNotation(self) → Spinnaker::GenApi::EDisplayNotation

virtual EDisplayNotation Spinnaker::GenApi::FloatNode::GetDisplayNotation() const

Get the way the float should be converted to a string

GetDisplayPrecision(self) → int64_t

virtual int64_t Spinnaker::GenApi::FloatNode::GetDisplayPrecision() const

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

GetEnumAlias(self) → [IEnumeration](#)

IEnumeration* Spinnaker::GenApi::FloatNode::GetEnumAlias()

gets the interface of an alias node.

GetInc(self) → double

virtual double Spinnaker::GenApi::FloatNode::GetInc()

Get the constant increment if there is any

GetIncMode(self) → Spinnaker::GenApi::EIncMode

virtual EIncMode Spinnaker::GenApi::FloatNode::GetIncMode()

Get increment mode

GetIntAlias(self) → [IInteger](#)

IInteger* Spinnaker::GenApi::FloatNode::GetIntAlias()

gets the interface of an alias node.

GetListOfValidValues(self, bounded=True) → *double_autovector_t*

Parameters

- **bounded** (bool)
- **virtual**

- **double_autovector_t**
- **bounded=true)** (*Spinnaker::GenApi::FloatNode::GetListOfValidValues(bool)*)
- **value** (*Get list of valid*)

GetMax(*self*) → double

virtual double Spinnaker::GenApi::FloatNode::GetMax()

Get maximum value allowed

GetMin(*self*) → double

virtual double Spinnaker::GenApi::FloatNode::GetMin()

Get minimum value allowed

GetRepresentation(*self*) → Spinnaker::GenApi::ERepresentation

virtual ERepresentation Spinnaker::GenApi::FloatNode::GetRepresentation()

Get recommended representation

GetUnit(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::FloatNode::GetUnit() const

Get the physical unit name

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → double

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::FloatNode::GetValue**(*bool* (*virtual double*))
- **Verify=false**
- **IgnoreCache=false)** (*bool*)
- **value** (*Get node*)
- **Parameters**
- **-----**
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false)**
- **read** (*The value*)

HasInc(*self*) → bool

virtual bool Spinnaker::GenApi::FloatNode::HasInc()

True if the float has a constant increment

ImposeMax(*self*, *Value*)

Parameters

- **Value** (*double*)

- **Value** (*virtual void Spinnaker::GenApi::FloatNode::ImposeMax(double)*)
- **value** (*Restrict maximum*)

ImposeMin(*self, Value*)

Parameters

- **Value** (*double*)
- **Value** (*virtual void Spinnaker::GenApi::FloatNode::ImposeMin(double)*)
- **value** (*Restrict minimum*)

SetReference(*self, pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase** (*virtual void Spinnaker::GenApi::FloatNode::SetReference(INode)*
—
- **Float** (*overload SetReference for*)

SetValue(*self, Value, Verify=True*)

Parameters

- **Value** (*The value to set*)
- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Value**
- **bool**
- **Verify=true)**
- **value** (*Set node*)
- **Parameters**
- -----
- **Value**
- **Verify**

property thisown

The membership flag

class PySpin.**FloatRegNode**(*args, **kwargs)

Bases: *FloatNode, RegisterNode*

Interface for string properties.

C++ includes: FloatRegNode.h

SetReference(*self, pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::FloatRegNode::SetReference(INode** (*virtual void*)
- ***pBase)**

- **Value** (overload SetReference for)

property thisown

The membership flag

`PySpin.GetErrorMessage()` → char const *

`PySpin.GetFiles(FileTemplate, DirectoriesOnly=False)`

Parameters

- **FileTemplate** (*Spinnaker::GenICam::gcstring const &*)
- **DirectoriesOnly** (*bool const*)
- **SPINNAKER_API**
- **&FileTemplate** (*void Spinnaker::GenICam::GetFiles(const gcstring)*)

:param : :param gcstring_vector &FileNames: :param const bool DirectoriesOnly=false): :param Gets a list of files or directories matching a given FileTemplate:

`PySpin.GetGenICamCLProtocolFolder()` → *gcstring*

SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamCLProtocolFolder(void)

Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling SetGenICamCLProtocolFolder(). If GetGenICamCLProtocolFolder() is called before SetGenICamCLProtocolFolder(), it will return the value of environment variable GENICAM_CLPROTOCOL. If this environment variable does not exist, an exception will be thrown.

`PySpin.GetGenICamCacheFolder()` → *gcstring*

SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamCacheFolder(void)

Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling SetGenICamCacheFolder(). If GetGenICamCacheFolder() is called before SetGenICamCacheFolder(), it will return the value of environment variable GENICAM_CACHE_Vx_y. If this environment variable does not exist, an exception will be thrown.

`PySpin.GetGenICamLogConfig()` → *gcstring*

SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamLogConfig(void)

Retrieve the path of the GenICam logging properties file

The path to the logging properties file can be stored by calling SetGenICamLogConfig(). If GetGenICamLogConfig() is called before SetGenICamLogConfig(), it will return the value of environment variable GENICAM_LOG_CONFIG_Vx_y. If this environment variable does not exist, an exception will be thrown.

`PySpin.GetInterfaceName(pBase)` → *gcstring*

Parameters

- **pBase** (*Spinnaker::GenApi::IBase **)
- ***pBase** (*GenICam::gcstring Spinnaker::GenApi::GetInterfaceName(IBase) –*
- **DEPRICATED** (*Returns the name of the main interface as string*)
- **use**
- **instead** (*IBase::GetPrincipalInterfaceType()*)

`PySpin.GetModulePathFromFunction(pFunction) → gcstring`

Parameters

- `pFunction` (`void *`)
- `gcstring` (`SPINNAKER_API`)
- `*pFunction` (`Spinnaker::GenICam::GetModulePathFromFunction(void)–`
- `only` (`true = only subdirectories (ex . and ..) are retrieved; false =`
- `retrieved` (`files are`)
- `given` (`Gets the full path to the module (DLL/SO) containing the`
- `found.` (`pFunction; empty string if not`)

`PySpin.GetValueOfEnvironmentVariable(VariableName) → gcstring`

Parameters

- `VariableName` (`Spinnaker::GenICam::gcstring const &`)
- `GetValueOfEnvironmentVariable(VariableName`
- `bool` (`SPINNAKER_API`)
- `VariableName`
- `VariableContent` (`Spinnaker::GenICam::gcstring &`)
- `bool`
- `gcstring` (`Spinnaker::GenICam::GetValueOfEnvironmentVariable(const)`
- `&VariableName`
- `&VariableContent`) (`gcstring`)
- `environment` (`Retrieve the value of an environment variable true if`)
- `found` (`variable was`)
- `false` (`otherwise`)

class `PySpin.H264Option`

Bases: `object`

Options for saving H264 files.

C++ includes: `SpinVideoDefs.h`

property `bitrate`

property `crf`

property `frameRate`

property `height`

property `reserved`

property `thisown`

The membership flag

property `useMP4`

property width**class** PySpin.**IBase**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::IBase class.

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode**property thisown**

The membership flag

class PySpin.**IBoolean**(*args, **kwargs)Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IBoolean class.

GetValue(self, Verify=False, IgnoreCache=False) → bool**Parameters**

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(self, Value, Verify=True)**Parameters**

- **Value** (bool)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.**ICameraBase**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::ICameraBase class.

BeginAcquisition(self)**DeInit**(self)**DiscoverMaxPacketSize**(self) → unsigned int**EndAcquisition**(self)**ForceIP**(self)**GetAccessMode**(self) → Spinnaker::GenApi::EAccessMode**GetActiveNumDataStreams**(self) → unsigned int**GetBufferOwnership**(self) → Spinnaker::BufferOwnership**GetDeviceID**(self) → *gcstring***GetGuiXml**(self) → *gcstring*

GetNextImage(*self*, grabTimeout=EVENT_TIMEOUT_INFINITE, streamIndex=0) → *ImagePtr*

Parameters

- **grabTimeout** (uint64_t)
- **streamIndex** (uint64_t)

GetNextImageSync(*self*, grabTimeout=EVENT_TIMEOUT_INFINITE) → *ImageList*

Parameters

grabTimeout (uint64_t)

GetNodeMap(*self*) → *INodeMap*

GetNumDataStreams(*self*) → unsigned int

GetNumImagesInUse(*self*) → unsigned int

GetTLDeviceNodeMap(*self*) → *INodeMap*

GetTLStreamNodeMap(*self*, streamIndex) → *INodeMap*

Parameters

streamIndex (uint64_t)

GetUniqueID(*self*) → *gcstring*

GetUserBufferCount(*self*) → uint64_t

GetUserBufferSize(*self*) → uint64_t

GetUserBufferTotalSize(*self*) → uint64_t

Init(*self*)

IsInitialized(*self*) → bool

IsStreaming(*self*) → bool

IsValid(*self*) → bool

ReadPort(*self*, iAddress) → PyObject *

Parameters

iAddress (uint64_t)

RegisterEventHandler(*self*, evtHandlerToRegister)

Parameters

- **evtHandlerToRegister** (Spinnaker::ImageEventHandler &)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **eventName**)
- **evtHandlerToRegister**
- **eventName** (Spinnaker::GenICam::gcstring const &)
- **RegisterEventHandler**(*self*

- **evtHandlerToRegister**
- **streamIndex**)
- **evtHandlerToRegister**
- **streamIndex** (*uint64_t*)

SetBufferOwnership(*self*, *mode*)

Parameters

mode (*enum Spinnaker::BufferOwnership const*)

SetUserBuffers(*self*, *pMemBuffers*, *totalSize*)

Parameters

- **pMemBuffers** (*void *const*)
- **totalSize** (*uint64_t*)
- **SetUserBuffers**(*self*
- **ppMemBuffers** (*void **const*)
- **bufferCount** (*uint64_t const*)
- **bufferSize**)
- **ppMemBuffers**
- **bufferCount**
- **bufferSize** (*uint64_t const*)

property TLDevice

property TLStream

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

WritePort(*self*, *iAddress*, *pBuffer*)

Parameters

- **iAddress** (*uint64_t*)
- **pBuffer** (*uint32_t*)

property thisown

The membership flag

class **PySpin.ICameraList**(*args, **kwargs)

Bases: *object*

Proxy of C++ *Spinnaker::ICameraList* class.

Add(*self*, *camera*)

Parameters

camera (*Spinnaker::CameraPtr*)

```

Append(self, list)

    Parameters
    list (Spinnaker::CameraList const &)

Clear(self)

GetByDeviceID(self, deviceID) → CameraPtr

    Parameters
    deviceID (std::string)

GetByIndex(self, index) → CameraPtr

    Parameters
    index (unsigned int)

GetBySerial(self, serialNumber) → CameraPtr

    Parameters
    serialNumber (std::string)

GetSize(self) → unsigned int

Remove(self, camera)

    Parameters
    camera (Spinnaker::CameraPtr)

RemoveByDeviceID(self, deviceID)

    Parameters
    deviceID (std::string)

RemoveByIndex(self, index)

    Parameters
    index (unsigned int)

RemoveBySerial(self, serialNumber)

    Parameters
    serialNumber (std::string)

property thisown
    The membership flag

class PySpin.ICategory(*args, **kwargs)
    Bases: IValue
    Proxy of C++ Spinnaker::GenApi::ICategory class.

GetFeatures(self)

property thisown
    The membership flag

class PySpin.IChunkData(*args, **kwargs)
    Bases: object
    Proxy of C++ Spinnaker::IChunkData class.

```

GetBlackLevel(*self*) → float64_t

GetCRC(*self*) → int64_t

GetCompressionMode(*self*) → int64_t

GetCompressionRatio(*self*) → float64_t

GetCounterValue(*self*) → int64_t

GetCurrentDataRate(*self*) → int64_t

GetEnable(*self*) → bool

GetEncoderValue(*self*) → int64_t

GetExposureEndLineStatusAll(*self*) → int64_t

GetExposureTime(*self*) → float64_t

GetFrameID(*self*) → int64_t

GetGain(*self*) → float64_t

GetHeight(*self*) → int64_t

GetImage(*self*) → int64_t

GetInferenceBoundingBoxResult(*self*) → *InferenceBoundingBoxResult*

GetInferenceConfidence(*self*) → float64_t

GetInferenceFrameId(*self*) → int64_t

GetInferenceResult(*self*) → int64_t

GetLinePitch(*self*) → int64_t

GetLineStatusAll(*self*) → int64_t

GetModeActive(*self*) → bool

GetOffsetX(*self*) → int64_t

GetOffsetY(*self*) → int64_t

GetPartSelector(*self*) → int64_t

GetPixelDynamicRangeMax(*self*) → int64_t

GetPixelDynamicRangeMin(*self*) → int64_t

GetScan3dAxisMax(*self*) → float64_t

GetScan3dAxisMin(*self*) → float64_t

GetScan3dCoordinateOffset(*self*) → float64_t

GetScan3dCoordinateReferenceValue(*self*) → float64_t

GetScan3dCoordinateScale(*self*) → float64_t

GetScan3dInvalidDataFlag(*self*) → bool
GetScan3dInvalidDataValue(*self*) → float64_t
GetScan3dTransformValue(*self*) → float64_t
GetScanLineSelector(*self*) → int64_t
GetSequencerSetActive(*self*) → int64_t
GetSerialData(*self*) → uint8_t *
GetSerialDataLength(*self*) → int64_t
GetSerialReceiveOverflow(*self*) → bool
GetStreamChannelID(*self*) → int64_t
GetTimerValue(*self*) → float64_t
GetTimestamp(*self*) → int64_t
GetTimestampLatchValue(*self*) → int64_t
GetTransferBlockID(*self*) → int64_t
GetTransferQueueCurrentBlockCount(*self*) → int64_t
GetWidth(*self*) → int64_t
SetChunks(*self*, *pNodeMap*)

Parameters

pNodeMap (*Spinnaker::GenApi::INodeMap &*)

property thisown

The membership flag

class PySpin. **ICommand**(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::ICommand class.

Execute(*self*, *Verify=True*)

Parameters

Verify (*bool*)

IsDone(*self*, *Verify=True*) → bool

Parameters

Verify (*bool*)

property thisown

The membership flag

class PySpin. **IDestroy**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::IDestroy class.

Destroy(*self*)

property thisown

The membership flag

class PySpin.IDeviceArrivalEventHandler(*args, **kwargs)

Bases: [EventHandler](#)

Proxy of C++ Spinnaker::IDeviceArrivalEventHandler class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class PySpin.IDeviceEventHandler(*args, **kwargs)

Bases: [EventHandler](#)

Proxy of C++ Spinnaker::IDeviceEventHandler class.

GetDeviceEventId(*self*) → uint64_t

GetDeviceEventName(*self*) → *gcstring*

OnDeviceEvent(*self*, *eventName*)

Parameters

eventName (*Spinnaker::GenICam::gcstring*)

property thisown

The membership flag

class PySpin.IDeviceInfo(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::IDeviceInfo class.

GetDeviceVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t* &)

GetGenApiVersion(*self*, *Version*, *Build*)

Parameters

• **Version** (*Spinnaker::GenICam::Version_t* &)

• **Build** (*uint16_t* &)

GetModelName(*self*) → *gcstring*

GetProductGuid(*self*) → *gcstring*

GetSchemaVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t* &)

GetStandardNameSpace(*self*) → *gcstring*

GetToolTip(*self*) → *gcstring*

GetVendorName(*self*) → *gcstring*

GetVersionGuid(*self*) → *gcstring*

property thisown

The membership flag

class PySpin.**IDeviceRemovalEventHandler**(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ Spinnaker::IDeviceRemovalEventHandler class.

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class PySpin.**IEnumEntry**(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IEnumEntry class.

GetNumericValue(*self*) → double

GetSymbolic(*self*) → *gcstring*

GetValue(*self*) → int64_t

IsSelfClearing(*self*) → bool

property thisown

The membership flag

class PySpin.**IEnumReference**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::IEnumReference class.

SetEnumReference(*self*, *Index*, *Name*)

Parameters

- **Index** (*int*)
- **Name** (*Spinnaker::GenICam::gcstring*)

SetNumEnums(*self*, *NumEnums*)

Parameters

NumEnums (*int*)

property thisown

The membership flag

class PySpin.IEnumeration(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IEnumeration class.

GetCurrentEntry(self, Verify=False, IgnoreCache=False) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntries(self)

GetEntry(self, IntValue) → *IEnumEntry*

Parameters

IntValue (*int64_t* const)

GetEntryByName(self, Symbolic) → *IEnumEntry*

Parameters

Symbolic (Spinnaker::GenICam::gcstring const &)

GetIntValue(self, Verify=False, IgnoreCache=False) → *int64_t*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetSymbolics(self, Symbolics)

Parameters

Symbolics (Spinnaker::GenApi::StringList_t &)

SetIntValue(self, Value, Verify=True)

Parameters

- **Value** (*int64_t*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.IEnumerationT_AcquisitionModeEnums(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AcquisitionModeEnums > class.

GetCurrentEntry(self, Verify=False, IgnoreCache=False) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AcquisitionModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AcquisitionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AcquisitionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_AcquisitionStatusSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< AcquisitionStatusSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AcquisitionStatusSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AcquisitionStatusSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AcquisitionStatusSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ActionSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ActionSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ActionSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ActionSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ActionSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ActionUnconditionalModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ActionUnconditionalModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ActionUnconditionalModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ActionUnconditionalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ActionUnconditionalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_AdcBitDepthEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AdcBitDepthEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AdcBitDepthEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AdcBitDepthEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::AdcBitDepthEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_AutoAlgorithmSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoAlgorithmSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoAlgorithmSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::AutoAlgorithmSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::AutoAlgorithmSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_AutoExposureControlPriorityEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoExposureControlPriorityEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::AutoExposureControlPriorityEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AutoExposureControlPriorityEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureControlPriorityEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_AutoExposureLightingModeEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< AutoExposureLightingModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::AutoExposureLightingModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AutoExposureLightingModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureLightingModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_AutoExposureMeteringModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoExposureMeteringModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoExposureMeteringModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::AutoExposureMeteringModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureMeteringModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_AutoExposureTargetGreyValueAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoExposureTargetGreyValueAutoEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoExposureTargetGreyValueAutoEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AutoExposureTargetGreyValueAutoEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureTargetGreyValueAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_BalanceRatioSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BalanceRatioSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BalanceRatioSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BalanceRatioSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::BalanceRatioSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_BalanceWhiteAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BalanceWhiteAutoEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BalanceWhiteAutoEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::BalanceWhiteAutoEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::BalanceWhiteAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_BalanceWhiteAutoProfileEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BalanceWhiteAutoProfileEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::BalanceWhiteAutoProfileEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BalanceWhiteAutoProfileEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BalanceWhiteAutoProfileEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_BinningHorizontalModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BinningHorizontalModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::BinningHorizontalModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BinningHorizontalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BinningHorizontalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_BinningSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BinningSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BinningSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BinningSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BinningSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_BinningVerticalModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BinningVerticalModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BinningVerticalModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BinningVerticalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BinningVerticalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_BlackLevelAutoBalanceEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BlackLevelAutoBalanceEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BlackLevelAutoBalanceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BlackLevelAutoBalanceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::BlackLevelAutoBalanceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_BlackLevelAutoEnums**(*args, **kwargs)

Bases: [*IEnumeration*](#), [*IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BlackLevelAutoEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [*IEnumEntry*](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [*IEnumEntry*](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BlackLevelAutoEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::BlackLevelAutoEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::BlackLevelAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_BlackLevelSelectorEnums**(*args, **kwargs)

Bases: [*IEnumeration*](#), [*IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BlackLevelSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [*IEnumEntry*](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::BlackLevelSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BlackLevelSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BlackLevelSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BsiFlatFieldCorrectionAutoEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionAutoEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BsiFlatFieldCorrectionAutoEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BsiFlatFieldCorrectionGainSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkBlackLevelSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkBlackLevelSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkBlackLevelSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkBlackLevelSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkBlackLevelSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_ChunkCounterSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkCounterSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkCounterSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkCounterSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::ChunkCounterSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkEncoderSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkEncoderSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ChunkEncoderSelectorEnums* const)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ChunkEncoderSelectorEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::ChunkEncoderSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkEncoderStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkEncoderStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkEncoderStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkEncoderStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkEncoderStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ChunkExposureTimeSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkExposureTimeSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkExposureTimeSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkExposureTimeSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ChunkExposureTimeSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkGainSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkGainSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkGainSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ChunkGainSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ChunkGainSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkImageComponentEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkImageComponentEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::ChunkImageComponentEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkImageComponentEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkImageComponentEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ChunkPixelFormatEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkPixelFormatEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::ChunkPixelFormatEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkPixelFormatEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ChunkPixelFormatEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkRegionIDEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkRegionIDEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkRegionIDEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ChunkRegionIDEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ChunkRegionIDEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateReferenceSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ChunkScan3dCoordinateSelectorEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dCoordinateSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateSystemEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dCoordinateSystemEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateSystemReferenceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemReferenceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_ChunkScan3dCoordinateTransformSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateTransformSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums

Parameters

- **Verify** (*bool*)

- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkScan3dDistanceUnitEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dDistanceUnitEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dDistanceUnitEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ChunkScan3dDistanceUnitEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dDistanceUnitEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkScan3dOutputModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dOutputModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)

- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dOutputModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dOutputModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dOutputModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ChunkSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkSourceIDEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkSourceIDEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkSourceIDEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkSourceIDEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkSourceIDEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ChunkTimerSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkTimerSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkTimerSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkTimerSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkTimerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ChunkTransferStreamIDEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkTransferStreamIDEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkTransferStreamIDEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkTransferStreamIDEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ChunkTransferStreamIDEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ClConfigurationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ClConfigurationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ClConfigurationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ClConfigurationEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ClConfigurationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ClTimeSlotsCountEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ClTimeSlotsCountEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ClTimeSlotsCountEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ClTimeSlotsCountEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ClTimeSlotsCountEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ColorTransformationSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ColorTransformationSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ColorTransformationSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ColorTransformationSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ColorTransformationSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ColorTransformationValueSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ColorTransformationValueSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ColorTransformationValueSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ColorTransformationValueSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ColorTransformationValueSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ComponentDestinationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ComponentDestinationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ComponentDestinationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ComponentDestinationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ComponentDestinationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_ComponentSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ComponentSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ComponentSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ComponentSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ComponentSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CompressionSaturationPriorityEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CompressionSaturationPriorityEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CompressionSaturationPriorityEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CompressionSaturationPriorityEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CompressionSaturationPriorityEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CounterEventActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterEventActivationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterEventActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterEventActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CounterEventActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_CounterEventSourceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterEventSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterEventSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterEventSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterEventSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CounterResetActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterResetActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterResetActivationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::CounterResetActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterResetActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CounterResetSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterResetSourceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterResetSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterResetSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CounterResetSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_CounterSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CounterStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterStatusEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::CounterStatusEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CounterTriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterTriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterTriggerActivationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterTriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CounterTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_CounterTriggerSourceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CounterTriggerSourceEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterTriggerSourceEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterTriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CxpConnectionTestModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CxpConnectionTestModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpConnectionTestModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::CxpConnectionTestModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CxpConnectionTestModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CxpLinkConfigurationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpLinkConfigurationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CxpLinkConfigurationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CxpLinkConfigurationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_CxpLinkConfigurationPreferredEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationPreferredEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpLinkConfigurationPreferredEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CxpLinkConfigurationPreferredEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CxpLinkConfigurationPreferredEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CxpLinkConfigurationStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpLinkConfigurationStatusEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::CxpLinkConfigurationStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CxpLinkConfigurationStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_CxpPoCxpStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CxpPoCxpStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpPoCxpStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CxpPoCxpStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CxpPoCxpStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_DecimationHorizontalModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DecimationHorizontalModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DecimationHorizontalModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DecimationHorizontalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DecimationHorizontalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DecimationSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DecimationSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DecimationSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DecimationSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DecimationSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DecimationVerticalModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DecimationVerticalModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DecimationVerticalModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DecimationVerticalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DecimationVerticalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_DefectCorrectionModeEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DefectCorrectionModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DefectCorrectionModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DefectCorrectionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DefectCorrectionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeinterlacingEnums**(*args, **kwargs)

Bases: [*IEnumeration*](#), [*IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeinterlacingEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [*IEnumEntry*](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [*IEnumEntry*](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeinterlacingEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeinterlacingEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeinterlacingEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceAccessStatusEnum**(*args, **kwargs)

Bases: [*IEnumeration*](#), [*IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceAccessStatusEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [*IEnumEntry*](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceAccessStatusEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceAccessStatusEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceAccessStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_DeviceCharacterSetEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceCharacterSetEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceCharacterSetEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceCharacterSetEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceCharacterSetEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceClockSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceClockSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceClockSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeviceClockSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceClockSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceConnectionStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceConnectionStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceConnectionStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceConnectionStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceConnectionStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_DeviceCurrentSpeedEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceCurrentSpeedEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceCurrentSpeedEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceCurrentSpeedEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceCurrentSpeedEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceEndiannessMechanismEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceEndiannessMechanismEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceEndiannessMechanismEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeviceEndiannessMechanismEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceEndiannessMechanismEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceIndicatorModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceIndicatorModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceIndicatorModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceIndicatorModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceIndicatorModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_DeviceLinkHeartbeatModeEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceLinkHeartbeatModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceLinkHeartbeatModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceLinkHeartbeatModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceLinkHeartbeatModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceLinkThroughputLimitModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceLinkThroughputLimitModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceLinkThroughputLimitModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceLinkThroughputLimitModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceLinkThroughputLimitModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DevicePowerSupplySelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DevicePowerSupplySelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::DevicePowerSupplySelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DevicePowerSupplySelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DevicePowerSupplySelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_DeviceRegistersEndiannessEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceRegistersEndiannessEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::DeviceRegistersEndiannessEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceRegistersEndiannessEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceRegistersEndiannessEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceScanTypeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceScanTypeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceScanTypeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceScanTypeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceScanTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceSensorChromaEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceSensorChromaEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceSensorChromaEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceSensorChromaEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceSensorChromaEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_DeviceSerialPortBaudRateEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceSerialPortBaudRateEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceSerialPortBaudRateEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceSerialPortBaudRateEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceSerialPortBaudRateEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceSerialPortSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceSerialPortSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceSerialPortSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeviceSerialPortSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceSerialPortSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceStreamChannelEndiannessEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceStreamChannelEndiannessEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceStreamChannelEndiannessEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceStreamChannelEndiannessEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceStreamChannelEndiannessEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_DeviceStreamChannelTypeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceStreamChannelTypeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceStreamChannelTypeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceStreamChannelTypeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceStreamChannelTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceTLTypeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTLTypeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTLTypeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeviceTLTypeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTLTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceTapGeometryEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTapGeometryEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTapGeometryEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceTapGeometryEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTapGeometryEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_DeviceTemperatureSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTemperatureSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTemperatureSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceTemperatureSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTemperatureSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceTypeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTypeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTypeEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::DeviceTypeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_DeviceTypeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTypeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTypeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceTypeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_EncoderModeEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::EncoderModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_EncoderOutputModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderOutputModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderOutputModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::EncoderOutputModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::EncoderOutputModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_EncoderResetActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderResetActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderResetActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderResetActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderResetActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_EncoderResetSourceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderResetSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderResetSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderResetSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::EncoderResetSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_EncoderSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::EncoderSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::EncoderSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_EncoderSourceAEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderSourceAEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderSourceAEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderSourceAEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderSourceAEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_EncoderSourceBEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderSourceBEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderSourceBEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderSourceBEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderSourceBEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_EncoderStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderStatusEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderStatusEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_EventNotificationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EventNotificationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EventNotificationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EventNotificationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EventNotificationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_EventSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< EventSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EventSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EventSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::EventSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ExposureActiveModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExposureActiveModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureActiveModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ExposureActiveModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ExposureActiveModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ExposureAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExposureAutoEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureAutoEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExposureAutoEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ExposureAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ExposureModeEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ExposureModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExposureModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ExposureModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ExposureTimeModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExposureTimeModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureTimeModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ExposureTimeModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ExposureTimeModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ExposureTimeSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExposureTimeSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureTimeSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExposureTimeSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ExposureTimeSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ExternalVoltageSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExternalVoltageSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExternalVoltageSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExternalVoltageSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ExternalVoltageSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_FLIRFilterDriverStatusEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FLIRFilterDriverStatusEnum > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FLIRFilterDriverStatusEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::FLIRFilterDriverStatusEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::FLIRFilterDriverStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_FfcModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FfcModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FfcModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FfcModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::FfcModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_FileOpenModeEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FileOpenModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FileOpenModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FileOpenModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::FileOpenModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_FileOperationSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FileOperationSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FileOperationSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::FileOperationSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::FileOperationSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_FileOperationStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FileOperationStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FileOperationStatusEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FileOperationStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::FileOperationStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_FileSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< FileSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FileSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FileSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::FileSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GUIXMLLocationEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GUIXMLLocationEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GUIXMLLocationEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::GUIXMLLocationEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GUIXMLLocationEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GainAutoBalanceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GainAutoBalanceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GainAutoBalanceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GainAutoBalanceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GainAutoBalanceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_GainAutoEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GainAutoEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GainAutoEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GainAutoEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GainAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GainConversionEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GainConversionEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GainConversionEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::GainConversionEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GainConversionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GainSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GainSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GainSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GainSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GainSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_GenICamXMLLocationEnum(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GenICamXMLLocationEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GenICamXMLLocationEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GenICamXMLLocationEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GenICamXMLLocationEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevCCPEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevCCPEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevCCPEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::GevCCPEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevCCPEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevCCPEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevCCPEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevCCPEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevCCPEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevCCPEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevCurrentPhysicalLinkConfigurationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevCurrentPhysicalLinkConfigurationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::GevCurrentPhysicalLinkConfigurationEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevCurrentPhysicalLinkConfigurationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevGVCPEExtendedStatusCodesSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevGVCPEExtendedStatusCodesSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevGVCPEExtendedStatusCodesSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::GevGVCPEExtendedStatusCodesSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevGVCPEExtendedStatusCodesSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevGVSPExtendedIDModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevGVSPExtendedIDModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)

- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::GevGVSPExtendedIDModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevGVSPExtendedIDModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevGVSPExtendedIDModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GevIEEE1588ClockAccuracyEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::GevIEEE1588ClockAccuracyEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevIEEE1588ClockAccuracyEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588ClockAccuracyEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevIEEE1588ModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevIEEE1588ModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevIEEE1588ModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevIEEE1588ModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588ModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevIEEE1588StatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevIEEE1588StatusEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevIEEE1588StatusEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevIEEE1588StatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588StatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< *GevIEEE1588StatusLatchedEnums* > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevIEEE1588StatusLatchedEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevIEEE1588StatusLatchedEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588StatusLatchedEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevIPConfigurationStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevIPConfigurationStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevIPConfigurationStatusEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::GevIPConfigurationStatusEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevIPConfigurationStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevPhysicalLinkConfigurationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevPhysicalLinkConfigurationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevPhysicalLinkConfigurationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevPhysicalLinkConfigurationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevPhysicalLinkConfigurationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_GevSCPDDirectionEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< *GevSCPDDirectionEnums* > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevSCPDDirectionEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevSCPDDirectionEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevSCPDirectionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_GevSupportedOptionSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevSupportedOptionSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevSupportedOptionSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::GevSupportedOptionSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevSupportedOptionSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ImageComponentSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ImageComponentSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ImageComponentSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ImageComponentSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ImageComponentSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_ImageCompressionJPEGFormatOptionEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ImageCompressionJPEGFormatOptionEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ImageCompressionJPEGFormatOptionEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::ImageCompressionJPEGFormatOptionEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ImageCompressionJPEGFormatOptionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ImageCompressionModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ImageCompressionModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ImageCompressionModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::ImageCompressionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ImageCompressionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_ImageCompressionRateOptionEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ImageCompressionRateOptionEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::ImageCompressionRateOptionEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ImageCompressionRateOptionEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ImageCompressionRateOptionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_InterfaceTypeEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< InterfaceTypeEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::InterfaceTypeEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::InterfaceTypeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::InterfaceTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LUTSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LUTSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LUTSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LUTSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LUTSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LensShadingCoefficientActiveSetEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LensShadingCoefficientActiveSetEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::LensShadingCoefficientActiveSetEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LensShadingCoefficientActiveSetEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LensShadingCoefficientActiveSetEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_LensShadingCorrectionModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LensShadingCorrectionModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::LensShadingCorrectionModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LensShadingCorrectionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::LensShadingCorrectionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LineFormatEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LineFormatEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineFormatEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::LineFormatEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::LineFormatEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LineInputFilterSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LineInputFilterSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::LineInputFilterSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineInputFilterSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LineInputFilterSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_LineModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< LineModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::LineModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::LineModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LineSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LineSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::LineSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::LineSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LineSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LineSourceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineSourceEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LineSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_LogicBlockLUTInputActivationEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTInputActivationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockLUTInputActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTInputActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LogicBlockLUTInputSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTInputSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LogicBlockLUTInputSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTInputSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LogicBlockLUTInputSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTInputSourceEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockLUTInputSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTInputSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_LogicBlockLUTSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< LogicBlockLUTSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockLUTSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_LogicBlockSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LogicBlockSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LogicBlockSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_MultiRoiConfigurationInvalidReasonEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< MultiRoiConfigurationInvalidReasonEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::MultiRoiConfigurationInvalidReasonEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::MultiRoiConfigurationInvalidReasonEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::MultiRoiConfigurationInvalidReasonEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_MultiRoiSelectorEnums**(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< MultiRoiSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::MultiRoiSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::MultiRoiSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::MultiRoiSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_POEStatusEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< POEStatusEnum > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::POEStatusEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::POEStatusEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::POEStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_PixelColorFilterEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< PixelColorFilterEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelFormatEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::PixelFormatEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::PixelFormatEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_PixelFormatEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< PixelFormatEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelFormatEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::PixelFormatEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::PixelFormatEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_PixelFormatInfoSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelFormatInfoSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::PixelFormatInfoSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::PixelFormatInfoSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_PixelSizeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< PixelSizeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelSizeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::PixelSizeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::PixelSizeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_RegionDestinationEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< RegionDestinationEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RegionDestinationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::RegionDestinationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::RegionDestinationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_RegionModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< RegionModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RegionModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::RegionModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::RegionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_RegionSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< RegionSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RegionSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::RegionSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::RegionSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_RgbTransformLightSourceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< RgbTransformLightSourceEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RgbTransformLightSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::RgbTransformLightSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::RgbTransformLightSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateReferenceSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateReferenceSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dCoordinateReferenceSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::Scan3dCoordinateReferenceSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateReferenceSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::Scan3dCoordinateSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dCoordinateSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_Scan3dCoordinateSystemEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSystemEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::Scan3dCoordinateSystemEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dCoordinateSystemEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateSystemEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateSystemReferenceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSystemReferenceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dCoordinateSystemReferenceEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::Scan3dCoordinateSystemReferenceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateSystemReferenceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateTransformSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateTransformSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::Scan3dCoordinateTransformSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::Scan3dCoordinateTransformSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateTransformSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_Scan3dDistanceUnitEnums**(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dDistanceUnitEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::Scan3dDistanceUnitEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dDistanceUnitEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dDistanceUnitEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_Scan3dOutputModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dOutputModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dOutputModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::Scan3dOutputModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dOutputModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SensorDigitizationTapsEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SensorDigitizationTapsEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SensorDigitizationTapsEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SensorDigitizationTapsEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SensorDigitizationTapsEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_SensorShutterModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SensorShutterModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SensorShutterModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SensorShutterModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SensorShutterModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SensorTapsEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SensorTapsEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SensorTapsEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::SensorTapsEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SensorTapsEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SequencerConfigurationModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerConfigurationModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerConfigurationModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerConfigurationModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SequencerConfigurationModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_SequencerConfigurationValidEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerConfigurationValidEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerConfigurationValidEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerConfigurationValidEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SequencerConfigurationValidEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SequencerModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::SequencerModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SequencerModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SequencerSetValidEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerSetValidEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerSetValidEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerSetValidEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SequencerSetValidEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_SequencerTriggerActivationEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SequencerTriggerActivationEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerTriggerActivationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerTriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SequencerTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SequencerTriggerSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerTriggerSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerTriggerSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SequencerTriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SerialPortBaudRateEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortBaudRateEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortBaudRateEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortBaudRateEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortBaudRateEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_SerialPortParityEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SerialPortParityEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortParityEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortParityEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortParityEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SerialPortSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SerialPortSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SerialPortSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_SerialPortStopBitsEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortStopBitsEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortStopBitsEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortStopBitsEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortStopBitsEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SoftwareSignalSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SoftwareSignalSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SoftwareSignalSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::SoftwareSignalSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SoftwareSignalSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_SourceSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SourceSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SourceSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SourceSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SourceSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_StereoResolutionEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< StereoResolutionEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::StereoResolutionEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::StereoResolutionEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::StereoResolutionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_StreamBufferCountModeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< StreamBufferCountModeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::StreamBufferCountModeEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::StreamBufferCountModeEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::StreamBufferCountModeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_StreamBufferHandlingModeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< StreamBufferHandlingModeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::StreamBufferHandlingModeEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::StreamBufferHandlingModeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::StreamBufferHandlingModeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_StreamModeEnum(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< StreamModeEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::StreamModeEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::StreamModeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::StreamModeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_StreamTypeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< StreamTypeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::StreamTypeEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::StreamTypeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::StreamTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TLTypeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TLTypeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::TLTypeEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TLTypeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TLTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_TeledyneGigeVisionFilterDriverStatusEnum(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TeledyneGigeVisionFilterDriverStatusEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::TeledyneGigeVisionFilterDriverStatusEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::TeledyneGigeVisionFilterDriverStatusEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TeledyneGigeVisionFilterDriverStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TestPatternEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TestPatternEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TestPatternEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TestPatternEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TestPatternEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TestPatternGeneratorSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TestPatternGeneratorSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TestPatternGeneratorSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TestPatternGeneratorSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TestPatternGeneratorSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_TimerSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TimerSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TimerSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TimerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TimerStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TimerStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerStatusEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::TimerStatusEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TimerStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TimerTriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TimerTriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerTriggerActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TimerTriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TimerTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_TimerTriggerSourceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TimerTriggerSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerTriggerSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TimerTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TimerTriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TransferComponentSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferComponentSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferComponentSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::TransferComponentSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferComponentSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TransferControlModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferControlModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferControlModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferControlModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferControlModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_TransferOperationModeEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TransferOperationModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferOperationModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferOperationModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferOperationModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TransferQueueModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferQueueModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferQueueModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::TransferQueueModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferQueueModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TransferSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::TransferSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_TransferStatusSelectorEnums`(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TransferStatusSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::TransferStatusSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferStatusSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferStatusSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TransferTriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferTriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerActivationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::TransferTriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TransferTriggerModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferTriggerModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferTriggerModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferTriggerModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_TransferTriggerSelectorEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TransferTriggerSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferTriggerSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferTriggerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TransferTriggerSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferTriggerSourceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerSourceEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::TransferTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferTriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerActivationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_TriggerModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TriggerModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TriggerModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TriggerOverlapEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerOverlapEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerOverlapEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::TriggerOverlapEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TriggerOverlapEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_TriggerSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.IEnumerationT_TriggerSourceEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerSourceEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_U3VCurrentSpeedEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< U3VCurrentSpeedEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::U3VCurrentSpeedEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::U3VCurrentSpeedEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::U3VCurrentSpeedEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_UserOutputSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< UserOutputSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::UserOutputSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::UserOutputSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::UserOutputSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IEnumerationT_UserSetDefaultEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< UserSetDefaultEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::UserSetDefaultEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::UserSetDefaultEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::UserSetDefaultEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_UserSetSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< UserSetSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::UserSetSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::UserSetSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::UserSetSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.**IEnumerationT_WhiteClipSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< WhiteClipSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::WhiteClipSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::WhiteClipSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::WhiteClipSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.IFloat*(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IFloat class.

GetDisplayNotation(*self*) → *Spinnaker::GenApi::EDisplayNotation*

GetDisplayPrecision(*self*) → *int64_t*

GetInc(*self*) → *double*

GetIncMode(*self*) → *Spinnaker::GenApi::EIncMode*

GetListOfValidValues(*self*, *bounded=True*) → *double_autovector_t*

Parameters

bounded (*bool*)

GetMax(*self*) → *double*

GetMin(*self*) → *double*

GetRepresentation(*self*) → *Spinnaker::GenApi::ERepresentation*

GetUnit(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *double*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

HasInc(*self*) → bool

ImposeMax(*self*, *Value*)

Parameters

Value (*double*)

ImposeMin(*self*, *Value*)

Parameters

Value (*double*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*double*)
- **Verify** (*bool*)

property **thisown**

The membership flag

class **PySpin.IImage**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IImage class.

CalculateChannelStatistics(*self*, *channel*) → *ChannelStatistics*

Parameters

channel (enum *Spinnaker::StatisticsChannel*)

CalculateStatistics(*self*, *pStatistics*)

Parameters

pStatistics (*Spinnaker::ImageStatistics &*)

CheckCRC(*self*) → bool

DeepCopy(*self*, *pSrcImage*)

Parameters

pSrcImage (*Spinnaker::ImagePtr const*)

GetBitsPerPixel(*self*) → *size_t*

GetBufferSize(*self*) → *size_t*

GetChunkData(*self*) → *ChunkData*

GetChunkLayoutId(*self*) → *uint64_t*

GetColorProcessing(*self*) → *Spinnaker::ColorProcessingAlgorithm*

GetData(*self*)

GetData(*self*) → *PyObject **

GetDataAbsoluteMax(*self*) → *float*

GetDataAbsoluteMin(*self*) → *float*

GetFrameID(*self*) → uint64_t
GetHeight(*self*) → size_t
GetID(*self*) → uint64_t
GetImagePayloadType(*self*) → Spinnaker::ImagePayloadType
GetImageSize(*self*) → size_t
GetImageStatus(*self*) → Spinnaker::ImageStatus
GetNDArray(*self*) → PyObject *
GetNumChannels(*self*) → size_t
GetPayloadType(*self*) → size_t
GetPixelFormat(*self*) → Spinnaker::PixelFormatEnums
GetPixelFormatIntType(*self*) → Spinnaker::PixelFormatIntType
GetPixelFormatName(*self*) → *gcstring*
GetPrivateData(*self*) → void *
GetStreamIndex(*self*) → uint64_t
GetStride(*self*) → size_t
GetTLPayloadType(*self*) → Spinnaker::TLPayloadType
GetTLPixelFormat(*self*) → uint64_t
GetTLPixelFormatNamespace(*self*) → Spinnaker::TLPixelFormatNamespace
GetTimeStamp(*self*) → uint64_t
GetValidPayloadSize(*self*) → size_t
GetWidth(*self*) → size_t
GetXOffset(*self*) → size_t
GetXPadding(*self*) → size_t
GetYOffset(*self*) → size_t
GetYPadding(*self*) → size_t
HasCRC(*self*) → bool
HasChunkData(*self*) → bool
IsCompressed(*self*) → bool
IsInUse(*self*) → bool
IsIncomplete(*self*) → bool
Release(*self*)

ResetImage(*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

Parameters

- **width** (*size_t*)
- **height** (*size_t*)
- **offsetX** (*size_t*)
- **offsetY** (*size_t*)
- **pixelFormat** (*enum Spinnaker::PixelFormatEnums*)
- **ResetImage**(*self*
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**)
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*void **)
- **ResetImage**(*self*
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType** (*enum Spinnaker::TLPayloadType*)
- **dataSize**)
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType**

- `dataSize (size_t)`

Save(*self*, *pFilename*, *format*=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT)

Parameters

- `pFilename (char const *)`
- `format (enum Spinnaker::ImageFileFormat)`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption (Spinnaker::BMPOption &)`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`

- **pFilename**
- **pOption)**
- **pFilename**
- **pOption**
- **Save(self**
- **pFilename**
- **option)**
- **pFilename**
- **option** (*Spinnaker::SIOption &*)

property thisown
The membership flag

class PySpin.**IImageEventHandler**(*args, **kwargs)
Bases: *EventHandler*
Proxy of C++ Spinnaker::IImageEventHandler class.

property thisown
The membership flag

class PySpin.**IImageList**(*args, **kwargs)
Bases: object
Proxy of C++ Spinnaker::IImageList class.

Add(self, image)

Parameters
image (*Spinnaker::ImagePtr*)

Append(self, list)

Parameters
list (*Spinnaker::ImageList const &*)

Clear(self)

GetByIndex(self, index) → ImagePtr

Parameters
index (*unsigned int*)

GetByPayloadType(self, payloadType) → ImagePtr

Parameters
payloadType (*enum Spinnaker::ImagePayloadType const*)

GetByPixelFormat(self, pixelFormat) → ImagePtr

Parameters
pixelFormat (*enum Spinnaker::PixelFormatEnums*)

GetByStreamIndex(self, streamIndex) → ImagePtr

Parameters
streamIndex (*uint64_t const*)

GetSize(*self*) → unsigned int

Release(*self*)

RemoveByIndex(*self*, *index*)

Parameters

index (unsigned int)

RemoveByPayloadType(*self*, *payloadType*)

Parameters

payloadType (enum Spinnaker::ImagePayloadType const)

RemoveByPixelFormat(*self*, *pixelFormat*)

Parameters

pixelFormat (enum Spinnaker::PixelFormatEnums)

RemoveByStreamIndex(*self*, *streamIndex*)

Parameters

streamIndex (uint64_t const)

Save(*self*, *filename*)

Parameters

filename (char const *)

property thisown

The membership flag

class PySpin.IImageListEventHandler(*args, **kwargs)

Bases: [EventHandler](#)

Proxy of C++ Spinnaker::IImageListEventHandler class.

property thisown

The membership flag

class PySpin.IImageProcessor(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IImageProcessor class.

ApplyGamma(*self*, *srcImage*, *gamma*, *isGammaInverse=False*) → *ImagePtr*

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **gamma** (float)
- **isGammaInverse** (bool)
- **ApplyGamma**(*self*
- **srcImage**
- **destImage** (Spinnaker::ImagePtr &)
- **gamma**
- **isGammaInverse=False**)

- **srcImage**
- **destImage**
- **gamma**
- **isGammaInverse**

Convert(*self*, *srcImage*, *destFormat*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **destFormat** (*enum Spinnaker::PixelFormatEnums*)
- **Convert**(*self*
- **srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)
- **destFormat**)
- **srcImage**
- **destImage**
- **destFormat**
- **Convert**(*self*
- **srcImageList** (*Spinnaker::ImageList const &*)
- **ImagePtr** (*destFormat*) ->)
- **srcImageList**
- **destFormat**
- **Convert**(*self*
- **srcImageList**
- **destImage**
- **destFormat**)
- **srcImageList**
- **destImage**
- **destFormat**

GetColorProcessing(*self*) → *Spinnaker::ColorProcessingAlgorithm*

GetNumDecompressionThreads(*self*) → unsigned int

SetColorProcessing(*self*, *colorAlgorithm*)

Parameters

- **colorAlgorithm** (*enum Spinnaker::ColorProcessingAlgorithm*)

SetNumDecompressionThreads(*self*, *numThreads*)

Parameters

- **numThreads** (*unsigned int*)

property thisown

The membership flag

class PySpin.IInteger(*args, **kwargs)Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IInteger class.

GetInc(self) → int64_t**GetIncMode**(self) → Spinnaker::GenApi::EIncMode**GetListOfValidValues**(self, bounded=True) → *int64_autovector_t***Parameters****bounded** (bool)**GetMax**(self) → int64_t**GetMin**(self) → int64_t**GetRepresentation**(self) → Spinnaker::GenApi::ERepresentation**GetUnit**(self) → *gcstring***GetValue**(self, Verify=False, IgnoreCache=False) → int64_t**Parameters**

- **Verify** (bool)
- **IgnoreCache** (bool)

ImposeMax(self, Value)**Parameters****Value** (int64_t)**ImposeMin**(self, Value)**Parameters****Value** (int64_t)**SetValue**(self, Value, Verify=True)**Parameters**

- **Value** (int64_t)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.IInterface(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IInterface class.

GetCameras(self, updateCameras=True) → *CameraList***Parameters****updateCameras** (bool)

GetTLNodeMap(*self*) → *INodeMap*

IsCameraInUse(*self*) → bool

IsValid(*self*) → bool

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

evtHandlerToRegister (*Spinnaker::EventHandler* &)

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime*=0, *requestAck*=False, *pResultSize*=None, *results*=0)

Parameters

- **deviceKey** (*unsigned int*)
- **groupKey** (*unsigned int*)
- **groupMask** (*unsigned int*)
- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)

property **TLInterface**

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler* &)

UpdateCameras(*self*) → bool

property **thisown**

The membership flag

class *PySpin.IInterfaceArrivalEventHandler*(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ *Spinnaker::IInterfaceArrivalEventHandler* class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property **thisown**

The membership flag

class *PySpin.IInterfaceEventHandler*(*args, **kwargs)

Bases: *IDeviceArrivalEventHandler*, *IDeviceRemovalEventHandler*

Proxy of C++ *Spinnaker::IInterfaceEventHandler* class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class *PySpin.IInterfaceList*(*args, **kwargs)

Bases: *object*

Proxy of C++ Spinnaker::IInterfaceList class.

Add(*self*, *iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

Append(*self*, *list*)

Parameters

list (*Spinnaker::InterfaceList const **)

Clear(*self*)

GetByIndex(*self*, *index*) → *InterfacePtr*

Parameters

index (*unsigned int*)

GetSize(*self*) → *unsigned int*

Remove(*self*, *iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class *PySpin.IInterfaceRemovalEventHandler*(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ Spinnaker::IInterfaceRemovalEventHandler class.

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class *PySpin.ILoggingEventHandler*(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ Spinnaker::ILoggingEventHandler class.

OnLogEvent(*self*, *eventPtr*)

Parameters

eventPtr (*Spinnaker::LoggingEventDataPtr*)

property thisown

The membership flag

class PySpin.INode(*args, **kwargs)

Bases: *ISelector*, *IReference*

Proxy of C++ Spinnaker::GenApi::INode class.

DeregisterCallback(self, hCallback) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

GetAlias(self) → *INode*

GetCachingMode(self) → Spinnaker::GenApi::ECachingMode

GetCastAlias(self) → *INode*

GetChildren(self, Children, LinkType=ctReadingChildren)

Parameters

- **Children** (Spinnaker::GenApi::NodeList_t &)
- **LinkType** (enum Spinnaker::GenApi::ELinkType)

GetDescription(self) → *gcstring*

GetDeviceName(self) → *gcstring*

GetDisplayName(self) → *gcstring*

GetDocuURL(self) → *gcstring*

GetEventID(self) → *gcstring*

GetLockNodes(self, LockNodes)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetName(self, FullQualified=False) → *gcstring*

Parameters

FullQualified (bool)

GetNameSpace(self) → Spinnaker::GenApi::ENamespace

GetNodeMap(self) → *INodeMap*

GetParents(self, Parents)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(self) → int64_t

GetPrincipalInterfaceType(self) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const &*)
- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **AttributeStr** (*Spinnaker::GenICam::gcstring &*)

GetPropertyNames(*self*)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsStreamable(*self*) → bool

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

property thisown

The membership flag

class *PySpin.INodeMap*(*args, **kwargs)

Bases: object

Proxy of C++ *Spinnaker::GenApi::INodeMap* class.

Connect(*self*, *pPort*, *PortName*) → bool

Parameters

- **pPort** (*IPort **)
- **PortName** (*Spinnaker::GenICam::gcstring const &*)
- **Connect**(*self*
- **bool** (*pPort*) →)
- **pPort**

GetDeviceName(*self*) → *gcstring*

GetNode(*self*, *Name*) → *INode*

Parameters

Name (*Spinnaker::GenICam::gcstring const &*)

GetNodes(*self*)

GetNumNodes(*self*) → *uint64_t*

InvalidateNodes(*self*)

Poll(*self*, *ElapsedTime*)

Parameters

ElapsedTime (*int64_t*)

property thisown

The membership flag

class **PySpin.INodeMapDyn**(*args, **kwargs)

Bases: *INodeMap*

Proxy of C++ Spinnaker::GenApi::INodeMapDyn class.

ClearAllNodes(*self*)

ExtractIndependentSubtree(*self*, *XMLData*, *InjectXMLData*, *SubTreeRootNodeName*, *ExtractedSubtree*)

Parameters

- **XMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **SubTreeRootNodeName** (*Spinnaker::GenICam::gcstring const &*)
- **ExtractedSubtree** (*Spinnaker::GenICam::gcstring &*)

GetSupportedSchemaVersions(*self*)

LoadXMLFromFile(*self*, *FileName*)

Parameters

FileName (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromFileInject(*self*, *TargetFileName*, *InjectFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectFileName** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromString(*self*, *XMLData*)

Parameters

XMLData (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromStringInject (*self*, *TargetXMLData*, *InjectXMLData*)

Parameters

- **TargetXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromZIPData (*self*, *zipData*, *zipSize*)

Parameters

- **zipData** (*void const **)
- **zipSize** (*size_t*)

LoadXMLFromZIPFile (*self*, *ZipFileName*)

Parameters

- **ZipFileName** (*Spinnaker::GenICam::gcstring const &*)

MergeXMLFiles (*self*, *TargetFileName*, *InjectedFileName*, *OutputFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectedFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)

PreprocessXMLFromFile (*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*,
XMLValidation=xvDefault)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

PreprocessXMLFromZIPFile (*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*,
XMLValidation=xvDefault)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

property thisown

The membership flag

class **PySpin.IPersistScript** (**args*, ***kwargs*)

Bases: **object**

Proxy of C++ **Spinnaker::GenApi::IPersistScript** class.

PersistFeature(*self*, *item*)

Parameters

item (*Spinnaker::GenApi::IValue* &)

SetInfo(*self*, *Info*)

Parameters

Info (*Spinnaker::GenICam::gcstring* &)

property thisown

The membership flag

class *PySpin.IPointCloud*(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IPointCloud class.

AddPoint(*self*, *point*)

Parameters

point (*Spinnaker::Stereo3DPoint* const)

GetNumPoints(*self*) → size_t

GetPoint(*self*, *index*) → *Stereo3DPoint*

Parameters

index (*unsigned int* const)

GetPointCloudData(*self*) → *Spinnaker::IPointCloud::PointCloudData* *

LoadPointCloudFromPly(*self*, *filename*)

Parameters

filename (*std::string* const &)

PrintPoints(*self*, *numPointsToPrint*)

Parameters

numPointsToPrint (*unsigned int*)

SavePointCloudAsPly(*self*, *arg2*)

Parameters

arg2 (*std::string* const &)

property thisown

The membership flag

class *PySpin.IReference*(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::IReference class.

SetReference(*self*, *pBase*)

Parameters

pBase (*INode* *)

property thisown

The membership flag


```
class PySpin.IRegister(*args, **kwargs)
```

Bases: [IValue](#)

Proxy of C++ Spinnaker::GenApi::IRegister class.

```
Get(self, pBuffer, Verify=False, IgnoreCache=False)
```

Parameters

- **pBuffer** (`uint8_t *`)
- **Verify** (`bool`)
- **IgnoreCache** (`bool`)

Gets a NumPy array representing the contents of the register, as 8-bit unsigned ints.

6.3 Parameters:

pBuffer: 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)

```
GetAddress(self) → int64_t
```

```
GetLength(self) → int64_t
```

```
Set(self, pBuffer, Verify=True)
```

Parameters

- **pBuffer** (`uint8_t const *`)
- **Verify** (`bool`)

Set the register's contents with the contents (as 8-bit unsigned ints) of the given array.

6.4 Parameters:

pBuffer: The NumPy array containing the data to set

Verify: Enables AccessMode and Range verification (default = true)

property thisown

The membership flag

```
class PySpin.ISelector(*args, **kwargs)
```

Bases: [IBase](#)

Proxy of C++ Spinnaker::GenApi::ISelector class.

```
GetSelectedFeatures(self, arg2)
```

Parameters

arg2 (`FeatureList_t &`)

```
GetSelectingFeatures(self, arg2)
```

Parameters

arg2 (`FeatureList_t &`)

IsSelector(*self*) → bool

property thisown

The membership flag

class PySpin.**ISectorDigit**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::ISectorDigit class.

GetSelectorList(*self*, *Incremental=False*)

Parameters

Incremental (bool)

Restore(*self*)

SetFirst(*self*) → bool

SetNext(*self*, *Tick=True*) → bool

Parameters

Tick (bool)

ToString(*self*) → *gcstring*

property thisown

The membership flag

class PySpin.**IString**(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IString class.

GetMaxLength(*self*) → int64_t

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.**ISystem**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::ISystem class.

GetCameras(*self*, *updateInterfaces=True*, *updateCameras=True*) → *CameraList*

Parameters

- **updateInterfaces** (*bool*)
- **updateCameras** (*bool*)

GetInterfaces(*self*, *updateInterface=True*) → *InterfaceList*

Parameters

updateInterface (*bool*)

GetLibraryVersion(*self*) → *LibraryVersion*

GetLoggingEventPriorityLevel(*self*) → *Spinnaker::SpinnakerLogLevel*

GetTLNodeMap(*self*) → *INodeMap*

IsInUse(*self*) → *bool*

RegisterEventHandler(*self*, *evtHandlerToRegister*, *updateInterface=False*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::EventHandler &*)
- **updateInterface** (*bool*)

RegisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler &*)

ReleaseInstance(*self*)

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *requestAck=False*, *pResultSize=None*, *results=0*)

Parameters

- **deviceKey** (*unsigned int*)
- **groupKey** (*unsigned int*)
- **groupMask** (*unsigned int*)
- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)

SetLoggingEventPriorityLevel(*self*, *level*)

Parameters

level (*enum Spinnaker::SpinnakerLogLevel*)

property *TLSystem*

UnregisterAllLoggingEventHandlers(*self*)

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler* &)

UnregisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler* &)

UpdateCameras(*self*, *updateInterfaces=True*) → bool

Parameters

updateInterfaces (*bool*)

UpdateInterfaceList(*self*)

property thisown

The membership flag

class *PySpin.ISystemEventHandler*(*args, **kwargs)

Bases: *IInterfaceArrivalEventHandler*, *IInterfaceRemovalEventHandler*

Proxy of C++ Spinnaker::ISystemEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class *PySpin.IValue*(*args, **kwargs)

Bases: *INode*

Proxy of C++ Spinnaker::GenApi::IValue class.

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

• **ValueStr** (*Spinnaker::GenICam::gcstring* const &)

• **Verify** (*bool*)

GetNode(*self*) → *INode*

IsValueCacheValid(*self*) → bool

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

• **Verify** (*bool*)

• **IgnoreCache** (*bool*)

property thisown

The membership flag

class PySpin.**Image**(*args, **kwargs)

Bases: *IImage*

The image object class.

C++ includes: Image.h

CheckCRC(self) → bool

bool Spinnaker::Image::CheckCRC() const

Checks if the computed checksum matches with chunk data's ImageCRC

Returns true if computed checksum matches with the chunk data's CRC and false otherwise.

static Create() → *ImagePtr*

static Create(image) → *ImagePtr*

Parameters

- **image** (*Spinnaker::ImagePtr const*)
- **Create**(width
- **height** (*size_t*)
- **offsetX** (*size_t*)
- **offsetY** (*size_t*)
- **pixelFormat** (*enum Spinnaker::PixelFormatEnums*)
- **ImagePtr** (*copied from another*)
- **width** (*or using*)
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*void **)
- **Create**(width
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType** (*enum Spinnaker::TLPayloadType*)
- **ImagePtr**
- **width**
- **height**
- **offsetX**

- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType**
- **dataSize** (*size_t*)
- **object** (*Creates a new Image*)
- **constructor** (*either using a default*)
- **ImagePtr**
- **width**
- **height**

:param : :param offset_x: :param offset_y: :param pixel format: :param and a NumPy array containing 8-bit unsigned ints representing the image data: :param (replaces the void* pData argument).:

DeepCopy(*self*, *pSrcImage*)

Parameters

- **pSrcImage** (*The Image to copy the data from.*)
- **void**
- **pSrcImage** (*Spinnaker::Image::DeepCopy(const ImagePtr)*)
- **operation** (*Performs a deep copy of the Image. After this*)
- **image** (*the*)
- **not** (*contents and member variables will be the same. The Images will*)
- **released.** (*share a buffer. The Image's current buffer will not be*)
- **Parameters**
- -----
- **pSrcImage**

GetBitsPerPixel(*self*) → *size_t*

size_t Spinnaker::Image::GetBitsPerPixel() const

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.

The number of bits used per pixel.

GetBufferSize(*self*) → *size_t*

size_t Spinnaker::Image::GetBufferSize() const

Gets the size of the buffer associated with the image in bytes.

The size of the buffer, in bytes.

GetChunkData(*self*) → *ChunkData*

const *ChunkData*& Spinnaker::Image::GetChunkData() const

Returns a pointer to a chunk data interface. No ownership is transferred, the chunk data interface reference is valid until Image::Release() is called on this image.

ChunkData interface that provides access to image chunks.

GetChunkLayoutId(*self*) → uint64_t

uint64_t Spinnaker::Image::GetChunkLayoutId() const

Returns the id of the chunk data layout.

uint64_t value representing the id of the chunk data layout.

GetColorProcessing(*self*) → Spinnaker::ColorProcessingAlgorithm

ColorProcessingAlgorithm Spinnaker::Image::GetColorProcessing() const

Gets the algorithm used to produce the image.

See: Convert()

The color processing algorithm used to produce the image.

GetDataAbsoluteMax(*self*) → float

GetDataAbsoluteMin(*self*) → float

GetFrameID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetFrameID() const

Gets the frame ID for this image.

The frame ID.

GetHeight(*self*) → size_t

size_t Spinnaker::Image::GetHeight() const

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.

The height in pixels.

GetID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetID() const

Gets a unique ID for this image. Each image in a stream will have a unique ID to help identify it.

The 64 bit unique id for this image.

GetImagePayloadType(*self*) → Spinnaker::ImagePayloadType

GetImageSize(*self*) → size_t

size_t Spinnaker::Image::GetImageSize() const

Returns the size of the image

The image size in bytes.

GetImageStatus(*self*) → Spinnaker::ImageStatus

ImageStatus Spinnaker::Image::GetImageStatus() const

Returns data integrity status of the image returned from GetNextImage()

Returns whether image has any data integrity issues.

static GetImageStatusDescription(*status*) → char const *

Parameters

status (enum Spinnaker::ImageStatus)

GetNumChannels(*self*) → size_t

GetPayloadType(*self*) → size_t

size_t Spinnaker::Image::GetPayloadType() const

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.

Device types specific payload type.

GetPixelFormat(*self*) → Spinnaker::PixelFormatEnums

Spinnaker::PixelFormatEnums Spinnaker::Image::GetPixelFormat() const

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: Convert()

enum value representing the PixelFormat.

GetPixelFormatIntType(*self*) → Spinnaker::PixelFormatIntType

GetPixelFormatName(*self*) → *gcstring*

GenICam::gcstring Spinnaker::Image::GetPixelFormatName() const

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.

string value representing the PixelFormat.

GetPrivateData(*self*) → void *

void* Spinnaker::Image::GetPrivateData() const

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.

A pointer to the user passed data pointer.

GetStreamIndex(*self*) → uint64_t

GetStride(*self*) → size_t

size_t Spinnaker::Image::GetStride() const

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.

The stride in bytes.

GetTLPayloadType(*self*) → Spinnaker::TLPayloadType

PayloadTypeInfoIDs Spinnaker::Image::GetTLPayloadType() const

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.

Transport Layer specific payload type.

GetTLPixelFormat(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTLPixelFormat() const

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: GetTLPixelFormatNamespace()

Transport Layer specific pixel format.

GetTLPixelFormatNamespace(*self*) → Spinnaker::TLPixelFormatNamespace

PixelFormatNamespaceID Spinnaker::Image::GetTLPixelFormatNamespace() const

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: GetTLPixelFormat()

enum value representing the PixelFormatNamespace.

GetTimeStamp(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTimeStamp() const

Gets the time stamp for the image in nanoseconds.

The time stamp of the image.

GetValidPayloadSize(*self*) → size_t

size_t Spinnaker::Image::GetValidPayloadSize() const

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: GetBufferSize()

size_t value representing valid payload.

GetWidth(*self*) → size_t

size_t Spinnaker::Image::GetWidth() const

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.

The width in pixels.

GetXOffset(*self*) → size_t

size_t Spinnaker::Image::GetXOffset() const

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.

The x offset in pixels.

GetXPadding(*self*) → size_t

size_t Spinnaker::Image::GetXPadding() const

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.

The x padding in bytes.

GetYOffset(*self*) → size_t

size_t Spinnaker::Image::GetYOffset() const

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.

The y offset in pixels.

GetYPadding(*self*) → size_t

size_t Spinnaker::Image::GetYPadding() const

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.

The y padding in bytes.

HasCRC(*self*) → bool

bool Spinnaker::Image::HasCRC() const

Checks if the image contains ImageCRC checksum from chunk data

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

HasChunkData(*self*) → bool

IsCompressed(*self*) → bool

IsInUse(*self*) → bool

bool Spinnaker::Image::IsInUse()

Returns true if the image is still in use by the stream

Returns true if the image is in use and false otherwise.

IsIncomplete(*self*) → bool

bool Spinnaker::Image::IsIncomplete() const

Returns a boolean value indicating if this image was incomplete. An image is marked as incomplete if the transport layer received less data than it requested.

Returns true if image is incomplete, false otherwise.

static Load(*pFilename*, *format*=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT) → *ImagePtr*

Parameters

- **pFilename** (*char const **)
- **format** (*enum Spinnaker::ImageFileFormat*)

Release(*self*)

void Spinnaker::Image::Release()

ResetImage(*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

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.*)
- `ResetImage(self`
 - `width`
 - `height`
 - `offsetX`
 - `offsetY`
 - `pixelFormat`
 - `pData`)
 - `width`
 - `height`
 - `offsetX`
 - `offsetY`
 - `pixelFormat`
 - `pData` (*Pointer to the image buffer.*)
- `ResetImage(self`
 - `width`
 - `height`
 - `offsetX`
 - `offsetY`
 - `pixelFormat`
 - `pData`
 - `dataPayloadType` (*enum Spinnaker::TLPayloadType*)
 - `dataSize`)
 - `width`
 - `height`
 - `offsetX`
 - `offsetY`
 - `pixelFormat`
 - `pData`
 - `dataPayloadType`
 - `dataSize` (*size_t*)
 - `void`
 - `width`
 - `height`
 - `size_t`

- `offsetX`
- `offsetY`
- `pixelFormat`
- `void`
- `*pData`)
- `object.` (*Sets new dimensions of the image*)
- `Parameters`
- `-----`
- `width`
- `height`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`

Save(*self*, *pFilename*, *format*=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT)

Parameters

- `pFilename` (*Filename to save image with.*)
- `format` (*enum Spinnaker::ImageFileFormat*)
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption` (*Options to use while saving image.*)
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`

- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- void
- *pFilename (Spinnaker::Image::Save(const char) –
- &pOption) ([BMPOption](#))
- specified. (Saves the image to the specified file name with the options)
- Parameters
- -----
- pFilename
- pOption

property thisown

The membership flag

class PySpin.ImageEventHandler

Bases: [IImageEventHandler](#)

Proxy of C++ Spinnaker::ImageEventHandler class.

OnImageEvent(*self*, *image*)

Parameters

image (*Spinnaker::ImagePtr*)

property **thisown**

The membership flag

class *PySpin.ImageList*(*args)

Bases: *IImageList*

Proxy of C++ *Spinnaker::ImageList* class.

Add(*self*, *image*)

Parameters

image (*Spinnaker::ImagePtr*)

Append(*self*, *list*)

Parameters

list (*Spinnaker::ImageList const &*)

Clear(*self*)

GetByIndex(*self*, *index*) → *ImagePtr*

Parameters

index (*unsigned int*)

GetByPayloadType(*self*, *payloadType*) → *ImagePtr*

Parameters

payloadType (*enum Spinnaker::ImagePayloadType const*)

GetByPixelFormat(*self*, *pixelFormat*) → *ImagePtr*

Parameters

pixelFormat (*enum Spinnaker::PixelFormatEnums*)

GetByStreamIndex(*self*, *streamIndex*) → *ImagePtr*

Parameters

streamIndex (*uint64_t const*)

GetSize(*self*) → *unsigned int*

static Load(*filename*) → *ImageList*

Parameters

filename (*char const **)

Release(*self*)

RemoveByIndex(*self*, *index*)

Parameters

index (*unsigned int*)

RemoveByPayloadType(*self*, *payloadType*)

Parameters

payloadType (*enum Spinnaker::ImagePayloadType const*)

RemoveByPixelFormat(*self*, *pixelFormat*)

Parameters

pixelFormat (enum *Spinnaker::PixelFormatEnums*)

RemoveByStreamIndex(*self*, *streamIndex*)

Parameters

streamIndex (uint64_t const)

Save(*self*, *filename*)

Parameters

filename (char const *)

property thisown

The membership flag

class *PySpin.ImageListEventHandler*

Bases: *IImageListEventHandler*

Proxy of C++ *Spinnaker::ImageListEventHandler* class.

OnImageListEvent(*self*, *imageList*)

Parameters

imageList (*Spinnaker::ImageList*)

property thisown

The membership flag

PySpin.ImageList_Load(*filename*) → *ImageList*

Parameters

filename (char const *)

class *PySpin.ImagePixel*

Bases: object

Proxy of C++ *Spinnaker::ImagePixel* class.

property thisown

The membership flag

property u

property v

class *PySpin.ImageProcessor*(*args)

Bases: *IImageProcessor*

Proxy of C++ *Spinnaker::ImageProcessor* class.

ApplyGamma(*self*, *srcImage*, *gamma*, *applyGammaInverse=False*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr* const &)
- **gamma** (float)
- **applyGammaInverse** (bool)
- **ApplyGamma**(*self*

- `srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `gamma`
- `applyGammaInverse=False)`
- `srcImage`
- `destImage`
- `gamma`
- `applyGammaInverse`

Convert(*self*, *srcImage*, *destFormat*) → *ImagePtr*

Parameters

- `srcImage (Spinnaker::ImagePtr const &)`
- `destFormat (enum Spinnaker::PixelFormatEnums)`
- `Convert(self`
- `srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `destFormat)`
- `srcImage`
- `destImage`
- `destFormat`
- `Convert(self`
- `srcImageList (Spinnaker::ImageList const &)`
- `ImagePtr (destFormat) ->)`
- `srcImageList`
- `destFormat`
- `Convert(self`
- `srcImageList`
- `destImage`
- `destFormat)`
- `srcImageList`
- `destImage`
- `destFormat`

GetColorProcessing(*self*) → `Spinnaker::ColorProcessingAlgorithm`

GetNumDecompressionThreads(*self*) → unsigned int

SetColorProcessing(*self*, *colorAlgorithm*)

Parameters

colorAlgorithm (enum *Spinnaker::ColorProcessingAlgorithm*)

SetNumDecompressionThreads(*self*, *numThreads*)

Parameters

numThreads (unsigned int)

property thisown

The membership flag

class *PySpin.ImagePtr*(*args)

Bases: *_SWIG_ImgPtr*

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.

C++ includes: *ImagePtr.h*

property thisown

The membership flag

class *PySpin.ImageUtility*

Bases: *object*

Proxy of C++ *Spinnaker::ImageUtility* class.

static CreateNormalized(*srcImage*, *destPixelFormat*, *srcDataRange*=*SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE*)
→ *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **destPixelFormat** (enum *Spinnaker::PixelFormatEnums const*)
- **srcDataRange** (enum *Spinnaker::SourceDataRange*)
- **CreateNormalized**(*srcImage*
- **min** (*double const*)
- **max** (*double const*)
- **ImagePtr** (*srcDataRange*=*SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE*)
→)
- **srcImage**
- **min**
- **max**
- **srcDataRange**
- **CreateNormalized**(*srcImage*
- **min**
- **max**
- **destPixelFormat**

- `ImagePtr`
- `srcImage`
- `min`
- `max`
- `destPixelFormat`
- `srcDataRange`
- `CreateNormalized(srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `srcDataRange`
- `CreateNormalized(srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange`

static `CreateScaled(srcImage, scalingAlg, scalingFactor) → ImagePtr`

Parameters

- `srcImage (Spinnaker::ImagePtr const &)`
- `scalingAlg (enum Spinnaker::ImageScalingAlgorithm)`
- `scalingFactor (double)`
- `CreateScaled(srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `scalingAlg`
- `scalingFactor)`
- `srcImage`
- `destImage`
- `scalingAlg`
- `scalingFactor`

property thisown

The membership flag

class PySpin.ImageUtilityCCM

Bases: object

Proxy of C++ Spinnaker::ImageUtilityCCM class.

static ApplicationToString(application) → std::string

Parameters

application (Spinnaker::CCMApplication const &)

static ColorSpaceToString(colorSpace) → std::string

Parameters

colorSpace (Spinnaker::CCMColorSpace const &)

static ColorTemperatureToString(colorTemperature) → std::string

Parameters

colorTemperature (Spinnaker::CCMColorTemperature const &)

static CreateColorCorrected(srcImage, settings) → ImagePtr

Parameters

- srcImage (Spinnaker::ImagePtr const &)
- settings (Spinnaker::CCMSettings const &)
- CreateColorCorrected(srcImage
- destImage (Spinnaker::ImagePtr &)
- settings)
- srcImage
- destImage
- settings

static EncryptColorCorrectionMatrix(ccmMatrixEntries) → std::string

Parameters

ccmMatrixEntries (std::string)

static SensorToString(sensor) → std::string

Parameters

sensor (Spinnaker::CCMSensor const &)

static TypeToString(type) → std::string

Parameters

type (Spinnaker::CCMType const &)

property thisown

The membership flag

PySpin.ImageUtilityCCM_ApplicationToString(application) → std::string

Parameters

application (Spinnaker::CCMApplication const &)

`PySpin.ImageUtilityCCM_ColorSpaceToString(colorSpace) → std::string`

Parameters

`colorSpace` (*Spinnaker::CCMColorSpace const &*)

`PySpin.ImageUtilityCCM_ColorTemperatureToString(colorTemperature) → std::string`

Parameters

`colorTemperature` (*Spinnaker::CCMColorTemperature const &*)

`PySpin.ImageUtilityCCM_CreateColorCorrected(srcImage, settings) → ImagePtr`

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `settings` (*Spinnaker::CCMSettings const &*)
- `ImageUtilityCCM_CreateColorCorrected(srcImage`
- `destImage` (*Spinnaker::ImagePtr &*)
- `settings`)
- `srcImage`
- `destImage`
- `settings`

`PySpin.ImageUtilityCCM_EncryptColorCorrectionMatrix(ccmMatrixEntries) → std::string`

Parameters

`ccmMatrixEntries` (*std::string*)

`PySpin.ImageUtilityCCM_SensorToString(sensor) → std::string`

Parameters

`sensor` (*Spinnaker::CCMSensor const &*)

`PySpin.ImageUtilityCCM_TypeToString(type) → std::string`

Parameters

`type` (*Spinnaker::CCMType const &*)

class `PySpin.ImageUtilityHeatmap`

Bases: `object`

Proxy of C++ `Spinnaker::ImageUtilityHeatmap` class.

static `CreateHeatmap(srcImage) → ImagePtr`

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `CreateHeatmap(srcImage`
- `min` (*float const*)
- `max` (*float const*)
- `lowColor` (*enum Spinnaker::HeatmapColor const*)
- `highColor` (*enum Spinnaker::HeatmapColor const*)
- `doCheckInvalidVal` (*bool const*)

- `ImagePtr (invalidVal) ->`
- `srcImage`
- `min`
- `max`
- `lowColor`
- `highColor`
- `doCheckInvalidVal`
- `invalidVal (unsigned int const)`
- `CreateHeatmap(srcImage`
- `destImage)`
- `srcImage`
- `destImage (Spinnaker::ImagePtr &)`

`static GetHeatmapColorGradient(currentLowColor, currentHighColor)`

Parameters

- `currentLowColor (Spinnaker::HeatmapColor &)`
- `currentHighColor (Spinnaker::HeatmapColor &)`

`static GetHeatmapRange(currentLowValue, currentHighValue)`

Parameters

- `currentLowValue (unsigned int &)`
- `currentHighValue (unsigned int &)`

`static SetHeatmapColorGradient(newLowColor, newHighColor)`

Parameters

- `newLowColor (enum Spinnaker::HeatmapColor const)`
- `newHighColor (enum Spinnaker::HeatmapColor const)`

`static SetHeatmapRange(newLowValue, newHighValue)`

Parameters

- `newLowValue (unsigned int const)`
- `newHighValue (unsigned int const)`

property `thisown`

The membership flag

`PySpin.ImageUtilityHeatmap_CreateHeatmap(srcImage) → ImagePtr`

Parameters

- `srcImage (Spinnaker::ImagePtr const &)`
- `ImageUtilityHeatmap_CreateHeatmap(srcImage`
- `min (float const)`

- `max` (*float const*)
- `lowColor` (*enum Spinnaker::HeatmapColor const*)
- `highColor` (*enum Spinnaker::HeatmapColor const*)
- `doCheckInvalidVal` (*bool const*)
- `ImagePtr` (*invalidVal*) \rightarrow
- `srcImage`
- `min`
- `max`
- `lowColor`
- `highColor`
- `doCheckInvalidVal`
- `invalidVal` (*unsigned int const*)
- `ImageUtilityHeatmap_CreateHeatmap` (`srcImage`
- `destImage`)
- `srcImage`
- `destImage` (*Spinnaker::ImagePtr &*)

`PySpin.ImageUtilityHeatmap_GetHeatmapColorGradient` (*currentLowColor, currentHighColor*)

Parameters

- `currentLowColor` (*Spinnaker::HeatmapColor &*)
- `currentHighColor` (*Spinnaker::HeatmapColor &*)

`PySpin.ImageUtilityHeatmap_GetHeatmapRange` (*currentLowValue, currentHighValue*)

Parameters

- `currentLowValue` (*unsigned int &*)
- `currentHighValue` (*unsigned int &*)

`PySpin.ImageUtilityHeatmap_SetHeatmapColorGradient` (*newLowColor, newHighColor*)

Parameters

- `newLowColor` (*enum Spinnaker::HeatmapColor const*)
- `newHighColor` (*enum Spinnaker::HeatmapColor const*)

`PySpin.ImageUtilityHeatmap_SetHeatmapRange` (*newLowValue, newHighValue*)

Parameters

- `newLowValue` (*unsigned int const*)
- `newHighValue` (*unsigned int const*)

class `PySpin.ImageUtilityPolarization`

Bases: `object`

Proxy of C++ `Spinnaker::ImageUtilityPolarization` class.

```
static CreateAolp(srcImage, colorProcessin-  

                 gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  

                 → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateAolp**(**srcImage**
- **destAolpImg** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destAolpImg**
- **colorProcessingAlg**

```
static CreateDolp(srcImage, colorProcessin-  

                 gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  

                 → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateDolp**(**srcImage**
- **destDolpImage** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destDolpImage**
- **colorProcessingAlg**

```
static CreateGlareReduced(srcImage) → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **CreateGlareReduced**(**srcImage**
- **destGlareReducedImage**)
- **srcImage**
- **destGlareReducedImage** (*Spinnaker::ImagePtr &*)

```
static CreateStokesS0(srcImage, colorProcessin-  

                    gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  

                    → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)

- **CreateStokesS0**(srcImage
- destStokesS0Image (Spinnaker::ImagePtr &)
- colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- srcImage
- destStokesS0Image
- colorProcessingAlg

static CreateStokesS1(srcImage, colorProcessin-
gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ ImagePtr

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **colorProcessingAlg** (enum Spinnaker::ColorProcessingAlgorithm const)
- **CreateStokesS1**(srcImage
- destStokesS1Image (Spinnaker::ImagePtr &)
- colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- srcImage
- destStokesS1Image
- colorProcessingAlg

static CreateStokesS2(srcImage, colorProcessin-
gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ ImagePtr

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **colorProcessingAlg** (enum Spinnaker::ColorProcessingAlgorithm const)
- **CreateStokesS2**(srcImage
- destStokesS2Image (Spinnaker::ImagePtr &)
- colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- srcImage
- destStokesS2Image
- colorProcessingAlg

static ExtractPolarQuadrant(srcImage, desiredQuadrant) → ImagePtr

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **desiredQuadrant** (enum Spinnaker::PolarizationQuadrant const)
- **ExtractPolarQuadrant**(srcImage
- destQuadImage (Spinnaker::ImagePtr &)
- desiredQuadrant)

- `srcImage`
- `destQuadImage`
- `desiredQuadrant`

property `thisown`

The membership flag

`PySpin.ImageUtilityPolarization_CreateAolp`(*srcImage*, *colorProcessingAlg*=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
→ *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `colorProcessingAlg` (*enum Spinnaker::ColorProcessingAlgorithm const*)
- `ImageUtilityPolarization_CreateAolp`(*srcImage*
- `destAolpImg` (*Spinnaker::ImagePtr &*)
- `colorProcessingAlg`=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
- `srcImage`
- `destAolpImg`
- `colorProcessingAlg`

`PySpin.ImageUtilityPolarization_CreateDolp`(*srcImage*, *colorProcessingAlg*=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
→ *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `colorProcessingAlg` (*enum Spinnaker::ColorProcessingAlgorithm const*)
- `ImageUtilityPolarization_CreateDolp`(*srcImage*
- `destDolpImage` (*Spinnaker::ImagePtr &*)
- `colorProcessingAlg`=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
- `srcImage`
- `destDolpImage`
- `colorProcessingAlg`

`PySpin.ImageUtilityPolarization_CreateGlareReduced`(*srcImage*) → *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `ImageUtilityPolarization_CreateGlareReduced`(*srcImage*
- `destGlareReducedImage`)
- `srcImage`
- `destGlareReducedImage` (*Spinnaker::ImagePtr &*)

`PySpin.ImageUtilityPolarization_CreateStokesS0`(*srcImage*, *colorProcessingAlg*=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
→ *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `colorProcessingAlg` (*enum Spinnaker::ColorProcessingAlgorithm const*)
- `ImageUtilityPolarization_CreateStokesS0`(`srcImage`
`destStokesS0Image` (*Spinnaker::ImagePtr &*)
`colorProcessingAlg`=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
- `srcImage`
- `destStokesS0Image`
- `colorProcessingAlg`

`PySpin.ImageUtilityPolarization_CreateStokesS1`(*srcImage*, *colorProcessingAlg*=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
→ *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `colorProcessingAlg` (*enum Spinnaker::ColorProcessingAlgorithm const*)
- `ImageUtilityPolarization_CreateStokesS1`(`srcImage`
`destStokesS1Image` (*Spinnaker::ImagePtr &*)
`colorProcessingAlg`=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
- `srcImage`
- `destStokesS1Image`
- `colorProcessingAlg`

`PySpin.ImageUtilityPolarization_CreateStokesS2`(*srcImage*, *colorProcessingAlg*=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
→ *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `colorProcessingAlg` (*enum Spinnaker::ColorProcessingAlgorithm const*)
- `ImageUtilityPolarization_CreateStokesS2`(`srcImage`
`destStokesS2Image` (*Spinnaker::ImagePtr &*)
`colorProcessingAlg`=`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`)
- `srcImage`
- `destStokesS2Image`
- `colorProcessingAlg`

`PySpin.ImageUtilityPolarization_ExtractPolarQuadrant(srcImage, desiredQuadrant) → ImagePtr`

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `desiredQuadrant` (*enum Spinnaker::PolarizationQuadrant const*)
- `ImageUtilityPolarization_ExtractPolarQuadrant(srcImage`
- `destQuadImage` (*Spinnaker::ImagePtr &*)
- `desiredQuadrant`)
- `srcImage`
- `destQuadImage`
- `desiredQuadrant`

`class PySpin.ImageUtilityStereo`

Bases: `object`

Proxy of C++ `Spinnaker::ImageUtilityStereo` class.

`static Compute3DPointFromPixel(disparity, stereoCameraParameters, stereo3DPoint) → bool`

Parameters

- `disparity` (*uint16_t const*)
- `stereoCameraParameters` (*Spinnaker::StereoCameraParameters const &*)
- `stereo3DPoint` (*Spinnaker::Stereo3DPoint &*)

`static ComputeDistanceBetweenPoints(disparityImage, stereoParam, imagePixel1, ImagePixel2, distance) → bool`

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `stereoParam` (*Spinnaker::StereoCameraParameters const &*)
- `imagePixel1` (*Spinnaker::ImagePixel const &*)
- `ImagePixel2` (*Spinnaker::ImagePixel const &*)
- `distance` (*float &*)

`static ComputeDistanceToPoint(disparityImage, stereoParam, imagePixel, distance) → bool`

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `stereoParam` (*Spinnaker::StereoCameraParameters const &*)
- `imagePixel` (*Spinnaker::ImagePixel const &*)
- `distance` (*float &*)

`static ComputePointCloud(disparityImage, rectifiedImage, pointCloudParameters, stereoCameraParameters) → PointCloud`

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)

- `rectifiedImage` (*Spinnaker::ImagePtr const &*)
- `pointCloudParameters` (*Spinnaker::PointCloudParameters const &*)
- `stereoCameraParameters` (*Spinnaker::StereoCameraParameters const &*)
- `ComputePointCloud`(`disparityImage`
 - `rectifiedImage`
 - `pointCloudParameters`
 - `stereoCameraParameters`
 - `pointCloud`)
 - `disparityImage`
 - `rectifiedImage`
 - `pointCloudParameters`
 - `stereoCameraParameters`
 - `pointCloud` (*Spinnaker::PointCloud &*)

static `CreateDepthImage`(*disparityImage, stereoCameraParameters, invalidDepthVal, minDepthVal, maxDepthVal*) → *ImagePtr*

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `stereoCameraParameters` (*Spinnaker::StereoCameraParameters const &*)
- `invalidDepthVal` (*uint16_t const*)
- `minDepthVal` (*float &*)
- `maxDepthVal` (*float &*)
- `CreateDepthImage`(`disparityImage`
 - `stereoCameraParameters`
 - `invalidDepthVal`
 - `depthImage` (*Spinnaker::ImagePtr &*)
 - `minDepthVal`
 - `maxDepthVal`)
 - `disparityImage`
 - `stereoCameraParameters`
 - `invalidDepthVal`
 - `depthImage`
 - `minDepthVal`
 - `maxDepthVal`

static `FilterSpeckles`(*disparityImage, maxSpeckleSize, speckleThreshold, disparityScaleFactor, invalidDataValue*) → *ImagePtr*

Parameters

- **disparityImage** (*Spinnaker::ImagePtr const &*)
- **maxSpeckleSize** (*int const*)
- **speckleThreshold** (*int const*)
- **disparityScaleFactor** (*float const*)
- **invalidDataValue** (*float const*)

static FilterSpecklesFromImage (*disparityImage, maxSpeckleSize, speckleThreshold, disparityScaleFactor, invalidDataValue*)

Parameters

- **disparityImage** (*Spinnaker::ImagePtr &*)
- **maxSpeckleSize** (*int const*)
- **speckleThreshold** (*int const*)
- **disparityScaleFactor** (*float const*)
- **invalidDataValue** (*float const*)

static IsStereoCamera (*pCamera*) → bool

Parameters

pCamera (*Spinnaker::CameraPtr*)

property maxDepthThresholdInMeter

property maxDepthThresholdInMm

property thisown

The membership flag

PySpin.ImageUtilityStereo_Compute3DPointFromPixel (*disparity, stereoCameraParameters, stereo3DPoint*) → bool

Parameters

- **disparity** (*uint16_t const*)
- **stereoCameraParameters** (*Spinnaker::StereoCameraParameters const &*)
- **stereo3DPoint** (*Spinnaker::Stereo3DPoint &*)

PySpin.ImageUtilityStereo_ComputeDistanceBetweenPoints (*disparityImage, stereoParam, imagePixel1, ImagePixel2, distance*) → bool

Parameters

- **disparityImage** (*Spinnaker::ImagePtr const &*)
- **stereoParam** (*Spinnaker::StereoCameraParameters const &*)
- **imagePixel1** (*Spinnaker::ImagePixel const &*)
- **ImagePixel2** (*Spinnaker::ImagePixel const &*)
- **distance** (*float &*)

`PySpin.ImageUtilityStereo_ComputeDistanceToPoint`(*disparityImage*, *stereoParam*, *imagePixel*, *distance*)
→ bool

Parameters

- **disparityImage** (*Spinnaker::ImagePtr* const &)
- **stereoParam** (*Spinnaker::StereoCameraParameters* const &)
- **imagePixel** (*Spinnaker::ImagePixel* const &)
- **distance** (*float* &)

`PySpin.ImageUtilityStereo_ComputePointCloud`(*disparityImage*, *rectifiedImage*, *pointCloudParameters*,
stereoCameraParameters) → *PointCloud*

Parameters

- **disparityImage** (*Spinnaker::ImagePtr* const &)
- **rectifiedImage** (*Spinnaker::ImagePtr* const &)
- **pointCloudParameters** (*Spinnaker::PointCloudParameters* const &)
- **stereoCameraParameters** (*Spinnaker::StereoCameraParameters* const &)
- **ImageUtilityStereo_ComputePointCloud**(**disparityImage**
rectifiedImage
pointCloudParameters
stereoCameraParameters
pointCloud)
- **disparityImage**
rectifiedImage
pointCloudParameters
stereoCameraParameters
pointCloud (*Spinnaker::PointCloud* &)

`PySpin.ImageUtilityStereo_CreateDepthImage`(*disparityImage*, *stereoCameraParameters*, *invalidDepthVal*,
minDepthVal, *maxDepthVal*) → *ImagePtr*

Parameters

- **disparityImage** (*Spinnaker::ImagePtr* const &)
- **stereoCameraParameters** (*Spinnaker::StereoCameraParameters* const &)
- **invalidDepthVal** (*uint16_t* const)
- **minDepthVal** (*float* &)
- **maxDepthVal** (*float* &)
- **ImageUtilityStereo_CreateDepthImage**(**disparityImage**
stereoCameraParameters
invalidDepthVal
depthImage (*Spinnaker::ImagePtr* &))

- `minDepthVal`
- `maxDepthVal`
- `disparityImage`
- `stereoCameraParameters`
- `invalidDepthVal`
- `depthImage`
- `minDepthVal`
- `maxDepthVal`

`PySpin.ImageUtilityStereo_FilterSpeckles`(*disparityImage, maxSpeckleSize, speckleThreshold, disparityScaleFactor, invalidDataValue*) → *ImagePtr*

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `maxSpeckleSize` (*int const*)
- `speckleThreshold` (*int const*)
- `disparityScaleFactor` (*float const*)
- `invalidDataValue` (*float const*)

`PySpin.ImageUtilityStereo_FilterSpecklesFromImage`(*disparityImage, maxSpeckleSize, speckleThreshold, disparityScaleFactor, invalidDataValue*)

Parameters

- `disparityImage` (*Spinnaker::ImagePtr &*)
- `maxSpeckleSize` (*int const*)
- `speckleThreshold` (*int const*)
- `disparityScaleFactor` (*float const*)
- `invalidDataValue` (*float const*)

`PySpin.ImageUtilityStereo_IsStereoCamera`(*pCamera*) → *bool*

Parameters

`pCamera` (*Spinnaker::CameraPtr*)

`PySpin.ImageUtility_CreateNormalized`(*srcImage, destPixelFormat, src-DataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE*) → *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `destPixelFormat` (*enum Spinnaker::PixelFormatEnums const*)
- `srcDataRange` (*enum Spinnaker::SourceDataRange*)
- `ImageUtility_CreateNormalized`(*srcImage*
- `min` (*double const*)
- `max` (*double const*)

- **ImagePtr** (*srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE*)
->)
- **srcImage**
- **min**
- **max**
- **srcDataRange**
- **ImageUtility_CreateNormalized(srcImage**
- **min**
- **max**
- **destPixelFormat**
- **ImagePtr**
- **srcImage**
- **min**
- **max**
- **destPixelFormat**
- **srcDataRange**
- **ImageUtility_CreateNormalized(srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)
- **srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE**)
- **srcImage**
- **destImage**
- **srcDataRange**
- **ImageUtility_CreateNormalized(srcImage**
- **destImage**
- **min**
- **max**
- **srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE**)
- **srcImage**
- **destImage**
- **min**
- **max**
- **srcDataRange**

PySpin.**ImageUtility_CreateScaled**(*srcImage, scalingAlg, scalingFactor*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **scalingAlg** (*enum Spinnaker::ImageScalingAlgorithm*)

- **scalingFactor** (*double*)
- **ImageUtility_CreateScaled**(srcImage
- **destImage** (*Spinnaker::ImagePtr &*)
- **scalingAlg**
- **scalingFactor**)
- **srcImage**
- **destImage**
- **scalingAlg**
- **scalingFactor**

PySpin.**Image_Create**() → *ImagePtr*

PySpin.**Image_Create**(*image*) → *ImagePtr*

Parameters

- **image** (*Spinnaker::ImagePtr const*)
- **Image_Create**(width
- **height** (*size_t*)
- **offsetX** (*size_t*)
- **offsetY** (*size_t*)
- **pixelFormat** (*enum Spinnaker::PixelFormatEnums*)
- **ImagePtr** (*copied from another*)
- **width** (*or using*)
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*void **)
- **Image_Create**(width
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType** (*enum Spinnaker::TLPayloadType*)
- **ImagePtr**
- **width**
- **height**
- **offsetX**

- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType**
- **dataSize** (*size_t*)
- **object** (*Creates a new Image*)
- **constructor** (*either using a default*)
- **ImagePtr**
- **width**
- **height**

:param : :param offset_x: :param offset_y: :param pixel format: :param and a NumPy array containing 8-bit unsigned ints representing the image data: :param (replaces the void* pData argument):.

PySpin.Image_GetImageStatusDescription(*status*) → char const *

Parameters

status (enum *Spinnaker::ImageStatus*)

PySpin.Image_Load(*pFilename*, *format=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT*) → *ImagePtr*

Parameters

- **pFilename** (*char const **)
- **format** (enum *Spinnaker::ImageFileFormat*)

class PySpin.InferenceBoundingBox

Bases: object

Proxy of C++ Spinnaker::InferenceBoundingBox class.

property boxType

property circle

property classId

property confidence

property rect

property rotatedRect

property thisown

The membership flag

class PySpin.InferenceBoundingBoxResult(*args)

Bases: object

Proxy of C++ Spinnaker::InferenceBoundingBoxResult class.

GetBoxAt(*self*, *index*) → *InferenceBoundingBox*

Parameters

index (uint16_t const)

GetBoxCount(*self*) → int16_t

GetBoxSize(*self*) → int8_t

GetVersion(*self*) → int8_t

property thisown

The membership flag

class PySpin.**InferenceBoxCircle**

Bases: object

Proxy of C++ Spinnaker::InferenceBoxCircle class.

property centerXCoord

property centerYCoord

property radius

property thisown

The membership flag

class PySpin.**InferenceBoxRect**

Bases: object

Proxy of C++ Spinnaker::InferenceBoxRect class.

property bottomRightXCoord

property bottomRightYCoord

property thisown

The membership flag

property topLeftXCoord

property topLeftYCoord

class PySpin.**InferenceBoxRotatedRect**

Bases: object

Proxy of C++ Spinnaker::InferenceBoxRotatedRect class.

property bottomRightXCoord

property bottomRightYCoord

property rotationAngle

property thisown

The membership flag

property topLeftXCoord

property topLeftYCoord

class PySpin.IntRegNode(*args, **kwargs)

Bases: [IntegerNode](#), [RegisterNode](#)

Interface for string properties.

C++ includes: IntRegNode.h

SetReference(self, pBase)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase** (*virtual void Spinnaker::GenApi::IntRegNode::SetReference(INode)*
—
- **Value** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.IntegerNode(*args, **kwargs)

Bases: [IInteger](#), [ValueNode](#)

Interface for string properties.

C++ includes: IntegerNode.h

GetFloatAlias(self) → [IFloat](#)

virtual IFloat* Spinnaker::GenApi::IntegerNode::GetFloatAlias()

gets the interface of an alias node.

GetInc(self) → int64_t

virtual int64_t Spinnaker::GenApi::IntegerNode::GetInc()

Get increment

GetIncMode(self) → Spinnaker::GenApi::EIncMode

virtual EIncMode Spinnaker::GenApi::IntegerNode::GetIncMode()

Get increment mode

GetListOfValidValues(self, bounded=True) → *int64_autovector_t*

Parameters

- **bounded** (*bool*)
- **virtual**
- **int64_autovector_t**
- **Spinnaker::GenApi::IntegerNode::GetListOfValidValues(bool**
bounded=true)
- **value** (*Get list of valid*)

GetMax(self) → int64_t

virtual int64_t Spinnaker::GenApi::IntegerNode::GetMax()

Get maximum value allowed

GetMin(*self*) → int64_t

virtual int64_t Spinnaker::GenApi::IntegerNode::GetMin()

Get minimum value allowed

GetRepresentation(*self*) → Spinnaker::GenApi::ERepresentation

virtual ERepresentation Spinnaker::GenApi::IntegerNode::GetRepresentation()

Get recommended representation

GetUnit(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::IntegerNode::GetUnit()

Get the physical unit name

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → int64_t

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::IntegerNode::GetValue**(bool (virtual int64_t)
- **Verify=false**
- **IgnoreCache=false**) (bool)
- **value** (*Get node*)
- **Parameters**
- -----
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

ImposeMax(*self*, *Value*)

Parameters

- **Value** (int64_t)
- **Value**) (virtual void Spinnaker::GenApi::IntegerNode::ImposeMax(int64_t)
- **value** (*Restrict maximum*)

ImposeMin(*self*, *Value*)

Parameters

- **Value** (int64_t)
- **Value**) (virtual void Spinnaker::GenApi::IntegerNode::ImposeMin(int64_t)
- **value** (*Restrict minimum*)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::IntegerNode::SetReference**(*INode (virtual void)*)
- ***pBase**
- **Integer** (*overload SetReference for*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*virtual void Spinnaker::GenApi::IntegerNode::SetValue(int64_t)*)
- **Verify** (*bool*)
- **Value**

:param : :param bool Verify=true): :param Set node value: :param Parameters: :param ———: :param Value: :type Value: The value to set :param Verify: :type Verify: Enables AccessMode and Range verification (default = true)

property thisown

The membership flag

class PySpin.**InterfaceArrivalEventHandler**

Bases: [*IInterfaceArrivalEventHandler*](#)

Proxy of C++ Spinnaker::InterfaceArrivalEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class PySpin.**InterfaceEventHandler**

Bases: [*IInterfaceEventHandler*](#)

Proxy of C++ Spinnaker::InterfaceEventHandler class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class PySpin.**InterfaceList**(*args)

Bases: [*IInterfaceList*](#)

A list of the available interfaces on the system.

C++ includes: InterfaceList.h

Add(self, iface)

Parameters

iface (*Spinnaker::InterfacePtr*)

Append(self, list)

Parameters

list (*Spinnaker::InterfaceList const **)

Clear(self)

void Spinnaker::InterfaceList::Clear()

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.

GetByIndex(self, index) → [*InterfacePtr*](#)

Parameters

- **index** (*The index at which to retrieve the Interface object*)
- **const** (*InterfacePtr Spinnaker::InterfaceList::GetByIndex(int index)*)
- **"index"**. (*Returns a pointer to an Interface object at the*)
- **Parameters**
- -----
- **index**
- **object.** (*A pointer to an Interface*)

GetByInterfaceID(self, interfaceID) → [*InterfacePtr*](#)

Parameters

interfaceID (*std::string*)

GetSize(self) → unsigned int

int Spinnaker::InterfaceList::GetSize() const

Returns the size of the interface list. The size is the number of Interface objects stored in the list.

An integer that represents the list size.

Remove(self, iface)

Parameters

iface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class PySpin.**InterfacePtr**(*args)

Bases: `_SWIG_IFacePtr`

A reference tracked pointer to the interface object.

C++ includes: `InterfacePtr.h`

property thisown

The membership flag

class PySpin.**InterfaceRemovalEventHandler**

Bases: `IInterfaceRemovalEventHandler`

Proxy of C++ Spinnaker::InterfaceRemovalEventHandler class.

OnInterfaceRemoval(self, pInterface)

Parameters

pInterface (`Spinnaker::InterfacePtr`)

property thisown

The membership flag

PySpin.**IsAvailable**(AccessMode) → bool

Parameters

- **AccessMode** (enum `Spinnaker::GenApi::EAccessMode`)
- **bool** (`IsAvailable(ptr) ->`)
- **p** (`Spinnaker::GenApi::IBase const *`)
- **bool**
- **r** (`Spinnaker::GenApi::IBase const &`)
- **bool**
- **ptr** (`Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &`)
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**

- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **T** (*Spinnaker::GenApi::IsAvailable(const Spinnaker::GenApi::CPointer<*
- **B**
- **&ptr)** (*>*)
- **Available** (*Checks if a node is*)

PySpin.IsCacheable(*CachingMode*) → bool

Parameters

- **CachingMode** (*enum Spinnaker::GenApi::ECachingMode*)
- **bool**
- **CachingMode)** (*Spinnaker::GenApi::IsCacheable(ECachingMode)*
- **Cacheability** (*Tests*)

PySpin.IsImplemented(*AccessMode*) → bool

Parameters

- **AccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **bool** (*IsImplemented(ptr) ->*)
- **p** (*Spinnaker::GenApi::IBase const **)
- **bool**
- **r** (*Spinnaker::GenApi::IBase const &*)
- **bool**
- **ptr** (*Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &*)
- **bool**
- **ptr**
- **bool**
- **ptr**

- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- T(*Spinnaker::GenApi::IsImplemented(const Spinnaker::GenApi::CPointer<*

```
:param : :param B > &ptr): :param Checks if a node is Implemented:
```

PySpin.IsReadable(*AccessMode*) → bool

Parameters

- **AccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **bool** (*IsReadable(ptr) ->*)
- **p** (*Spinnaker::GenApi::IBase const **)
- **bool**
- **r** (*Spinnaker::GenApi::IBase const &*)
- **bool**
- **ptr** (*Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &*)
- **bool**
- **ptr**
- **bool**

- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- T (*Spinnaker::GenApi::IsReadable(const Spinnaker::GenApi::CPointer<*
- B
- &ptr) (>)
- readable (*Checks if a node is*)

PySpin.IsVisible(*Visibility, MaxVisiblity*) → bool

Parameters

- **Visibility** (*Spinnaker::GenApi::IsVisible(EVisibility)*)
- **MaxVisiblity** (*enum Spinnaker::GenApi::EVisibility*)
- **bool**
- **Visibility**
- **EVisibility**
- **MaxVisiblity**
- **CAVE** (*Tests Visibility*)

PySpin.IsWritable(*AccessMode*) → bool

Parameters

- **AccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **bool** (*IsWritable(ptr) ->*)
- **p** (*Spinnaker::GenApi::IBase const **)
- **bool**
- **r** (*Spinnaker::GenApi::IBase const &*)
- **bool**
- **ptr** (*Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &*)
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **T** (*Spinnaker::GenApi::IsWritable(const Spinnaker::GenApi::CPointer<*

- **B**
- **&ptr) (>)**
- **Writable** (*Checks if a node is*)

class PySpin.JPEGOption

Bases: object

Options for saving JPEG image.

C++ includes: SpinnakerDefs.h

property progressive

property quality

property reserved

property thisown

The membership flag

class PySpin.JPG20Option

Bases: object

Options for saving JPEG2000 image.

C++ includes: SpinnakerDefs.h

property quality

property reserved

property thisown

The membership flag

class PySpin.LibraryVersion

Bases: object

Proxy of C++ Spinnaker::LibraryVersion class.

property build

property major

property minor

property thisown

The membership flag

property type

class PySpin.LoggingEventData(*args, **kwargs)

Bases: object

The LoggingEventData object.

C++ includes: LoggingEventData.h

GetCategoryName(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetCategoryName()

Gets the logging event category name.

The category name

GetLogMessage(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetLogMessage()

Gets the logging event message.

The log message

GetNDC(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetNDC()

Gets the logging event's Nested Diagnostic Context (NDC).

The log event's NDC

GetPriority(*self*) → int const

const int Spinnaker::LoggingEventData::GetPriority()

Gets the logging event priority.

The log priority

GetPriorityName(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetPriorityName()

Gets the logging event priority name.

The priority name of the log

GetThreadName(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetThreadName()

Gets the logging event thread name.

The thread name

GetTimestamp(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetTimestamp()

Gets the logging event time stamp.

The time stamp of the log

property thisown

The membership flag

class PySpin.**LoggingEventDataPtr**(*args)

Bases: _SWIG_LogPtr

A reference tracked pointer to the LoggingEvent object.

C++ includes: LoggingEventDataPtr.h

property thisown

The membership flag

class PySpin.LoggingEventHandlerBases: [ILoggingEventHandler](#)

Proxy of C++ Spinnaker::LoggingEventHandler class.

OnLogEvent(*self*, *eventPtr*)**Parameters****eventPtr** (*Spinnaker::LoggingEventDataPtr*)**property thisown**

The membership flag

class PySpin.MJPGOption

Bases: object

Options for saving MJPG files.

C++ includes: SpinVideoDefs.h

property frameRate**property height****property quality****property reserved****property thisown**

The membership flag

property width**class** PySpin.Node(*args, **kwargs)Bases: [INode](#)

class common to all nodes

C++ includes: Node.h

DeregisterCallback(*self*, *hCallback*) → bool**Parameters**

- **hCallback** (*Spinnaker::GenApi::CallbackHandleType*)
- **bool** (*virtual*)
- **Spinnaker::GenApi::Node::DeregisterCallback(CallbackHandleType**
- **hCallback)**
- **the** (*De register change callback Destroys CNodeCallback object true if*)
- **valid** (*callback handle was*)

GetAccessMode(*self*) → *Spinnaker::GenApi::EAccessMode**virtual EAccessMode Spinnaker::GenApi::Node::GetAccessMode() const*

Base interface overrides.

Get the access mode of the node

GetAlias(*self*) → *INode*

virtual INode* Spinnaker::GenApi::Node::GetAlias() const

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

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

virtual ECachingMode Spinnaker::GenApi::Node::GetCachingMode() const

Get Caching Mode

GetCastAlias(*self*) → *INode*

virtual INode* Spinnaker::GenApi::Node::GetCastAlias() const

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

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

- **LinkType** (*The link type*)
- **virtual**
- **Spinnaker::GenApi::Node::GetChildren(GenApi::NodeList_t (void)**
- **&Children**
- **const** (*ELinkType LinkType=ctReadingChildren*)
- **on.** (*Get all nodes this node directly depends*)
- **Parameters**
- **-----**
- **Children** (*List of children nodes*)
- **LinkType**

GetDescription(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDescription() const

Get a long description of the node

GetDeviceName(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDeviceName() const

Get a name of the device

GetDisplayName(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDisplayName() const

Get a name string for display

GetDocuURL(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDocuURL() const

Gets a URL pointing to the documentation of that feature

GetEventID(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetEventID() const

Get the EventId of the node

GetLockNodes(*self*)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

- **FullQualified** (*bool*)
- **virtual**
- **Spinnaker::GenApi::Node::GetName**(*bool* (*GenICam::gcstring*))
- **const** (*FullQualified=false*)
- **name** (*Get node*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

virtual GenApi::ENamespace Spinnaker::GenApi::Node::GetNameSpace() *const*

Get name space

GetNodeHandle(*self*) → *std::shared_ptr< Spinnaker::GenApi::Node::NodeImpl >*

std::shared_ptr<Node::NodeImpl> Spinnaker::GenApi::Node::GetNodeHandle() *const*

Get Node handle

GetNodeMap(*self*) → *INodeMap*

virtual INodeMap Spinnaker::GenApi::Node::GetNodeMap()* *const*

Retrieves the central node map

GetParents(*self*)

virtual void Spinnaker::GenApi::Node::GetParents(GenApi::NodeList_t &Parents) *const*

Gets all nodes this node is directly depending on.

6.5 Parameters:

Parents: List of parent nodes

GetPollingTime(*self*) → *int64_t*

virtual int64_t Spinnaker::GenApi::Node::GetPollingTime() *const*

recommended polling time (for not cacheable nodes)

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

virtual EInterfaceType Spinnaker::GenApi::Node::GetPrincipalInterfaceType() *const*

Get the type of the main interface of a node

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const &*)
- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **AttributeStr** (*Spinnaker::GenICam::gcstring &*)
- **virtual**
- **GenICam::gcstring** (*bool Spinnaker::GenApi::Node::GetProperty(const)*)
- **&PropertyName**
- **&ValueStr** (*GenICam::gcstring*)

- **GenICam::gcstring**
- **&AttributeStr)**
- **a** (*Retrieves a property plus an additional attribute by name If*)
- **as** (*property has multiple values/attribute they come with Tabs*)
- **delimiters**

GetPropertyNames(*self*)

virtual void Spinnaker::GenApi::Node::GetPropertyNames(GenICam::gcstring_vector &PropertyNames)
const

Returns a list of the names all properties set during initialization

GetSelectedFeatures(*self*)

virtual void Spinnaker::GenApi::Node::GetSelectedFeatures(FeatureList_t &) const

retrieve the group of selected features

GetSelectingFeatures(*self*)

virtual void Spinnaker::GenApi::Node::GetSelectingFeatures(FeatureList_t &) const

retrieve the group of features selecting this node

GetToolTip(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetToolTip() const

Get a short description of the node

GetVisibility(*self*) → Spinnaker::GenApi::EVisibility

virtual EVisibility Spinnaker::GenApi::Node::GetVisibility() const

Get the recommended visibility of the node

ImposeAccessMode(*self*, *ImposedAccessMode*)**Parameters**

- **ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **Spinnaker::GenApi::Node::ImposeAccessMode**(**EAccessMode** (*virtual void*))
- **ImposedAccessMode**)
- **node** (*Imposes an access mode to the natural access mode of the*)

ImposeVisibility(*self*, *ImposedVisibility*)**Parameters**

- **ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*)
- **Spinnaker::GenApi::Node::ImposeVisibility**(**EVisibility** (*virtual void*))
- **ImposedVisibility**)
- **node** (*Imposes a visibility to the natural visibility of the*)

InvalidateNode(*self*)

virtual void Spinnaker::GenApi::Node::InvalidateNode()

Indicates that the node's value may have changed. Fires the callback on this and all dependent nodes

IsAccessModeCacheable(*self*) → Spinnaker::GenApi::EYesNo

virtual EYesNo Spinnaker::GenApi::Node::IsAccessModeCacheable() const

True if the AccessMode can be cached

IsCachable(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsCachable() const

Is the node value cacheable

IsDeprecated(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsDeprecated() const

True if the node should not be used any more

IsFeature(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsFeature() const

True if the node can be reached via category nodes from a category node named “Root”

IsSelector(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsSelector() const

Selector interface overrides.

true if this feature selects a group of features

IsStreamable(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsStreamable() const

True if the node is streamable

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

- **pCallback** (*Spinnaker::GenApi::CNodeCallback **)
- **CallbackHandleType** (*virtual*)
- ***pCallback** (*Spinnaker::GenApi::Node::RegisterCallback(CNodeCallback)*
—
- **object** (*Register change callback Takes ownership of the CNodeCallback*)

SetNodeHandle(*self*, *pNodeHandle*)

Parameters

- **pNodeHandle** (*std::shared_ptr< Spinnaker::GenApi::Node::NodeImpl >*)
- **void**
- **Node::NodeImpl** (*Spinnaker::GenApi::Node::SetNodeHandle(std::shared_ptr<*
>))
- **pNodeHandle** (*>*)
- **handle** (*Set Node*)

SetNodeMap(*self*, *pNodeMap*)

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap **)

- **void**
- ***pNodeMap** (*Spinnaker::GenApi::Node::SetNodeMap*([INodeMap](#)) –

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::ISelector **)
- **SetReference**(*self*
- **pBase**)
- **pBase**
- **virtual**
- ***pBase** (*void Spinnaker::GenApi::Node::SetReference*([ISelector](#)) –

property thisown

The membership flag

class `PySpin.NodeCallback`

Bases: `object`

Proxy of C++ `NodeCallback` class.

CallbackFunction(*self*, *node*)

Parameters

node (*Spinnaker::GenApi::INode **)

Callback function used in node callbacks (see `NodeMapCallback` example for more details). Users should override this function when using node callbacks.

6.6 Parameters:

node: `INode` passed to the function during the callback.

property thisown

The membership flag

class `PySpin.NodeMap(*args)`

Bases: [INodeMap](#), [IDeviceInfo](#)

Smart pointer template for `NodeMaps` with `create` function.

6.7 Parameters:

TCameraParams: The camera specific parameter class (auto generated from camera xml file)

C++ includes: `NodeMap.h`

static `ClearXMLCache()` → `bool`

Connect(*self*, *pPort*, *PortName*) → `bool`

Parameters

- **pPort** (*IPort **)
- **PortName** (*Spinnaker::GenICam::gcstring const &*)

- **Connect**(self
- **bool** (pPort) ->)
- **pPort**
- **virtual**
- **const** (bool Spinnaker::GenApi::NodeMap::Connect(IPort *pPort)) –
- **"Device"** (Connects a port to the standard port)

Destroy(self)

void Spinnaker::GenApi::NodeMap::Destroy()

Destroys the node map

GetDeviceName(self) → gcstring

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetDeviceName()

Get device name

GetDeviceVersion(self, Version)**Parameters**

- **Version** (Spinnaker::GenICam::Version_t &)
- **void** (virtual)
- **Spinnaker::GenApi::NodeMap::GetDeviceVersion**(GenICam::Version_t
- **&Version**)
- **file** (Get the version of the device description)

GetGenApiVersion(self, Version, Build)**Parameters**

- **Version** (Spinnaker::GenICam::Version_t &)
- **Build** (uint16_t &)
- **void** (virtual)
- **Spinnaker::GenApi::NodeMap::GetGenApiVersion**(GenICam::Version_t
- **&Version**
- **&Build** (uint16_t)
- **implementation** (Get the version of the DLL's GenApi)

GetModelName(self) → gcstring

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetModelName()

Get the model name

GetNode(self, key) → INode**Parameters**

- **key** (Spinnaker::GenICam::gcstring const &)
- **virtual**
- **GenICam::gcstring** (INode* Spinnaker::GenApi::NodeMap::GetNode(const)

- **const** (&key))
- **name** (Retrieves the node from the central map by)

GetNodeMapHandle(self) → void *

void* Spinnaker::GenApi::NodeMap::GetNodeMapHandle() const

GetNodes(self)

virtual void Spinnaker::GenApi::NodeMap::GetNodes(NodeList_t &Nodes) const

Retrieves all nodes in the node map

GetNumNodes(self) → uint64_t

virtual uint64_t Spinnaker::GenApi::NodeMap::GetNumNodes() const

Get the number of nodes in the map

GetProductGuid(self) → gcstring

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetProductGuid()

Get the GUID describing the product

GetSchemaVersion(self, Version)

Parameters

- **Version** (Spinnaker::GenICam::Version_t &)
- **void** (virtual)
- **Spinnaker::GenApi::NodeMap::GetSchemaVersion**(GenICam::Version_t
- **&Version**)
- **number** (Get the schema version)

GetStandardNameSpace(self) → gcstring

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetStandardNameSpace()

Get the standard name space

GetSupportedSchemaVersions(self)

virtual void Spinnaker::GenApi::NodeMap::GetSupportedSchemaVersions(GenICam::gcstring_vector &SchemaVersions)

! Loads an XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromFile(const GenICam::gcstring& XMLFileName, const GenICam::gcstring& StyleSheetFileName, const GenICam::gcstring& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Loads a Zipped XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromZIPFile(const GenICam::gcstring& ZIPFileName, const GenICam::gcstring& StyleSheetFileName, const GenICam::gcstring& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Injects an XML file into a target file virtual void MergeXMLFiles(const GenICam::gcstring& TargetFileName, *< Name of the target XML file to process const GenICam::gcstring& InjectedFileName, *< Name of the Injected XML file to process const GenICam::gcstring& OutputFileName *< Name of the output file);

! Extract independent subtree virtual void ExtractIndependentSubtree(const GenICam::gcstring& XMLData, *< The XML data the subtree is extracted from. const GenICam::gcstring& InjectXMLData, *< Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed. const GenICam::gcstring& SubTreeRootNodeName,*< The name of the node that represents the root of the subtree that shall be extracted. GenICam::gcstring& ExtractedSubtree *< The returned extracted subtree

as string.); Gets a list of supported schema versions Each list entry is a string with the format “{Major}.{Minor}” where {Major} and {Minor} are integers Example: {“1.1”, “1.2”} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

GetToolTip(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetToolTip()

Get tool tip

GetVendorName(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetVendorName()

Get the vendor name

GetVersionGuid(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetVersionGuid()

Get the GUID describing the product version

InvalidateNodes(*self*)

virtual void Spinnaker::GenApi::NodeMap::InvalidateNodes() const

Invalidates all nodes

LoadXMLFromFile(*self*, *FileName*)

Parameters

- **FileName** (*Spinnaker::GenICam::gcstring*)
- **Spinnaker::GenApi::NodeMap::LoadXMLFromFile**(*GenICam::gcstring* (*void*)
- **FileName**)
- **if** (! Creates the object from the default DLL ! note Can only be used)
- **xml** (the class *TCameraParams* was auto generated from a specific camera)
- **LoadDLL(void);** (*file void*)
- **and** (! Creates the object from a DLL whose name is deduced from vendor)
- **VendorName** (*model name void LoadDLL*(*GenICam::gcstring*)

:param : :param GenICam::gcstring ModelName); :param ! Creates the object from a DLL with given file name void: :param LoadDLL(GenICam::gcstring FileName); Creates the object from a XML: :param file with given file name:

LoadXMLFromFileInject(*self*, *TargetFileName*, *InjectFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring*)
- **InjectFileName** (*Spinnaker::GenICam::gcstring*)
- **void**
- **Spinnaker::GenApi::NodeMap::LoadXMLFromFileInject**(*GenICam::gcstring*
- **TargetFileName**

- **InjectFileName** (*GenICam::gcstring*)
- **given** (*Creates the object from a XML target and an inject file with*)
- **name** (*file*)

LoadXMLFromString(*self, XMLData*)

Parameters

- **XMLData** (*Spinnaker::GenICam::gcstring const &*)
- **Spinnaker::GenApi::NodeMap::LoadXMLFromString**(*const (void)*)
- **&XMLData** (*GenICam::gcstring*)
- **string** (*Creates the object from XML data given in a*)

LoadXMLFromStringInject(*self, TargetXMLDataconst, InjectXMLData*)

Parameters

- **TargetXMLDataconst** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **void**
- **Spinnaker::GenApi::NodeMap::LoadXMLFromStringInject**(*const*)
- **&TargetXMLDataconst** (*GenICam::gcstring*)
- **GenICam::gcstring** (*const*)
- **&InjectXMLData**
- **injection** (*Creates the object from XML data given in a string with*)

LoadXMLFromZIPData(*self, zipData, zipSize*)

Parameters

- **zipData** (*void const **)
- **zipSize** (*size_t*)
- **void** (*void Spinnaker::GenApi::NodeMap::LoadXMLFromZIPData (const)*)
- ***zipData**
- **zipSize** (*size_t*)
- **string** (*Creates the object from a ZIP'd XML file given in a*)

LoadXMLFromZIPFile(*self, ZipFileName*)

Parameters

- **ZipFileName** (*Spinnaker::GenICam::gcstring*)
- **void**
- **Spinnaker::GenApi::NodeMap::LoadXMLFromZIPFile**(*GenICam::gcstring*)
- **ZipFileName**
- **name** (*Creates the object from a ZIP'd XML file with given file*)

Poll(*self*, *ElapsedTime*)

Parameters

- **ElapsedTime** (*int64_t*)
- **void** (*virtual*)
- **ElapsedTime** (*Spinnaker::GenApi::NodeMap::Poll(int64_t)*)
- **time** (*Fires nodes which have a polling*)

property thisown

The membership flag

PySpin.**NodeMap_ClearXMLCache**() → bool

class PySpin.**PGMOption**

Bases: object

Options for saving PGM images.

C++ includes: SpinnakerDefs.h

property binaryFile

property reserved

property thisown

The membership flag

class PySpin.**PNGOption**

Bases: object

Options for saving PNG images.

C++ includes: SpinnakerDefs.h

property compressionLevel

property interlaced

property reserved

property thisown

The membership flag

class PySpin.**PPMOption**

Bases: object

Options for saving PPM images.

C++ includes: SpinnakerDefs.h

property binaryFile

property reserved

property thisown

The membership flag

class PySpin.PointCloudBases: *IPointCloud*

Proxy of C++ Spinnaker::PointCloud class.

AddPoint(*self*, *point*)

Parameters

point (*Spinnaker::Stereo3DPoint const*)**GetNumPoints**(*self*) → *size_t***GetPoint**(*self*, *index*) → *Stereo3DPoint*

Parameters

index (*unsigned int const*)**GetPointCloudData**(*self*) → *Spinnaker::IPointCloud::PointCloudData ****LoadPointCloudFromPly**(*self*, *filename*)

Parameters

filename (*std::string const &*)**PrintPoints**(*self*, *numPointsToPrint*)

Parameters

numPointsToPrint (*unsigned int*)**SavePointCloudAsPly**(*self*, *arg0*)

Parameters

arg0 (*std::string const &*)**property thisown**

The membership flag

class PySpin.PointCloudParametersBases: *object*

Proxy of C++ Spinnaker::PointCloudParameters class.

property ROIImageBottom**property ROIImageLeft****property ROIImageRight****property ROIImageTop****property ROIWorldCoordinatesXMax****property ROIWorldCoordinatesXMin****property ROIWorldCoordinatesYMax****property ROIWorldCoordinatesYMin****property ROIWorldCoordinatesZMax****property ROIWorldCoordinatesZMin**

property decimationFactor

property thisown

The membership flag

class PySpin.RegisterNode(*args, **kwargs)

Bases: *IRegister*, *ValueNode*

Interface for string properties.

C++ includes: RegisterNode.h

Get(self, pBuffer, Verify=False, IgnoreCache=False)

Parameters

- **pBuffer** (*The buffer receiving the data to read*)
- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **virtual**
- ***pBuffer** (*void Spinnaker::GenApi::RegisterNode::Get(uint8_t) -*
- **int64_t**
- **Length** (*The number of bytes to retrieve*)
- **Verify=false** (*bool*)
- **IgnoreCache=false** (*bool*)
- **contents** (*Fills a buffer with the register's*)
- **Parameters**
- **-----**
- **pBuffer**
- **Length**
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

GetAddress(self) → int64_t

virtual int64_t Spinnaker::GenApi::RegisterNode::GetAddress()

Retrieves the Address of the register

GetLength(self) → int64_t

virtual int64_t Spinnaker::GenApi::RegisterNode::GetLength()

Retrieves the Length of the register [Bytes]

Set(*self*, *pBuffer*, *Verify=True*)

Parameters

- **pBuffer** (*uint8_t const **)
- **Verify** (*bool*)
- **virtual**
- ***pBuffer** (*void Spinnaker::GenApi::RegisterNode::Set(const uint8_t) –*

:param : :param int64_t Length: :param bool Verify=true): :param Set the register's contents: :param
Parameters: :param ———: :param pBuffer: :type pBuffer: The buffer containing the data to set :param
Length: :type Length: The number of bytes in pBuffer :param Verify: :type Verify: Enables AccessMode
and Range verification (default = true)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::RegisterNode::SetReference(INode (virtual void)**
- ***pBase**)
- **Register** (*overload SetReference for*)

property thisown

The membership flag

PySpin.RegisterNodeCallback(*pNode*, *f*)

Parameters

- **pNode** (*Spinnaker::GenApi::INode **)
- **f** (*NodeCallback &*)

PySpin.ReplaceEnvironmentVariables(*Buffer*, *ReplaceBlankBy20=False*)

Parameters

- **Buffer** (*Spinnaker::GenICam::gcstring &*)
- **ReplaceBlankBy20** (*bool*)
- **void** (*SPINNAKER_API*)
- **&Buffer** (*Spinnaker::GenICam::ReplaceEnvironmentVariables(gcstring)*
- **bool**
- **ReplaceBlankBy20=false**)
- **%20** (*Replaces in a string and replace ' ' with*)

class PySpin.SIOption

Bases: object

Proxy of C++ Spinnaker::SIOption class.

property reserved

property thisown

The membership flag

`PySpin.SetGenICamCLProtocolFolder(path)`

Parameters

- **path** (*Spinnaker::GenICam::gcstring const &*)
- **void** (*SPINNAKER_API*)
- **&path** (*Spinnaker::GenICam::SetGenICamCLProtocolFolder(const gcstring)*)
- **folder** (*Stores the path of the CLProtocol*)

`PySpin.SetGenICamCacheFolder(path)`

Parameters

- **path** (*Spinnaker::GenICam::gcstring const &*)
- **Spinnaker::GenICam::SetGenICamCacheFolder(const (SPINNAKER_API void)**
- **&path** (*gcstring*)
- **folder** (*Stores the path of the GenICam cache*)

`PySpin.SetGenICamLogConfig(path)`

Parameters

- **path** (*Spinnaker::GenICam::gcstring const &*)
- **Spinnaker::GenICam::SetGenICamLogConfig(const (SPINNAKER_API void)**
- **&path** (*gcstring*)
- **file** (*Stores the path of the GenICam logging properties*)

`PySpin.SetMessageCallback(cb)`

Adds a callback to the updatator to handle messages from the updatator. Only gets called if the -P switch is present in the arguments passed to UpdateFirmware[Console]!

Parameters

cb – Function to use as callback; this function must take exactly 1 argument.

`PySpin.SetProgressCallback(cb)`

Adds a callback to the updatator to represent update progress. Only gets called if the -P switch is present in the arguments passed to UpdateFirmware[Console]!

Parameters

cb – Function to use as callback; this function must take exactly 4 arguments.

`PySpin.SpinUpdate_SetMsgCallback(messageCallbackFunction)`

Parameters

messageCallbackFunction (*SpinUpdate::UpdatatorMessageCallback*)

`PySpin.SpinUpdate_SetProgCallback(progressCallbackFunction)`

Parameters

progressCallbackFunction (*SpinUpdate::UpdatatorProgressCallback*)

class PySpin.SpinVideo

Bases: object

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

C++ includes: SpinVideo.h

Append(*self*, *pImage*)

Parameters

- **pImage** (*The image to append.*)
- **virtual**
- **pImage**) (*void Spinnaker::Video::SpinVideo::Append(ImagePtr)*)
- **file.** (*Append an image to the AVI/MP4*)
- **Parameters**
- -----
- **pImage**

Close(*self*)

virtual void Spinnaker::Video::SpinVideo::Close()

Close the AVI/MP4 file.

See: Open()

Open(*self*, *pFileName*, *pOption*)

Parameters

- **pFileName** (*The filename of the MP4 file.*)
- **pOption** (*H264 options to apply to the MP4 file.*)
- **Open**(*self*
- **pFileName**
- **pOption**)
- **pFileName**
- **pOption**
- **Open**(*self*
- **pFileName**
- **pOption**)
- **pFileName**
- **pOption**
- **void** (*virtual*)
- ***pFileName** (*Spinnaker::Video::SpinVideo::Open(const char) –*
- **Video::H264Option**
- **&pOption**)
- **The** (*Open an H264 MP4 file in preparation for writing Images to disk.*
-)

- **automatically** (size of MP4 files is limited to 2GB. The filenames are)
- **specified.** (generated using the filename)
- **Parameters**
- -----
- **pFileName**
- **pOption**
- See ([H264Option](#))
- See

SetMaximumFileSize(*self*, *size*)

Parameters

size (*unsigned int*)

property thisown

The membership flag

class `PySpin.Stereo3DPoint`

Bases: `object`

Proxy of C++ Spinnaker::Stereo3DPoint class.

property b

property g

property pixel

property r

property thisown

The membership flag

property x

property y

property z

class `PySpin.StereoCameraParameters`

Bases: `object`

Proxy of C++ Spinnaker::StereoCameraParameters class.

property baseline

property coordinateOffset

property disparityScaleFactor

property focalLength

property invalidDataFlag

property invalidDataValue

property principalPointU

property principalPointV

property thisown

The membership flag

class PySpin.StringNode(*args, **kwargs)

Bases: *IString*, *ValueNode*

Interface for string properties.

C++ includes: StringNode.h

GetMaxLength(self) → int64_t

virtual int64_t Spinnaker::GenApi::StringNode::GetMaxLength()

Retrieves the maximum length of the string in bytes

GetValue(self, Verify=False, IgnoreCache=False) → *gcstring*

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::StringNode::GetValue**(bool *GenICam::gcstring*) (virtual)
- **Verify=false**
- **IgnoreCache=false**) (bool)
- **value** (*Get node*)
- **Parameters**
- -----
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

SetReference(self, pBase)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase**) (virtual void Spinnaker::GenApi::StringNode::SetReference(*INode*)
—
- **Value** (*overload SetReference for*)

SetValue(self, Value, Verify=True)

Parameters

- **Value** (*The value to set*)

- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Spinnaker::GenApi::StringNode::SetValue(const (virtual void)**
- **&Value** (*GenICam::gcstring*)
- **Verify=true)** (*bool*)
- **value** (*Set node*)
- **Parameters**
- -----
- **Value**
- **Verify**

property thisown

The membership flag

class PySpin.**StringRegNode**(*args, **kwargs)

Bases: [StringNode](#), [RegisterNode](#)

Interface for string properties.

C++ includes: StringRegNode.h

SetReference(self, pBase)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::StringRegNode::SetReference(INode (virtual void)**
- ***pBase)**
- **Value** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.**System**(*args, **kwargs)

Bases: [ISystem](#)

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

C++ includes: System.h

GetCameras(self, updateInterfaces=True, updateCameras=True) → [CameraList](#)

Parameters

- **updateInterfaces** (*Determines whether or not updateInterfaceList() is*)
- **updateCameras** (*Determines whether or not UpdateCameras() is called*)
- **CameraList**
- **updateInterfaces=true** (*Spinnaker::System::GetCameras(bool)*)
- **bool**
- **updateCameras=true)**
- **call** (*Returns a list of cameras that are available on the system. This*)

- **interfaces.** (returns both GigE Vision and Usb3 Vision cameras from all)
- **It** (The camera list object will reference count the cameras it returns.)
- **before** (is important that the camera list is destroyed or is cleared)
- **system->** (calling system-> ReleaseInstance() or else the call to
- **a** (ReleaseInstance() will result in an error message thrown that)
- **held.** (reference to the camera is still)
- **See** (CameraList::Clear())
- **See**
- **Parameters**
- -----
- **updateInterfaces**
- **system** (before getting cameras from available interfaces on the)
- **updateCameras**
- **system**
- **cameras.** (An CameraList object that contains a list of all)

static **GetInstance()** → *SystemPtr*

GetInterfaces(*self*, *updateInterface=True*) → *InterfaceList*

Parameters

- **updateInterface** (Determines whether or not UpdateInterfaceList() is)
- **Spinnaker::System::GetInterfaces**(**bool** (*InterfaceList*)
- **updateInterface=true**)
- **call** (Returns a list of interfaces available on the system. This)
- **interfaces.** (An InterfaceList object that contains a list of all)
- **Parameters**
- -----
- **updateInterface**
- **interfaces** (called before getting available)
- **interfaces.**

GetLibraryVersion(*self*) → *LibraryVersion*

GetLoggingEventPriorityLevel(*self*) → Spinnaker::SpinnakerLogLevel

SpinnakerLogLevel Spinnaker::System::GetLoggingEventPriorityLevel()

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: `SpinnakerLogLevel`

Level The threshold level

GetTLNodeMap(*self*) → *INodeMap*

IsInUse(*self*) → bool

bool `Spinnaker::System::IsInUse()`

Checks if the system is in use by any interface or camera objects.

Returns true if the system is in use and false otherwise.

RegisterEventHandler(*self*, *evtHandlerToRegister*, *updateInterface=False*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::EventHandler &*)
- **updateInterface** (*bool*)

RegisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler &*)

ReleaseInstance(*self*)

void `Spinnaker::System::ReleaseInstance()`

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_RESOURCE_IN_USE`.

See: `Error`

See: `GetInstance()`

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *requestAck=False*, *pResultSize=None*, *results=0*)

Parameters

- **deviceKey** (*Spinnaker::System::SendActionCommand(unsigned int)*)
- **groupKey** (*int*)
- **groupMask** (*unsigned int*)
- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)
- **void**
- **deviceKey**

- **unsigned**
- **groupKey**
- **groupMask**
- **actionTime=0** (*unsigned long long*)

:param : :param unsigned int *pResultSize=0: :param ActionCommandResult results[]=NULL): :param Broadcast an Action Command to all devices on system: :param Parameters: :param ———: :param deviceKey: :type deviceKey: The Action Command's device key :param groupKey: :type groupKey: The Action Command's group key :param groupMask: :type groupMask: The Action Command's group mask :param actionTime: :type actionTime: (Optional) Time when to assert a future action. Zero :param means immediate action.: :param pResultSize: :type pResultSize: (Optional) The number of results in the results array. :param The value passed should be equal to the expected number of devices: :param that acknowledge the command. Returns the number of received results.: :param results: :type results: (Optional) An Array with *pResultSize elements to hold the :param action command result status. The buffer is filled starting from index: :param 0. If received results are less than expected number of devices that: :param acknowledge the command: :param remaining results are not changed. If: :param received results are more than expected number of devices that: :param acknowledge the command: :param extra results are ignored and not appended to: :param array. This parameter is ignored if pResultSize is 0. Thus this: :param parameter can be NULL if pResultSize is 0 or NULL.:

SetLoggingEventPriorityLevel(*self, level*)

Parameters

- **level** (*enum Spinnaker::SpinnakerLogLevel*)
- **void**
- **Spinnaker::System::SetLoggingEventPriorityLevel**(*SpinnakerLogLevel*
- *level*)
- **events** (*Sets a threshold priority level for logging event. Logging*)
- **callbacks.** (*below such level will not trigger*)
- **logging** (*Spinnaker uses five levels of*)
- **intervention.** (*Warning - failures that are recoverable without user*)
- **intervention.**
- **removal** (*Notice - information about events such as camera arrival and*)

:param : :param initialization and deinitialization: :param starting and stopping image: :param acquisition: :param and feature modification.: :param Info - information about recurring events that are generated regularly: :param such as information on individual images.: :param Debug - information that can be used to troubleshoot the system.: :param See: :type See: SpinnakerLogLevel :param Parameters: :param ———: :param level: :type level: The threshold level

UnregisterAllLoggingEventHandlers(*self*)

UnregisterEventHandler(*self, evtHandlerToUnregister*)

Parameters

- **evtHandlerToUnregister** (*Spinnaker::EventHandler &*)

UnregisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler* &)

UpdateCameras(*self*, *updateInterfaces=True*) → bool

Parameters

- **updateInterfaces** (*bool*)
- **bool**
- **updateInterfaces=true** (*Spinnaker::System::UpdateCameras(bool)*)
- **that** (*Updates the list of cameras on the system. Note*)
- **each** (*System::GetCameras()* internally calls *UpdateCameras()* for)
- **the** (*interface it enumerates. If the list changed between this call and*)
- **true** (*last time UpdateCameras was called then the return value will be*)

:param : :param otherwise it is false.: :param See: :type See: *GetCameras()* :param Parameters: :param
 ———: :param *updateInterfaces*: :type *updateInterfaces*: Determines whether or not *UpdateInterfaceList()* is :param called before updating cameras for available interfaces on the system: :param True
 if cameras changed on interface and false otherwise.:

UpdateInterfaceList(*self*)

property thisown

The membership flag

class *PySpin.SystemEventHandler*

Bases: *ISystemEventHandler*

Proxy of C++ *Spinnaker::SystemEventHandler* class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class *PySpin.SystemPtr*(*args)

Bases: *_SWIG_SysPtr*

A reference tracked pointer to a system object.

C++ includes: *SystemPtr.h*

property thisown

The membership flag

PySpin.**System_GetInstance**() → *SystemPtr*

class PySpin.**TIFFOption**

Bases: object

Options for saving TIFF images.

C++ includes: SpinnakerDefs.h

property **compression**

property **reserved**

property **thisown**

The membership flag

PySpin.**ThrowBadAlloc**()

SPINNAKER_API void Spinnaker::GenICam::ThrowBadAlloc()

PySpin.**Tokenize**(*str*, *delimiters*=' ')

Parameters

- **str** (*Spinnaker::GenICam::gcstring const &*)
- **delimiters** (*Spinnaker::GenICam::gcstring const &*)
- **SPINNAKER_API**
- **&str** (*void Spinnaker::GenICam::Tokenize(const gcstring)*)
- **gcstring_vector**
- **&tokens**
- **")** (*const gcstring &delimiters=""*)
- **delimiter** (*splits str input string into a list of tokens using the*)

class PySpin.**TransportLayerDevice**(*nodeMapTLDevice*)

Bases: object

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

C++ includes: TransportLayerDevice.h

property **DeviceAccessStatus**

property **DeviceBootloaderVersion**

property **DeviceCurrentSpeed**

property **DeviceDisplayName**

property **DeviceDriverVersion**

property **DeviceEndiannessMechanism**

property **DeviceID**

property **DeviceInstanceId**

property **DeviceIsUpdater**

property DeviceLinkSpeed
property DeviceLocation
property DeviceModelName
property DeviceMulticastMonitorMode
property DevicePortId
property DeviceReset
property DeviceSerialNumber
property DeviceType
property DeviceU3VProtocol
property DeviceUserID
property DeviceVendorName
property DeviceVersion
property GUIXMLLocation
property GUIXMLPath
property GenICamXMLLocation
property GenICamXMLPath
property GevCCP
property GevDeviceAutoForceIP
property GevDeviceDiscoverMaximumPacketSize
property GevDeviceForceGateway
property GevDeviceForceIP
property GevDeviceForceIPAddress
property GevDeviceForceSubnetMask
property GevDeviceGateway
property GevDeviceIPAddress
property GevDeviceIsWrongSubnet
property GevDeviceMACAddress
property GevDeviceMaximumPacketSize
property GevDeviceMaximumRetryCount
property GevDeviceModeIsBigEndian
property GevDevicePort

property `GevDeviceReadAndWriteTimeout`

property `GevDeviceSubnetMask`

property `GevVersionMajor`

property `GevVersionMinor`

property `StreamID`

property `StreamSelector`

property `thisown`

The membership flag

class `PySpin.TransportLayerInterface(nodeMapTLDevice)`

Bases: `object`

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

C++ includes: `TransportLayerInterface.h`

property `ActionCommand`

property `DeviceAccessStatus`

property `DeviceCount`

property `DeviceID`

property `DeviceModelName`

property `DeviceSelector`

property `DeviceSerialNumber`

property `DeviceUnlock`

property `DeviceUpdateList`

property `DeviceVendorName`

property `FLIRFilterDriverStatus`

property `GevActionAckRequired`

property `GevActionDeviceKey`

property `GevActionGroupKey`

property `GevActionGroupMask`

property `GevActionTime`

property `GevDeviceAutoForceIP`

property `GevDeviceDisableDiscovery`

property `GevDeviceDiscoveryEnabled`

property `GevDeviceEnableDiscovery`

property `GevDeviceForceGateway`
property `GevDeviceForceIP`
property `GevDeviceForceIPAddress`
property `GevDeviceForceSubnetMask`
property `GevDeviceGateway`
property `GevDeviceIPAddress`
property `GevDeviceMACAddress`
property `GevDeviceSubnetMask`
property `GevInterfaceGateway`
property `GevInterfaceGatewaySelector`
property `GevInterfaceIsIPConflict`
property `GevInterfaceMACAddress`
property `GevInterfaceMTU`
property `GevInterfaceReceiveLinkSpeed`
property `GevInterfaceSubnetIPAddress`
property `GevInterfaceSubnetMask`
property `GevInterfaceSubnetSelector`
property `GevInterfaceTransmitLinkSpeed`
property `HostAdapterDriverVersion`
property `HostAdapterName`
property `HostAdapterVendor`
property `IncompatibleDeviceCount`
property `IncompatibleDeviceID`
property `IncompatibleDeviceModelName`
property `IncompatibleDeviceSelector`
property `IncompatibleDeviceVendorName`
property `IncompatibleGevDeviceIPAddress`
property `IncompatibleGevDeviceMACAddress`
property `IncompatibleGevDeviceSubnetMask`
property `InterfaceDisplayName`
property `InterfaceID`

property `InterfaceType`

property `POEStatus`

property `TeledyneGigeVisionFilterDriverStatus`

property `thisown`

The membership flag

class `PySpin.TransportLayerStream`(*nodeMapTLDevice*)

Bases: `object`

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

C++ includes: `TransportLayerStream.h`

property `StreamAnnounceBufferMinimum`

property `StreamAnnouncedBufferCount`

property `StreamBlockTransferSize`

property `StreamBlocksProcessingTimeLast`

property `StreamBlocksProcessingTimeMax`

property `StreamBlocksProcessingTimeMin`

property `StreamBlocksReceptionTimeLast`

property `StreamBlocksReceptionTimeMax`

property `StreamBlocksReceptionTimeMin`

property `StreamBufferAlignment`

property `StreamBufferCountManual`

property `StreamBufferCountMax`

property `StreamBufferCountMode`

property `StreamBufferCountResult`

property `StreamBufferHandlingMode`

property `StreamCRCCheckEnable`

property `StreamChunkCountMaximum`

property `StreamDeliveredFrameCount`

property `StreamDroppedFrameCount`

property `StreamID`

property `StreamIncompleteFrameCount`

property `StreamInputBufferCount`

property `StreamIsGrabbing`

property StreamLostFrameCount
 property StreamMissedPacketCount
 property StreamMode
 property StreamOutputBufferCount
 property StreamPacketResendEnable
 property StreamPacketResendMaxRequests
 property StreamPacketResendReceivedPacketCount
 property StreamPacketResendRequestCount
 property StreamPacketResendRequestTimeoutCount
 property StreamPacketResendRequestedPacketCount
 property StreamPacketResendTimeout
 property StreamPacketsDuplicatedCount
 property StreamPacketsNotYetAvailableCount
 property StreamPacketsPerFrameCount
 property StreamPacketsTemporarilyUnavailableCount
 property StreamPacketsTimeoutCount
 property StreamPacketsUnavailableCount
 property StreamReceivedFrameCount
 property StreamReceivedPacketCount
 property StreamStartedFrameCount
 property StreamType
 property thisown

The membership flag

class PySpin.TransportLayerSystem(*nodeMapTLDevice*)

Bases: object

Part of the QuickSpin API to provide access to system information.

C++ includes: TransportLayerSystem.h

property EnumerateGEVInterfaces
 property EnumerateGen2Cameras
 property EnumerateUSBInterfaces
 property GEVAutoAssignIPEnable
 property GenTlSFNCVersionMajor

property GenTlSFNCVersionMinor
property GenTlSFNCVersionSubMinor
property GenTlVersionMajor
property GenTlVersionMinor
property GevInterfaceDefaultGateway
property GevInterfaceDefaultIPAddress
property GevInterfaceDefaultSubnetMask
property GevInterfaceMACAddress
property GevVersionMajor
property GevVersionMinor
property InterfaceDisplayName
property InterfaceID
property InterfaceSelector
property InterfaceUpdateList
property TlDisplayName
property TlFileName
property TlID
property TlModelName
property TlPath
property TlType
property TlVendorName
property TlVersion
property thisown

The membership flag

PySpin.UpdateFirmware(*args*) → int

Parameters

args (*std::vector< std::string >*)

PySpin.UpdateFirmwareConsole(*numArgs*) → int

Parameters

numArgs (*unsigned int*)

PySpin.UpdateFirmwareGUI(*args*) → int

Parameters

args (*std::string*)

PySpin.UrlDecode(*Input*) → *gcstring*

Parameters

- **Input** (*Spinnaker::GenICam::gcstring const &*)
- **SPINNAKER_API**
- **&Input** (*gcstring Spinnaker::GenICam::UrlDecode(const gcstring)*)
- **equivalent** (*Replaces xx escapes by their char*)

PySpin.UrlEncode(*Input*) → *gcstring*

Parameters

- **Input** (*Spinnaker::GenICam::gcstring const &*)
- **SPINNAKER_API**
- **&Input** (*gcstring Spinnaker::GenICam::UrlEncode(const gcstring)*)
- **xx** (*Converts \ to / and replaces all unsafe characters by their*)
- **equivalent**

class PySpin.ValueNode(**args*, ***kwargs*)

Bases: *IValue*, *Node*

Interface for value properties.

C++ includes: ValueNode.h

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

- **ValueStr** (*The value to set*)
- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Spinnaker::GenApi::ValueNode::FromString(const (virtual void)**
- **&ValueStr** (*GenICam::gcstring*)
- **Verify=true**) (*bool*)
- **string** (*Set content of the node as*)
- **Parameters**
- **-----**
- **ValueStr**
- **Verify**

GetNode(*self*) → *INode*

virtual INode* Spinnaker::GenApi::ValueNode::GetNode()

IsValueCacheValid(*self*) → bool

virtual bool Spinnaker::GenApi::ValueNode::IsValueCacheValid() const

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

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase** (*virtual void Spinnaker::GenApi::ValueNode::SetReference(INode)*
—
- **Value** (*overload SetReference for*)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::ValueNode::ToString**(*bool* (virtual
GenICam::gcstring)
- **Verify=false**
- **IgnoreCache=false**) (*bool*)
- **string** (*Get content of the node as*)
- **Parameters**
- -----
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

property thisown

The membership flag

class **PySpin.Version_t**

Bases: *object*

Version

C++ includes: *GCTypes.h*

property **Major**

property **Minor**

property **SubMinor**

property thisown

The membership flag

class PySpin.**double_autovector_t**(*args)

Bases: object

Vector of doubles with reference counting.

C++ includes: Autovector.h

size(self) → size_t

size_t Spinnaker::GenApi::double_autovector_t::size() const

property thisown

The membership flag

class PySpin.**gcstring**(*args)

Bases: object

Proxy of C++ Spinnaker::GenICam::gcstring class.

append(self, str) → *gcstring*

Parameters

- **str** (Spinnaker::GenICam::gcstring const &)
- **append**(self
- **count** (gcstring& Spinnaker::GenICam::gcstring::append(size_t)
- **gcstring** (ch) ->)
- **count**
- **ch** (char)
- **virtual**
- **count**
- **ch** (char)

assign(self, str) → *gcstring*

Parameters

- **str** (Spinnaker::GenICam::gcstring const &)
- **assign**(self
- **count** (size_t)
- **gcstring** (n) ->)
- **count**
- **ch** (char)
- **assign**(self
- **gcstring**
- **pc** (char const *)
- **assign**(self
- **pc**
- **gcstring**

- **pc**
- **n** (*size_t*)
- **virtual**
- ***pc** (*gcstring& Spinnaker::GenICam::gcstring::assign(const char) –*
- **size_t**
- **n**)

c_str(*self*) → *char const **

virtual const char Spinnaker::GenICam::gcstring::c_str(void) const*

compare(*self, str*) → *int*

Parameters

- **str** (*Spinnaker::GenICam::gcstring const &*)
- **virtual**
- **const** (*int Spinnaker::GenICam::gcstring::compare(const gcstring &str)*)

empty(*self*) → *bool*

virtual bool Spinnaker::GenICam::gcstring::empty(void) const

find(*self, ch, offset=0*) → *size_t*

Parameters

- **ch** (*char*)
- **offset** (*size_t*)
- **find**(*self*
- **str** (*Spinnaker::GenICam::gcstring const &*)
- **size_t** (*count*) →)
- **str**
- **offset**
- **find**(*self*
- **str**
- **offset**
- **size_t**
- **str**
- **offset**
- **count** (*size_t*)
- **find**(*self*
- **pc** (*char const **)
- **size_t**
- **pc**

- `offset`
- `find(self`
- `pc`
- `offset`
- `size_t`
- `pc`
- `offset`
- `count`
- `virtual`
- `*pc(size_t Spinnaker::GenICam::gcstring::find(const char) –`
- `size_t`
- `offset`
- `const(size_t count))`

`find_first_not_of(self, str, offset=0) → size_t`

Parameters

- `str` (`Spinnaker::GenICam::gcstring const &`)
- `offset` (`size_t`)
- `Spinnaker::GenICam::gcstring::find_first_not_of(const (virtual size_t)`
- `&str` (`gcstring`)
- `const(size_t offset=0))`

`find_first_of(self, str, offset=0) → size_t`

Parameters

- `str` (`Spinnaker::GenICam::gcstring const &`)
- `offset` (`size_t`)
- `Spinnaker::GenICam::gcstring::find_first_of(const (virtual size_t)`
- `&str` (`gcstring`)
- `const(size_t offset=0))`

`length(self) → size_t`

`virtual size_t Spinnaker::GenICam::gcstring::length(void) const`

`max_size(self) → size_t`

`virtual size_t Spinnaker::GenICam::gcstring::max_size() const`

`npos = 18446744073709551615`

`resize(self, n)`

Parameters

- `n` (`size_t`)

- **virtual**

- **n** (*void Spinnaker::GenICam::gcstring::resize(size_t)*)

size(*self*) → *size_t*

virtual size_t Spinnaker::GenICam::gcstring::size(void) const

substr(*self*, *offset=0*, *count=size_t(-1)*) → *gcstring*

Parameters

- **offset** (*size_t*)

- **count** (*size_t*)

- **virtual**

- **offset=0** (*gcstring Spinnaker::GenICam::gcstring::substr(size_t)*)

- **size_t**

- **const** (*count=GCSTRING_NPOS*)

swap(*self*, *Right*)

Parameters

- **Right** (*Spinnaker::GenICam::gcstring &*)

- **virtual**

- **&Right** (*void Spinnaker::GenICam::gcstring::swap(gcstring)*)

property thisown

The membership flag

PySpin.gcstring__npos() → *size_t*

class **PySpin.int64_autovector_t**(*args)

Bases: object

Vector of integers with reference counting.

C++ includes: Autovector.h

size(*self*) → *size_t*

size_t Spinnaker::GenApi::int64_autovector_t::size() const

property thisown

The membership flag

class **PySpin.node_vector**(*args)

Bases: object

Proxy of C++ Spinnaker::GenApi::node_vector class.

assign(*self*, *n*, *val*)

Parameters

- **n** (*size_t*)

- **val** (*Spinnaker::GenApi::node_vector::T const &*)

at(*self*, *uiIndex*) → *INode*

Parameters

- **uiIndex** (*size_t*)
- **at**(*self*
- **INode** (*uiIndex*) ->)
- **uiIndex**

back(*self*) → *INode*

back(*self*) → *INode*

begin(*self*) → Spinnaker::GenApi::node_vector::iterator

begin(*self*) → Spinnaker::GenApi::node_vector::const_iterator

capacity(*self*) → *size_t*

clear(*self*)

empty(*self*) → bool

end(*self*) → Spinnaker::GenApi::node_vector::iterator

end(*self*) → Spinnaker::GenApi::node_vector::const_iterator

erase(*self*, *pos*) → Spinnaker::GenApi::node_vector::iterator

Parameters

- **pos** (Spinnaker::GenApi::node_vector::iterator)
- **erase**(*self*
- **uiIndex**)
- **uiIndex** (*size_t*)

front(*self*) → *INode*

front(*self*) → *INode*

insert(*self*, *pos*, *val*) → Spinnaker::GenApi::node_vector::iterator

Parameters

- **pos** (Spinnaker::GenApi::node_vector::iterator)
- **val** (Spinnaker::GenApi::node_vector::T const &)
- **insert**(*self*
- **uiIndex** (*size_t*)
- **val**)
- **uiIndex**
- **val**

max_size(*self*) → *size_t*

pop_back(*self*)

push_back(*self*, *val*)

Parameters

val (Spinnaker::GenApi::node_vector::T const &)

reserve(*self*, *uiSize*)

Parameters

uiSize (size_t)

resize(*self*, *uiSize*)

Parameters

uiSize (size_t)

size(*self*) → size_t

property thisown

The membership flag

class PySpin.**value_vector**(*args)

Bases: object

Proxy of C++ Spinnaker::GenApi::value_vector class.

assign(*self*, *n*, *obj*)

Parameters

- **n** (size_t)
- **obj** (Spinnaker::GenApi::value_vector::T const &)

at(*self*, *uiIndex*) → *IValue*

Parameters

- **uiIndex** (size_t)
- **at**(*self*
- **IValue** (*uiIndex*) ->)
- **uiIndex**

back(*self*) → *IValue*

back(*self*) → *IValue*

begin(*self*) → Spinnaker::GenApi::value_vector::iterator

begin(*self*) → Spinnaker::GenApi::value_vector::const_iterator

capacity(*self*) → size_t

clear(*self*)

empty(*self*) → bool

end(*self*) → Spinnaker::GenApi::value_vector::iterator

end(*self*) → Spinnaker::GenApi::value_vector::const_iterator

erase(*self*, *pos*) → Spinnaker::GenApi::value_vector::iterator

Parameters

- **pos** (*Spinnaker::GenApi::value_vector::iterator*)
- **erase**(*self*
- **uiIndex**)
- **uiIndex** (*size_t*)

front(*self*) → *IValue*

front(*self*) → *IValue*

insert(*self*, *pos*, *val*) → Spinnaker::GenApi::value_vector::iterator

Parameters

- **pos** (*Spinnaker::GenApi::value_vector::iterator*)
- **val** (*Spinnaker::GenApi::value_vector::T const &*)
- **insert**(*self*
- **uiIndex** (*size_t*)
- **val**)
- **uiIndex**
- **val**

max_size(*self*) → *size_t*

pop_back(*self*)

push_back(*self*, *val*)

Parameters

- **val** (*Spinnaker::GenApi::value_vector::T const &*)

reserve(*self*, *uiSize*)

Parameters

- **uiSize** (*size_t*)

resize(*self*, *uiSize*, *val*)

Parameters

- **uiSize** (*size_t*)
- **val** (*Spinnaker::GenApi::value_vector::T const &*)

size(*self*) → *size_t*

property thisown

The membership flag

PYTHON MODULE INDEX

p

PySpin, [83](#)

A

- AasRoiEnable (*PySpin.Camera* property), 10, 117
- AasRoiHeight (*PySpin.Camera* property), 10, 117
- AasRoiOffsetX (*PySpin.Camera* property), 10, 117
- AasRoiOffsetY (*PySpin.Camera* property), 10, 117
- AasRoiWidth (*PySpin.Camera* property), 10, 117
- AcquisitionAbort (*PySpin.Camera* property), 10, 117
- AcquisitionArm (*PySpin.Camera* property), 10, 117
- AcquisitionBurstFrameCount (*PySpin.Camera* property), 10, 117
- AcquisitionFrameCount (*PySpin.Camera* property), 10, 117
- AcquisitionFrameRate (*PySpin.Camera* property), 10, 117
- AcquisitionFrameRateEnable (*PySpin.Camera* property), 10, 118
- AcquisitionFrameRatePersistence (*PySpin.Camera* property), 10, 118
- AcquisitionLineRate (*PySpin.Camera* property), 10, 118
- AcquisitionMode (*PySpin.Camera* property), 10, 118
- AcquisitionResultingFrameRate (*PySpin.Camera* property), 10, 118
- AcquisitionStart (*PySpin.Camera* property), 10, 118
- AcquisitionStatus (*PySpin.Camera* property), 10, 118
- AcquisitionStatusSelector (*PySpin.Camera* property), 10, 118
- AcquisitionStop (*PySpin.Camera* property), 10, 118
- AcquisitionTransferFrameRate (*PySpin.Camera* property), 11, 118
- ActionCommand (*PySpin.TransportLayerInterface* property), 79, 412
- ActionCommandResult (class in *PySpin*), 83
- ActionDeviceKey (*PySpin.Camera* property), 11, 118
- ActionGroupKey (*PySpin.Camera* property), 11, 118
- ActionGroupMask (*PySpin.Camera* property), 11, 118
- ActionQueueEmpty (*PySpin.Camera* property), 11, 118
- ActionQueueSize (*PySpin.Camera* property), 11, 118
- ActionSelector (*PySpin.Camera* property), 11, 118
- ActionSignalSize (*PySpin.Camera* property), 11, 118
- ActionUnconditionalMode (*PySpin.Camera* property), 11, 118
- AdaptiveCompressionEnable (*PySpin.Camera* property), 11, 118
- AdcBitDepth (*PySpin.Camera* property), 11, 118
- Add() (*PySpin.CameraList* method), 39, 147
- Add() (*PySpin.ICameraList* method), 176
- Add() (*PySpin.IImageList* method), 330
- Add() (*PySpin.IInterfaceList* method), 335
- Add() (*PySpin.ImageList* method), 55, 354
- Add() (*PySpin.InterfaceList* method), 67, 379
- AddPoint() (*PySpin.IPointCloud* method), 340
- AddPoint() (*PySpin.PointCloud* method), 68, 398
- aPAUSEMACtrlFramesReceived (*PySpin.Camera* property), 35, 142
- aPAUSEMACtrlFramesTransmitted (*PySpin.Camera* property), 35, 142
- Append() (*PySpin.CameraList* method), 39, 147
- append() (*PySpin.gcstring* method), 419
- Append() (*PySpin.ICameraList* method), 176
- Append() (*PySpin.IImageList* method), 330
- Append() (*PySpin.IInterfaceList* method), 335
- Append() (*PySpin.ImageList* method), 55, 354
- Append() (*PySpin.InterfaceList* method), 67, 379
- Append() (*PySpin.SpinVideo* method), 69, 402
- Application (*PySpin.CCMSettings* property), 87
- ApplicationToString() (*PySpin.ImageUtilityCCM* static method), 59, 359
- ApplyGamma() (*PySpin.IImageProcessor* method), 331
- ApplyGamma() (*PySpin.ImageProcessor* method), 56, 355
- assign() (*PySpin.gcstring* method), 419
- assign() (*PySpin.node_vector* method), 422
- assign() (*PySpin.value_vector* method), 424
- at() (*PySpin.node_vector* method), 422
- at() (*PySpin.value_vector* method), 424
- AutoAlgorithmSelector (*PySpin.Camera* property), 11, 118
- AutoExposureControlLoopDamping (*PySpin.Camera* property), 11, 118
- AutoExposureControlPriority (*PySpin.Camera* property), 11, 118
- AutoExposureEVCompensation (*PySpin.Camera* property), 11, 118

erty), 11, 118
 AutoExposureExposureTimeLowerLimit (PySpin.Camera property), 11, 118
 AutoExposureExposureTimeUpperLimit (PySpin.Camera property), 11, 118
 AutoExposureGainLowerLimit (PySpin.Camera property), 11, 118
 AutoExposureGainUpperLimit (PySpin.Camera property), 11, 118
 AutoExposureGreyValueLowerLimit (PySpin.Camera property), 11, 118
 AutoExposureGreyValueUpperLimit (PySpin.Camera property), 11, 118
 AutoExposureLightingMode (PySpin.Camera property), 11, 118
 AutoExposureMeteringMode (PySpin.Camera property), 11, 119
 AutoExposureTargetGreyValue (PySpin.Camera property), 11, 119
 AutoExposureTargetGreyValueAuto (PySpin.Camera property), 11, 119
 AVIOption (class in PySpin), 83

B

b (PySpin.Stereo3DPoint property), 403
 back() (PySpin.node_vector method), 423
 back() (PySpin.value_vector method), 424
 BalanceRatio (PySpin.Camera property), 11, 119
 BalanceRatioSelector (PySpin.Camera property), 11, 119
 BalanceWhiteAuto (PySpin.Camera property), 11, 119
 BalanceWhiteAutoDamping (PySpin.Camera property), 11, 119
 BalanceWhiteAutoLowerLimit (PySpin.Camera property), 11, 119
 BalanceWhiteAutoProfile (PySpin.Camera property), 11, 119
 BalanceWhiteAutoUpperLimit (PySpin.Camera property), 12, 119
 baseline (PySpin.StereoCameraParameters property), 403
 begin() (PySpin.node_vector method), 423
 begin() (PySpin.value_vector method), 424
 BeginAcquisition() (PySpin.CameraBase method), 35, 142
 BeginAcquisition() (PySpin.ICameraBase method), 174
 binaryFile (PySpin.PGMOption property), 397
 binaryFile (PySpin.PPMOption property), 397
 BinningHorizontal (PySpin.Camera property), 12, 119
 BinningHorizontalMode (PySpin.Camera property), 12, 119
 BinningSelector (PySpin.Camera property), 12, 119

BinningVertical (PySpin.Camera property), 12, 119
 BinningVerticalMode (PySpin.Camera property), 12, 119
 bitrate (PySpin.H264Option property), 173
 BlackLevel (PySpin.Camera property), 12, 119
 BlackLevelAuto (PySpin.Camera property), 12, 119
 BlackLevelAutoBalance (PySpin.Camera property), 12, 119
 BlackLevelClampingEnable (PySpin.Camera property), 12, 119
 BlackLevelRaw (PySpin.Camera property), 12, 119
 BlackLevelSelector (PySpin.Camera property), 12, 119
 BMPOption (class in PySpin), 83
 BooleanNode (class in PySpin), 83
 bottomRightXCoord (PySpin.InferenceBoxRect property), 375
 bottomRightXCoord (PySpin.InferenceBoxRotatedRect property), 375
 bottomRightYCoord (PySpin.InferenceBoxRect property), 375
 bottomRightYCoord (PySpin.InferenceBoxRotatedRect property), 375
 boxType (PySpin.InferenceBoundingBox property), 374
 BsiFlatFieldCorrectionAuto (PySpin.Camera property), 12, 119
 BsiFlatFieldCorrectionAutoDamping (PySpin.Camera property), 12, 119
 BsiFlatFieldCorrectionEnable (PySpin.Camera property), 12, 119
 BsiFlatFieldCorrectionGain (PySpin.Camera property), 12, 119
 BsiFlatFieldCorrectionGainSelector (PySpin.Camera property), 12, 119
 BufferedBurstFrameCountMax (PySpin.Camera property), 12, 119
 BufferedBurstMode (PySpin.Camera property), 12, 119
 build (PySpin.LibraryVersion property), 385

C

c_str() (PySpin.gcstring method), 420
 CalculateChannelStatistics() (PySpin.IImage method), 326
 CalculateStatistics() (PySpin.IImage method), 326
 CallbackFunction() (PySpin.NodeCallback method), 392
 Camera (class in PySpin), 10, 117
 CameraBase (class in PySpin), 35, 142
 CameraList (class in PySpin), 39, 147
 CameraPtr (class in PySpin), 41, 149
 capacity() (PySpin.node_vector method), 423
 capacity() (PySpin.value_vector method), 424
 CategoryNode (class in PySpin), 149

- CBasePtr (class in PySpin), 10, 85
- CBooleanPtr (class in PySpin), 85
- CCategoryPtr (class in PySpin), 88
- CCMSettings (class in PySpin), 87
- CCommandPtr (class in PySpin), 90
- CDeviceInfoPtr (class in PySpin), 93
- centerXCoord (PySpin.InferenceBoxCircle property), 375
- centerYCoord (PySpin.InferenceBoxCircle property), 375
- CEnumEntryPtr (class in PySpin), 93
- CEnumerationPtr (class in PySpin), 96
- CFeatureBag (class in PySpin), 99
- CFloatPtr (class in PySpin), 100
- channel (PySpin.ChannelStatistics property), 42, 150
- ChannelStatistics (class in PySpin), 42, 149
- CheckCRC() (PySpin.Image method), 326
- CheckCRC() (PySpin.Image method), 46, 345
- ChunkBlackLevel (PySpin.Camera property), 12, 119
- ChunkBlackLevelSelector (PySpin.Camera property), 12, 119
- ChunkCompressionMode (PySpin.Camera property), 12, 120
- ChunkCompressionRatio (PySpin.Camera property), 12, 120
- ChunkCounterSelector (PySpin.Camera property), 12, 120
- ChunkCounterValue (PySpin.Camera property), 12, 120
- ChunkCRC (PySpin.Camera property), 12, 119
- ChunkCurrentDatarate (PySpin.Camera property), 12, 120
- ChunkData (class in PySpin), 42, 150
- ChunkEnable (PySpin.Camera property), 12, 120
- ChunkEncoderSelector (PySpin.Camera property), 12, 120
- ChunkEncoderStatus (PySpin.Camera property), 12, 120
- ChunkEncoderValue (PySpin.Camera property), 12, 120
- ChunkExposureEndLineStatusAll (PySpin.Camera property), 13, 120
- ChunkExposureTime (PySpin.Camera property), 13, 120
- ChunkExposureTimeSelector (PySpin.Camera property), 13, 120
- ChunkFrameID (PySpin.Camera property), 13, 120
- ChunkGain (PySpin.Camera property), 13, 120
- ChunkGainSelector (PySpin.Camera property), 13, 120
- ChunkHeight (PySpin.Camera property), 13, 120
- ChunkImage (PySpin.Camera property), 13, 120
- ChunkImageComponent (PySpin.Camera property), 13, 120
- ChunkInferenceBoundingBoxResult (PySpin.Camera property), 13, 120
- ChunkInferenceConfidence (PySpin.Camera property), 13, 120
- ChunkInferenceFrameId (PySpin.Camera property), 13, 120
- ChunkInferenceResult (PySpin.Camera property), 13, 120
- ChunkLinePitch (PySpin.Camera property), 13, 120
- ChunkLineStatusAll (PySpin.Camera property), 13, 120
- ChunkModeActive (PySpin.Camera property), 13, 120
- ChunkOffsetX (PySpin.Camera property), 13, 120
- ChunkOffsetY (PySpin.Camera property), 13, 120
- ChunkPartSelector (PySpin.Camera property), 13, 120
- ChunkPixelDynamicRangeMax (PySpin.Camera property), 13, 120
- ChunkPixelDynamicRangeMin (PySpin.Camera property), 13, 120
- ChunkPixelFormat (PySpin.Camera property), 13, 120
- ChunkRegionID (PySpin.Camera property), 13, 121
- ChunkScan3dAxisMax (PySpin.Camera property), 13, 121
- ChunkScan3dAxisMin (PySpin.Camera property), 13, 121
- ChunkScan3dCoordinateOffset (PySpin.Camera property), 13, 121
- ChunkScan3dCoordinateReferenceSelector (PySpin.Camera property), 13, 121
- ChunkScan3dCoordinateReferenceValue (PySpin.Camera property), 13, 121
- ChunkScan3dCoordinateScale (PySpin.Camera property), 13, 121
- ChunkScan3dCoordinateSelector (PySpin.Camera property), 13, 121
- ChunkScan3dCoordinateSystem (PySpin.Camera property), 13, 121
- ChunkScan3dCoordinateSystemReference (PySpin.Camera property), 14, 121
- ChunkScan3dCoordinateTransformSelector (PySpin.Camera property), 14, 121
- ChunkScan3dDistanceUnit (PySpin.Camera property), 14, 121
- ChunkScan3dInvalidDataFlag (PySpin.Camera property), 14, 121
- ChunkScan3dInvalidDataValue (PySpin.Camera property), 14, 121
- ChunkScan3dOutputMode (PySpin.Camera property), 14, 121
- ChunkScan3dTransformValue (PySpin.Camera property), 14, 121
- ChunkScanLineSelector (PySpin.Camera property), 14, 121

- ChunkSelector (*PySpin.Camera* property), 14, 121
- ChunkSequencerSetActive (*PySpin.Camera* property), 14, 121
- ChunkSerialData (*PySpin.Camera* property), 14, 121
- ChunkSerialDataLength (*PySpin.Camera* property), 14, 121
- ChunkSerialReceiveOverflow (*PySpin.Camera* property), 14, 121
- ChunkSourceID (*PySpin.Camera* property), 14, 121
- ChunkStreamChannelID (*PySpin.Camera* property), 14, 121
- ChunkTimerSelector (*PySpin.Camera* property), 14, 121
- ChunkTimerValue (*PySpin.Camera* property), 14, 121
- ChunkTimestamp (*PySpin.Camera* property), 14, 121
- ChunkTimestampLatchValue (*PySpin.Camera* property), 14, 121
- ChunkTransferBlockID (*PySpin.Camera* property), 14, 121
- ChunkTransferQueueCurrentBlockCount (*PySpin.Camera* property), 14, 121
- ChunkTransferStreamID (*PySpin.Camera* property), 14, 121
- ChunkWidth (*PySpin.Camera* property), 14, 122
- CIntegerPtr (class in *PySpin*), 100
- circle (*PySpin.InferenceBoundingBox* property), 374
- classId (*PySpin.InferenceBoundingBox* property), 374
- ClConfiguration (*PySpin.Camera* property), 14, 122
- Clear() (*PySpin.CameraList* method), 40, 147
- Clear() (*PySpin.ICameraList* method), 177
- Clear() (*PySpin.IImageList* method), 330
- Clear() (*PySpin.IInterfaceList* method), 335
- Clear() (*PySpin.ImageList* method), 55, 354
- Clear() (*PySpin.InterfaceList* method), 67, 379
- clear() (*PySpin.node_vector* method), 423
- clear() (*PySpin.value_vector* method), 424
- ClearAllNodes() (*PySpin.CNodeMapDynPtr* method), 103
- ClearAllNodes() (*PySpin.INodeMapDyn* method), 338
- ClearXMLCache() (*PySpin.NodeMap* static method), 392
- Close() (*PySpin.SpinVideo* method), 69, 402
- ClTimeSlotsCount (*PySpin.Camera* property), 14, 122
- CNodeMapDynPtr (class in *PySpin*), 103
- CNodeMapPtr (class in *PySpin*), 105
- CNodePtr (class in *PySpin*), 106
- ColorSpace (*PySpin.CCMSettings* property), 88
- ColorSpaceToString() (*PySpin.ImageUtilityCCM* static method), 60, 359
- ColorTemperature (*PySpin.CCMSettings* property), 88
- ColorTemperatureToString() (*PySpin.ImageUtilityCCM* static method), 60, 359
- ColorTransformationEnable (*PySpin.Camera* property), 14, 122
- ColorTransformationSelector (*PySpin.Camera* property), 14, 122
- ColorTransformationValue (*PySpin.Camera* property), 14, 122
- ColorTransformationValueSelector (*PySpin.Camera* property), 14, 122
- Combine() (in module *PySpin*), 153
- CommandNode (class in *PySpin*), 154
- compare() (*PySpin.gcstring* method), 420
- ComponentActiveCount (*PySpin.Camera* property), 14, 122
- ComponentDestination (*PySpin.Camera* property), 14, 122
- ComponentEnable (*PySpin.Camera* property), 15, 122
- ComponentSelector (*PySpin.Camera* property), 15, 122
- CompressedFrameDropCount (*PySpin.Camera* property), 15, 122
- compression (*PySpin.TIFFOption* property), 410
- compressionLevel (*PySpin.PNGOption* property), 397
- CompressionSaturationPriority (*PySpin.Camera* property), 15, 122
- Compute3DPointFromPixel() (*PySpin.ImageUtilityStereo* static method), 64, 367
- ComputeDistanceBetweenPoints() (*PySpin.ImageUtilityStereo* static method), 64, 367
- ComputeDistanceToPoint() (*PySpin.ImageUtilityStereo* static method), 64, 367
- ComputePointCloud() (*PySpin.ImageUtilityStereo* static method), 64, 367
- confidence (*PySpin.DeviceEventInferenceData* property), 156
- confidence (*PySpin.InferenceBoundingBox* property), 374
- Connect() (*PySpin.CNodeMapDynPtr* method), 103
- Connect() (*PySpin.CNodeMapPtr* method), 105
- Connect() (*PySpin.INodeMap* method), 337
- Connect() (*PySpin.NodeMap* method), 392
- ControlPacketsReservedBandwidth (*PySpin.Camera* property), 15, 122
- Convert() (*PySpin.IImageProcessor* method), 332
- Convert() (*PySpin.ImageProcessor* method), 57, 356
- coordinateOffset (*PySpin.StereoCameraParameters* property), 403
- CounterDelay (*PySpin.Camera* property), 15, 122
- CounterDuration (*PySpin.Camera* property), 15, 122
- CounterEventActivation (*PySpin.Camera* property), 15, 122
- CounterEventSource (*PySpin.Camera* property), 15,

- 122
- CounterReset (*PySpin.Camera* property), 15, 122
- CounterResetActivation (*PySpin.Camera* property), 15, 122
- CounterResetSource (*PySpin.Camera* property), 15, 122
- CounterSelector (*PySpin.Camera* property), 15, 122
- CounterStatus (*PySpin.Camera* property), 15, 122
- CounterTriggerActivation (*PySpin.Camera* property), 15, 122
- CounterTriggerSource (*PySpin.Camera* property), 15, 122
- CounterValue (*PySpin.Camera* property), 15, 122
- CounterValueAtReset (*PySpin.Camera* property), 15, 122
- Create() (*PySpin.Image* static method), 46, 345
- CreateAolp() (*PySpin.ImageUtilityPolarization* static method), 62, 362
- CreateColorCorrected() (*PySpin.ImageUtilityCCM* static method), 60, 359
- CreateDepthImage() (*PySpin.ImageUtilityStereo* static method), 65, 368
- CreateDolp() (*PySpin.ImageUtilityPolarization* static method), 62, 363
- CreateGlareReduced() (*PySpin.ImageUtilityPolarization* static method), 62, 363
- CreateHeatmap() (*PySpin.ImageUtilityHeatmap* static method), 60, 360
- CreateNormalized() (*PySpin.ImageUtility* static method), 58, 357
- CreateScaled() (*PySpin.ImageUtility* static method), 59, 358
- CreateStokesS0() (*PySpin.ImageUtilityPolarization* static method), 62, 363
- CreateStokesS1() (*PySpin.ImageUtilityPolarization* static method), 63, 364
- CreateStokesS2() (*PySpin.ImageUtilityPolarization* static method), 63, 364
- CRegisterPtr (class in *PySpin*), 108
- crf (*PySpin.H264Option* property), 173
- CSelectorPtr (class in *PySpin*), 111
- CSelectorSet (class in *PySpin*), 111
- CStringPtr (class in *PySpin*), 112
- CustomCCMCode (*PySpin.CCMSettings* property), 88
- CValuePtr (class in *PySpin*), 115
- CxpConnectionSelector (*PySpin.Camera* property), 15, 122
- CxpConnectionTestErrorCount (*PySpin.Camera* property), 15, 122
- CxpConnectionTestMode (*PySpin.Camera* property), 15, 122
- CxpConnectionTestPacketCount (*PySpin.Camera* property), 15, 122
- CxpLinkConfiguration (*PySpin.Camera* property), 15, 123
- CxpLinkConfigurationPreferred (*PySpin.Camera* property), 15, 123
- CxpLinkConfigurationStatus (*PySpin.Camera* property), 15, 123
- CxpPoCxpAuto (*PySpin.Camera* property), 15, 123
- CxpPoCxpStatus (*PySpin.Camera* property), 15, 123
- CxpPoCxpTripReset (*PySpin.Camera* property), 15, 123
- CxpPoCxpTurnOff (*PySpin.Camera* property), 15, 123
- ## D
- decimationFactor (*PySpin.PointCloudParameters* property), 398
- DecimationHorizontal (*PySpin.Camera* property), 15, 123
- DecimationHorizontalMode (*PySpin.Camera* property), 15, 123
- DecimationSelector (*PySpin.Camera* property), 16, 123
- DecimationVertical (*PySpin.Camera* property), 16, 123
- DecimationVerticalMode (*PySpin.Camera* property), 16, 123
- DeepCopy() (*PySpin.IImage* method), 326
- DeepCopy() (*PySpin.Image* method), 47, 346
- DefectCorrectionMode (*PySpin.Camera* property), 16, 123
- DefectCorrectStaticEnable (*PySpin.Camera* property), 16, 123
- DefectTableApply (*PySpin.Camera* property), 16, 123
- DefectTableCoordinateX (*PySpin.Camera* property), 16, 123
- DefectTableCoordinateY (*PySpin.Camera* property), 16, 123
- DefectTableFactoryRestore (*PySpin.Camera* property), 16, 123
- DefectTableIndex (*PySpin.Camera* property), 16, 123
- DefectTablePixelCount (*PySpin.Camera* property), 16, 123
- DefectTableSave (*PySpin.Camera* property), 16, 123
- DefectTableSensor (*PySpin.Camera* property), 16, 123
- DeInit() (*PySpin.CameraBase* method), 35, 142
- DeInit() (*PySpin.ICameraBase* method), 174
- Deinterlacing (*PySpin.Camera* property), 16, 123
- DeregisterCallback() (*PySpin.CBooleanPtr* method), 85
- DeregisterCallback() (*PySpin.CCategoryPtr* method), 88
- DeregisterCallback() (*PySpin.CCommandPtr* method), 90

DeregisterCallback()	(PySpin.CEnumEntryPtr method), 93	DeviceEventExposureEndData (class in PySpin), 155
DeregisterCallback()	(PySpin.CEnumerationPtr method), 96	DeviceEventHandler (class in PySpin), 5, 155
DeregisterCallback()	(PySpin.CIntegerPtr method), 100	DeviceEventInferenceData (class in PySpin), 155
DeregisterCallback()	(PySpin.CNodePtr method), 106	DeviceFamilyName (PySpin.Camera property), 16, 123
DeregisterCallback()	(PySpin.CRegisterPtr method), 108	DeviceFeaturePersistenceEnd (PySpin.Camera property), 16, 124
DeregisterCallback()	(PySpin.CStringPtr method), 112	DeviceFeaturePersistenceStart (PySpin.Camera property), 16, 124
DeregisterCallback()	(PySpin.CValuePtr method), 115	DeviceFirmwareVersion (PySpin.Camera property), 16, 124
DeregisterCallback()	(PySpin.INode method), 336	DeviceGenCPVersionMajor (PySpin.Camera property), 16, 124
DeregisterCallback()	(PySpin.Node method), 387	DeviceGenCPVersionMinor (PySpin.Camera property), 16, 124
DeregisterNodeCallback()	(in module PySpin), 155	DeviceID (PySpin.Camera property), 16, 124
Destroy()	(PySpin.IDestroy method), 179	DeviceID (PySpin.TransportLayerDevice property), 77, 410
Destroy()	(PySpin.NodeMap method), 393	DeviceID (PySpin.TransportLayerInterface property), 79, 412
DeviceAccessStatus	(PySpin.TransportLayerDevice property), 77, 410	DeviceIndicatorMode (PySpin.Camera property), 16, 124
DeviceAccessStatus	(PySpin.TransportLayerInterface property), 79, 412	DeviceInstanceId (PySpin.TransportLayerDevice property), 77, 410
DeviceAddress	(PySpin.ActionCommandResult property), 83	DeviceIsUpdater (PySpin.TransportLayerDevice property), 77, 410
DeviceArrivalEventHandler	(class in PySpin), 5, 155	DeviceLinkBandwidthReserve (PySpin.Camera property), 16, 124
DeviceBootloaderVersion	(PySpin.TransportLayerDevice property), 77, 410	DeviceLinkCommandTimeout (PySpin.Camera property), 16, 124
DeviceCharacterSet	(PySpin.Camera property), 16, 123	DeviceLinkConnectionCount (PySpin.Camera property), 17, 124
DeviceClockFrequency	(PySpin.Camera property), 16, 123	DeviceLinkCurrentThroughput (PySpin.Camera property), 17, 124
DeviceClockSelector	(PySpin.Camera property), 16, 123	DeviceLinkHeartbeatMode (PySpin.Camera property), 17, 124
DeviceConnectionSelector	(PySpin.Camera property), 16, 123	DeviceLinkHeartbeatTimeout (PySpin.Camera property), 17, 124
DeviceConnectionSpeed	(PySpin.Camera property), 16, 123	DeviceLinkSelector (PySpin.Camera property), 17, 124
DeviceConnectionStatus	(PySpin.Camera property), 16, 123	DeviceLinkSpeed (PySpin.Camera property), 17, 124
DeviceCount	(PySpin.TransportLayerInterface property), 79, 412	DeviceLinkSpeed (PySpin.TransportLayerDevice property), 77, 410
DeviceCurrentSpeed	(PySpin.TransportLayerDevice property), 77, 410	DeviceLinkThroughputLimit (PySpin.Camera property), 17, 124
DeviceDisplayName	(PySpin.TransportLayerDevice property), 77, 410	DeviceLinkThroughputLimitMode (PySpin.Camera property), 17, 124
DeviceDriverVersion	(PySpin.TransportLayerDevice property), 77, 410	DeviceLocation (PySpin.TransportLayerDevice property), 77, 411
DeviceEndianessMechanism	(PySpin.TransportLayerDevice property), 77, 410	DeviceManifestEntrySelector (PySpin.Camera property), 17, 124
DeviceEventChannelCount	(PySpin.Camera property), 16, 123	DeviceManifestPrimaryURL (PySpin.Camera property), 17, 124
		DeviceManifestSchemaMajorVersion (PySpin.Camera property), 17, 124

DeviceManifestSchemaMinorVersion (PySpin.Camera property), 17, 124
 DeviceManifestSecondaryURL (PySpin.Camera property), 17, 124
 DeviceManifestXMLMajorVersion (PySpin.Camera property), 17, 124
 DeviceManifestXMLMinorVersion (PySpin.Camera property), 17, 124
 DeviceManifestXMLSubMinorVersion (PySpin.Camera property), 17, 124
 DeviceManufacturerInfo (PySpin.Camera property), 17, 124
 DeviceMaxThroughput (PySpin.Camera property), 17, 124
 DeviceModelName (PySpin.Camera property), 17, 124
 DeviceModelName (PySpin.TransportLayerDevice property), 77, 411
 DeviceModelName (PySpin.TransportLayerInterface property), 79, 412
 DeviceMulticastMonitorMode (PySpin.TransportLayerDevice property), 77, 411
 DevicePortId (PySpin.TransportLayerDevice property), 77, 411
 DevicePowerSupplySelector (PySpin.Camera property), 17, 124
 DeviceRegistersCheck (PySpin.Camera property), 17, 124
 DeviceRegistersEndianness (PySpin.Camera property), 17, 124
 DeviceRegistersStreamingEnd (PySpin.Camera property), 17, 125
 DeviceRegistersStreamingStart (PySpin.Camera property), 17, 125
 DeviceRegistersValid (PySpin.Camera property), 17, 125
 DeviceRemovalEventHandler (class in PySpin), 6, 156
 DeviceReset (PySpin.Camera property), 17, 125
 DeviceReset (PySpin.TransportLayerDevice property), 77, 411
 DeviceScanType (PySpin.Camera property), 17, 125
 DeviceSelector (PySpin.TransportLayerInterface property), 79, 412
 DeviceSensorChroma (PySpin.Camera property), 17, 125
 DeviceSerialNumber (PySpin.Camera property), 18, 125
 DeviceSerialNumber (PySpin.TransportLayerDevice property), 77, 411
 DeviceSerialNumber (PySpin.TransportLayerInterface property), 79, 412
 DeviceSerialPortBaudRate (PySpin.Camera property), 18, 125
 DeviceSerialPortSelector (PySpin.Camera property), 18, 125
 DeviceSFNCVersionMajor (PySpin.Camera property), 17, 125
 DeviceSFNCVersionMinor (PySpin.Camera property), 17, 125
 DeviceSFNCVersionSubMinor (PySpin.Camera property), 17, 125
 DeviceStreamChannelCount (PySpin.Camera property), 18, 125
 DeviceStreamChannelEndianness (PySpin.Camera property), 18, 125
 DeviceStreamChannelLink (PySpin.Camera property), 18, 125
 DeviceStreamChannelPacketSize (PySpin.Camera property), 18, 125
 DeviceStreamChannelSelector (PySpin.Camera property), 18, 125
 DeviceStreamChannelType (PySpin.Camera property), 18, 125
 DeviceTapGeometry (PySpin.Camera property), 18, 125
 DeviceTemperature (PySpin.Camera property), 18, 125
 DeviceTemperatureSelector (PySpin.Camera property), 18, 125
 DeviceTLType (PySpin.Camera property), 18, 125
 DeviceTLVersionMajor (PySpin.Camera property), 18, 125
 DeviceTLVersionMinor (PySpin.Camera property), 18, 125
 DeviceTLVersionSubMinor (PySpin.Camera property), 18, 125
 DeviceType (PySpin.Camera property), 18, 125
 DeviceType (PySpin.TransportLayerDevice property), 78, 411
 DeviceU3VProtocol (PySpin.TransportLayerDevice property), 78, 411
 DeviceUnlock (PySpin.TransportLayerInterface property), 79, 412
 DeviceUpdateList (PySpin.TransportLayerInterface property), 79, 412
 DeviceUptime (PySpin.Camera property), 18, 125
 DeviceUserID (PySpin.Camera property), 18, 125
 DeviceUserID (PySpin.TransportLayerDevice property), 78, 411
 DeviceVendorName (PySpin.Camera property), 18, 125
 DeviceVendorName (PySpin.TransportLayerDevice property), 78, 411
 DeviceVendorName (PySpin.TransportLayerInterface property), 79, 412
 DeviceVersion (PySpin.Camera property), 18, 125
 DeviceVersion (PySpin.TransportLayerDevice property), 78, 411
 DiscoverMaxPacketSize() (PySpin.CameraBase

- method*), 35, 142
- DiscoverMaxPacketSize() (*PySpin.ICameraBase method*), 174
- disparityScaleFactor (*PySpin.StereoCameraParameters property*), 403
- DoesEnvironmentVariableExist() (*in module PySpin*), 156
- double_autovector_t (*class in PySpin*), 418
- ## E
- EAccessModeClass (*class in PySpin*), 156
- EAccessModeClass_FromString() (*in module PySpin*), 157
- EAccessModeClass_ToString() (*in module PySpin*), 157
- EatComments() (*in module PySpin*), 166
- ECachingModeClass (*class in PySpin*), 157
- ECachingModeClass_FromString() (*in module PySpin*), 157
- ECachingModeClass_ToString() (*in module PySpin*), 157
- EDisplayNotationClass (*class in PySpin*), 158
- EDisplayNotationClass_FromString() (*in module PySpin*), 158
- EDisplayNotationClass_ToString() (*in module PySpin*), 158
- EEndianessClass (*class in PySpin*), 158
- EEndianessClass_FromString() (*in module PySpin*), 159
- EEndianessClass_ToString() (*in module PySpin*), 159
- EGenApiSchemaVersionClass (*class in PySpin*), 159
- EGenApiSchemaVersionClass_FromString() (*in module PySpin*), 160
- EGenApiSchemaVersionClass_ToString() (*in module PySpin*), 160
- EInputDirectionClass (*class in PySpin*), 160
- EInputDirectionClass_FromString() (*in module PySpin*), 160
- EInputDirectionClass_ToString() (*in module PySpin*), 160
- empty() (*PySpin.gcstring method*), 420
- empty() (*PySpin.node_vector method*), 423
- empty() (*PySpin.value_vector method*), 424
- ENamespaceClass (*class in PySpin*), 161
- ENamespaceClass_FromString() (*in module PySpin*), 161
- ENamespaceClass_ToString() (*in module PySpin*), 161
- EncoderDivider (*PySpin.Camera property*), 18, 125
- EncoderMode (*PySpin.Camera property*), 18, 126
- EncoderOutputMode (*PySpin.Camera property*), 18, 126
- EncoderReset (*PySpin.Camera property*), 18, 126
- EncoderResetActivation (*PySpin.Camera property*), 18, 126
- EncoderResetSource (*PySpin.Camera property*), 18, 126
- EncoderSelector (*PySpin.Camera property*), 18, 126
- EncoderSourceA (*PySpin.Camera property*), 18, 126
- EncoderSourceB (*PySpin.Camera property*), 18, 126
- EncoderStatus (*PySpin.Camera property*), 18, 126
- EncoderTimeout (*PySpin.Camera property*), 19, 126
- EncoderValue (*PySpin.Camera property*), 19, 126
- EncoderValueAtReset (*PySpin.Camera property*), 19, 126
- EncryptColorCorrectionMatrix() (*PySpin.ImageUtilityCCM static method*), 60, 359
- end() (*PySpin.node_vector method*), 423
- end() (*PySpin.value_vector method*), 424
- EndAcquisition() (*PySpin.CameraBase method*), 35, 143
- EndAcquisition() (*PySpin.ICameraBase method*), 174
- EnumEntryNode (*class in PySpin*), 166
- EnumerateGen2Cameras (*PySpin.TransportLayerSystem property*), 415
- EnumerateGEVInterfaces (*PySpin.TransportLayerSystem property*), 415
- EnumerateUSBInterfaces (*PySpin.TransportLayerSystem property*), 415
- EnumerationCount (*PySpin.Camera property*), 19, 126
- EnumNode (*class in PySpin*), 167
- erase() (*PySpin.node_vector method*), 423
- erase() (*PySpin.value_vector method*), 424
- ERepresentationClass (*class in PySpin*), 161
- ERepresentationClass_FromString() (*in module PySpin*), 162
- ERepresentationClass_ToString() (*in module PySpin*), 162
- errorcode (*PySpin.SpinnakerException attribute*), 69
- ESignClass (*class in PySpin*), 162
- ESignClass_FromString() (*in module PySpin*), 162
- ESignClass_ToString() (*in module PySpin*), 163
- ESlopeClass (*class in PySpin*), 163
- ESlopeClass_FromString() (*in module PySpin*), 163
- ESlopeClass_ToString() (*in module PySpin*), 163
- EStandardNameSpaceClass (*class in PySpin*), 164
- EStandardNameSpaceClass_FromString() (*in module PySpin*), 164
- EStandardNameSpaceClass_ToString() (*in module PySpin*), 164
- EventAcquisitionEnd (*PySpin.Camera property*), 19, 126

EventAcquisitionEndFrameID (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionEndTimestamp (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionError (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionErrorFrameID (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionErrorTimestamp (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionStart (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionStartFrameID (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionStartTimestamp (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTransferEnd (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTransferEndFrameID (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTransferEndTimestamp (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTransferStart (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTransferStartFrameID (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTransferStartTimestamp (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTrigger (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTriggerFrameID (<i>PySpin.Camera property</i>), 19, 126	
EventAcquisitionTriggerTimestamp (<i>PySpin.Camera property</i>), 19, 126	
EventActionLate (<i>PySpin.Camera property</i>), 19, 127	
EventActionLateFrameID (<i>PySpin.Camera property</i>), 19, 127	
EventActionLateTimestamp (<i>PySpin.Camera property</i>), 19, 127	
EventCounter0End (<i>PySpin.Camera property</i>), 19, 127	
EventCounter0EndFrameID (<i>PySpin.Camera property</i>), 19, 127	
EventCounter0EndTimestamp (<i>PySpin.Camera property</i>), 19, 127	
EventCounter0Start (<i>PySpin.Camera property</i>), 19, 127	
EventCounter0StartFrameID (<i>PySpin.Camera property</i>), 19, 127	
EventCounter0StartTimestamp (<i>PySpin.Camera property</i>), 19, 127	
EventCounter1End (<i>PySpin.Camera property</i>), 20, 127	
EventCounter1EndFrameID (<i>PySpin.Camera property</i>), 20, 127	
EventCounter1EndTimestamp (<i>PySpin.Camera property</i>), 20, 127	
EventCounter1Start (<i>PySpin.Camera property</i>), 20, 127	
EventCounter1StartFrameID (<i>PySpin.Camera property</i>), 20, 127	
EventCounter1StartTimestamp (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder0Restarted (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder0RestartedFrameID (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder0RestartedTimestamp (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder0Stopped (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder0StoppedFrameID (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder0StoppedTimestamp (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder1Restarted (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder1RestartedFrameID (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder1RestartedTimestamp (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder1Stopped (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder1StoppedFrameID (<i>PySpin.Camera property</i>), 20, 127	
EventEncoder1StoppedTimestamp (<i>PySpin.Camera property</i>), 20, 127	
EventError (<i>PySpin.Camera property</i>), 20, 127	
EventErrorCode (<i>PySpin.Camera property</i>), 20, 127	
EventErrorFrameID (<i>PySpin.Camera property</i>), 20, 127	
EventErrorTimestamp (<i>PySpin.Camera property</i>), 20, 127	
EventExposureEnd (<i>PySpin.Camera property</i>), 20, 128	
EventExposureEndFrameID (<i>PySpin.Camera property</i>), 20, 128	
EventExposureEndTimestamp (<i>PySpin.Camera property</i>), 20, 128	
EventExposureStart (<i>PySpin.Camera property</i>), 20, 128	
EventExposureStartFrameID (<i>PySpin.Camera property</i>), 20, 128	
EventExposureStartTimestamp (<i>PySpin.Camera property</i>), 20, 128	
EventFrameBurstEnd (<i>PySpin.Camera property</i>), 20, 128	
EventFrameBurstEndFrameID (<i>PySpin.Camera property</i>), 20, 128	
EventFrameBurstEndTimestamp (<i>PySpin.Camera property</i>), 20, 128	

EventFrameBurstStart (<i>PySpin.Camera</i> property), 21, 128	129
EventFrameBurstStartFrameID (<i>PySpin.Camera</i> property), 21, 128	EventLine1AnyEdgeFrameID (<i>PySpin.Camera</i> property), 21, 129
EventFrameBurstStartTimestamp (<i>PySpin.Camera</i> property), 21, 128	EventLine1AnyEdgeTimestamp (<i>PySpin.Camera</i> property), 21, 129
EventFrameEnd (<i>PySpin.Camera</i> property), 21, 128	EventLine1FallingEdge (<i>PySpin.Camera</i> property), 21, 129
EventFrameEndFrameID (<i>PySpin.Camera</i> property), 21, 128	EventLine1FallingEdgeFrameID (<i>PySpin.Camera</i> property), 22, 129
EventFrameEndTimestamp (<i>PySpin.Camera</i> property), 21, 128	EventLine1FallingEdgeTimestamp (<i>PySpin.Camera</i> property), 22, 129
EventFrameStart (<i>PySpin.Camera</i> property), 21, 128	EventLine1RisingEdge (<i>PySpin.Camera</i> property), 22, 129
EventFrameStartFrameID (<i>PySpin.Camera</i> property), 21, 128	EventLine1RisingEdgeFrameID (<i>PySpin.Camera</i> property), 22, 129
EventFrameStartTimestamp (<i>PySpin.Camera</i> property), 21, 128	EventLine1RisingEdgeTimestamp (<i>PySpin.Camera</i> property), 22, 129
EventFrameTransferEnd (<i>PySpin.Camera</i> property), 21, 128	EventLinkSpeedChange (<i>PySpin.Camera</i> property), 22, 129
EventFrameTransferEndFrameID (<i>PySpin.Camera</i> property), 21, 128	EventLinkSpeedChangeFrameID (<i>PySpin.Camera</i> property), 22, 129
EventFrameTransferEndTimestamp (<i>PySpin.Camera</i> property), 21, 128	EventLinkSpeedChangeTimestamp (<i>PySpin.Camera</i> property), 22, 129
EventFrameTransferStart (<i>PySpin.Camera</i> property), 21, 128	EventLinkTrigger0 (<i>PySpin.Camera</i> property), 22, 129
EventFrameTransferStartFrameID (<i>PySpin.Camera</i> property), 21, 128	EventLinkTrigger0FrameID (<i>PySpin.Camera</i> property), 22, 129
EventFrameTransferStartTimestamp (<i>PySpin.Camera</i> property), 21, 128	EventLinkTrigger0Timestamp (<i>PySpin.Camera</i> property), 22, 129
EventFrameTrigger (<i>PySpin.Camera</i> property), 21, 128	EventLinkTrigger1 (<i>PySpin.Camera</i> property), 22, 129
EventFrameTriggerFrameID (<i>PySpin.Camera</i> property), 21, 128	EventLinkTrigger1FrameID (<i>PySpin.Camera</i> property), 22, 129
EventFrameTriggerTimestamp (<i>PySpin.Camera</i> property), 21, 128	EventLinkTrigger1Timestamp (<i>PySpin.Camera</i> property), 22, 129
EventHandler (class in <i>PySpin</i>), 6, 168	EventNotification (<i>PySpin.Camera</i> property), 22, 129
EventLine0AnyEdge (<i>PySpin.Camera</i> property), 21, 128	EventSelector (<i>PySpin.Camera</i> property), 22, 129
EventLine0AnyEdgeFrameID (<i>PySpin.Camera</i> property), 21, 128	EventSequencerSetChange (<i>PySpin.Camera</i> property), 22, 129
EventLine0AnyEdgeTimestamp (<i>PySpin.Camera</i> property), 21, 128	EventSequencerSetChangeFrameID (<i>PySpin.Camera</i> property), 22, 129
EventLine0FallingEdge (<i>PySpin.Camera</i> property), 21, 128	EventSequencerSetChangeTimestamp (<i>PySpin.Camera</i> property), 22, 129
EventLine0FallingEdgeFrameID (<i>PySpin.Camera</i> property), 21, 129	EventSerialData (<i>PySpin.Camera</i> property), 22, 129
EventLine0FallingEdgeTimestamp (<i>PySpin.Camera</i> property), 21, 129	EventSerialDataLength (<i>PySpin.Camera</i> property), 22, 129
EventLine0RisingEdge (<i>PySpin.Camera</i> property), 21, 129	EventSerialPortReceive (<i>PySpin.Camera</i> property), 22, 129
EventLine0RisingEdgeFrameID (<i>PySpin.Camera</i> property), 21, 129	EventSerialPortReceiveTimestamp (<i>PySpin.Camera</i> property), 22, 130
EventLine0RisingEdgeTimestamp (<i>PySpin.Camera</i> property), 21, 129	EventSerialReceiveOverflow (<i>PySpin.Camera</i> property), 22, 130
EventLine1AnyEdge (<i>PySpin.Camera</i> property), 21, 129	EventStream0TransferBlockEnd (<i>PySpin.Camera</i> property), 22, 130

- property*), 22, 130
- EventStream@TransferBlockEndFrameID
(*PySpin.Camera property*), 22, 130
- EventStream@TransferBlockEndTimestamp
(*PySpin.Camera property*), 22, 130
- EventStream@TransferBlockStart (*PySpin.Camera property*), 22, 130
- EventStream@TransferBlockStartFrameID
(*PySpin.Camera property*), 22, 130
- EventStream@TransferBlockStartTimestamp
(*PySpin.Camera property*), 22, 130
- EventStream@TransferBlockTrigger
(*PySpin.Camera property*), 22, 130
- EventStream@TransferBlockTriggerFrameID
(*PySpin.Camera property*), 23, 130
- EventStream@TransferBlockTriggerTimestamp
(*PySpin.Camera property*), 23, 130
- EventStream@TransferBurstEnd (*PySpin.Camera property*), 23, 130
- EventStream@TransferBurstEndFrameID
(*PySpin.Camera property*), 23, 130
- EventStream@TransferBurstEndTimestamp
(*PySpin.Camera property*), 23, 130
- EventStream@TransferBurstStart (*PySpin.Camera property*), 23, 130
- EventStream@TransferBurstStartFrameID
(*PySpin.Camera property*), 23, 130
- EventStream@TransferBurstStartTimestamp
(*PySpin.Camera property*), 23, 130
- EventStream@TransferEnd (*PySpin.Camera property*), 23, 130
- EventStream@TransferEndFrameID (*PySpin.Camera property*), 23, 130
- EventStream@TransferEndTimestamp
(*PySpin.Camera property*), 23, 130
- EventStream@TransferOverflow (*PySpin.Camera property*), 23, 130
- EventStream@TransferOverflowFrameID
(*PySpin.Camera property*), 23, 130
- EventStream@TransferOverflowTimestamp
(*PySpin.Camera property*), 23, 130
- EventStream@TransferPause (*PySpin.Camera property*), 23, 130
- EventStream@TransferPauseFrameID
(*PySpin.Camera property*), 23, 130
- EventStream@TransferPauseTimestamp
(*PySpin.Camera property*), 23, 130
- EventStream@TransferResume (*PySpin.Camera property*), 23, 130
- EventStream@TransferResumeFrameID
(*PySpin.Camera property*), 23, 130
- EventStream@TransferResumeTimestamp
(*PySpin.Camera property*), 23, 130
- EventStream@TransferStart (*PySpin.Camera property*), 23, 130
- EventStream@TransferStartFrameID
(*PySpin.Camera property*), 23, 130
- EventStream@TransferStartTimestamp
(*PySpin.Camera property*), 23, 131
- EventTest (*PySpin.Camera property*), 23, 131
- EventTestTimestamp (*PySpin.Camera property*), 23, 131
- EventTimer@End (*PySpin.Camera property*), 23, 131
- EventTimer@EndFrameID (*PySpin.Camera property*), 23, 131
- EventTimer@EndTimestamp (*PySpin.Camera property*), 23, 131
- EventTimer@Start (*PySpin.Camera property*), 23, 131
- EventTimer@StartFrameID (*PySpin.Camera property*), 23, 131
- EventTimer@StartTimestamp (*PySpin.Camera property*), 23, 131
- EventTimer1End (*PySpin.Camera property*), 24, 131
- EventTimer1EndFrameID (*PySpin.Camera property*), 24, 131
- EventTimer1EndTimestamp (*PySpin.Camera property*), 24, 131
- EventTimer1Start (*PySpin.Camera property*), 24, 131
- EventTimer1StartFrameID (*PySpin.Camera property*), 24, 131
- EventTimer1StartTimestamp (*PySpin.Camera property*), 24, 131
- EVisibilityClass (class in *PySpin*), 164
- EVisibilityClass_FromString() (in module *PySpin*), 165
- EVisibilityClass_ToString() (in module *PySpin*), 165
- Execute() (*PySpin.CCommandPtr method*), 90
- Execute() (*PySpin.CommandNode method*), 154
- Execute() (*PySpin.ICommand method*), 179
- ExposureActiveMode (*PySpin.Camera property*), 24, 131
- ExposureAuto (*PySpin.Camera property*), 24, 131
- ExposureMode (*PySpin.Camera property*), 24, 131
- ExposureTime (*PySpin.Camera property*), 24, 131
- ExposureTimeMode (*PySpin.Camera property*), 24, 131
- ExposureTimeSelector (*PySpin.Camera property*), 24, 131
- ExternalVoltageEnable (*PySpin.Camera property*), 24, 131
- ExternalVoltageSelector (*PySpin.Camera property*), 24, 131
- ExternalVoltageValue (*PySpin.Camera property*), 24, 131
- ExtractIndependentSubtree()
(*PySpin.CNodeMapDynPtr method*), 103
- ExtractIndependentSubtree()
(*PySpin.INodeMapDyn method*), 338

- ExtractPolarQuadrant() (PySpin.ImageUtilityPolarization static method), 63, 364
- EYesNoClass (class in PySpin), 165
- EYesNoClass_FromString() (in module PySpin), 165
- EYesNoClass_ToString() (in module PySpin), 166
- ## F
- FactoryReset (PySpin.Camera property), 24, 131
- FfcEnable (PySpin.Camera property), 24, 131
- FfcMode (PySpin.Camera property), 24, 131
- FfcUserGain (PySpin.Camera property), 24, 131
- FfcUserOffset (PySpin.Camera property), 24, 131
- FfcUserTableReset (PySpin.Camera property), 24, 131
- FfcUserTableSave (PySpin.Camera property), 24, 131
- FfcUserTableXCoordinate (PySpin.Camera property), 24, 132
- FileAccessBuffer (PySpin.Camera property), 24, 132
- FileAccessLength (PySpin.Camera property), 24, 132
- FileAccessOffset (PySpin.Camera property), 24, 132
- FileOpenMode (PySpin.Camera property), 24, 132
- FileOperationExecute (PySpin.Camera property), 24, 132
- FileOperationResult (PySpin.Camera property), 24, 132
- FileOperationSelector (PySpin.Camera property), 24, 132
- FileOperationStatus (PySpin.Camera property), 24, 132
- FileSelector (PySpin.Camera property), 25, 132
- FileSize (PySpin.Camera property), 25, 132
- FilterSpeckles() (PySpin.ImageUtilityStereo static method), 66, 368
- FilterSpecklesFromImage() (PySpin.ImageUtilityStereo static method), 66, 369
- find() (PySpin.gcstring method), 420
- find_first_not_of() (PySpin.gcstring method), 421
- find_first_of() (PySpin.gcstring method), 421
- FLIRFilterDriverStatus (PySpin.TransportLayerInterface property), 79, 412
- FloatNode (class in PySpin), 169
- FloatRegNode (class in PySpin), 171
- focalLength (PySpin.StereoCameraParameters property), 403
- ForceIP() (PySpin.CameraBase method), 36, 143
- ForceIP() (PySpin.ICameraBase method), 174
- frameID (PySpin.DeviceEventExposureEndData property), 155
- frameID (PySpin.DeviceEventInferenceData property), 156
- frameRate (PySpin.AVIOption property), 83
- frameRate (PySpin.H264Option property), 173
- frameRate (PySpin.MJPGOption property), 387
- FromString() (PySpin.CBooleanPtr method), 85
- FromString() (PySpin.CCategoryPtr method), 88
- FromString() (PySpin.CCommandPtr method), 90
- FromString() (PySpin.CEnumEntryPtr method), 94
- FromString() (PySpin.CEnumerationPtr method), 96
- FromString() (PySpin.CIntegerPtr method), 100
- FromString() (PySpin.CRegisterPtr method), 108
- FromString() (PySpin.CStringPtr method), 112
- FromString() (PySpin.CValuePtr method), 115
- FromString() (PySpin.EAccessModeClass static method), 156
- FromString() (PySpin.ECachingModeClass static method), 157
- FromString() (PySpin.EDisplayNotationClass static method), 158
- FromString() (PySpin.EEndianessClass static method), 158
- FromString() (PySpin.EGenApiSchemaVersionClass static method), 159
- FromString() (PySpin.EInputDirectionClass static method), 160
- FromString() (PySpin.ENamespaceClass static method), 161
- FromString() (PySpin.ERepresentationClass static method), 161
- FromString() (PySpin.ESignClass static method), 162
- FromString() (PySpin.ESlopeClass static method), 163
- FromString() (PySpin.EStandardNameSpaceClass static method), 164
- FromString() (PySpin.EVisibilityClass static method), 164
- FromString() (PySpin.EYesNoClass static method), 165
- FromString() (PySpin.IValue method), 344
- FromString() (PySpin.ValueNode method), 417
- front() (PySpin.node_vector method), 423
- front() (PySpin.value_vector method), 425
- fullmessage (PySpin.SpinnakerException attribute), 69
- ## G
- g (PySpin.Stereo3DPoint property), 403
- Gain (PySpin.Camera property), 25, 132
- GainAuto (PySpin.Camera property), 25, 132
- GainAutoBalance (PySpin.Camera property), 25, 132
- GainConversion (PySpin.Camera property), 25, 132
- GainSelector (PySpin.Camera property), 25, 132
- Gamma (PySpin.Camera property), 25, 132
- GammaEnable (PySpin.Camera property), 25, 132
- gcstring (class in PySpin), 419
- gcstring__npos() (in module PySpin), 422
- GenICamXMLLocation (PySpin.TransportLayerDevice property), 78, 411

- GenICamXMLPath (*PySpin.TransportLayerDevice* property), 78, 411
- GenTlsSFNCVersionMajor (*PySpin.TransportLayerSystem* property), 415
- GenTlsSFNCVersionMinor (*PySpin.TransportLayerSystem* property), 415
- GenTlsSFNCVersionSubMinor (*PySpin.TransportLayerSystem* property), 416
- GenTlVersionMajor (*PySpin.TransportLayerSystem* property), 416
- GenTlVersionMinor (*PySpin.TransportLayerSystem* property), 416
- Get() (*PySpin.CRegisterPtr* method), 108
- Get() (*PySpin.IRegister* method), 341
- Get() (*PySpin.RegisterNode* method), 399
- GetAccessMode() (*PySpin.CameraBase* method), 36, 143
- GetAccessMode() (*PySpin.CBasePtr* method), 10, 85
- GetAccessMode() (*PySpin.CBooleanPtr* method), 85
- GetAccessMode() (*PySpin.CCategoryPtr* method), 88
- GetAccessMode() (*PySpin.CCommandPtr* method), 90
- GetAccessMode() (*PySpin.CEnumEntryPtr* method), 94
- GetAccessMode() (*PySpin.CEnumerationPtr* method), 96
- GetAccessMode() (*PySpin.CIntegerPtr* method), 100
- GetAccessMode() (*PySpin.CNodePtr* method), 106
- GetAccessMode() (*PySpin.CRegisterPtr* method), 109
- GetAccessMode() (*PySpin.CSelectorPtr* method), 111
- GetAccessMode() (*PySpin.CStringPtr* method), 112
- GetAccessMode() (*PySpin.CValuePtr* method), 115
- GetAccessMode() (*PySpin.IBase* method), 174
- GetAccessMode() (*PySpin.ICameraBase* method), 174
- GetAccessMode() (*PySpin.Node* method), 387
- GetActiveNumDataStreams() (*PySpin.CameraBase* method), 36, 143
- GetActiveNumDataStreams() (*PySpin.ICameraBase* method), 174
- GetAddress() (*PySpin.CRegisterPtr* method), 109
- GetAddress() (*PySpin.IRegister* method), 341
- GetAddress() (*PySpin.RegisterNode* method), 399
- GetAlias() (*PySpin.CBooleanPtr* method), 85
- GetAlias() (*PySpin.CCategoryPtr* method), 88
- GetAlias() (*PySpin.CCommandPtr* method), 91
- GetAlias() (*PySpin.CEnumEntryPtr* method), 94
- GetAlias() (*PySpin.CEnumerationPtr* method), 96
- GetAlias() (*PySpin.CIntegerPtr* method), 101
- GetAlias() (*PySpin.CNodePtr* method), 106
- GetAlias() (*PySpin.CRegisterPtr* method), 109
- GetAlias() (*PySpin.CStringPtr* method), 113
- GetAlias() (*PySpin.CValuePtr* method), 115
- GetAlias() (*PySpin.INode* method), 336
- GetAlias() (*PySpin.Node* method), 387
- GetBitsPerPixel() (*PySpin.IImage* method), 326
- GetBitsPerPixel() (*PySpin.Image* method), 47, 346
- GetBlackLevel() (*PySpin.ChunkData* method), 42, 150
- GetBlackLevel() (*PySpin.IChunkData* method), 177
- GetBoxAt() (*PySpin.InferenceBoundingBoxResult* method), 374
- GetBoxCount() (*PySpin.InferenceBoundingBoxResult* method), 375
- GetBoxSize() (*PySpin.InferenceBoundingBoxResult* method), 375
- GetBufferOwnership() (*PySpin.CameraBase* method), 36, 143
- GetBufferOwnership() (*PySpin.ICameraBase* method), 174
- GetBufferSize() (*PySpin.IImage* method), 326
- GetBufferSize() (*PySpin.Image* method), 48, 346
- GetByDeviceID() (*PySpin.CameraList* method), 40, 147
- GetByDeviceID() (*PySpin.ICameraList* method), 177
- GetByIndex() (*PySpin.CameraList* method), 40, 147
- GetByIndex() (*PySpin.ICameraList* method), 177
- GetByIndex() (*PySpin.IImageList* method), 330
- GetByIndex() (*PySpin.IInterfaceList* method), 335
- GetByIndex() (*PySpin.ImageList* method), 55, 354
- GetByIndex() (*PySpin.InterfaceList* method), 67, 379
- GetByInterfaceID() (*PySpin.InterfaceList* method), 68, 379
- GetByPayloadType() (*PySpin.IImageList* method), 330
- GetByPayloadType() (*PySpin.ImageList* method), 55, 354
- GetByPixelFormat() (*PySpin.IImageList* method), 330
- GetByPixelFormat() (*PySpin.ImageList* method), 55, 354
- GetBySerial() (*PySpin.CameraList* method), 40, 147
- GetBySerial() (*PySpin.ICameraList* method), 177
- GetByStreamIndex() (*PySpin.IImageList* method), 330
- GetByStreamIndex() (*PySpin.ImageList* method), 55, 354
- GetCachingMode() (*PySpin.CBooleanPtr* method), 85
- GetCachingMode() (*PySpin.CCategoryPtr* method), 88
- GetCachingMode() (*PySpin.CCommandPtr* method), 91
- GetCachingMode() (*PySpin.CEnumEntryPtr* method), 94
- GetCachingMode() (*PySpin.CEnumerationPtr* method), 96
- GetCachingMode() (*PySpin.CIntegerPtr* method), 101
- GetCachingMode() (*PySpin.CNodePtr* method), 106
- GetCachingMode() (*PySpin.CRegisterPtr* method), 109
- GetCachingMode() (*PySpin.CStringPtr* method), 113
- GetCachingMode() (*PySpin.CValuePtr* method), 115
- GetCachingMode() (*PySpin.INode* method), 336
- GetCachingMode() (*PySpin.Node* method), 388
- GetCameras() (*PySpin.IInterface* method), 66, 333
- GetCameras() (*PySpin.ISystem* method), 342

`GetCameras()` (*PySpin.System* method), 71, 405
`GetCastAlias()` (*PySpin.CBooleanPtr* method), 85
`GetCastAlias()` (*PySpin.CCategoryPtr* method), 88
`GetCastAlias()` (*PySpin.CCommandPtr* method), 91
`GetCastAlias()` (*PySpin.CEnumEntryPtr* method), 94
`GetCastAlias()` (*PySpin.CEnumerationPtr* method), 96
`GetCastAlias()` (*PySpin.CIntegerPtr* method), 101
`GetCastAlias()` (*PySpin.CNodePtr* method), 106
`GetCastAlias()` (*PySpin.CRegisterPtr* method), 109
`GetCastAlias()` (*PySpin.CStringPtr* method), 113
`GetCastAlias()` (*PySpin.CValuePtr* method), 115
`GetCastAlias()` (*PySpin.INode* method), 336
`GetCastAlias()` (*PySpin.Node* method), 388
`GetCategoryName()` (*PySpin.LoggingEventData* method), 385
`GetChildren()` (*PySpin.CBooleanPtr* method), 85
`GetChildren()` (*PySpin.CCategoryPtr* method), 88
`GetChildren()` (*PySpin.CCommandPtr* method), 91
`GetChildren()` (*PySpin.CEnumEntryPtr* method), 94
`GetChildren()` (*PySpin.CEnumerationPtr* method), 96
`GetChildren()` (*PySpin.CIntegerPtr* method), 101
`GetChildren()` (*PySpin.CNodePtr* method), 106
`GetChildren()` (*PySpin.CRegisterPtr* method), 109
`GetChildren()` (*PySpin.CStringPtr* method), 113
`GetChildren()` (*PySpin.CValuePtr* method), 115
`GetChildren()` (*PySpin.INode* method), 336
`GetChildren()` (*PySpin.Node* method), 388
`GetChunkData()` (*PySpin.IImage* method), 326
`GetChunkData()` (*PySpin.Image* method), 48, 346
`GetChunkLayoutId()` (*PySpin.IImage* method), 326
`GetChunkLayoutId()` (*PySpin.Image* method), 48, 347
`GetColorProcessing()` (*PySpin.IImage* method), 326
`GetColorProcessing()` (*PySpin.IImageProcessor* method), 332
`GetColorProcessing()` (*PySpin.Image* method), 48, 347
`GetColorProcessing()` (*PySpin.ImageProcessor* method), 57, 356
`GetCompressionMode()` (*PySpin.ChunkData* method), 42, 150
`GetCompressionMode()` (*PySpin.IChunkData* method), 178
`GetCompressionRatio()` (*PySpin.ChunkData* method), 42, 150
`GetCompressionRatio()` (*PySpin.IChunkData* method), 178
`GetCounterValue()` (*PySpin.ChunkData* method), 42, 150
`GetCounterValue()` (*PySpin.IChunkData* method), 178
`GetCRC()` (*PySpin.ChunkData* method), 42, 150
`GetCRC()` (*PySpin.IChunkData* method), 178
`GetCurrentDatarate()` (*PySpin.ChunkData* method), 43, 150
`GetCurrentDatarate()` (*PySpin.IChunkData* method), 178
`GetCurrentEntry()` (*PySpin.CEnumerationPtr* method), 96
`GetCurrentEntry()` (*PySpin.EnumNode* method), 167
`GetCurrentEntry()` (*PySpin.IEnumeration* method), 182
`GetCurrentEntry()` (*PySpin.IEnumerationT_AcquisitionModeEnums* method), 182
`GetCurrentEntry()` (*PySpin.IEnumerationT_AcquisitionStatusSelectorEnums* method), 183
`GetCurrentEntry()` (*PySpin.IEnumerationT_ActionSelectorEnums* method), 184
`GetCurrentEntry()` (*PySpin.IEnumerationT_ActionUnconditionalModeEnums* method), 184
`GetCurrentEntry()` (*PySpin.IEnumerationT_AdcBitDepthEnums* method), 185
`GetCurrentEntry()` (*PySpin.IEnumerationT_AutoAlgorithmSelectorEnums* method), 186
`GetCurrentEntry()` (*PySpin.IEnumerationT_AutoExposureControlPriorities* method), 186
`GetCurrentEntry()` (*PySpin.IEnumerationT_AutoExposureLightingModes* method), 187
`GetCurrentEntry()` (*PySpin.IEnumerationT_AutoExposureMeteringModes* method), 188
`GetCurrentEntry()` (*PySpin.IEnumerationT_AutoExposureTargetGreyValues* method), 188
`GetCurrentEntry()` (*PySpin.IEnumerationT_BalanceRatioSelectorEnums* method), 189
`GetCurrentEntry()` (*PySpin.IEnumerationT_BalanceWhiteAutoEnums* method), 190
`GetCurrentEntry()` (*PySpin.IEnumerationT_BalanceWhiteAutoProfileEnums* method), 190
`GetCurrentEntry()` (*PySpin.IEnumerationT_BinningHorizontalModeEnums* method), 191
`GetCurrentEntry()` (*PySpin.IEnumerationT_BinningSelectorEnums* method), 192
`GetCurrentEntry()` (*PySpin.IEnumerationT_BinningVerticalModeEnums* method), 192
`GetCurrentEntry()` (*PySpin.IEnumerationT_BlackLevelAutoBalanceEnums* method), 193
`GetCurrentEntry()` (*PySpin.IEnumerationT_BlackLevelAutoEnums* method), 194
`GetCurrentEntry()` (*PySpin.IEnumerationT_BlackLevelSelectorEnums* method), 194
`GetCurrentEntry()` (*PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums* method), 195
`GetCurrentEntry()` (*PySpin.IEnumerationT_BsiFlatFieldCorrectionGainEnums* method), 196
`GetCurrentEntry()` (*PySpin.IEnumerationT_ChunkBlackLevelSelectorEnums* method), 196
`GetCurrentEntry()` (*PySpin.IEnumerationT_ChunkCounterSelectorEnums* method), 197
`GetCurrentEntry()` (*PySpin.IEnumerationT_ChunkEncoderSelectorEnums* method), 197

method), 198

GetCurrentEntry() (PySpin.IEnumerationT_ChunkEncodingEnums method), 198

GetCurrentEntry() (PySpin.IEnumerationT_ChunkExposureEnums method), 199

GetCurrentEntry() (PySpin.IEnumerationT_ChunkGainEnums method), 200

GetCurrentEntry() (PySpin.IEnumerationT_ChunkImageEnums method), 200

GetCurrentEntry() (PySpin.IEnumerationT_ChunkPixelFormatEnums method), 201

GetCurrentEntry() (PySpin.IEnumerationT_ChunkRegionEnums method), 202

GetCurrentEntry() (PySpin.IEnumerationT_ChunkScanEnums method), 202

GetCurrentEntry() (PySpin.IEnumerationT_ChunkScanEnums method), 203

GetCurrentEntry() (PySpin.IEnumerationT_ChunkScanEnums method), 204

GetCurrentEntry() (PySpin.IEnumerationT_ChunkScanEnums method), 204

GetCurrentEntry() (PySpin.IEnumerationT_ChunkScanEnums method), 205

GetCurrentEntry() (PySpin.IEnumerationT_ChunkScanEnums method), 206

GetCurrentEntry() (PySpin.IEnumerationT_ChunkScanEnums method), 206

GetCurrentEntry() (PySpin.IEnumerationT_ChunkSelectorEnums method), 207

GetCurrentEntry() (PySpin.IEnumerationT_ChunkSourceEnums method), 208

GetCurrentEntry() (PySpin.IEnumerationT_ChunkTimerEnums method), 208

GetCurrentEntry() (PySpin.IEnumerationT_ChunkTransferEnums method), 209

GetCurrentEntry() (PySpin.IEnumerationT_ClConfigurationEnums method), 210

GetCurrentEntry() (PySpin.IEnumerationT_ClTimeSlotsEnums method), 210

GetCurrentEntry() (PySpin.IEnumerationT_ColorTransferEnums method), 211

GetCurrentEntry() (PySpin.IEnumerationT_ColorTransferEnums method), 212

GetCurrentEntry() (PySpin.IEnumerationT_ComponentLocationEnums method), 212

GetCurrentEntry() (PySpin.IEnumerationT_ComponentLocationEnums method), 213

GetCurrentEntry() (PySpin.IEnumerationT_CompressionEnums method), 214

GetCurrentEntry() (PySpin.IEnumerationT_CounterEventsEnums method), 214

GetCurrentEntry() (PySpin.IEnumerationT_CounterEventsEnums method), 215

GetCurrentEntry() (PySpin.IEnumerationT_CounterResetSourceEnums method), 216

GetCurrentEntry() (PySpin.IEnumerationT_CounterResetSourceEnums method), 216

GetCurrentEntry() (PySpin.IEnumerationT_CounterSelectorEnums method), 217

GetCurrentEntry() (PySpin.IEnumerationT_CounterStatusEnums method), 218

GetCurrentEntry() (PySpin.IEnumerationT_CounterTriggerActivationEnums method), 218

GetCurrentEntry() (PySpin.IEnumerationT_CounterTriggerSourceEnums method), 219

GetCurrentEntry() (PySpin.IEnumerationT_CxpConnectionTestModeEnums method), 220

GetCurrentEntry() (PySpin.IEnumerationT_CxpLinkConfigurationEnums method), 220

GetCurrentEntry() (PySpin.IEnumerationT_CxpLinkConfigurationPreferencesEnums method), 221

GetCurrentEntry() (PySpin.IEnumerationT_CxpLinkConfigurationStatusEnums method), 222

GetCurrentEntry() (PySpin.IEnumerationT_CxpPoCxpStatusEnums method), 222

GetCurrentEntry() (PySpin.IEnumerationT_DecimationHorizontalModeEnums method), 223

GetCurrentEntry() (PySpin.IEnumerationT_DecimationSelectorEnums method), 224

GetCurrentEntry() (PySpin.IEnumerationT_DecimationVerticalModeEnums method), 224

GetCurrentEntry() (PySpin.IEnumerationT_DefectCorrectionModeEnums method), 225

GetCurrentEntry() (PySpin.IEnumerationT_DeinterlacingEnums method), 226

GetCurrentEntry() (PySpin.IEnumerationT_DeviceAccessStatusEnums method), 226

GetCurrentEntry() (PySpin.IEnumerationT_DeviceCharacterSetEnums method), 227

GetCurrentEntry() (PySpin.IEnumerationT_DeviceClockSelectorEnums method), 228

GetCurrentEntry() (PySpin.IEnumerationT_DeviceConnectionStatusEnums method), 228

GetCurrentEntry() (PySpin.IEnumerationT_DeviceCurrentSpeedEnums method), 229

GetCurrentEntry() (PySpin.IEnumerationT_DeviceEndianessMechanismEnums method), 230

GetCurrentEntry() (PySpin.IEnumerationT_DeviceIndicatorModeEnums method), 230

GetCurrentEntry() (PySpin.IEnumerationT_DeviceLinkHeartbeatModelEnums method), 231

GetCurrentEntry() (PySpin.IEnumerationT_DeviceLinkThroughputLimitsEnums method), 232

GetCurrentEntry() (PySpin.IEnumerationT_DevicePowerSupplySelectorEnums method), 232

GetCurrentEntry() (PySpin.IEnumerationT_DeviceRegistersEndiannessEnums method), 233

GetCurrentEntry() (PySpin.IEnumerationT_DeviceScanTypeEnums method), 233

method), 234

GetCurrentEntry() (PySpin.IEnumerationT_DeviceSensorTypeEnums *method*), 234

GetCurrentEntry() (PySpin.IEnumerationT_DeviceSerialDataRateEnums *method*), 235

GetCurrentEntry() (PySpin.IEnumerationT_DeviceSerialDataTransferEnums *method*), 236

GetCurrentEntry() (PySpin.IEnumerationT_DeviceStreamTypeEnums *method*), 236

GetCurrentEntry() (PySpin.IEnumerationT_DeviceStreamTypeEnums *method*), 237

GetCurrentEntry() (PySpin.IEnumerationT_DeviceTapGainEnums *method*), 238

GetCurrentEntry() (PySpin.IEnumerationT_DeviceTemperatureEnums *method*), 239

GetCurrentEntry() (PySpin.IEnumerationT_DeviceTLTypeEnums *method*), 238

GetCurrentEntry() (PySpin.IEnumerationT_DeviceTypeEnums *method*), 240

GetCurrentEntry() (PySpin.IEnumerationT_DeviceTypeEnums *method*), 240

GetCurrentEntry() (PySpin.IEnumerationT_EncoderModeEnums *method*), 241

GetCurrentEntry() (PySpin.IEnumerationT_EncoderOutputEnums *method*), 242

GetCurrentEntry() (PySpin.IEnumerationT_EncoderResolutionEnums *method*), 242

GetCurrentEntry() (PySpin.IEnumerationT_EncoderResolutionEnums *method*), 243

GetCurrentEntry() (PySpin.IEnumerationT_EncoderSelectorEnums *method*), 244

GetCurrentEntry() (PySpin.IEnumerationT_EncoderSourceEnums *method*), 244

GetCurrentEntry() (PySpin.IEnumerationT_EncoderSourceEnums *method*), 245

GetCurrentEntry() (PySpin.IEnumerationT_EncoderStatusEnums *method*), 246

GetCurrentEntry() (PySpin.IEnumerationT_EventNotificationEnums *method*), 246

GetCurrentEntry() (PySpin.IEnumerationT_EventSelectorEnums *method*), 247

GetCurrentEntry() (PySpin.IEnumerationT_ExposureActionEnums *method*), 248

GetCurrentEntry() (PySpin.IEnumerationT_ExposureAutomationEnums *method*), 248

GetCurrentEntry() (PySpin.IEnumerationT_ExposureModeEnums *method*), 249

GetCurrentEntry() (PySpin.IEnumerationT_ExposureTimeEnums *method*), 250

GetCurrentEntry() (PySpin.IEnumerationT_ExposureTimeEnums *method*), 250

GetCurrentEntry() (PySpin.IEnumerationT_ExternalVoltageEnums *method*), 251

GetCurrentEntry() (PySpin.IEnumerationT_FfcModeEnums *method*), 252

GetCurrentEntry() (PySpin.IEnumerationT_FileOpenModeEnums *method*), 253

GetCurrentEntry() (PySpin.IEnumerationT_FileOperationSelectorEnums *method*), 254

GetCurrentEntry() (PySpin.IEnumerationT_FileOperationStatusEnums *method*), 254

GetCurrentEntry() (PySpin.IEnumerationT_FileSelectorEnums *method*), 255

GetCurrentEntry() (PySpin.IEnumerationT_FLIRFilterDriverStatusEnums *method*), 252

GetCurrentEntry() (PySpin.IEnumerationT_GainAutoBalanceEnums *method*), 256

GetCurrentEntry() (PySpin.IEnumerationT_GainAutoEnums *method*), 257

GetCurrentEntry() (PySpin.IEnumerationT_GainConversionEnums *method*), 258

GetCurrentEntry() (PySpin.IEnumerationT_GainSelectorEnums *method*), 258

GetCurrentEntry() (PySpin.IEnumerationT_GenICamXMLLocationEnums *method*), 259

GetCurrentEntry() (PySpin.IEnumerationT_GevCCPEnum *method*), 260

GetCurrentEntry() (PySpin.IEnumerationT_GevCCPEnum *method*), 260

GetCurrentEntry() (PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums *method*), 261

GetCurrentEntry() (PySpin.IEnumerationT_GevGVCPExtendedStatusConfigurationEnums *method*), 262

GetCurrentEntry() (PySpin.IEnumerationT_GevGVSPExtendedIDModeEnums *method*), 262

GetCurrentEntry() (PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnums *method*), 263

GetCurrentEntry() (PySpin.IEnumerationT_GevIEEE1588ModeEnums *method*), 264

GetCurrentEntry() (PySpin.IEnumerationT_GevIEEE1588StatusEnums *method*), 264

GetCurrentEntry() (PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnums *method*), 265

GetCurrentEntry() (PySpin.IEnumerationT_GevIPConfigurationStatusEnums *method*), 266

GetCurrentEntry() (PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnums *method*), 266

GetCurrentEntry() (PySpin.IEnumerationT_GevSCPDDirectionEnums *method*), 267

GetCurrentEntry() (PySpin.IEnumerationT_GevSupportedOptionSelectorEnums *method*), 268

GetCurrentEntry() (PySpin.IEnumerationT_GUIXMLLocationEnums *method*), 256

GetCurrentEntry() (PySpin.IEnumerationT_ImageComponentSelectorEnums *method*), 268

GetCurrentEntry() (PySpin.IEnumerationT_ImageCompressionJPEGFormatEnums *method*), 269

GetCurrentEntry() (PySpin.IEnumerationT_ImageCompressionModeEnums *method*), 269

445

method), 306

GetCurrentEntry() (PySpin.IEnumerationT_TeledyneGigEData method), 324

method), 307

GetCurrentEntry() (PySpin.IEnumerationT_TestPatternData method), 326

method), 308

GetCurrentEntry() (PySpin.IEnumerationT_TestPatternData method), 326

method), 308

GetCurrentEntry() (PySpin.IEnumerationT_TimerSelectorEnums method), 326

method), 309

GetCurrentEntry() (PySpin.IEnumerationT_TimerStatusEnums method), 326

method), 310

GetCurrentEntry() (PySpin.IEnumerationT_TimerTriggerEnums method), 326

method), 310

GetCurrentEntry() (PySpin.IEnumerationT_TimerTriggerEnums method), 326

method), 311

GetCurrentEntry() (PySpin.IEnumerationT_TLTypeEnums method), 326

method), 306

GetCurrentEntry() (PySpin.IEnumerationT_TransferCommandEnums method), 326

method), 312

GetCurrentEntry() (PySpin.IEnumerationT_TransferCommandEnums method), 326

method), 312

GetCurrentEntry() (PySpin.IEnumerationT_TransferOperationEnums method), 326

method), 313

GetCurrentEntry() (PySpin.IEnumerationT_TransferQueueEnums method), 326

method), 314

GetCurrentEntry() (PySpin.IEnumerationT_TransferSelectorEnums method), 326

method), 314

GetCurrentEntry() (PySpin.IEnumerationT_TransferStatusSelectorEnums method), 326

method), 315

GetCurrentEntry() (PySpin.IEnumerationT_TransferTriggerActivationEnums method), 326

method), 316

GetCurrentEntry() (PySpin.IEnumerationT_TransferTriggerModeEnums method), 326

method), 316

GetCurrentEntry() (PySpin.IEnumerationT_TransferTriggerModeEnums method), 326

method), 317

GetCurrentEntry() (PySpin.IEnumerationT_TransferTriggerModeEnums method), 326

method), 318

GetCurrentEntry() (PySpin.IEnumerationT_TransferTriggerModeEnums method), 326

method), 318

GetCurrentEntry() (PySpin.IEnumerationT_TriggerActivationEnums method), 326

method), 318

GetCurrentEntry() (PySpin.IEnumerationT_TriggerModeEnums method), 326

method), 319

GetCurrentEntry() (PySpin.IEnumerationT_TriggerOverloadEnums method), 326

method), 320

GetCurrentEntry() (PySpin.IEnumerationT_TriggerSelectorEnums method), 326

method), 320

GetCurrentEntry() (PySpin.IEnumerationT_TriggerSourceEnums method), 326

method), 321

GetCurrentEntry() (PySpin.IEnumerationT_U3VCurrentEnums method), 326

method), 322

GetCurrentEntry() (PySpin.IEnumerationT_UserOutputEnums method), 326

method), 322

GetCurrentEntry() (PySpin.IEnumerationT_UserSetDefaultEnums method), 326

method), 323

GetCurrentEntry() (PySpin.IEnumerationT_UserSetSelectorEnums method), 326

method), 93

method), 324

GetCurrentEntry() (PySpin.IEnumerationT_WhiteClipSelectorEnums method), 324

method), 324

GetData() (PySpin.IImage method), 326

GetDataAbsoluteMax() (PySpin.IImage method), 326

GetDataAbsoluteMax() (PySpin.Image method), 48,

347

GetDataAbsoluteMin() (PySpin.IImage method), 326

GetDataAbsoluteMin() (PySpin.Image method), 48,

347

GetDescription() (PySpin.CBooleanPtr method), 85

GetDescription() (PySpin.CCategoryPtr method), 88

GetDescription() (PySpin.CCommandPtr method), 91

GetDescription() (PySpin.CEnumEntryPtr method), 94

GetDescription() (PySpin.CEnumerationPtr method), 96

GetDescription() (PySpin.CIntegerPtr method), 101

GetDescription() (PySpin.CNodePtr method), 106

GetDescription() (PySpin.CRegisterPtr method), 109

GetDescription() (PySpin.CStringPtr method), 113

GetDescription() (PySpin.CValuePtr method), 115

GetDescription() (PySpin.INode method), 336

GetDescription() (PySpin.Node method), 388

GetDeviceEventId() (PySpin.DeviceEventHandler method), 5, 155

GetDeviceEventId() (PySpin.IDeviceEventHandler method), 180

GetDeviceEventName() (PySpin.DeviceEventHandler method), 5, 155

GetDeviceEventName() (PySpin.IDeviceEventHandler method), 180

GetDeviceID() (PySpin.CameraBase method), 36, 143

GetDeviceID() (PySpin.ICameraBase method), 174

GetDeviceName() (PySpin.CBooleanPtr method), 85

GetDeviceName() (PySpin.CCategoryPtr method), 88

GetDeviceName() (PySpin.CCommandPtr method), 91

GetDeviceName() (PySpin.CEnumEntryPtr method), 94

GetDeviceName() (PySpin.CEnumerationPtr method), 96

GetDeviceName() (PySpin.CIntegerPtr method), 101

GetDeviceName() (PySpin.CNodeMapDynPtr method), 104

GetDeviceName() (PySpin.CNodeMapPtr method), 106

GetDeviceName() (PySpin.CNodePtr method), 106

GetDeviceName() (PySpin.CRegisterPtr method), 109

GetDeviceName() (PySpin.CStringPtr method), 113

GetDeviceName() (PySpin.CValuePtr method), 115

GetDeviceName() (PySpin.INode method), 336

GetDeviceName() (PySpin.INodeMap method), 337

GetDeviceName() (PySpin.Node method), 388

GetDeviceName() (PySpin.NodeMap method), 393

GetDeviceVersion() (PySpin.CDeviceInfoPtr method), 93

- GetDeviceVersion() (*PySpin.IDeviceInfo* method), 180
- GetDeviceVersion() (*PySpin.NodeMap* method), 393
- GetDisplayName() (*PySpin.CBooleanPtr* method), 85
- GetDisplayName() (*PySpin.CCategoryPtr* method), 88
- GetDisplayName() (*PySpin.CCommandPtr* method), 91
- GetDisplayName() (*PySpin.CEnumEntryPtr* method), 94
- GetDisplayName() (*PySpin.CEnumerationPtr* method), 97
- GetDisplayName() (*PySpin.CIntegerPtr* method), 101
- GetDisplayName() (*PySpin.CNodePtr* method), 106
- GetDisplayName() (*PySpin.CRegisterPtr* method), 109
- GetDisplayName() (*PySpin.CStringPtr* method), 113
- GetDisplayName() (*PySpin.CValuePtr* method), 115
- GetDisplayName() (*PySpin.INode* method), 336
- GetDisplayName() (*PySpin.Node* method), 388
- GetDisplayNotation() (*PySpin.FloatNode* method), 169
- GetDisplayNotation() (*PySpin.IFloat* method), 325
- GetDisplayPrecision() (*PySpin.FloatNode* method), 169
- GetDisplayPrecision() (*PySpin.IFloat* method), 325
- GetDocuURL() (*PySpin.CBooleanPtr* method), 85
- GetDocuURL() (*PySpin.CCategoryPtr* method), 88
- GetDocuURL() (*PySpin.CCommandPtr* method), 91
- GetDocuURL() (*PySpin.CEnumEntryPtr* method), 94
- GetDocuURL() (*PySpin.CEnumerationPtr* method), 97
- GetDocuURL() (*PySpin.CIntegerPtr* method), 101
- GetDocuURL() (*PySpin.CNodePtr* method), 106
- GetDocuURL() (*PySpin.CRegisterPtr* method), 109
- GetDocuURL() (*PySpin.CStringPtr* method), 113
- GetDocuURL() (*PySpin.CValuePtr* method), 115
- GetDocuURL() (*PySpin.INode* method), 336
- GetDocuURL() (*PySpin.Node* method), 388
- GetEnable() (*PySpin.ChunkData* method), 43, 150
- GetEnable() (*PySpin.IChunkData* method), 178
- GetEncoderValue() (*PySpin.ChunkData* method), 43, 150
- GetEncoderValue() (*PySpin.IChunkData* method), 178
- GetEntries() (*PySpin.CEnumerationPtr* method), 97
- GetEntries() (*PySpin.EnumNode* method), 167
- GetEntries() (*PySpin.IEnumeration* method), 182
- GetEntry() (*PySpin.CEnumerationPtr* method), 97
- GetEntry() (*PySpin.EnumNode* method), 167
- GetEntry() (*PySpin.IEnumeration* method), 182
- GetEntry() (*PySpin.IEnumerationT_AcquisitionModeEnums* method), 182
- GetEntry() (*PySpin.IEnumerationT_AcquisitionStatusSelectorEnums* method), 183
- GetEntry() (*PySpin.IEnumerationT_ActionSelectorEnums* method), 184
- GetEntry() (*PySpin.IEnumerationT_ActionUnconditionalModeEnums* method), 184
- GetEntry() (*PySpin.IEnumerationT_AdcBitDepthEnums* method), 185
- GetEntry() (*PySpin.IEnumerationT_AutoAlgorithmSelectorEnums* method), 186
- GetEntry() (*PySpin.IEnumerationT_AutoExposureControlPriorityEnums* method), 186
- GetEntry() (*PySpin.IEnumerationT_AutoExposureLightingModeEnums* method), 187
- GetEntry() (*PySpin.IEnumerationT_AutoExposureMeteringModeEnums* method), 188
- GetEntry() (*PySpin.IEnumerationT_AutoExposureTargetGreyValueAutoEnums* method), 188
- GetEntry() (*PySpin.IEnumerationT_BalanceRatioSelectorEnums* method), 189
- GetEntry() (*PySpin.IEnumerationT_BalanceWhiteAutoEnums* method), 190
- GetEntry() (*PySpin.IEnumerationT_BalanceWhiteAutoProfileEnums* method), 190
- GetEntry() (*PySpin.IEnumerationT_BinningHorizontalModeEnums* method), 191
- GetEntry() (*PySpin.IEnumerationT_BinningSelectorEnums* method), 192
- GetEntry() (*PySpin.IEnumerationT_BinningVerticalModeEnums* method), 192
- GetEntry() (*PySpin.IEnumerationT_BlackLevelAutoBalanceEnums* method), 193
- GetEntry() (*PySpin.IEnumerationT_BlackLevelAutoEnums* method), 194
- GetEntry() (*PySpin.IEnumerationT_BlackLevelSelectorEnums* method), 194
- GetEntry() (*PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums* method), 195
- GetEntry() (*PySpin.IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums* method), 196
- GetEntry() (*PySpin.IEnumerationT_ChunkBlackLevelSelectorEnums* method), 196
- GetEntry() (*PySpin.IEnumerationT_ChunkCounterSelectorEnums* method), 197
- GetEntry() (*PySpin.IEnumerationT_ChunkEncoderSelectorEnums* method), 198
- GetEntry() (*PySpin.IEnumerationT_ChunkEncoderStatusEnums* method), 198
- GetEntry() (*PySpin.IEnumerationT_ChunkExposureTimeSelectorEnums* method), 199
- GetEntry() (*PySpin.IEnumerationT_ChunkGainSelectorEnums* method), 200
- GetEntry() (*PySpin.IEnumerationT_ChunkImageComponentEnums* method), 200
- GetEntry() (*PySpin.IEnumerationT_ChunkPixelFormatEnums* method), 201
- GetEntry() (*PySpin.IEnumerationT_ChunkRegionIDEnums* method), 202
- GetEntry() (*PySpin.IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums* method), 202

GetEntry() (PySpin.IEnumerationT_ChunkScan3dCoordinates method), 203
 GetEntry() (PySpin.IEnumerationT_ChunkScan3dCoordinates method), 204
 GetEntry() (PySpin.IEnumerationT_ChunkScan3dCoordinates method), 204
 GetEntry() (PySpin.IEnumerationT_ChunkScan3dCoordinates method), 205
 GetEntry() (PySpin.IEnumerationT_ChunkScan3dDistance method), 206
 GetEntry() (PySpin.IEnumerationT_ChunkScan3dOutputMediaFormat method), 207
 GetEntry() (PySpin.IEnumerationT_ChunkSelectorEnums method), 207
 GetEntry() (PySpin.IEnumerationT_ChunkSourceIDEnums method), 208
 GetEntry() (PySpin.IEnumerationT_ChunkTimerSelectorEnums method), 208
 GetEntry() (PySpin.IEnumerationT_ChunkTransferStreamIDEnums method), 209
 GetEntry() (PySpin.IEnumerationT_ClConfigurationEnums method), 210
 GetEntry() (PySpin.IEnumerationT_ClTimeSlotsCountEnums method), 210
 GetEntry() (PySpin.IEnumerationT_ColorTransformation method), 211
 GetEntry() (PySpin.IEnumerationT_ColorTransformation method), 212
 GetEntry() (PySpin.IEnumerationT_ComponentDestinationEnums method), 212
 GetEntry() (PySpin.IEnumerationT_ComponentSelectorEnums method), 213
 GetEntry() (PySpin.IEnumerationT_CompressionSaturation method), 214
 GetEntry() (PySpin.IEnumerationT_CounterEventActivation method), 214
 GetEntry() (PySpin.IEnumerationT_CounterEventSourceEnums method), 215
 GetEntry() (PySpin.IEnumerationT_CounterResetActivation method), 216
 GetEntry() (PySpin.IEnumerationT_CounterResetSourceEnums method), 216
 GetEntry() (PySpin.IEnumerationT_CounterSelectorEnums method), 217
 GetEntry() (PySpin.IEnumerationT_CounterStatusEnums method), 218
 GetEntry() (PySpin.IEnumerationT_CounterTriggerActivation method), 218
 GetEntry() (PySpin.IEnumerationT_CounterTriggerSourceEnums method), 219
 GetEntry() (PySpin.IEnumerationT_CxpConnectionTestMode method), 220
 GetEntry() (PySpin.IEnumerationT_CxpLinkConfigurationPreferredEnum method), 221
 GetEntry() (PySpin.IEnumerationT_CxpLinkConfigurationStatusEnums method), 222
 GetEntry() (PySpin.IEnumerationT_CxpPoCxpStatusEnums method), 222
 GetEntry() (PySpin.IEnumerationT_DecimationHorizontalModeEnums method), 223
 GetEntry() (PySpin.IEnumerationT_DecimationSelectorEnums method), 224
 GetEntry() (PySpin.IEnumerationT_DecimationVerticalModeEnums method), 224
 GetEntry() (PySpin.IEnumerationT_DefectCorrectionModeEnums method), 225
 GetEntry() (PySpin.IEnumerationT_DeinterlacingEnums method), 226
 GetEntry() (PySpin.IEnumerationT_DeviceAccessStatusEnum method), 226
 GetEntry() (PySpin.IEnumerationT_DeviceCharacterSetEnums method), 227
 GetEntry() (PySpin.IEnumerationT_DeviceClockSelectorEnums method), 228
 GetEntry() (PySpin.IEnumerationT_DeviceConnectionStatusEnums method), 228
 GetEntry() (PySpin.IEnumerationT_DeviceCurrentSpeedEnum method), 229
 GetEntry() (PySpin.IEnumerationT_DeviceEndianessMechanismEnum method), 230
 GetEntry() (PySpin.IEnumerationT_DeviceIndicatorModeEnums method), 230
 GetEntry() (PySpin.IEnumerationT_DeviceLinkHeartbeatModeEnums method), 231
 GetEntry() (PySpin.IEnumerationT_DeviceLinkThroughputLimitModeEnums method), 232
 GetEntry() (PySpin.IEnumerationT_DevicePowerSupplySelectorEnums method), 232
 GetEntry() (PySpin.IEnumerationT_DeviceRegistersEndiannessEnums method), 233
 GetEntry() (PySpin.IEnumerationT_DeviceScanTypeEnums method), 234
 GetEntry() (PySpin.IEnumerationT_DeviceSensorChromaEnums method), 234
 GetEntry() (PySpin.IEnumerationT_DeviceSerialPortBaudRateEnums method), 235
 GetEntry() (PySpin.IEnumerationT_DeviceSerialPortSelectorEnums method), 236
 GetEntry() (PySpin.IEnumerationT_DeviceStreamChannelEndiannessEnums method), 236
 GetEntry() (PySpin.IEnumerationT_DeviceStreamChannelTypeEnums method), 237
 GetEntry() (PySpin.IEnumerationT_DeviceTapGeometryEnums method), 238
 GetEntry() (PySpin.IEnumerationT_DeviceTemperatureSelectorEnums method), 239

GetEntry() (PySpin.IEnumerationT_DeviceTLTypeEnums method), 238
 GetEntry() (PySpin.IEnumerationT_DeviceTypeEnum method), 240
 GetEntry() (PySpin.IEnumerationT_DeviceTypeEnums method), 240
 GetEntry() (PySpin.IEnumerationT_EncoderModeEnums method), 241
 GetEntry() (PySpin.IEnumerationT_EncoderOutputModeEnums method), 242
 GetEntry() (PySpin.IEnumerationT_EncoderResetActivationEnums method), 242
 GetEntry() (PySpin.IEnumerationT_EncoderResetSourceEnums method), 243
 GetEntry() (PySpin.IEnumerationT_EncoderSelectorEnums method), 244
 GetEntry() (PySpin.IEnumerationT_EncoderSourceAEnums method), 244
 GetEntry() (PySpin.IEnumerationT_EncoderSourceBEnums method), 245
 GetEntry() (PySpin.IEnumerationT_EncoderStatusEnums method), 246
 GetEntry() (PySpin.IEnumerationT_EventNotificationEnums method), 246
 GetEntry() (PySpin.IEnumerationT_EventSelectorEnums method), 247
 GetEntry() (PySpin.IEnumerationT_ExposureActiveModeEnums method), 248
 GetEntry() (PySpin.IEnumerationT_ExposureAutoEnums method), 248
 GetEntry() (PySpin.IEnumerationT_ExposureModeEnums method), 249
 GetEntry() (PySpin.IEnumerationT_ExposureTimeModeEnums method), 250
 GetEntry() (PySpin.IEnumerationT_ExposureTimeSelectorEnums method), 250
 GetEntry() (PySpin.IEnumerationT_ExternalVoltageSelectorEnums method), 251
 GetEntry() (PySpin.IEnumerationT_FfcModeEnums method), 252
 GetEntry() (PySpin.IEnumerationT_FileOpenModeEnums method), 253
 GetEntry() (PySpin.IEnumerationT_FileOperationSelectorEnums method), 254
 GetEntry() (PySpin.IEnumerationT_FileOperationStatusEnums method), 254
 GetEntry() (PySpin.IEnumerationT_FileSelectorEnums method), 255
 GetEntry() (PySpin.IEnumerationT_FLIRFilterDriverStatusEnums method), 252
 GetEntry() (PySpin.IEnumerationT_GainAutoBalanceEnums method), 256
 GetEntry() (PySpin.IEnumerationT_GainAutoEnums method), 257
 GetEntry() (PySpin.IEnumerationT_GainConversionEnums method), 258
 GetEntry() (PySpin.IEnumerationT_GainSelectorEnums method), 258
 GetEntry() (PySpin.IEnumerationT_GenICamXMLLocationEnums method), 259
 GetEntry() (PySpin.IEnumerationT_GevCCPEnum method), 260
 GetEntry() (PySpin.IEnumerationT_GevCCPEnums method), 260
 GetEntry() (PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums method), 261
 GetEntry() (PySpin.IEnumerationT_GevGVCPExtendedStatusCodesSelectors method), 262
 GetEntry() (PySpin.IEnumerationT_GevGVSPExtendedIDModeEnums method), 263
 GetEntry() (PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnums method), 263
 GetEntry() (PySpin.IEnumerationT_GevIEEE1588ModeEnums method), 264
 GetEntry() (PySpin.IEnumerationT_GevIEEE1588StatusEnums method), 264
 GetEntry() (PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnums method), 265
 GetEntry() (PySpin.IEnumerationT_GevIPConfigurationStatusEnums method), 266
 GetEntry() (PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnums method), 266
 GetEntry() (PySpin.IEnumerationT_GevSCPDirectionEnums method), 267
 GetEntry() (PySpin.IEnumerationT_GevSupportedOptionSelectorEnums method), 268
 GetEntry() (PySpin.IEnumerationT_GUIXMLLocationEnums method), 256
 GetEntry() (PySpin.IEnumerationT_ImageComponentSelectorEnums method), 268
 GetEntry() (PySpin.IEnumerationT_ImageCompressionJPEGFormatOptions method), 269
 GetEntry() (PySpin.IEnumerationT_ImageCompressionModeEnums method), 270
 GetEntry() (PySpin.IEnumerationT_ImageCompressionRateOptionEnums method), 270
 GetEntry() (PySpin.IEnumerationT_InterfaceTypeEnum method), 271
 GetEntry() (PySpin.IEnumerationT_LensShadingCoefficientActiveSetEnums method), 272
 GetEntry() (PySpin.IEnumerationT_LensShadingCorrectionModeEnums method), 273
 GetEntry() (PySpin.IEnumerationT_LineFormatEnums method), 274
 GetEntry() (PySpin.IEnumerationT_LineInputFilterSelectorEnums method), 274
 GetEntry() (PySpin.IEnumerationT_LineModeEnums method), 275

<code>GetEntry()</code> (PySpin.IEnumerationT_LineSelectorEnums method), 276	<code>GetEntry()</code> (PySpin.IEnumerationT_SensorShutterModeEnums method), 293
<code>GetEntry()</code> (PySpin.IEnumerationT_LineSourceEnums method), 276	<code>GetEntry()</code> (PySpin.IEnumerationT_SensorTapsEnums method), 294
<code>GetEntry()</code> (PySpin.IEnumerationT_LogicBlockLUTInputEnums method), 277	<code>GetEntry()</code> (PySpin.IEnumerationT_SequencerConfigurationModeEnums method), 294
<code>GetEntry()</code> (PySpin.IEnumerationT_LogicBlockLUTInputEnums method), 278	<code>GetEntry()</code> (PySpin.IEnumerationT_SequencerConfigurationValidEnums method), 295
<code>GetEntry()</code> (PySpin.IEnumerationT_LogicBlockLUTInputEnums method), 278	<code>GetEntry()</code> (PySpin.IEnumerationT_SequencerModeEnums method), 296
<code>GetEntry()</code> (PySpin.IEnumerationT_LogicBlockLUTSelectorEnums method), 279	<code>GetEntry()</code> (PySpin.IEnumerationT_SequencerSetValidEnums method), 296
<code>GetEntry()</code> (PySpin.IEnumerationT_LogicBlockSelectorEnums method), 280	<code>GetEntry()</code> (PySpin.IEnumerationT_SequencerTriggerActivationEnums method), 297
<code>GetEntry()</code> (PySpin.IEnumerationT_LUTSelectorEnums method), 272	<code>GetEntry()</code> (PySpin.IEnumerationT_SequencerTriggerSourceEnums method), 298
<code>GetEntry()</code> (PySpin.IEnumerationT_MultiRoiConfigurationEnums method), 280	<code>GetEntry()</code> (PySpin.IEnumerationT_SerialPortBaudRateEnums method), 298
<code>GetEntry()</code> (PySpin.IEnumerationT_MultiRoiSelectorEnums method), 281	<code>GetEntry()</code> (PySpin.IEnumerationT_SerialPortParityEnums method), 299
<code>GetEntry()</code> (PySpin.IEnumerationT_PixelColorFilterEnums method), 282	<code>GetEntry()</code> (PySpin.IEnumerationT_SerialPortSelectorEnums method), 300
<code>GetEntry()</code> (PySpin.IEnumerationT_PixelFormatEnums method), 283	<code>GetEntry()</code> (PySpin.IEnumerationT_SerialPortSourceEnums method), 300
<code>GetEntry()</code> (PySpin.IEnumerationT_PixelFormatInfoSelectorEnums method), 284	<code>GetEntry()</code> (PySpin.IEnumerationT_SerialPortStopBitsEnums method), 301
<code>GetEntry()</code> (PySpin.IEnumerationT_PixelSizeEnums method), 284	<code>GetEntry()</code> (PySpin.IEnumerationT_SoftwareSignalSelectorEnums method), 302
<code>GetEntry()</code> (PySpin.IEnumerationT_POEStatusEnum method), 282	<code>GetEntry()</code> (PySpin.IEnumerationT_SourceSelectorEnums method), 302
<code>GetEntry()</code> (PySpin.IEnumerationT_RegionDestinationEnums method), 285	<code>GetEntry()</code> (PySpin.IEnumerationT_StereoResolutionEnums method), 303
<code>GetEntry()</code> (PySpin.IEnumerationT_RegionModeEnums method), 286	<code>GetEntry()</code> (PySpin.IEnumerationT_StreamBufferCountModeEnum method), 304
<code>GetEntry()</code> (PySpin.IEnumerationT_RegionSelectorEnums method), 286	<code>GetEntry()</code> (PySpin.IEnumerationT_StreamBufferHandlingModeEnum method), 304
<code>GetEntry()</code> (PySpin.IEnumerationT_RgbTransformLightSourceEnums method), 287	<code>GetEntry()</code> (PySpin.IEnumerationT_StreamModeEnum method), 305
<code>GetEntry()</code> (PySpin.IEnumerationT_Scan3dCoordinateReferenceEnums method), 288	<code>GetEntry()</code> (PySpin.IEnumerationT_StreamTypeEnum method), 306
<code>GetEntry()</code> (PySpin.IEnumerationT_Scan3dCoordinateSelectorEnums method), 288	<code>GetEntry()</code> (PySpin.IEnumerationT_TeledyneGigeVisionFilterDriverStatusEnums method), 307
<code>GetEntry()</code> (PySpin.IEnumerationT_Scan3dCoordinateSystemEnums method), 289	<code>GetEntry()</code> (PySpin.IEnumerationT_TestPatternEnums method), 308
<code>GetEntry()</code> (PySpin.IEnumerationT_Scan3dCoordinateSystemEnums method), 290	<code>GetEntry()</code> (PySpin.IEnumerationT_TestPatternGeneratorSelectorEnums method), 308
<code>GetEntry()</code> (PySpin.IEnumerationT_Scan3dCoordinateTransferEnums method), 290	<code>GetEntry()</code> (PySpin.IEnumerationT_TimerSelectorEnums method), 309
<code>GetEntry()</code> (PySpin.IEnumerationT_Scan3dDistanceUnitEnums method), 291	<code>GetEntry()</code> (PySpin.IEnumerationT_TimerStatusEnums method), 310
<code>GetEntry()</code> (PySpin.IEnumerationT_Scan3dOutputModeEnums method), 292	<code>GetEntry()</code> (PySpin.IEnumerationT_TimerTriggerActivationEnums method), 310
<code>GetEntry()</code> (PySpin.IEnumerationT_SensorDigitizationTapsEnums method), 292	<code>GetEntry()</code> (PySpin.IEnumerationT_TimerTriggerSourceEnums method), 311

GetEntry() (PySpin.IEnumerationT_TLTypeEnum method), 306
 GetEntry() (PySpin.IEnumerationT_TransferComponentSelectorEnums method), 312
 GetEntry() (PySpin.IEnumerationT_TransferControlModeEnums method), 312
 GetEntry() (PySpin.IEnumerationT_TransferOperationModeEnums method), 313
 GetEntry() (PySpin.IEnumerationT_TransferQueueModeEnums method), 314
 GetEntry() (PySpin.IEnumerationT_TransferSelectorEnums method), 314
 GetEntry() (PySpin.IEnumerationT_TransferStatusSelectorEnums method), 315
 GetEntry() (PySpin.IEnumerationT_TransferTriggerActivationEnums method), 316
 GetEntry() (PySpin.IEnumerationT_TransferTriggerModeEnums method), 316
 GetEntry() (PySpin.IEnumerationT_TransferTriggerSelectorEnums method), 317
 GetEntry() (PySpin.IEnumerationT_TransferTriggerSourceEnums method), 318
 GetEntry() (PySpin.IEnumerationT_TriggerActivationEnums method), 318
 GetEntry() (PySpin.IEnumerationT_TriggerModeEnums method), 319
 GetEntry() (PySpin.IEnumerationT_TriggerOverlapEnums method), 320
 GetEntry() (PySpin.IEnumerationT_TriggerSelectorEnums method), 320
 GetEntry() (PySpin.IEnumerationT_TriggerSourceEnums method), 321
 GetEntry() (PySpin.IEnumerationT_U3VCurrentSpeedEnums method), 322
 GetEntry() (PySpin.IEnumerationT_UserOutputSelectorEnums method), 322
 GetEntry() (PySpin.IEnumerationT_UserSetDefaultEnums method), 323
 GetEntry() (PySpin.IEnumerationT_UserSetSelectorEnums method), 324
 GetEntry() (PySpin.IEnumerationT_WhiteClipSelectorEnums method), 324
 GetEntryByName() (PySpin.CEnumerationPtr method), 97
 GetEntryByName() (PySpin.EnumNode method), 167
 GetEntryByName() (PySpin.IEnumeration method), 182
 GetEnumAlias() (PySpin.CFloatPtr method), 100
 GetEnumAlias() (PySpin.FloatNode method), 169
 GetErrorMessage() (in module PySpin), 172
 GetEventID() (PySpin.CBooleanPtr method), 86
 GetEventID() (PySpin.CCategoryPtr method), 88
 GetEventID() (PySpin.CCommandPtr method), 91
 GetEventID() (PySpin.CEnumEntryPtr method), 94
 GetEventID() (PySpin.CEnumerationPtr method), 97
 GetEventID() (PySpin.CIntegerPtr method), 101
 GetEventID() (PySpin.CNodePtr method), 106
 GetEventID() (PySpin.CRegisterPtr method), 109
 GetEventID() (PySpin.CStringPtr method), 113
 GetEventID() (PySpin.CValuePtr method), 115
 GetEventID() (PySpin.INode method), 336
 GetEventID() (PySpin.Node method), 388
 GetEventPayloadData() (PySpin.EventHandler method), 6, 169
 GetEventPayloadDataSize() (PySpin.EventHandler method), 6, 169
 GetEventType() (PySpin.EventHandler method), 6, 169
 GetExposureEndLineStatusAll() (PySpin.ChunkData method), 43, 150
 GetExposureEndLineStatusAll() (PySpin.IChunkData method), 178
 GetExposureTime() (PySpin.ChunkData method), 43, 150
 GetExposureTime() (PySpin.IChunkData method), 178
 GetFeatureBagHandle() (PySpin.CFeatureBag method), 99
 GetFeatures() (PySpin.CategoryNode method), 149
 GetFeatures() (PySpin.CCategoryPtr method), 88
 GetFeatures() (PySpin.ICategory method), 177
 GetFiles() (in module PySpin), 172
 GetFloatAlias() (PySpin.IntegerNode method), 376
 GetFrameID() (PySpin.ChunkData method), 43, 151
 GetFrameID() (PySpin.IChunkData method), 178
 GetFrameID() (PySpin.IImage method), 326
 GetFrameID() (PySpin.Image method), 48, 347
 GetGain() (PySpin.ChunkData method), 43, 151
 GetGain() (PySpin.IChunkData method), 178
 GetGenApiVersion() (PySpin.CDeviceInfoPtr method), 93
 GetGenApiVersion() (PySpin.IDeviceInfo method), 180
 GetGenApiVersion() (PySpin.NodeMap method), 393
 GetGenICamCacheFolder() (in module PySpin), 172
 GetGenICamCLProtocolFolder() (in module PySpin), 172
 GetGenICamLogConfig() (in module PySpin), 172
 GetGuiXml() (PySpin.CameraBase method), 36, 143
 GetGuiXml() (PySpin.ICameraBase method), 174
 GetHeatmapColorGradient() (PySpin.ImageUtilityHeatmap static method), 61, 361
 GetHeatmapRange() (PySpin.ImageUtilityHeatmap static method), 61, 361
 GetHeight() (PySpin.ChunkData method), 43, 151
 GetHeight() (PySpin.IChunkData method), 178
 GetHeight() (PySpin.IImage method), 327
 GetHeight() (PySpin.Image method), 48, 347
 GetID() (PySpin.IImage method), 327
 GetID() (PySpin.Image method), 48, 347

`GetImage()` (*PySpin.ChunkData* method), 43, 151
`GetImage()` (*PySpin.IChunkData* method), 178
`GetImagePayloadType()` (*PySpin.IImage* method), 327
`GetImagePayloadType()` (*PySpin.Image* method), 48, 347
`GetImageSize()` (*PySpin.IImage* method), 327
`GetImageSize()` (*PySpin.Image* method), 49, 347
`GetImageStatus()` (*PySpin.IImage* method), 327
`GetImageStatus()` (*PySpin.Image* method), 49, 347
`GetImageStatusDescription()` (*PySpin.Image* static method), 49, 347
`GetInc()` (*PySpin.CIntegerPtr* method), 101
`GetInc()` (*PySpin.FloatNode* method), 169
`GetInc()` (*PySpin.IFloat* method), 325
`GetInc()` (*PySpin.IInteger* method), 333
`GetInc()` (*PySpin.IntegerNode* method), 376
`GetIncMode()` (*PySpin.CIntegerPtr* method), 101
`GetIncMode()` (*PySpin.FloatNode* method), 169
`GetIncMode()` (*PySpin.IFloat* method), 325
`GetIncMode()` (*PySpin.IInteger* method), 333
`GetIncMode()` (*PySpin.IntegerNode* method), 376
`GetInferenceBoundingBoxResult()` (*PySpin.ChunkData* method), 43, 151
`GetInferenceBoundingBoxResult()` (*PySpin.IChunkData* method), 178
`GetInferenceConfidence()` (*PySpin.ChunkData* method), 43, 151
`GetInferenceConfidence()` (*PySpin.IChunkData* method), 178
`GetInferenceFrameId()` (*PySpin.ChunkData* method), 43, 151
`GetInferenceFrameId()` (*PySpin.IChunkData* method), 178
`GetInferenceResult()` (*PySpin.ChunkData* method), 43, 151
`GetInferenceResult()` (*PySpin.IChunkData* method), 178
`GetInstance()` (*PySpin.System* static method), 71, 406
`GetIntAlias()` (*PySpin.CFloatPtr* method), 100
`GetIntAlias()` (*PySpin.FloatNode* method), 169
`GetInterfaceName()` (in module *PySpin*), 172
`GetInterfaces()` (*PySpin.ISystem* method), 343
`GetInterfaces()` (*PySpin.System* method), 71, 406
`GetIntValue()` (*PySpin.CEnumerationPtr* method), 97
`GetIntValue()` (*PySpin.EnumNode* method), 167
`GetIntValue()` (*PySpin.IEnumeration* method), 182
`GetLength()` (*PySpin.CRegisterPtr* method), 109
`GetLength()` (*PySpin.IRegister* method), 341
`GetLength()` (*PySpin.RegisterNode* method), 399
`GetLibraryVersion()` (*PySpin.ISystem* method), 343
`GetLibraryVersion()` (*PySpin.System* method), 72, 406
`GetLinePitch()` (*PySpin.ChunkData* method), 43, 151
`GetLinePitch()` (*PySpin.IChunkData* method), 178
`GetLineStatusAll()` (*PySpin.ChunkData* method), 43, 151
`GetLineStatusAll()` (*PySpin.IChunkData* method), 178
`GetListOfValidValues()` (*PySpin.CIntegerPtr* method), 101
`GetListOfValidValues()` (*PySpin.FloatNode* method), 169
`GetListOfValidValues()` (*PySpin.IFloat* method), 325
`GetListOfValidValues()` (*PySpin.IInteger* method), 333
`GetListOfValidValues()` (*PySpin.IntegerNode* method), 376
`GetLockNodes()` (*PySpin.CBooleanPtr* method), 86
`GetLockNodes()` (*PySpin.CCategoryPtr* method), 88
`GetLockNodes()` (*PySpin.CCommandPtr* method), 91
`GetLockNodes()` (*PySpin.CEnumEntryPtr* method), 94
`GetLockNodes()` (*PySpin.CEnumerationPtr* method), 97
`GetLockNodes()` (*PySpin.CIntegerPtr* method), 101
`GetLockNodes()` (*PySpin.CNodePtr* method), 106
`GetLockNodes()` (*PySpin.CRegisterPtr* method), 109
`GetLockNodes()` (*PySpin.CStringPtr* method), 113
`GetLockNodes()` (*PySpin.CValuePtr* method), 115
`GetLockNodes()` (*PySpin.INode* method), 336
`GetLockNodes()` (*PySpin.Node* method), 388
`GetLoggingEventPriorityLevel()` (*PySpin.ISystem* method), 343
`GetLoggingEventPriorityLevel()` (*PySpin.System* method), 72, 406
`GetLogMessage()` (*PySpin.LoggingEventData* method), 386
`GetMax()` (*PySpin.CIntegerPtr* method), 101
`GetMax()` (*PySpin.FloatNode* method), 170
`GetMax()` (*PySpin.IFloat* method), 325
`GetMax()` (*PySpin.IInteger* method), 333
`GetMax()` (*PySpin.IntegerNode* method), 376
`GetMaxLength()` (*PySpin.CStringPtr* method), 113
`GetMaxLength()` (*PySpin.IString* method), 342
`GetMaxLength()` (*PySpin.StringNode* method), 404
`GetMin()` (*PySpin.CIntegerPtr* method), 101
`GetMin()` (*PySpin.FloatNode* method), 170
`GetMin()` (*PySpin.IFloat* method), 325
`GetMin()` (*PySpin.IInteger* method), 333
`GetMin()` (*PySpin.IntegerNode* method), 376
`GetModeActive()` (*PySpin.ChunkData* method), 43, 151
`GetModeActive()` (*PySpin.IChunkData* method), 178
`GetModelName()` (*PySpin.CDeviceInfoPtr* method), 93
`GetModelName()` (*PySpin.IDeviceInfo* method), 180
`GetModelName()` (*PySpin.NodeMap* method), 393
`GetModulePathFromFunction()` (in module *PySpin*), 172
`GetName()` (*PySpin.CBooleanPtr* method), 86
`GetName()` (*PySpin.CCategoryPtr* method), 89

- [GetName\(\) \(PySpin.CCommandPtr method\)](#), 91
- [GetName\(\) \(PySpin.CEnumEntryPtr method\)](#), 94
- [GetName\(\) \(PySpin.CEnumerationPtr method\)](#), 97
- [GetName\(\) \(PySpin.CIntegerPtr method\)](#), 101
- [GetName\(\) \(PySpin.CNodePtr method\)](#), 107
- [GetName\(\) \(PySpin.CRegisterPtr method\)](#), 109
- [GetName\(\) \(PySpin.CStringPtr method\)](#), 113
- [GetName\(\) \(PySpin.CValuePtr method\)](#), 116
- [GetName\(\) \(PySpin.INode method\)](#), 336
- [GetName\(\) \(PySpin.Node method\)](#), 388
- [GetNameSpace\(\) \(PySpin.CBooleanPtr method\)](#), 86
- [GetNameSpace\(\) \(PySpin.CCategoryPtr method\)](#), 89
- [GetNameSpace\(\) \(PySpin.CCommandPtr method\)](#), 91
- [GetNameSpace\(\) \(PySpin.CEnumEntryPtr method\)](#), 94
- [GetNameSpace\(\) \(PySpin.CEnumerationPtr method\)](#), 97
- [GetNameSpace\(\) \(PySpin.CIntegerPtr method\)](#), 101
- [GetNameSpace\(\) \(PySpin.CNodePtr method\)](#), 107
- [GetNameSpace\(\) \(PySpin.CRegisterPtr method\)](#), 109
- [GetNameSpace\(\) \(PySpin.CStringPtr method\)](#), 113
- [GetNameSpace\(\) \(PySpin.CValuePtr method\)](#), 116
- [GetNameSpace\(\) \(PySpin.INode method\)](#), 336
- [GetNameSpace\(\) \(PySpin.Node method\)](#), 389
- [GetNDArray\(\) \(PySpin.IImage method\)](#), 327
- [GetNDC\(\) \(PySpin.LoggingEventData method\)](#), 386
- [GetNextImage\(\) \(PySpin.CameraBase method\)](#), 36, 143
- [GetNextImage\(\) \(PySpin.ICameraBase method\)](#), 174
- [GetNextImageSync\(\) \(PySpin.CameraBase method\)](#), 36, 144
- [GetNextImageSync\(\) \(PySpin.ICameraBase method\)](#), 175
- [GetNode\(\) \(PySpin.CBooleanPtr method\)](#), 86
- [GetNode\(\) \(PySpin.CCategoryPtr method\)](#), 89
- [GetNode\(\) \(PySpin.CCommandPtr method\)](#), 91
- [GetNode\(\) \(PySpin.CEnumEntryPtr method\)](#), 94
- [GetNode\(\) \(PySpin.CEnumerationPtr method\)](#), 97
- [GetNode\(\) \(PySpin.CIntegerPtr method\)](#), 101
- [GetNode\(\) \(PySpin.CNodeMapDynPtr method\)](#), 104
- [GetNode\(\) \(PySpin.CNodeMapPtr method\)](#), 106
- [GetNode\(\) \(PySpin.CRegisterPtr method\)](#), 109
- [GetNode\(\) \(PySpin.CStringPtr method\)](#), 113
- [GetNode\(\) \(PySpin.CValuePtr method\)](#), 116
- [GetNode\(\) \(PySpin.INodeMap method\)](#), 338
- [GetNode\(\) \(PySpin.IValue method\)](#), 344
- [GetNode\(\) \(PySpin.NodeMap method\)](#), 393
- [GetNode\(\) \(PySpin.ValueNode method\)](#), 417
- [GetNodeHandle\(\) \(PySpin.Node method\)](#), 389
- [GetNodeMap\(\) \(PySpin.CameraBase method\)](#), 37, 144
- [GetNodeMap\(\) \(PySpin.CBooleanPtr method\)](#), 86
- [GetNodeMap\(\) \(PySpin.CCategoryPtr method\)](#), 89
- [GetNodeMap\(\) \(PySpin.CCommandPtr method\)](#), 91
- [GetNodeMap\(\) \(PySpin.CEnumEntryPtr method\)](#), 94
- [GetNodeMap\(\) \(PySpin.CEnumerationPtr method\)](#), 97
- [GetNodeMap\(\) \(PySpin.CIntegerPtr method\)](#), 101
- [GetNodeMap\(\) \(PySpin.CNodePtr method\)](#), 107
- [GetNodeMap\(\) \(PySpin.CRegisterPtr method\)](#), 109
- [GetNodeMap\(\) \(PySpin.CStringPtr method\)](#), 113
- [GetNodeMap\(\) \(PySpin.CValuePtr method\)](#), 116
- [GetNodeMap\(\) \(PySpin.INode method\)](#), 336
- [GetNodeMap\(\) \(PySpin.Node method\)](#), 389
- [GetNodeMap\(\) \(PySpin.CRegisterPtr method\)](#), 109
- [GetNodeMap\(\) \(PySpin.CStringPtr method\)](#), 113
- [GetNodeMap\(\) \(PySpin.CValuePtr method\)](#), 116
- [GetNodeMap\(\) \(PySpin.INode method\)](#), 336
- [GetNodeMap\(\) \(PySpin.Node method\)](#), 389
- [GetNodeMapHandle\(\) \(PySpin.NodeMap method\)](#), 394
- [GetNodes\(\) \(PySpin.CNodeMapDynPtr method\)](#), 104
- [GetNodes\(\) \(PySpin.CNodeMapPtr method\)](#), 106
- [GetNodes\(\) \(PySpin.INodeMap method\)](#), 338
- [GetNodes\(\) \(PySpin.NodeMap method\)](#), 394
- [GetNumChannels\(\) \(PySpin.IImage method\)](#), 327
- [GetNumChannels\(\) \(PySpin.Image method\)](#), 49, 347
- [GetNumDataStreams\(\) \(PySpin.CameraBase method\)](#), 37, 144
- [GetNumDataStreams\(\) \(PySpin.ICameraBase method\)](#), 175
- [GetNumDecompressionThreads\(\) \(PySpin.IImageProcessor method\)](#), 332
- [GetNumDecompressionThreads\(\) \(PySpin.ImageProcessor method\)](#), 57, 356
- [GetNumericValue\(\) \(PySpin.CEnumEntryPtr method\)](#), 94
- [GetNumericValue\(\) \(PySpin.EnumEntryNode method\)](#), 166
- [GetNumericValue\(\) \(PySpin.IEnumEntry method\)](#), 181
- [GetNumImagesInUse\(\) \(PySpin.CameraBase method\)](#), 37, 144
- [GetNumImagesInUse\(\) \(PySpin.ICameraBase method\)](#), 175
- [GetNumNodes\(\) \(PySpin.CNodeMapDynPtr method\)](#), 104
- [GetNumNodes\(\) \(PySpin.CNodeMapPtr method\)](#), 106
- [GetNumNodes\(\) \(PySpin.INodeMap method\)](#), 338
- [GetNumNodes\(\) \(PySpin.NodeMap method\)](#), 394
- [GetNumPoints\(\) \(PySpin.IPointCloud method\)](#), 340
- [GetNumPoints\(\) \(PySpin.PointCloud method\)](#), 68, 398
- [GetOffsetX\(\) \(PySpin.ChunkData method\)](#), 44, 151
- [GetOffsetX\(\) \(PySpin.IChunkData method\)](#), 178
- [GetOffsetY\(\) \(PySpin.ChunkData method\)](#), 44, 151
- [GetOffsetY\(\) \(PySpin.IChunkData method\)](#), 178
- [GetParents\(\) \(PySpin.CBooleanPtr method\)](#), 86
- [GetParents\(\) \(PySpin.CCategoryPtr method\)](#), 89
- [GetParents\(\) \(PySpin.CCommandPtr method\)](#), 91
- [GetParents\(\) \(PySpin.CEnumEntryPtr method\)](#), 94
- [GetParents\(\) \(PySpin.CEnumerationPtr method\)](#), 97
- [GetParents\(\) \(PySpin.CIntegerPtr method\)](#), 101
- [GetParents\(\) \(PySpin.CNodePtr method\)](#), 107
- [GetParents\(\) \(PySpin.CRegisterPtr method\)](#), 109
- [GetParents\(\) \(PySpin.CStringPtr method\)](#), 113
- [GetParents\(\) \(PySpin.CValuePtr method\)](#), 116
- [GetParents\(\) \(PySpin.INode method\)](#), 336
- [GetParents\(\) \(PySpin.Node method\)](#), 389

`GetPartSelector()` (*PySpin.ChunkData* method), 44, 151

`GetPartSelector()` (*PySpin.IChunkData* method), 178

`GetPayloadType()` (*PySpin.IImage* method), 327

`GetPayloadType()` (*PySpin.Image* method), 49, 347

`GetPixelDynamicRangeMax()` (*PySpin.ChunkData* method), 44, 151

`GetPixelDynamicRangeMax()` (*PySpin.IChunkData* method), 178

`GetPixelDynamicRangeMin()` (*PySpin.ChunkData* method), 44, 152

`GetPixelDynamicRangeMin()` (*PySpin.IChunkData* method), 178

`GetPixelFormat()` (*PySpin.IImage* method), 327

`GetPixelFormat()` (*PySpin.Image* method), 49, 348

`GetPixelFormatIntType()` (*PySpin.IImage* method), 327

`GetPixelFormatIntType()` (*PySpin.Image* method), 49, 348

`GetPixelFormatName()` (*PySpin.IImage* method), 327

`GetPixelFormatName()` (*PySpin.Image* method), 49, 348

`GetPoint()` (*PySpin.IPointCloud* method), 340

`GetPoint()` (*PySpin.PointCloud* method), 68, 398

`GetPointCloudData()` (*PySpin.IPointCloud* method), 340

`GetPointCloudData()` (*PySpin.PointCloud* method), 69, 398

`GetPollingTime()` (*PySpin.CBooleanPtr* method), 86

`GetPollingTime()` (*PySpin.CCategoryPtr* method), 89

`GetPollingTime()` (*PySpin.CCommandPtr* method), 91

`GetPollingTime()` (*PySpin.CEnumEntryPtr* method), 94

`GetPollingTime()` (*PySpin.CEnumerationPtr* method), 97

`GetPollingTime()` (*PySpin.CIntegerPtr* method), 101

`GetPollingTime()` (*PySpin.CNodePtr* method), 107

`GetPollingTime()` (*PySpin.CRegisterPtr* method), 109

`GetPollingTime()` (*PySpin.CStringPtr* method), 113

`GetPollingTime()` (*PySpin.CValuePtr* method), 116

`GetPollingTime()` (*PySpin.INode* method), 336

`GetPollingTime()` (*PySpin.Node* method), 389

`GetPrincipalInterfaceType()` (*PySpin.CBooleanPtr* method), 86

`GetPrincipalInterfaceType()` (*PySpin.CCategoryPtr* method), 89

`GetPrincipalInterfaceType()` (*PySpin.CCommandPtr* method), 91

`GetPrincipalInterfaceType()` (*PySpin.CEnumEntryPtr* method), 94

`GetPrincipalInterfaceType()` (*PySpin.CEnumerationPtr* method), 97

`GetPrincipalInterfaceType()` (*PySpin.CIntegerPtr* method), 101

`GetPrincipalInterfaceType()` (*PySpin.CNodePtr* method), 107

`GetPrincipalInterfaceType()` (*PySpin.CRegisterPtr* method), 109

`GetPrincipalInterfaceType()` (*PySpin.CStringPtr* method), 113

`GetPrincipalInterfaceType()` (*PySpin.CValuePtr* method), 116

`GetPrincipalInterfaceType()` (*PySpin.INode* method), 336

`GetPrincipalInterfaceType()` (*PySpin.Node* method), 389

`GetPriority()` (*PySpin.LoggingEventData* method), 386

`GetPriorityName()` (*PySpin.LoggingEventData* method), 386

`GetPrivateData()` (*PySpin.IImage* method), 327

`GetPrivateData()` (*PySpin.Image* method), 49, 348

`GetProductGuid()` (*PySpin.CDeviceInfoPtr* method), 93

`GetProductGuid()` (*PySpin.IDeviceInfo* method), 180

`GetProductGuid()` (*PySpin.NodeMap* method), 394

`GetProperty()` (*PySpin.CBooleanPtr* method), 86

`GetProperty()` (*PySpin.CCategoryPtr* method), 89

`GetProperty()` (*PySpin.CCommandPtr* method), 91

`GetProperty()` (*PySpin.CEnumEntryPtr* method), 94

`GetProperty()` (*PySpin.CEnumerationPtr* method), 97

`GetProperty()` (*PySpin.CIntegerPtr* method), 101

`GetProperty()` (*PySpin.CNodePtr* method), 107

`GetProperty()` (*PySpin.CRegisterPtr* method), 109

`GetProperty()` (*PySpin.CStringPtr* method), 113

`GetProperty()` (*PySpin.CValuePtr* method), 116

`GetProperty()` (*PySpin.INode* method), 336

`GetProperty()` (*PySpin.Node* method), 389

`GetPropertyNames()` (*PySpin.CBooleanPtr* method), 86

`GetPropertyNames()` (*PySpin.CCategoryPtr* method), 89

`GetPropertyNames()` (*PySpin.CCommandPtr* method), 91

`GetPropertyNames()` (*PySpin.CEnumEntryPtr* method), 95

`GetPropertyNames()` (*PySpin.CEnumerationPtr* method), 97

`GetPropertyNames()` (*PySpin.CIntegerPtr* method), 102

`GetPropertyNames()` (*PySpin.CNodePtr* method), 107

`GetPropertyNames()` (*PySpin.CRegisterPtr* method), 110

`GetPropertyNames()` (*PySpin.CStringPtr* method), 113

`GetPropertyNames()` (*PySpin.CValuePtr* method), 116

`GetPropertyNames()` (*PySpin.INode* method), 337

`GetPropertyNames()` (*PySpin.Node* method), 390

`GetRepresentation()` (*PySpin.CIntegerPtr* method),

102			
GetRepresentation()	(PySpin.FloatNode method),	GetSelectedFeatures()	(PySpin.CEnumEntryPtr method), 95
170		GetSelectedFeatures()	(PySpin.CEnumerationPtr method), 98
GetRepresentation()	(PySpin.IFloat method), 325	GetSelectedFeatures()	(PySpin.CIntegerPtr method), 102
GetRepresentation()	(PySpin.IInteger method), 333	GetSelectedFeatures()	(PySpin.CNodePtr method), 107
GetRepresentation()	(PySpin.IntegerNode method), 377	GetSelectedFeatures()	(PySpin.CRegisterPtr method), 110
GetScan3dAxisMax()	(PySpin.ChunkData method), 44, 152	GetSelectedFeatures()	(PySpin.CSelectorPtr method), 111
GetScan3dAxisMax()	(PySpin.IChunkData method), 178	GetSelectedFeatures()	(PySpin.CStringPtr method), 113
GetScan3dAxisMin()	(PySpin.ChunkData method), 44, 152	GetSelectedFeatures()	(PySpin.CValuePtr method), 116
GetScan3dAxisMin()	(PySpin.IChunkData method), 178	GetSelectedFeatures()	(PySpin.ISelector method), 341
GetScan3dCoordinateOffset()	(PySpin.ChunkData method), 44, 152	GetSelectedFeatures()	(PySpin.Node method), 390
GetScan3dCoordinateOffset()	(PySpin.IChunkData method), 178	GetSelectingFeatures()	(PySpin.CBooleanPtr method), 86
GetScan3dCoordinateReferenceValue()	(PySpin.ChunkData method), 44, 152	GetSelectingFeatures()	(PySpin.CCategoryPtr method), 89
GetScan3dCoordinateReferenceValue()	(PySpin.IChunkData method), 178	GetSelectingFeatures()	(PySpin.CCommandPtr method), 92
GetScan3dCoordinateScale()	(PySpin.ChunkData method), 44, 152	GetSelectingFeatures()	(PySpin.CEnumEntryPtr method), 95
GetScan3dCoordinateScale()	(PySpin.IChunkData method), 178	GetSelectingFeatures()	(PySpin.CEnumerationPtr method), 98
GetScan3dInvalidDataFlag()	(PySpin.ChunkData method), 44, 152	GetSelectingFeatures()	(PySpin.CIntegerPtr method), 102
GetScan3dInvalidDataFlag()	(PySpin.IChunkData method), 178	GetSelectingFeatures()	(PySpin.CNodePtr method), 107
GetScan3dInvalidDataValue()	(PySpin.ChunkData method), 44, 152	GetSelectingFeatures()	(PySpin.CRegisterPtr method), 110
GetScan3dInvalidDataValue()	(PySpin.IChunkData method), 179	GetSelectingFeatures()	(PySpin.CSelectorPtr method), 111
GetScan3dTransformValue()	(PySpin.ChunkData method), 45, 152	GetSelectingFeatures()	(PySpin.CStringPtr method), 114
GetScan3dTransformValue()	(PySpin.IChunkData method), 179	GetSelectingFeatures()	(PySpin.CValuePtr method), 116
GetScanLineSelector()	(PySpin.ChunkData method), 45, 152	GetSelectingFeatures()	(PySpin.ISelector method), 341
GetScanLineSelector()	(PySpin.IChunkData method), 179	GetSelectingFeatures()	(PySpin.Node method), 390
GetSchemaVersion()	(PySpin.CDeviceInfoPtr method), 93	GetSelectorList()	(PySpin.CSelectorSet method), 112
GetSchemaVersion()	(PySpin.IDeviceInfo method), 180	GetSelectorList()	(PySpin.ISelectorDigit method), 342
GetSchemaVersion()	(PySpin.NodeMap method), 394	GetSequencerSetActive()	(PySpin.ChunkData method), 45, 152
GetSelectedFeatures()	(PySpin.CBooleanPtr method), 86	GetSequencerSetActive()	(PySpin.IChunkData method), 179
GetSelectedFeatures()	(PySpin.CCategoryPtr method), 89	GetSerialData()	(PySpin.ChunkData method), 45, 153
GetSelectedFeatures()	(PySpin.CCommandPtr method), 91	GetSerialData()	(PySpin.IChunkData method), 179

GetSerialDataLength() (<i>PySpin.ChunkData</i> method), 45, 153	GetTLDeviceNodeMap() (<i>PySpin.CameraBase</i> method), 37, 144
GetSerialDataLength() (<i>PySpin.IChunkData</i> method), 179	GetTLDeviceNodeMap() (<i>PySpin.ICameraBase</i> method), 175
GetSerialReceiveOverflow() (<i>PySpin.ChunkData</i> method), 45, 153	GetTLNodeMap() (<i>PySpin.IInterface</i> method), 66, 333
GetSerialReceiveOverflow() (<i>PySpin.IChunkData</i> method), 179	GetTLNodeMap() (<i>PySpin.ISystem</i> method), 343
GetSize() (<i>PySpin.CameraList</i> method), 40, 148	GetTLNodeMap() (<i>PySpin.System</i> method), 72, 407
GetSize() (<i>PySpin.ICameraList</i> method), 177	GetTLPayloadType() (<i>PySpin.IImage</i> method), 327
GetSize() (<i>PySpin.IImageList</i> method), 330	GetTLPayloadType() (<i>PySpin.Image</i> method), 50, 348
GetSize() (<i>PySpin.IInterfaceList</i> method), 335	GetTLPixelFormat() (<i>PySpin.IImage</i> method), 327
GetSize() (<i>PySpin.ImageList</i> method), 55, 354	GetTLPixelFormat() (<i>PySpin.Image</i> method), 50, 348
GetSize() (<i>PySpin.InterfaceList</i> method), 68, 379	GetTLPixelFormatNamespace() (<i>PySpin.IImage</i> method), 327
GetStandardNameSpace() (<i>PySpin.CDeviceInfoPtr</i> method), 93	GetTLPixelFormatNamespace() (<i>PySpin.Image</i> method), 50, 349
GetStandardNameSpace() (<i>PySpin.IDeviceInfo</i> method), 180	GetTLStreamNodeMap() (<i>PySpin.CameraBase</i> method), 37, 144
GetStandardNameSpace() (<i>PySpin.NodeMap</i> method), 394	GetTLStreamNodeMap() (<i>PySpin.ICameraBase</i> method), 175
GetStreamChannelID() (<i>PySpin.ChunkData</i> method), 45, 153	GetToolTip() (<i>PySpin.CBooleanPtr</i> method), 86
GetStreamChannelID() (<i>PySpin.IChunkData</i> method), 179	GetToolTip() (<i>PySpin.CCategoryPtr</i> method), 89
GetStreamIndex() (<i>PySpin.IImage</i> method), 327	GetToolTip() (<i>PySpin.CCommandPtr</i> method), 92
GetStreamIndex() (<i>PySpin.Image</i> method), 50, 348	GetToolTip() (<i>PySpin.CDeviceInfoPtr</i> method), 93
GetStride() (<i>PySpin.IImage</i> method), 327	GetToolTip() (<i>PySpin.CEnumEntryPtr</i> method), 95
GetStride() (<i>PySpin.Image</i> method), 50, 348	GetToolTip() (<i>PySpin.CEnumerationPtr</i> method), 98
GetSupportedSchemaVersions() (<i>PySpin.CNodeMapDynPtr</i> method), 104	GetToolTip() (<i>PySpin.CIntegerPtr</i> method), 102
GetSupportedSchemaVersions() (<i>PySpin.INodeMapDyn</i> method), 338	GetToolTip() (<i>PySpin.CNodePtr</i> method), 107
GetSupportedSchemaVersions() (<i>PySpin.NodeMap</i> method), 394	GetToolTip() (<i>PySpin.CRegisterPtr</i> method), 110
GetSymbolic() (<i>PySpin.CEnumEntryPtr</i> method), 95	GetToolTip() (<i>PySpin.CStringPtr</i> method), 114
GetSymbolic() (<i>PySpin.EnumEntryNode</i> method), 166	GetToolTip() (<i>PySpin.CValuePtr</i> method), 116
GetSymbolic() (<i>PySpin.IEnumEntry</i> method), 181	GetToolTip() (<i>PySpin.IDeviceInfo</i> method), 181
GetSymbolics() (<i>PySpin.CEnumerationPtr</i> method), 98	GetToolTip() (<i>PySpin.INode</i> method), 337
GetSymbolics() (<i>PySpin.EnumNode</i> method), 168	GetToolTip() (<i>PySpin.Node</i> method), 390
GetSymbolics() (<i>PySpin.IEnumeration</i> method), 182	GetToolTip() (<i>PySpin.NodeMap</i> method), 395
GetThreadName() (<i>PySpin.LoggingEventData</i> method), 386	GetTransferBlockID() (<i>PySpin.ChunkData</i> method), 45, 153
GetTimerValue() (<i>PySpin.ChunkData</i> method), 45, 153	GetTransferBlockID() (<i>PySpin.IChunkData</i> method), 179
GetTimerValue() (<i>PySpin.IChunkData</i> method), 179	GetTransferQueueCurrentBlockCount() (<i>PySpin.ChunkData</i> method), 45, 153
GetTimestamp() (<i>PySpin.ChunkData</i> method), 45, 153	GetTransferQueueCurrentBlockCount() (<i>PySpin.IChunkData</i> method), 179
GetTimestamp() (<i>PySpin.IChunkData</i> method), 179	GetUniqueID() (<i>PySpin.CameraBase</i> method), 37, 145
GetTimeStamp() (<i>PySpin.IImage</i> method), 327	GetUniqueID() (<i>PySpin.ICameraBase</i> method), 175
GetTimeStamp() (<i>PySpin.Image</i> method), 50, 349	GetUnit() (<i>PySpin.CIntegerPtr</i> method), 102
GetTimestamp() (<i>PySpin.LoggingEventData</i> method), 386	GetUnit() (<i>PySpin.FloatNode</i> method), 170
GetTimestampLatchValue() (<i>PySpin.ChunkData</i> method), 45, 153	GetUnit() (<i>PySpin.IFloat</i> method), 325
GetTimestampLatchValue() (<i>PySpin.IChunkData</i> method), 179	GetUnit() (<i>PySpin.IInteger</i> method), 333
	GetUnit() (<i>PySpin.IntegerNode</i> method), 377
	GetUserBufferCount() (<i>PySpin.CameraBase</i> method), 37, 145
	GetUserBufferCount() (<i>PySpin.ICameraBase</i> method), 175

<code>GetUserBufferSize()</code> (<i>PySpin.CameraBase</i> method), 37, 145	<code>GetValue()</code> (<i>PySpin.IEnumerationT_BlackLevelAutoEnums</i> method), 194
<code>GetUserBufferSize()</code> (<i>PySpin.ICameraBase</i> method), 175	<code>GetValue()</code> (<i>PySpin.IEnumerationT_BlackLevelSelectorEnums</i> method), 195
<code>GetUserBufferTotalSize()</code> (<i>PySpin.CameraBase</i> method), 37, 145	<code>GetValue()</code> (<i>PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums</i> method), 195
<code>GetUserBufferTotalSize()</code> (<i>PySpin.ICameraBase</i> method), 175	<code>GetValue()</code> (<i>PySpin.IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums</i> method), 196
<code>GetValidPayloadSize()</code> (<i>PySpin.IImage</i> method), 327	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkBlackLevelSelectorEnums</i> method), 197
<code>GetValidPayloadSize()</code> (<i>PySpin.Image</i> method), 50, 349	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkCounterSelectorEnums</i> method), 197
<code>GetValue()</code> (<i>PySpin.BooleanNode</i> method), 84	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkEncoderSelectorEnums</i> method), 198
<code>GetValue()</code> (<i>PySpin.CBooleanPtr</i> method), 86	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkEncoderStatusEnums</i> method), 199
<code>GetValue()</code> (<i>PySpin.CEnumEntryPtr</i> method), 95	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkExposureTimeSelectorEnums</i> method), 199
<code>GetValue()</code> (<i>PySpin.CIntegerPtr</i> method), 102	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkGainSelectorEnums</i> method), 200
<code>GetValue()</code> (<i>PySpin.CStringPtr</i> method), 114	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkImageComponentEnums</i> method), 201
<code>GetValue()</code> (<i>PySpin.EnumEntryNode</i> method), 166	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkPixelFormatEnums</i> method), 201
<code>GetValue()</code> (<i>PySpin.FloatNode</i> method), 170	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkRegionIDEnums</i> method), 202
<code>GetValue()</code> (<i>PySpin.IBoolean</i> method), 174	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkScan3dCoordinateReferenceEnums</i> method), 203
<code>GetValue()</code> (<i>PySpin.IEnumEntry</i> method), 181	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkScan3dCoordinateSelectorEnums</i> method), 203
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AcquisitionModeEnums</i> method), 183	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkScan3dCoordinateSystemEnums</i> method), 204
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AcquisitionStatusSelectorEnums</i> method), 183	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkScan3dCoordinateSystemReferenceEnums</i> method), 205
<code>GetValue()</code> (<i>PySpin.IEnumerationT_ActionSelectorEnums</i> method), 184	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkScan3dCoordinateTransformEnums</i> method), 205
<code>GetValue()</code> (<i>PySpin.IEnumerationT_ActionUnconditionalMethodEnums</i> method), 185	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkScan3dDistanceUnitEnums</i> method), 206
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AdcBitDepthEnums</i> method), 185	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkScan3dOutputModeEnums</i> method), 207
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AutoAlgorithmSelectorEnums</i> method), 186	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkSelectorEnums</i> method), 207
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AutoExposureControlEnums</i> method), 187	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkSourceIDEnums</i> method), 208
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AutoExposureLightingEnums</i> method), 187	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkTimerSelectorEnums</i> method), 209
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AutoExposureMeteringEnums</i> method), 188	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ChunkTransferStreamIDEnums</i> method), 209
<code>GetValue()</code> (<i>PySpin.IEnumerationT_AutoExposureTargetEnums</i> method), 189	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ClConfigurationEnums</i> method), 210
<code>GetValue()</code> (<i>PySpin.IEnumerationT_BalanceRatioSelectorEnums</i> method), 189	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ClTimeSlotsCountEnums</i> method), 211
<code>GetValue()</code> (<i>PySpin.IEnumerationT_BalanceWhiteAutoEnums</i> method), 190	<code>GetValue()</code> (<i>PySpin.IEnumerationT_ColorTransformationSelectorEnums</i> method), 211
<code>GetValue()</code> (<i>PySpin.IEnumerationT_BalanceWhiteAutoProgramEnums</i> method), 191	
<code>GetValue()</code> (<i>PySpin.IEnumerationT_BinningHorizontalModeEnums</i> method), 191	
<code>GetValue()</code> (<i>PySpin.IEnumerationT_BinningSelectorEnums</i> method), 192	
<code>GetValue()</code> (<i>PySpin.IEnumerationT_BinningVerticalModeEnums</i> method), 193	
<code>GetValue()</code> (<i>PySpin.IEnumerationT_BlackLevelAutoBalanceEnums</i> method), 193	

`GetValue()` (PySpin.IEnumerationT_ColorTransformationEnums method), 212

`GetValue()` (PySpin.IEnumerationT_ComponentDestinationEnums method), 213

`GetValue()` (PySpin.IEnumerationT_ComponentSelectorEnums method), 213

`GetValue()` (PySpin.IEnumerationT_CompressionSaturationEnums method), 214

`GetValue()` (PySpin.IEnumerationT_CounterEventActivationEnums method), 215

`GetValue()` (PySpin.IEnumerationT_CounterEventSourceEnums method), 215

`GetValue()` (PySpin.IEnumerationT_CounterResetActivationEnums method), 216

`GetValue()` (PySpin.IEnumerationT_CounterResetSourceEnums method), 217

`GetValue()` (PySpin.IEnumerationT_CounterSelectorEnums method), 217

`GetValue()` (PySpin.IEnumerationT_CounterStatusEnums method), 218

`GetValue()` (PySpin.IEnumerationT_CounterTriggerActivationEnums method), 219

`GetValue()` (PySpin.IEnumerationT_CounterTriggerSourceEnums method), 219

`GetValue()` (PySpin.IEnumerationT_CxpConnectionTestModeEnums method), 220

`GetValue()` (PySpin.IEnumerationT_CxpLinkConfigurationEnums method), 221

`GetValue()` (PySpin.IEnumerationT_CxpLinkConfigurationEnums method), 221

`GetValue()` (PySpin.IEnumerationT_CxpLinkConfigurationEnums method), 222

`GetValue()` (PySpin.IEnumerationT_CxpPoCxpStatusEnums method), 223

`GetValue()` (PySpin.IEnumerationT_DecimationHorizontalModeEnums method), 223

`GetValue()` (PySpin.IEnumerationT_DecimationSelectorEnums method), 224

`GetValue()` (PySpin.IEnumerationT_DecimationVerticalModeEnums method), 225

`GetValue()` (PySpin.IEnumerationT_DefectCorrectionModeEnums method), 225

`GetValue()` (PySpin.IEnumerationT_DeinterlacingEnums method), 226

`GetValue()` (PySpin.IEnumerationT_DeviceAccessStatusEnums method), 227

`GetValue()` (PySpin.IEnumerationT_DeviceCharacterSetEnums method), 227

`GetValue()` (PySpin.IEnumerationT_DeviceClockSelectorEnums method), 228

`GetValue()` (PySpin.IEnumerationT_DeviceConnectionStatusEnums method), 229

`GetValue()` (PySpin.IEnumerationT_DeviceCurrentSpeedEnums method), 229

`GetValue()` (PySpin.IEnumerationT_DeviceEndiannessMechanismEnums method), 230

`GetValue()` (PySpin.IEnumerationT_DeviceIndicatorModeEnums method), 231

`GetValue()` (PySpin.IEnumerationT_DeviceLinkHeartbeatModeEnums method), 231

`GetValue()` (PySpin.IEnumerationT_DeviceLinkThroughputLimitModeEnums method), 232

`GetValue()` (PySpin.IEnumerationT_DevicePowerSupplySelectorEnums method), 233

`GetValue()` (PySpin.IEnumerationT_DeviceRegistersEndiannessEnums method), 233

`GetValue()` (PySpin.IEnumerationT_DeviceScanTypeEnums method), 234

`GetValue()` (PySpin.IEnumerationT_DeviceSensorChromaEnums method), 235

`GetValue()` (PySpin.IEnumerationT_DeviceSerialPortBaudRateEnums method), 235

`GetValue()` (PySpin.IEnumerationT_DeviceSerialPortSelectorEnums method), 236

`GetValue()` (PySpin.IEnumerationT_DeviceStreamChannelEndiannessEnums method), 237

`GetValue()` (PySpin.IEnumerationT_DeviceStreamChannelTypeEnums method), 237

`GetValue()` (PySpin.IEnumerationT_DeviceTapGeometryEnums method), 239

`GetValue()` (PySpin.IEnumerationT_DeviceTemperatureSelectorEnums method), 239

`GetValue()` (PySpin.IEnumerationT_DeviceTLTypeEnums method), 238

`GetValue()` (PySpin.IEnumerationT_DeviceTypeEnums method), 240

`GetValue()` (PySpin.IEnumerationT_DeviceTypeEnums method), 241

`GetValue()` (PySpin.IEnumerationT_EncoderModeEnums method), 241

`GetValue()` (PySpin.IEnumerationT_EncoderOutputModeEnums method), 242

`GetValue()` (PySpin.IEnumerationT_EncoderResetActivationEnums method), 243

`GetValue()` (PySpin.IEnumerationT_EncoderResetSourceEnums method), 243

`GetValue()` (PySpin.IEnumerationT_EncoderSelectorEnums method), 244

`GetValue()` (PySpin.IEnumerationT_EncoderSourceAEnums method), 245

`GetValue()` (PySpin.IEnumerationT_EncoderSourceBEnums method), 245

`GetValue()` (PySpin.IEnumerationT_EncoderStatusEnums method), 246

`GetValue()` (PySpin.IEnumerationT_EventNotificationEnums method), 247

`GetValue()` (PySpin.IEnumerationT_EventSelectorEnums method), 247

GetValue() (PySpin.IEnumerationT_ExposureActiveModeEnums method), 248
 GetValue() (PySpin.IEnumerationT_ExposureAutoEnums method), 249
 GetValue() (PySpin.IEnumerationT_ExposureModeEnums method), 249
 GetValue() (PySpin.IEnumerationT_ExposureTimeModeEnums method), 250
 GetValue() (PySpin.IEnumerationT_ExposureTimeSelectorEnums method), 251
 GetValue() (PySpin.IEnumerationT_ExternalVoltageSelectorEnums method), 251
 GetValue() (PySpin.IEnumerationT_FfcModeEnums method), 253
 GetValue() (PySpin.IEnumerationT_FileOpenModeEnums method), 253
 GetValue() (PySpin.IEnumerationT_FileOperationSelectorEnums method), 254
 GetValue() (PySpin.IEnumerationT_FileOperationStatusEnums method), 255
 GetValue() (PySpin.IEnumerationT_FileSelectorEnums method), 255
 GetValue() (PySpin.IEnumerationT_FLIRFilterDriverStatusEnums method), 252
 GetValue() (PySpin.IEnumerationT_GainAutoBalanceEnums method), 257
 GetValue() (PySpin.IEnumerationT_GainAutoEnums method), 257
 GetValue() (PySpin.IEnumerationT_GainConversionEnums method), 258
 GetValue() (PySpin.IEnumerationT_GainSelectorEnums method), 259
 GetValue() (PySpin.IEnumerationT_GenICamXMLLocationEnums method), 259
 GetValue() (PySpin.IEnumerationT_GevCCPEnum method), 260
 GetValue() (PySpin.IEnumerationT_GevCCPEnums method), 261
 GetValue() (PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums method), 261
 GetValue() (PySpin.IEnumerationT_GevGVCPExtendedStatusEnums method), 262
 GetValue() (PySpin.IEnumerationT_GevGVSPExtendedIDEnums method), 263
 GetValue() (PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnums method), 263
 GetValue() (PySpin.IEnumerationT_GevIEEE1588ModeEnums method), 264
 GetValue() (PySpin.IEnumerationT_GevIEEE1588StatusEnums method), 265
 GetValue() (PySpin.IEnumerationT_GevIEEE1588StatusLocationEnums method), 265
 GetValue() (PySpin.IEnumerationT_GevIPConfigurationStatusEnums method), 266
 GetValue() (PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnums method), 267
 GetValue() (PySpin.IEnumerationT_GevSCPDDirectionEnums method), 267
 GetValue() (PySpin.IEnumerationT_GevSupportedOptionSelectorEnums method), 268
 GetValue() (PySpin.IEnumerationT_GUIXMLLocationEnum method), 256
 GetValue() (PySpin.IEnumerationT_ImageComponentSelectorEnums method), 269
 GetValue() (PySpin.IEnumerationT_ImageCompressionJPEGFormatOptionsEnums method), 269
 GetValue() (PySpin.IEnumerationT_ImageCompressionModeEnums method), 270
 GetValue() (PySpin.IEnumerationT_ImageCompressionRateOptionEnums method), 271
 GetValue() (PySpin.IEnumerationT_InterfaceTypeEnum method), 271
 GetValue() (PySpin.IEnumerationT_LensShadingCoefficientActiveSetEnums method), 273
 GetValue() (PySpin.IEnumerationT_LensShadingCorrectionModeEnums method), 273
 GetValue() (PySpin.IEnumerationT_LineFormatEnums method), 274
 GetValue() (PySpin.IEnumerationT_LineInputFilterSelectorEnums method), 275
 GetValue() (PySpin.IEnumerationT_LineModeEnums method), 275
 GetValue() (PySpin.IEnumerationT_LineSelectorEnums method), 276
 GetValue() (PySpin.IEnumerationT_LineSourceEnums method), 277
 GetValue() (PySpin.IEnumerationT_LogicBlockLUTInputActivationEnums method), 277
 GetValue() (PySpin.IEnumerationT_LogicBlockLUTInputSelectorEnums method), 278
 GetValue() (PySpin.IEnumerationT_LogicBlockLUTInputSourceEnums method), 279
 GetValue() (PySpin.IEnumerationT_LogicBlockLUTSelectorEnums method), 279
 GetValue() (PySpin.IEnumerationT_LogicBlockSelectorEnums method), 280
 GetValue() (PySpin.IEnumerationT_LUTSelectorEnums method), 272
 GetValue() (PySpin.IEnumerationT_MultiRoiConfigurationInvalidReasonEnums method), 281
 GetValue() (PySpin.IEnumerationT_MultiRoiSelectorEnums method), 281
 GetValue() (PySpin.IEnumerationT_PixelColorFilterEnums method), 283
 GetValue() (PySpin.IEnumerationT_PixelFormatEnums method), 283
 GetValue() (PySpin.IEnumerationT_PixelFormatInfoSelectorEnums method), 284

<code>GetValue()</code> (PySpin.IEnumerationT_PixelSizeEnums method), 285	<code>GetValue()</code> (PySpin.IEnumerationT_SoftwareSignalSelectorEnums method), 302
<code>GetValue()</code> (PySpin.IEnumerationT_POEStatusEnum method), 282	<code>GetValue()</code> (PySpin.IEnumerationT_SourceSelectorEnums method), 303
<code>GetValue()</code> (PySpin.IEnumerationT_RegionDestinationEnum method), 285	<code>GetValue()</code> (PySpin.IEnumerationT_StereoResolutionEnums method), 303
<code>GetValue()</code> (PySpin.IEnumerationT_RegionModeEnums method), 286	<code>GetValue()</code> (PySpin.IEnumerationT_StreamBufferCountModeEnum method), 304
<code>GetValue()</code> (PySpin.IEnumerationT_RegionSelectorEnum method), 287	<code>GetValue()</code> (PySpin.IEnumerationT_StreamBufferHandlingModeEnum method), 305
<code>GetValue()</code> (PySpin.IEnumerationT_RgbTransformLightSource method), 287	<code>GetValue()</code> (PySpin.IEnumerationT_StreamModeEnum method), 305
<code>GetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateReference method), 288	<code>GetValue()</code> (PySpin.IEnumerationT_StreamTypeEnum method), 306
<code>GetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateSelector method), 289	<code>GetValue()</code> (PySpin.IEnumerationT_TeledyneGigeVisionFilterDriverStatus method), 307
<code>GetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateSystem method), 289	<code>GetValue()</code> (PySpin.IEnumerationT_TestPatternEnums method), 308
<code>GetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateSystemSelector method), 290	<code>GetValue()</code> (PySpin.IEnumerationT_TestPatternGeneratorSelectorEnums method), 309
<code>GetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateTransform method), 291	<code>GetValue()</code> (PySpin.IEnumerationT_TimerSelectorEnums method), 309
<code>GetValue()</code> (PySpin.IEnumerationT_Scan3dDistanceUnitEnum method), 291	<code>GetValue()</code> (PySpin.IEnumerationT_TimerStatusEnums method), 310
<code>GetValue()</code> (PySpin.IEnumerationT_Scan3dOutputModeEnum method), 292	<code>GetValue()</code> (PySpin.IEnumerationT_TimerTriggerActivationEnums method), 311
<code>GetValue()</code> (PySpin.IEnumerationT_SensorDigitizationTap method), 293	<code>GetValue()</code> (PySpin.IEnumerationT_TimerTriggerSourceEnums method), 311
<code>GetValue()</code> (PySpin.IEnumerationT_SensorShutterModeEnum method), 293	<code>GetValue()</code> (PySpin.IEnumerationT_TLTypeEnum method), 307
<code>GetValue()</code> (PySpin.IEnumerationT_SensorTapsEnums method), 294	<code>GetValue()</code> (PySpin.IEnumerationT_TransferComponentSelectorEnums method), 312
<code>GetValue()</code> (PySpin.IEnumerationT_SequencerConfiguration method), 295	<code>GetValue()</code> (PySpin.IEnumerationT_TransferControlModeEnums method), 313
<code>GetValue()</code> (PySpin.IEnumerationT_SequencerConfigurationSelector method), 295	<code>GetValue()</code> (PySpin.IEnumerationT_TransferOperationModeEnums method), 313
<code>GetValue()</code> (PySpin.IEnumerationT_SequencerModeEnum method), 296	<code>GetValue()</code> (PySpin.IEnumerationT_TransferQueueModeEnums method), 314
<code>GetValue()</code> (PySpin.IEnumerationT_SequencerSetValidEnum method), 297	<code>GetValue()</code> (PySpin.IEnumerationT_TransferSelectorEnums method), 315
<code>GetValue()</code> (PySpin.IEnumerationT_SequencerTriggerActivation method), 297	<code>GetValue()</code> (PySpin.IEnumerationT_TransferStatusSelectorEnums method), 315
<code>GetValue()</code> (PySpin.IEnumerationT_SequencerTriggerSource method), 298	<code>GetValue()</code> (PySpin.IEnumerationT_TransferTriggerActivationEnums method), 316
<code>GetValue()</code> (PySpin.IEnumerationT_SerialPortBaudRateEnum method), 299	<code>GetValue()</code> (PySpin.IEnumerationT_TransferTriggerModeEnums method), 317
<code>GetValue()</code> (PySpin.IEnumerationT_SerialPortParityEnum method), 299	<code>GetValue()</code> (PySpin.IEnumerationT_TransferTriggerSelectorEnums method), 317
<code>GetValue()</code> (PySpin.IEnumerationT_SerialPortSelectorEnum method), 300	<code>GetValue()</code> (PySpin.IEnumerationT_TransferTriggerSourceEnums method), 318
<code>GetValue()</code> (PySpin.IEnumerationT_SerialPortSourceEnum method), 301	<code>GetValue()</code> (PySpin.IEnumerationT_TriggerActivationEnums method), 319
<code>GetValue()</code> (PySpin.IEnumerationT_SerialPortStopBitsEnum method), 301	<code>GetValue()</code> (PySpin.IEnumerationT_TriggerModeEnums method), 319

GetValue() (PySpin.IEnumerationT_TriggerOverlapEnum method), 320	GetYOffset() (PySpin.Image method), 51, 350
GetValue() (PySpin.IEnumerationT_TriggerSelectorEnum method), 321	GetYPadding() (PySpin.Image method), 327
GetValue() (PySpin.IEnumerationT_TriggerSourceEnums method), 321	GetYPadding() (PySpin.Image method), 51, 350
GetValue() (PySpin.IEnumerationT_U3VCurrentSpeedEnum method), 322	GevActionAckRequired
GetValue() (PySpin.IEnumerationT_UserOutputSelectorEnum method), 323	GevActionDeviceKey (PySpin.TransportLayerInterface property), 79, 412
GetValue() (PySpin.IEnumerationT_UserSetDefaultEnum method), 323	GevActionGroupKey (PySpin.TransportLayerInterface property), 79, 412
GetValue() (PySpin.IEnumerationT_UserSetSelectorEnum method), 324	GevActionGroupMask (PySpin.TransportLayerInterface property), 79, 412
GetValue() (PySpin.IEnumerationT_WhiteClipSelectorEnum method), 325	GevActionTime (PySpin.TransportLayerInterface property), 79, 412
GetValue() (PySpin.IFloat method), 325	GevActiveLinkCount (PySpin.Camera property), 25, 132
GetValue() (PySpin.Integer method), 333	GEVAutoAssignIPEnable
GetValue() (PySpin.IntegerNode method), 377	(PySpin.TransportLayerSystem property), 415
GetValue() (PySpin.IString method), 342	GevCCP (PySpin.Camera property), 25, 132
GetValue() (PySpin.StringNode method), 404	GevCCP (PySpin.TransportLayerDevice property), 78, 411
GetValueOfEnvironmentVariable() (in module PySpin), 173	GevCurrentDefaultGateway (PySpin.Camera property), 25, 132
GetVendorName() (PySpin.CDeviceInfoPtr method), 93	GevCurrentIPAddress (PySpin.Camera property), 25, 132
GetVendorName() (PySpin.IDeviceInfo method), 181	GevCurrentIPConfigurationDHCP (PySpin.Camera property), 25, 132
GetVendorName() (PySpin.NodeMap method), 395	GevCurrentIPConfigurationLLA (PySpin.Camera property), 25, 132
GetVersion() (PySpin.InferenceBoundingBoxResult method), 375	GevCurrentIPConfigurationPersistentIP (PySpin.Camera property), 25, 132
GetVersionGuid() (PySpin.CDeviceInfoPtr method), 93	GevCurrentPhysicalLinkConfiguration (PySpin.Camera property), 25, 132
GetVersionGuid() (PySpin.IDeviceInfo method), 181	GevCurrentSubnetMask (PySpin.Camera property), 25, 132
GetVersionGuid() (PySpin.NodeMap method), 395	GevDeviceAutoForceIP
GetVisibility() (PySpin.CBooleanPtr method), 86	(PySpin.TransportLayerDevice property), 78, 411
GetVisibility() (PySpin.CCategoryPtr method), 89	GevDeviceAutoForceIP
GetVisibility() (PySpin.CCommandPtr method), 92	(PySpin.TransportLayerInterface property), 79, 412
GetVisibility() (PySpin.CEnumEntryPtr method), 95	GevDeviceDisableDiscovery
GetVisibility() (PySpin.CEnumerationPtr method), 98	(PySpin.TransportLayerInterface property), 79, 412
GetVisibility() (PySpin.CIntegerPtr method), 102	GevDeviceDiscoverMaximumPacketSize
GetVisibility() (PySpin.CNodePtr method), 107	(PySpin.TransportLayerDevice property), 78, 411
GetVisibility() (PySpin.CRegisterPtr method), 110	GevDeviceDiscoveryEnabled
GetVisibility() (PySpin.CStringPtr method), 114	(PySpin.TransportLayerInterface property), 79, 412
GetVisibility() (PySpin.CValuePtr method), 116	GevDeviceEnabledDiscovery
GetVisibility() (PySpin.INode method), 337	(PySpin.TransportLayerInterface property), 79, 412
GetVisibility() (PySpin.Node method), 390	
GetWidth() (PySpin.ChunkData method), 45, 153	
GetWidth() (PySpin.IChunkData method), 179	
GetWidth() (PySpin.Image method), 327	
GetWidth() (PySpin.Image method), 50, 349	
GetXOffset() (PySpin.Image method), 327	
GetXOffset() (PySpin.Image method), 51, 349	
GetXPadding() (PySpin.Image method), 327	
GetXPadding() (PySpin.Image method), 51, 349	
GetYOffset() (PySpin.Image method), 327	

GevDeviceForceGateway (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevDeviceSubnetMask (<i>PySpin.TransportLayerInterface</i> property), 79, 413
GevDeviceForceGateway (<i>PySpin.TransportLayerInterface</i> property), 79, 412	GevDiscoveryAckDelay (<i>PySpin.Camera</i> property), 25, 132
GevDeviceForceIP (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevFirstURL (<i>PySpin.Camera</i> property), 25, 132
GevDeviceForceIP (<i>PySpin.TransportLayerInterface</i> property), 79, 413	GevGVCPExtendedStatusCodes (<i>PySpin.Camera</i> prop- erty), 25, 132
GevDeviceForceIPAddress (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevGVCPExtendedStatusCodesSelector (<i>PySpin.Camera</i> property), 25, 132
GevDeviceForceIPAddress (<i>PySpin.TransportLayerInterface</i> property), 79, 413	GevGVCPHeartbeatDisable (<i>PySpin.Camera</i> prop- erty), 25, 133
GevDeviceForceSubnetMask (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevGVCPPendingAck (<i>PySpin.Camera</i> property), 25, 133
GevDeviceForceSubnetMask (<i>PySpin.TransportLayerInterface</i> property), 79, 413	GevGVCPPendingTimeout (<i>PySpin.Camera</i> property), 25, 133
GevDeviceGateway (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevGVSPExtendedIDMode (<i>PySpin.Camera</i> property), 25, 133
GevDeviceGateway (<i>PySpin.TransportLayerInterface</i> property), 79, 413	GevHeartbeatTimeout (<i>PySpin.Camera</i> property), 25, 133
GevDeviceIPAddress (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevIEEE1588 (<i>PySpin.Camera</i> property), 25, 133
GevDeviceIPAddress (<i>PySpin.TransportLayerInterface</i> property), 79, 413	GevIEEE1588ClockAccuracy (<i>PySpin.Camera</i> prop- erty), 25, 133
GevDeviceIsWrongSubnet (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevIEEE1588ClockId (<i>PySpin.Camera</i> property), 25, 133
GevDeviceMACAddress (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevIEEE1588DataSetLatch (<i>PySpin.Camera</i> prop- erty), 25, 133
GevDeviceMACAddress (<i>PySpin.TransportLayerInterface</i> property), 79, 413	GevIEEE1588Mode (<i>PySpin.Camera</i> property), 26, 133
GevDeviceMaximumPacketSize (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevIEEE1588OffsetFromMasterLatched (<i>PySpin.Camera</i> property), 26, 133
GevDeviceMaximumRetryCount (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevIEEE1588ParentClockIdLatched (<i>PySpin.Camera</i> property), 26, 133
GevDeviceModeIsBigEndian (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevIEEE1588Status (<i>PySpin.Camera</i> property), 26, 133
GevDevicePort (<i>PySpin.TransportLayerDevice</i> prop- erty), 78, 411	GevIEEE1588StatusLatched (<i>PySpin.Camera</i> prop- erty), 26, 133
GevDeviceReadAndWriteTimeout (<i>PySpin.TransportLayerDevice</i> property), 78, 411	GevInterfaceDefaultGateway (<i>PySpin.TransportLayerSystem</i> property), 416
GevDeviceSubnetMask (<i>PySpin.TransportLayerDevice</i> property), 78, 412	GevInterfaceDefaultIPAddress (<i>PySpin.TransportLayerSystem</i> property), 416
	GevInterfaceDefaultSubnetMask (<i>PySpin.TransportLayerSystem</i> property), 416
	GevInterfaceGateway (<i>PySpin.TransportLayerInterface</i> property), 80, 413
	GevInterfaceGatewaySelector (<i>PySpin.TransportLayerInterface</i> property), 80, 413
	GevInterfaceIsIPConflict (<i>PySpin.TransportLayerInterface</i> property), 80, 413

- `GevInterfaceMACAddress` (*PySpin.TransportLayerInterface* property), 80, 413
- `GevInterfaceMACAddress` (*PySpin.TransportLayerSystem* property), 416
- `GevInterfaceMTU` (*PySpin.TransportLayerInterface* property), 80, 413
- `GevInterfaceReceiveLinkSpeed` (*PySpin.TransportLayerInterface* property), 80, 413
- `GevInterfaceSelector` (*PySpin.Camera* property), 26, 133
- `GevInterfaceSubnetIPAddress` (*PySpin.TransportLayerInterface* property), 80, 413
- `GevInterfaceSubnetMask` (*PySpin.TransportLayerInterface* property), 80, 413
- `GevInterfaceSubnetSelector` (*PySpin.TransportLayerInterface* property), 80, 413
- `GevInterfaceTransmitLinkSpeed` (*PySpin.TransportLayerInterface* property), 80, 413
- `GevIPConfigurationStatus` (*PySpin.Camera* property), 26, 133
- `GevMACAddress` (*PySpin.Camera* property), 26, 133
- `GevMCDA` (*PySpin.Camera* property), 26, 133
- `GevMCPHostPort` (*PySpin.Camera* property), 26, 133
- `GevMCRC` (*PySpin.Camera* property), 26, 133
- `GevMCSP` (*PySpin.Camera* property), 26, 133
- `GevMCTT` (*PySpin.Camera* property), 26, 133
- `GevNumberOfActiveLinks` (*PySpin.Camera* property), 26, 133
- `GevNumberOfInterfaces` (*PySpin.Camera* property), 26, 133
- `GevPAUSEFrameReception` (*PySpin.Camera* property), 26, 133
- `GevPAUSEFrameTransmission` (*PySpin.Camera* property), 26, 133
- `GevPersistentDefaultGateway` (*PySpin.Camera* property), 26, 133
- `GevPersistentIPAddress` (*PySpin.Camera* property), 26, 133
- `GevPersistentSubnetMask` (*PySpin.Camera* property), 26, 133
- `GevPhysicalLinkConfiguration` (*PySpin.Camera* property), 26, 133
- `GevPhysicalLinkConfigurationCapability` (*PySpin.Camera* property), 26, 133
- `GevPrimaryApplicationIPAddress` (*PySpin.Camera* property), 26, 134
- `GevPrimaryApplicationSocket` (*PySpin.Camera* property), 26, 134
- `GevPrimaryApplicationSwitchoverKey` (*PySpin.Camera* property), 26, 134
- `GevSCCFGAllInTransmission` (*PySpin.Camera* property), 26, 134
- `GevSCCFGExtendedChunkData` (*PySpin.Camera* property), 26, 134
- `GevSCCFGPacketResendDestination` (*PySpin.Camera* property), 26, 134
- `GevSCCFGUnconditionalStreaming` (*PySpin.Camera* property), 26, 134
- `GevSCDA` (*PySpin.Camera* property), 26, 134
- `GevSCPD` (*PySpin.Camera* property), 26, 134
- `GevSCPDDirection` (*PySpin.Camera* property), 27, 134
- `GevSCPHostPort` (*PySpin.Camera* property), 27, 134
- `GevSCPIInterfaceIndex` (*PySpin.Camera* property), 27, 134
- `GevSCPSBigEndian` (*PySpin.Camera* property), 27, 134
- `GevSCPSDoNotFragment` (*PySpin.Camera* property), 27, 134
- `GevSCPSFireTestPacket` (*PySpin.Camera* property), 27, 134
- `GevSCPSPacketSize` (*PySpin.Camera* property), 27, 134
- `GevSCSP` (*PySpin.Camera* property), 27, 134
- `GevSCZoneConfigurationLock` (*PySpin.Camera* property), 27, 134
- `GevSCZoneCount` (*PySpin.Camera* property), 27, 134
- `GevSCZoneDirectionAll` (*PySpin.Camera* property), 27, 134
- `GevSecondURL` (*PySpin.Camera* property), 27, 134
- `GevStreamChannelSelector` (*PySpin.Camera* property), 27, 134
- `GevSupportedOption` (*PySpin.Camera* property), 27, 134
- `GevSupportedOptionSelector` (*PySpin.Camera* property), 27, 134
- `GevTimestampTickFrequency` (*PySpin.Camera* property), 27, 134
- `GevVersionMajor` (*PySpin.TransportLayerDevice* property), 78, 412
- `GevVersionMajor` (*PySpin.TransportLayerSystem* property), 416
- `GevVersionMinor` (*PySpin.TransportLayerDevice* property), 78, 412
- `GevVersionMinor` (*PySpin.TransportLayerSystem* property), 416
- `GUIXMLLocation` (*PySpin.TransportLayerDevice* property), 78, 411
- `GuiXmlManifestAddress` (*PySpin.Camera* property), 27, 134
- `GUIXMLPath` (*PySpin.TransportLayerDevice* property), 78, 411

H

H264Option (class in PySpin), 173
 HasChunkData() (PySpin.Image method), 327
 HasChunkData() (PySpin.Image method), 51, 350
 HasCRC() (PySpin.Image method), 327
 HasCRC() (PySpin.Image method), 51, 350
 HasInc() (PySpin.FloatNode method), 170
 HasInc() (PySpin.IFloat method), 325
 height (PySpin.AVIOption property), 83
 Height (PySpin.Camera property), 27, 134
 height (PySpin.H264Option property), 173
 height (PySpin.MJPGOption property), 387
 HeightMax (PySpin.Camera property), 27, 134
 histogram (PySpin.ChannelStatistics property), 42, 150
 HostAdapterDriverVersion
 (PySpin.TransportLayerInterface property), 80, 413
 HostAdapterName (PySpin.TransportLayerInterface property), 80, 413
 HostAdapterVendor (PySpin.TransportLayerInterface property), 80, 413

I

IBase (class in PySpin), 174
 IBoolean (class in PySpin), 174
 ICameraBase (class in PySpin), 174
 ICameraList (class in PySpin), 176
 ICategory (class in PySpin), 177
 IChunkData (class in PySpin), 177
 ICommand (class in PySpin), 179
 IDestroy (class in PySpin), 179
 IDeviceArrivalEventHandler (class in PySpin), 180
 IDeviceEventHandler (class in PySpin), 180
 IDeviceInfo (class in PySpin), 180
 IDeviceRemovalEventHandler (class in PySpin), 181
 IEnumEntry (class in PySpin), 181
 IEnumeration (class in PySpin), 181
 IEnumerationT_AcquisitionModeEnums (class in PySpin), 182
 IEnumerationT_AcquisitionStatusSelectorEnums (class in PySpin), 183
 IEnumerationT_ActionSelectorEnums (class in PySpin), 184
 IEnumerationT_ActionUnconditionalModeEnums (class in PySpin), 184
 IEnumerationT_AdcBitDepthEnums (class in PySpin), 185
 IEnumerationT_AutoAlgorithmSelectorEnums (class in PySpin), 186
 IEnumerationT_AutoExposureControlPriorityEnums (class in PySpin), 186
 IEnumerationT_AutoExposureLightingModeEnums (class in PySpin), 187
 IEnumerationT_AutoExposureMeteringModeEnums (class in PySpin), 188
 IEnumerationT_AutoExposureTargetGreyValueAutoEnums (class in PySpin), 188
 IEnumerationT_BalanceRatioSelectorEnums (class in PySpin), 189
 IEnumerationT_BalanceWhiteAutoEnums (class in PySpin), 190
 IEnumerationT_BalanceWhiteAutoProfileEnums (class in PySpin), 190
 IEnumerationT_BinningHorizontalModeEnums (class in PySpin), 191
 IEnumerationT_BinningSelectorEnums (class in PySpin), 192
 IEnumerationT_BinningVerticalModeEnums (class in PySpin), 192
 IEnumerationT_BlackLevelAutoBalanceEnums (class in PySpin), 193
 IEnumerationT_BlackLevelAutoEnums (class in PySpin), 194
 IEnumerationT_BlackLevelSelectorEnums (class in PySpin), 194
 IEnumerationT_BsiFlatFieldCorrectionAutoEnums (class in PySpin), 195
 IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums (class in PySpin), 196
 IEnumerationT_ChunkBlackLevelSelectorEnums (class in PySpin), 196
 IEnumerationT_ChunkCounterSelectorEnums (class in PySpin), 197
 IEnumerationT_ChunkEncoderSelectorEnums (class in PySpin), 198
 IEnumerationT_ChunkEncoderStatusEnums (class in PySpin), 198
 IEnumerationT_ChunkExposureTimeSelectorEnums (class in PySpin), 199
 IEnumerationT_ChunkGainSelectorEnums (class in PySpin), 200
 IEnumerationT_ChunkImageComponentEnums (class in PySpin), 200
 IEnumerationT_ChunkPixelFormatEnums (class in PySpin), 201
 IEnumerationT_ChunkRegionIDEnums (class in PySpin), 202
 IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums (class in PySpin), 202
 IEnumerationT_ChunkScan3dCoordinateSelectorEnums (class in PySpin), 203
 IEnumerationT_ChunkScan3dCoordinateSystemEnums (class in PySpin), 204
 IEnumerationT_ChunkScan3dCoordinateSystemReferenceEnums (class in PySpin), 204
 IEnumerationT_ChunkScan3dCoordinateTransformSelectorEnums (class in PySpin), 205

<code>IEnumerationT_ChunkScan3dDistanceUnitEnums</code> (class in <code>PySpin</code>), 206	<code>IEnumerationT_DecimationSelectorEnums</code> (class in <code>PySpin</code>), 224
<code>IEnumerationT_ChunkScan3dOutputModeEnums</code> (class in <code>PySpin</code>), 206	<code>IEnumerationT_DecimationVerticalModeEnums</code> (class in <code>PySpin</code>), 224
<code>IEnumerationT_ChunkSelectorEnums</code> (class in <code>PySpin</code>), 207	<code>IEnumerationT_DefectCorrectionModeEnums</code> (class in <code>PySpin</code>), 225
<code>IEnumerationT_ChunkSourceIDEnums</code> (class in <code>PySpin</code>), 208	<code>IEnumerationT_DeinterlacingEnums</code> (class in <code>PySpin</code>), 226
<code>IEnumerationT_ChunkTimerSelectorEnums</code> (class in <code>PySpin</code>), 208	<code>IEnumerationT_DeviceAccessStatusEnum</code> (class in <code>PySpin</code>), 226
<code>IEnumerationT_ChunkTransferStreamIDEnums</code> (class in <code>PySpin</code>), 209	<code>IEnumerationT_DeviceCharacterSetEnums</code> (class in <code>PySpin</code>), 227
<code>IEnumerationT_ClConfigurationEnums</code> (class in <code>PySpin</code>), 210	<code>IEnumerationT_DeviceClockSelectorEnums</code> (class in <code>PySpin</code>), 228
<code>IEnumerationT_ClTimeSlotsCountEnums</code> (class in <code>PySpin</code>), 210	<code>IEnumerationT_DeviceConnectionStatusEnums</code> (class in <code>PySpin</code>), 228
<code>IEnumerationT_ColorTransformationSelectorEnums</code> (class in <code>PySpin</code>), 211	<code>IEnumerationT_DeviceCurrentSpeedEnum</code> (class in <code>PySpin</code>), 229
<code>IEnumerationT_ColorTransformationValueSelectorEnums</code> (class in <code>PySpin</code>), 212	<code>IEnumerationT_DeviceEndiannessMechanismEnum</code> (class in <code>PySpin</code>), 230
<code>IEnumerationT_ComponentDestinationEnums</code> (class in <code>PySpin</code>), 212	<code>IEnumerationT_DeviceIndicatorModeEnums</code> (class in <code>PySpin</code>), 230
<code>IEnumerationT_ComponentSelectorEnums</code> (class in <code>PySpin</code>), 213	<code>IEnumerationT_DeviceLinkHeartbeatModeEnums</code> (class in <code>PySpin</code>), 231
<code>IEnumerationT_CompressionSaturationPriorityEnums</code> (class in <code>PySpin</code>), 214	<code>IEnumerationT_DeviceLinkThroughputLimitModeEnums</code> (class in <code>PySpin</code>), 232
<code>IEnumerationT_CounterEventActivationEnums</code> (class in <code>PySpin</code>), 214	<code>IEnumerationT_DevicePowerSupplySelectorEnums</code> (class in <code>PySpin</code>), 232
<code>IEnumerationT_CounterEventSourceEnums</code> (class in <code>PySpin</code>), 215	<code>IEnumerationT_DeviceRegistersEndiannessEnums</code> (class in <code>PySpin</code>), 233
<code>IEnumerationT_CounterResetActivationEnums</code> (class in <code>PySpin</code>), 216	<code>IEnumerationT_DeviceScanTypeEnums</code> (class in <code>PySpin</code>), 234
<code>IEnumerationT_CounterResetSourceEnums</code> (class in <code>PySpin</code>), 216	<code>IEnumerationT_DeviceSensorChromaEnums</code> (class in <code>PySpin</code>), 234
<code>IEnumerationT_CounterSelectorEnums</code> (class in <code>PySpin</code>), 217	<code>IEnumerationT_DeviceSerialPortBaudRateEnums</code> (class in <code>PySpin</code>), 235
<code>IEnumerationT_CounterStatusEnums</code> (class in <code>PySpin</code>), 218	<code>IEnumerationT_DeviceSerialPortSelectorEnums</code> (class in <code>PySpin</code>), 236
<code>IEnumerationT_CounterTriggerActivationEnums</code> (class in <code>PySpin</code>), 218	<code>IEnumerationT_DeviceStreamChannelEndiannessEnums</code> (class in <code>PySpin</code>), 236
<code>IEnumerationT_CounterTriggerSourceEnums</code> (class in <code>PySpin</code>), 219	<code>IEnumerationT_DeviceStreamChannelTypeEnums</code> (class in <code>PySpin</code>), 237
<code>IEnumerationT_CxpConnectionTestModeEnums</code> (class in <code>PySpin</code>), 220	<code>IEnumerationT_DeviceTapGeometryEnums</code> (class in <code>PySpin</code>), 238
<code>IEnumerationT_CxpLinkConfigurationEnums</code> (class in <code>PySpin</code>), 220	<code>IEnumerationT_DeviceTemperatureSelectorEnums</code> (class in <code>PySpin</code>), 239
<code>IEnumerationT_CxpLinkConfigurationPreferredEnums</code> (class in <code>PySpin</code>), 221	<code>IEnumerationT_DeviceTLTypeEnums</code> (class in <code>PySpin</code>), 238
<code>IEnumerationT_CxpLinkConfigurationStatusEnums</code> (class in <code>PySpin</code>), 222	<code>IEnumerationT_DeviceTypeEnum</code> (class in <code>PySpin</code>), 240
<code>IEnumerationT_CxpPoCxpStatusEnums</code> (class in <code>PySpin</code>), 222	<code>IEnumerationT_DeviceTypeEnum</code> (class in <code>PySpin</code>), 240
<code>IEnumerationT_DecimationHorizontalModeEnums</code> (class in <code>PySpin</code>), 223	<code>IEnumerationT_EncoderModeEnums</code> (class in <code>PySpin</code>), 241

<code>IEnumerationT_EncoderOutputModeEnums</code> (class in <code>PySpin</code>), 242	<code>IEnumerationT_GevGVCPEExtendedStatusCodesSelectorEnums</code> (class in <code>PySpin</code>), 262
<code>IEnumerationT_EncoderResetActivationEnums</code> (class in <code>PySpin</code>), 242	<code>IEnumerationT_GevGVSPExtendedIDModeEnums</code> (class in <code>PySpin</code>), 262
<code>IEnumerationT_EncoderResetSourceEnums</code> (class in <code>PySpin</code>), 243	<code>IEnumerationT_GevIEEE1588ClockAccuracyEnums</code> (class in <code>PySpin</code>), 263
<code>IEnumerationT_EncoderSelectorEnums</code> (class in <code>PySpin</code>), 244	<code>IEnumerationT_GevIEEE1588ModeEnums</code> (class in <code>PySpin</code>), 264
<code>IEnumerationT_EncoderSourceAEnums</code> (class in <code>PySpin</code>), 244	<code>IEnumerationT_GevIEEE1588StatusEnums</code> (class in <code>PySpin</code>), 264
<code>IEnumerationT_EncoderSourceBEnums</code> (class in <code>PySpin</code>), 245	<code>IEnumerationT_GevIEEE1588StatusLatchedEnums</code> (class in <code>PySpin</code>), 265
<code>IEnumerationT_EncoderStatusEnums</code> (class in <code>PySpin</code>), 246	<code>IEnumerationT_GevIPConfigurationStatusEnums</code> (class in <code>PySpin</code>), 266
<code>IEnumerationT_EventNotificationEnums</code> (class in <code>PySpin</code>), 246	<code>IEnumerationT_GevPhysicalLinkConfigurationEnums</code> (class in <code>PySpin</code>), 266
<code>IEnumerationT_EventSelectorEnums</code> (class in <code>PySpin</code>), 247	<code>IEnumerationT_GevSCPDDirectionEnums</code> (class in <code>PySpin</code>), 267
<code>IEnumerationT_ExposureActiveModeEnums</code> (class in <code>PySpin</code>), 248	<code>IEnumerationT_GevSupportedOptionSelectorEnums</code> (class in <code>PySpin</code>), 268
<code>IEnumerationT_ExposureAutoEnums</code> (class in <code>PySpin</code>), 248	<code>IEnumerationT_GUIXMLLocationEnum</code> (class in <code>PySpin</code>), 256
<code>IEnumerationT_ExposureModeEnums</code> (class in <code>PySpin</code>), 249	<code>IEnumerationT_ImageComponentSelectorEnums</code> (class in <code>PySpin</code>), 268
<code>IEnumerationT_ExposureTimeModeEnums</code> (class in <code>PySpin</code>), 250	<code>IEnumerationT_ImageCompressionJPEGFormatOptionEnums</code> (class in <code>PySpin</code>), 269
<code>IEnumerationT_ExposureTimeSelectorEnums</code> (class in <code>PySpin</code>), 250	<code>IEnumerationT_ImageCompressionModeEnums</code> (class in <code>PySpin</code>), 270
<code>IEnumerationT_ExternalVoltageSelectorEnums</code> (class in <code>PySpin</code>), 251	<code>IEnumerationT_ImageCompressionRateOptionEnums</code> (class in <code>PySpin</code>), 270
<code>IEnumerationT_FfcModeEnums</code> (class in <code>PySpin</code>), 252	<code>IEnumerationT_InterfaceTypeEnum</code> (class in <code>PySpin</code>), 271
<code>IEnumerationT_FileOpenModeEnums</code> (class in <code>PySpin</code>), 253	<code>IEnumerationT_LensShadingCoefficientActiveSetEnums</code> (class in <code>PySpin</code>), 272
<code>IEnumerationT_FileOperationSelectorEnums</code> (class in <code>PySpin</code>), 254	<code>IEnumerationT_LensShadingCorrectionModeEnums</code> (class in <code>PySpin</code>), 273
<code>IEnumerationT_FileOperationStatusEnums</code> (class in <code>PySpin</code>), 254	<code>IEnumerationT_LineFormatEnums</code> (class in <code>PySpin</code>), 274
<code>IEnumerationT_FileSelectorEnums</code> (class in <code>PySpin</code>), 255	<code>IEnumerationT_LineInputFilterSelectorEnums</code> (class in <code>PySpin</code>), 274
<code>IEnumerationT_FLIRFilterDriverStatusEnum</code> (class in <code>PySpin</code>), 252	<code>IEnumerationT_LineModeEnums</code> (class in <code>PySpin</code>), 275
<code>IEnumerationT_GainAutoBalanceEnums</code> (class in <code>PySpin</code>), 256	<code>IEnumerationT_LineSelectorEnums</code> (class in <code>PySpin</code>), 276
<code>IEnumerationT_GainAutoEnums</code> (class in <code>PySpin</code>), 257	<code>IEnumerationT_LineSourceEnums</code> (class in <code>PySpin</code>), 276
<code>IEnumerationT_GainConversionEnums</code> (class in <code>PySpin</code>), 258	<code>IEnumerationT_LogicBlockLUTInputActivationEnums</code> (class in <code>PySpin</code>), 277
<code>IEnumerationT_GainSelectorEnums</code> (class in <code>PySpin</code>), 258	<code>IEnumerationT_LogicBlockLUTInputSelectorEnums</code> (class in <code>PySpin</code>), 278
<code>IEnumerationT_GenICamXMLLocationEnum</code> (class in <code>PySpin</code>), 259	<code>IEnumerationT_LogicBlockLUTInputSourceEnums</code> (class in <code>PySpin</code>), 278
<code>IEnumerationT_GevCCPEnum</code> (class in <code>PySpin</code>), 260	<code>IEnumerationT_LogicBlockLUTSelectorEnums</code> (class in <code>PySpin</code>), 279
<code>IEnumerationT_GevCCPEnums</code> (class in <code>PySpin</code>), 260	<code>IEnumerationT_LogicBlockSelectorEnums</code> (class in <code>PySpin</code>), 261
<code>IEnumerationT_GevCurrentPhysicalLinkConfigurationEnum</code> (class in <code>PySpin</code>), 261	

- PySpin*), 280
- IEnumerationT_LUTSelectorEnums* (class in *PySpin*), 272
- IEnumerationT_MultiRoiConfigurationInvalidReasonEnums* *PySpin*), 298
(class in *PySpin*), 280
- IEnumerationT_MultiRoiSelectorEnums* (class in *PySpin*), 281
- IEnumerationT_PixelColorFilterEnums* (class in *PySpin*), 282
- IEnumerationT_PixelFormatEnums* (class in *PySpin*), 283
- IEnumerationT_PixelFormatInfoSelectorEnums* (class in *PySpin*), 284
- IEnumerationT_PixelSizeEnums* (class in *PySpin*), 284
- IEnumerationT_POEStatusEnum* (class in *PySpin*), 282
- IEnumerationT_RegionDestinationEnums* (class in *PySpin*), 285
- IEnumerationT_RegionModeEnums* (class in *PySpin*), 286
- IEnumerationT_RegionSelectorEnums* (class in *PySpin*), 286
- IEnumerationT_RgbTransformLightSourceEnums* (class in *PySpin*), 287
- IEnumerationT_Scan3dCoordinateReferenceSelectorEnums* (class in *PySpin*), 288
- IEnumerationT_Scan3dCoordinateSelectorEnums* (class in *PySpin*), 288
- IEnumerationT_Scan3dCoordinateSystemEnums* (class in *PySpin*), 289
- IEnumerationT_Scan3dCoordinateSystemReferenceEnums* (class in *PySpin*), 290
- IEnumerationT_Scan3dCoordinateTransformSelectorEnums* (class in *PySpin*), 290
- IEnumerationT_Scan3dDistanceUnitEnums* (class in *PySpin*), 291
- IEnumerationT_Scan3dOutputModeEnums* (class in *PySpin*), 292
- IEnumerationT_SensorDigitizationTapsEnums* (class in *PySpin*), 292
- IEnumerationT_SensorShutterModeEnums* (class in *PySpin*), 293
- IEnumerationT_SensorTapsEnums* (class in *PySpin*), 294
- IEnumerationT_SequencerConfigurationModeEnums* (class in *PySpin*), 294
- IEnumerationT_SequencerConfigurationValidEnums* (class in *PySpin*), 295
- IEnumerationT_SequencerModeEnums* (class in *PySpin*), 296
- IEnumerationT_SequencerSetValidEnums* (class in *PySpin*), 296
- IEnumerationT_SequencerTriggerActivationEnums* (class in *PySpin*), 297
- IEnumerationT_SequencerTriggerSourceEnums* (class in *PySpin*), 298
- IEnumerationT_SerialPortBaudRateEnums* (class in *PySpin*), 299
- IEnumerationT_SerialPortParityEnums* (class in *PySpin*), 299
- IEnumerationT_SerialPortSelectorEnums* (class in *PySpin*), 300
- IEnumerationT_SerialPortSourceEnums* (class in *PySpin*), 300
- IEnumerationT_SerialPortStopBitsEnums* (class in *PySpin*), 301
- IEnumerationT_SoftwareSignalSelectorEnums* (class in *PySpin*), 302
- IEnumerationT_SourceSelectorEnums* (class in *PySpin*), 302
- IEnumerationT_StereoResolutionEnums* (class in *PySpin*), 303
- IEnumerationT_StreamBufferCountModeEnum* (class in *PySpin*), 304
- IEnumerationT_StreamBufferHandlingModeEnum* (class in *PySpin*), 304
- IEnumerationT_StreamModeEnum* (class in *PySpin*), 305
- IEnumerationT_StreamTypeEnum* (class in *PySpin*), 306
- IEnumerationT_TeledyneGigeVisionFilterDriverStatusEnum* (class in *PySpin*), 307
- IEnumerationT_TestPatternEnums* (class in *PySpin*), 308
- IEnumerationT_TestPatternGeneratorSelectorEnums* (class in *PySpin*), 308
- IEnumerationT_TimerSelectorEnums* (class in *PySpin*), 309
- IEnumerationT_TimerStatusEnums* (class in *PySpin*), 310
- IEnumerationT_TimerTriggerActivationEnums* (class in *PySpin*), 310
- IEnumerationT_TimerTriggerSourceEnums* (class in *PySpin*), 311
- IEnumerationT_TLTypeEnum* (class in *PySpin*), 306
- IEnumerationT_TransferComponentSelectorEnums* (class in *PySpin*), 312
- IEnumerationT_TransferControlModeEnums* (class in *PySpin*), 312
- IEnumerationT_TransferOperationModeEnums* (class in *PySpin*), 313
- IEnumerationT_TransferQueueModeEnums* (class in *PySpin*), 314
- IEnumerationT_TransferSelectorEnums* (class in *PySpin*), 314
- IEnumerationT_TransferStatusSelectorEnums* (class in *PySpin*), 315
- IEnumerationT_TransferTriggerActivationEnums*

- (class in *PySpin*), 316
- IEnumerationT_TransferTriggerModeEnums* (class in *PySpin*), 316
- IEnumerationT_TransferTriggerSelectorEnums* (class in *PySpin*), 317
- IEnumerationT_TransferTriggerSourceEnums* (class in *PySpin*), 318
- IEnumerationT_TriggerActivationEnums* (class in *PySpin*), 318
- IEnumerationT_TriggerModeEnums* (class in *PySpin*), 319
- IEnumerationT_TriggerOverlapEnums* (class in *PySpin*), 320
- IEnumerationT_TriggerSelectorEnums* (class in *PySpin*), 320
- IEnumerationT_TriggerSourceEnums* (class in *PySpin*), 321
- IEnumerationT_U3VCurrentSpeedEnums* (class in *PySpin*), 322
- IEnumerationT_UserOutputSelectorEnums* (class in *PySpin*), 322
- IEnumerationT_UserSetDefaultEnums* (class in *PySpin*), 323
- IEnumerationT_UserSetSelectorEnums* (class in *PySpin*), 324
- IEnumerationT_WhiteClipSelectorEnums* (class in *PySpin*), 324
- IEnumReference* (class in *PySpin*), 181
- IFloat* (class in *PySpin*), 325
- IImage* (class in *PySpin*), 326
- IImageEventHandler* (class in *PySpin*), 330
- IImageList* (class in *PySpin*), 330
- IImageListEventHandler* (class in *PySpin*), 331
- IImageProcessor* (class in *PySpin*), 331
- IInteger* (class in *PySpin*), 333
- IInterface* (class in *PySpin*), 66, 333
- IInterfaceArrivalEventHandler* (class in *PySpin*), 334
- IInterfaceEventHandler* (class in *PySpin*), 334
- IInterfaceList* (class in *PySpin*), 335
- IInterfaceRemovalEventHandler* (class in *PySpin*), 335
- ILoggingEventHandler* (class in *PySpin*), 335
- Image* (class in *PySpin*), 46, 345
- Image_Create()* (in module *PySpin*), 373
- Image_GetImageStatusDescription()* (in module *PySpin*), 374
- Image_Load()* (in module *PySpin*), 374
- ImageComponentEnable* (*PySpin.Camera* property), 27, 134
- ImageComponentSelector* (*PySpin.Camera* property), 27, 134
- ImageCompressionBitrate* (*PySpin.Camera* property), 27, 134
- ImageCompressionJPEGFormatOption* (*PySpin.Camera* property), 27, 135
- ImageCompressionMode* (*PySpin.Camera* property), 27, 135
- ImageCompressionQuality* (*PySpin.Camera* property), 27, 135
- ImageCompressionRateOption* (*PySpin.Camera* property), 27, 135
- ImageEventHandler* (class in *PySpin*), 6, 353
- ImageList* (class in *PySpin*), 55, 354
- ImageList_Load()* (in module *PySpin*), 355
- ImageListEventHandler* (class in *PySpin*), 7, 355
- ImagePixel* (class in *PySpin*), 355
- ImageProcessor* (class in *PySpin*), 56, 355
- ImagePtr* (class in *PySpin*), 58, 357
- ImageUtility* (class in *PySpin*), 58, 357
- ImageUtility_CreateNormalized()* (in module *PySpin*), 371
- ImageUtility_CreateScaled()* (in module *PySpin*), 372
- ImageUtilityCCM* (class in *PySpin*), 59, 359
- ImageUtilityCCM_ApplicationToString()* (in module *PySpin*), 359
- ImageUtilityCCM_ColorSpaceToString()* (in module *PySpin*), 359
- ImageUtilityCCM_ColorTemperatureToString()* (in module *PySpin*), 360
- ImageUtilityCCM_CreateColorCorrected()* (in module *PySpin*), 360
- ImageUtilityCCM_EncryptColorCorrectionMatrix()* (in module *PySpin*), 360
- ImageUtilityCCM_SensorToString()* (in module *PySpin*), 360
- ImageUtilityCCM_TypeToString()* (in module *PySpin*), 360
- ImageUtilityHeatmap* (class in *PySpin*), 60, 360
- ImageUtilityHeatmap_CreateHeatmap()* (in module *PySpin*), 361
- ImageUtilityHeatmap_GetHeatmapColorGradient()* (in module *PySpin*), 362
- ImageUtilityHeatmap_GetHeatmapRange()* (in module *PySpin*), 362
- ImageUtilityHeatmap_SetHeatmapColorGradient()* (in module *PySpin*), 362
- ImageUtilityHeatmap_SetHeatmapRange()* (in module *PySpin*), 362
- ImageUtilityPolarization* (class in *PySpin*), 62, 362
- ImageUtilityPolarization_CreateAoIp()* (in module *PySpin*), 365
- ImageUtilityPolarization_CreateDoIp()* (in module *PySpin*), 365
- ImageUtilityPolarization_CreateGlareReduced()* (in module *PySpin*), 365
- ImageUtilityPolarization_CreateStokesS0()* (in

- module PySpin*), 365
- `ImageUtilityPolarization_CreateStokesS1()` (*in module PySpin*), 366
- `ImageUtilityPolarization_CreateStokesS2()` (*in module PySpin*), 366
- `ImageUtilityPolarization_ExtractPolarQuadrant()` (*in module PySpin*), 366
- `ImageUtilityStereo` (*class in PySpin*), 64, 367
- `ImageUtilityStereo_Compute3DPointFromPixel()` (*in module PySpin*), 369
- `ImageUtilityStereo_ComputeDistanceBetweenPoints()` (*in module PySpin*), 369
- `ImageUtilityStereo_ComputeDistanceToPoint()` (*in module PySpin*), 369
- `ImageUtilityStereo_ComputePointCloud()` (*in module PySpin*), 370
- `ImageUtilityStereo_CreateDepthImage()` (*in module PySpin*), 370
- `ImageUtilityStereo_FilterSpeckles()` (*in module PySpin*), 371
- `ImageUtilityStereo_FilterSpecklesFromImage()` (*in module PySpin*), 371
- `ImageUtilityStereo_IsStereoCamera()` (*in module PySpin*), 371
- `ImposeAccessMode()` (*PySpin.CBooleanPtr method*), 86
- `ImposeAccessMode()` (*PySpin.CCategoryPtr method*), 89
- `ImposeAccessMode()` (*PySpin.CCommandPtr method*), 92
- `ImposeAccessMode()` (*PySpin.CEnumEntryPtr method*), 95
- `ImposeAccessMode()` (*PySpin.CEnumerationPtr method*), 98
- `ImposeAccessMode()` (*PySpin.CIntegerPtr method*), 102
- `ImposeAccessMode()` (*PySpin.CNodePtr method*), 107
- `ImposeAccessMode()` (*PySpin.CRegisterPtr method*), 110
- `ImposeAccessMode()` (*PySpin.CStringPtr method*), 114
- `ImposeAccessMode()` (*PySpin.CValuePtr method*), 116
- `ImposeAccessMode()` (*PySpin.INode method*), 337
- `ImposeAccessMode()` (*PySpin.Node method*), 390
- `ImposeMax()` (*PySpin.CIntegerPtr method*), 102
- `ImposeMax()` (*PySpin.FloatNode method*), 170
- `ImposeMax()` (*PySpin.IFloat method*), 326
- `ImposeMax()` (*PySpin.IInteger method*), 333
- `ImposeMax()` (*PySpin.IntegerNode method*), 377
- `ImposeMin()` (*PySpin.CIntegerPtr method*), 102
- `ImposeMin()` (*PySpin.FloatNode method*), 171
- `ImposeMin()` (*PySpin.IFloat method*), 326
- `ImposeMin()` (*PySpin.IInteger method*), 333
- `ImposeMin()` (*PySpin.IntegerNode method*), 377
- `ImposeVisibility()` (*PySpin.CBooleanPtr method*), 87
- `ImposeVisibility()` (*PySpin.CCategoryPtr method*), 89
- `ImposeVisibility()` (*PySpin.CCommandPtr method*), 92
- `ImposeVisibility()` (*PySpin.CEnumEntryPtr method*), 95
- `ImposeVisibility()` (*PySpin.CEnumerationPtr method*), 98
- `ImposeVisibility()` (*PySpin.CIntegerPtr method*), 102
- `ImposeVisibility()` (*PySpin.CNodePtr method*), 107
- `ImposeVisibility()` (*PySpin.CRegisterPtr method*), 110
- `ImposeVisibility()` (*PySpin.CStringPtr method*), 114
- `ImposeVisibility()` (*PySpin.CValuePtr method*), 116
- `ImposeVisibility()` (*PySpin.INode method*), 337
- `ImposeVisibility()` (*PySpin.Node method*), 390
- `IncompatibleDeviceCount` (*PySpin.TransportLayerInterface property*), 80, 413
- `IncompatibleDeviceID` (*PySpin.TransportLayerInterface property*), 80, 413
- `IncompatibleDeviceModelName` (*PySpin.TransportLayerInterface property*), 80, 413
- `IncompatibleDeviceSelector` (*PySpin.TransportLayerInterface property*), 80, 413
- `IncompatibleDeviceVendorName` (*PySpin.TransportLayerInterface property*), 80, 413
- `IncompatibleGevDeviceIPAddress` (*PySpin.TransportLayerInterface property*), 80, 413
- `IncompatibleGevDeviceMACAddress` (*PySpin.TransportLayerInterface property*), 80, 413
- `IncompatibleGevDeviceSubnetMask` (*PySpin.TransportLayerInterface property*), 80, 413
- `indexedColor_8bit` (*PySpin.BMPOption property*), 83
- `InferenceBoundingBox` (*class in PySpin*), 374
- `InferenceBoundingBoxResult` (*class in PySpin*), 374
- `InferenceBoxCircle` (*class in PySpin*), 375
- `InferenceBoxRect` (*class in PySpin*), 375
- `InferenceBoxRotatedRect` (*class in PySpin*), 375
- `Init()` (*PySpin.Camera method*), 27, 135
- `Init()` (*PySpin.CameraBase method*), 38, 145
- `Init()` (*PySpin.ICameraBase method*), 175
- `INode` (*class in PySpin*), 336
- `INodeMap` (*class in PySpin*), 337
- `INodeMapDyn` (*class in PySpin*), 338

- insert() (*PySpin.node_vector* method), 423
 insert() (*PySpin.value_vector* method), 425
 int64_autovector_t (class in *PySpin*), 422
 IntegerNode (class in *PySpin*), 376
 InterfaceArrivalEventHandler (class in *PySpin*), 7, 378
 InterfaceDisplayName
 (*PySpin.TransportLayerInterface* property), 80, 413
 InterfaceDisplayName
 (*PySpin.TransportLayerSystem* property), 416
 InterfaceEventHandler (class in *PySpin*), 7, 378
 InterfaceID (*PySpin.TransportLayerInterface* property), 80, 413
 InterfaceID (*PySpin.TransportLayerSystem* property), 416
 InterfaceList (class in *PySpin*), 67, 378
 InterfacePtr (class in *PySpin*), 68, 379
 InterfaceRemovalEventHandler (class in *PySpin*), 7, 380
 InterfaceSelector (*PySpin.TransportLayerSystem* property), 416
 InterfaceType (*PySpin.TransportLayerInterface* property), 80, 413
 InterfaceUpdateList (*PySpin.TransportLayerSystem* property), 416
 interlaced (*PySpin.PNGOption* property), 397
 IntRegNode (class in *PySpin*), 375
 InvalidateNode() (*PySpin.CBooleanPtr* method), 87
 InvalidateNode() (*PySpin.CCategoryPtr* method), 89
 InvalidateNode() (*PySpin.CCommandPtr* method), 92
 InvalidateNode() (*PySpin.CEnumEntryPtr* method), 95
 InvalidateNode() (*PySpin.CEnumerationPtr* method), 98
 InvalidateNode() (*PySpin.CIntegerPtr* method), 102
 InvalidateNode() (*PySpin.CNodePtr* method), 107
 InvalidateNode() (*PySpin.CRegisterPtr* method), 110
 InvalidateNode() (*PySpin.CStringPtr* method), 114
 InvalidateNode() (*PySpin.CValuePtr* method), 116
 InvalidateNode() (*PySpin.INode* method), 337
 InvalidateNode() (*PySpin.Node* method), 390
 InvalidateNodes() (*PySpin.CNodeMapDynPtr* method), 104
 InvalidateNodes() (*PySpin.CNodeMapPtr* method), 106
 InvalidateNodes() (*PySpin.INodeMap* method), 338
 InvalidateNodes() (*PySpin.NodeMap* method), 395
 invalidDataFlag (*PySpin.StereoCameraParameters* property), 403
 invalidDataValue (*PySpin.StereoCameraParameters* property), 403
 IPersistScript (class in *PySpin*), 339
 IPointCloud (class in *PySpin*), 340
 IReference (class in *PySpin*), 340
 IRegister (class in *PySpin*), 340
 IsAccessModeCacheable() (*PySpin.CBooleanPtr* method), 87
 IsAccessModeCacheable() (*PySpin.CCategoryPtr* method), 89
 IsAccessModeCacheable() (*PySpin.CCommandPtr* method), 92
 IsAccessModeCacheable() (*PySpin.CEnumEntryPtr* method), 95
 IsAccessModeCacheable() (*PySpin.CEnumerationPtr* method), 98
 IsAccessModeCacheable() (*PySpin.CIntegerPtr* method), 102
 IsAccessModeCacheable() (*PySpin.CNodePtr* method), 107
 IsAccessModeCacheable() (*PySpin.CRegisterPtr* method), 110
 IsAccessModeCacheable() (*PySpin.CStringPtr* method), 114
 IsAccessModeCacheable() (*PySpin.CValuePtr* method), 116
 IsAccessModeCacheable() (*PySpin.INode* method), 337
 IsAccessModeCacheable() (*PySpin.Node* method), 390
 IsAvailable() (in module *PySpin*), 380
 IsCacheable() (*PySpin.CBooleanPtr* method), 87
 IsCacheable() (*PySpin.CCategoryPtr* method), 90
 IsCacheable() (*PySpin.CCommandPtr* method), 92
 IsCacheable() (*PySpin.CEnumEntryPtr* method), 95
 IsCacheable() (*PySpin.CEnumerationPtr* method), 98
 IsCacheable() (*PySpin.CIntegerPtr* method), 102
 IsCacheable() (*PySpin.CNodePtr* method), 107
 IsCacheable() (*PySpin.CRegisterPtr* method), 110
 IsCacheable() (*PySpin.CStringPtr* method), 114
 IsCacheable() (*PySpin.CValuePtr* method), 117
 IsCacheable() (*PySpin.INode* method), 337
 IsCacheable() (*PySpin.Node* method), 391
 IsCacheable() (in module *PySpin*), 381
 IsCameraInUse() (*PySpin.IInterface* method), 66, 334
 IsCompressed() (*PySpin.IImage* method), 327
 IsCompressed() (*PySpin.Image* method), 51, 350
 IsDeprecated() (*PySpin.CBooleanPtr* method), 87
 IsDeprecated() (*PySpin.CCategoryPtr* method), 90
 IsDeprecated() (*PySpin.CCommandPtr* method), 92
 IsDeprecated() (*PySpin.CEnumEntryPtr* method), 95
 IsDeprecated() (*PySpin.CEnumerationPtr* method), 98
 IsDeprecated() (*PySpin.CIntegerPtr* method), 102
 IsDeprecated() (*PySpin.CNodePtr* method), 108
 IsDeprecated() (*PySpin.CRegisterPtr* method), 110
 IsDeprecated() (*PySpin.CStringPtr* method), 114
 IsDeprecated() (*PySpin.CValuePtr* method), 117

- IsDeprecated() (*PySpin.INode* method), 337
- IsDeprecated() (*PySpin.Node* method), 391
- IsDone() (*PySpin.CCommandPtr* method), 92
- IsDone() (*PySpin.CommandNode* method), 154
- IsDone() (*PySpin.ICommand* method), 179
- ISelector (class in *PySpin*), 341
- ISelectorDigit (class in *PySpin*), 342
- IsEmpty() (*PySpin.CSelectorSet* method), 112
- IsFeature() (*PySpin.CBooleanPtr* method), 87
- IsFeature() (*PySpin.CCategoryPtr* method), 90
- IsFeature() (*PySpin.CCommandPtr* method), 92
- IsFeature() (*PySpin.CEnumEntryPtr* method), 95
- IsFeature() (*PySpin.CEnumerationPtr* method), 98
- IsFeature() (*PySpin.CIntegerPtr* method), 103
- IsFeature() (*PySpin.CNodePtr* method), 108
- IsFeature() (*PySpin.CRegisterPtr* method), 110
- IsFeature() (*PySpin.CStringPtr* method), 114
- IsFeature() (*PySpin.CValuePtr* method), 117
- IsFeature() (*PySpin.INode* method), 337
- IsFeature() (*PySpin.Node* method), 391
- IsImplemented() (in module *PySpin*), 381
- IsIncomplete() (*PySpin.IImage* method), 327
- IsIncomplete() (*PySpin.Image* method), 51, 350
- IsInitialized() (*PySpin.CameraBase* method), 38, 145
- IsInitialized() (*PySpin.ICameraBase* method), 175
- IsInUse() (*PySpin.IImage* method), 327
- IsInUse() (*PySpin.Image* method), 51, 350
- IsInUse() (*PySpin.ISystem* method), 343
- IsInUse() (*PySpin.System* method), 72, 407
- IspEnable (*PySpin.Camera* property), 27, 135
- IsReadable() (in module *PySpin*), 382
- IsSelector() (*PySpin.CBooleanPtr* method), 87
- IsSelector() (*PySpin.CCategoryPtr* method), 90
- IsSelector() (*PySpin.CCommandPtr* method), 92
- IsSelector() (*PySpin.CEnumEntryPtr* method), 95
- IsSelector() (*PySpin.CEnumerationPtr* method), 98
- IsSelector() (*PySpin.CIntegerPtr* method), 103
- IsSelector() (*PySpin.CNodePtr* method), 108
- IsSelector() (*PySpin.CRegisterPtr* method), 110
- IsSelector() (*PySpin.CSelectorPtr* method), 111
- IsSelector() (*PySpin.CStringPtr* method), 114
- IsSelector() (*PySpin.CValuePtr* method), 117
- IsSelector() (*PySpin.ISelector* method), 341
- IsSelector() (*PySpin.Node* method), 391
- IsSelfClearing() (*PySpin.CEnumEntryPtr* method), 95
- IsSelfClearing() (*PySpin.EnumEntryNode* method), 166
- IsSelfClearing() (*PySpin.IEnumEntry* method), 181
- IsStereoCamera() (*PySpin.ImageUtilityStereo* static method), 66, 369
- IsStreamable() (*PySpin.CBooleanPtr* method), 87
- IsStreamable() (*PySpin.CCategoryPtr* method), 90
- IsStreamable() (*PySpin.CCommandPtr* method), 92
- IsStreamable() (*PySpin.CEnumEntryPtr* method), 95
- IsStreamable() (*PySpin.CEnumerationPtr* method), 98
- IsStreamable() (*PySpin.CIntegerPtr* method), 103
- IsStreamable() (*PySpin.CNodePtr* method), 108
- IsStreamable() (*PySpin.CRegisterPtr* method), 110
- IsStreamable() (*PySpin.CStringPtr* method), 114
- IsStreamable() (*PySpin.CValuePtr* method), 117
- IsStreamable() (*PySpin.INode* method), 337
- IsStreamable() (*PySpin.Node* method), 391
- IsStreaming() (*PySpin.CameraBase* method), 38, 145
- IsStreaming() (*PySpin.ICameraBase* method), 175
- IString (class in *PySpin*), 342
- IsValid() (*PySpin.CameraBase* method), 38, 145
- IsValid() (*PySpin.CBasePtr* method), 10, 85
- IsValid() (*PySpin.CBooleanPtr* method), 87
- IsValid() (*PySpin.CCategoryPtr* method), 90
- IsValid() (*PySpin.CCommandPtr* method), 92
- IsValid() (*PySpin.CDeviceInfoPtr* method), 93
- IsValid() (*PySpin.CEnumEntryPtr* method), 95
- IsValid() (*PySpin.CEnumerationPtr* method), 98
- IsValid() (*PySpin.CIntegerPtr* method), 103
- IsValid() (*PySpin.CNodeMapDynPtr* method), 104
- IsValid() (*PySpin.CNodeMapPtr* method), 106
- IsValid() (*PySpin.CNodePtr* method), 108
- IsValid() (*PySpin.CRegisterPtr* method), 110
- IsValid() (*PySpin.CSelectorPtr* method), 111
- IsValid() (*PySpin.CStringPtr* method), 114
- IsValid() (*PySpin.CValuePtr* method), 117
- IsValid() (*PySpin.ICameraBase* method), 175
- IsValid() (*PySpin.IInterface* method), 66, 334
- IsValueCacheValid() (*PySpin.CBooleanPtr* method), 87
- IsValueCacheValid() (*PySpin.CCategoryPtr* method), 90
- IsValueCacheValid() (*PySpin.CCommandPtr* method), 92
- IsValueCacheValid() (*PySpin.CEnumEntryPtr* method), 95
- IsValueCacheValid() (*PySpin.CEnumerationPtr* method), 98
- IsValueCacheValid() (*PySpin.CIntegerPtr* method), 103
- IsValueCacheValid() (*PySpin.CRegisterPtr* method), 110
- IsValueCacheValid() (*PySpin.CStringPtr* method), 114
- IsValueCacheValid() (*PySpin.CValuePtr* method), 117
- IsValueCacheValid() (*PySpin.IValue* method), 344
- IsValueCacheValid() (*PySpin.ValueNode* method), 417
- IsVisible() (in module *PySpin*), 383
- IsWritable() (in module *PySpin*), 383

ISystem (class in PySpin), 342

ISystemEventHandler (class in PySpin), 344

IValue (class in PySpin), 344

J

JPEGOption (class in PySpin), 385

JPG2Option (class in PySpin), 385

L

LargePenalty (PySpin.Camera property), 28, 135

length() (PySpin.gcstring method), 421

LensShadingCoefficientActiveSet
(PySpin.Camera property), 28, 135

LensShadingCorrectionCalibration
(PySpin.Camera property), 28, 135

LensShadingCorrectionCalibrationGainLimit
(PySpin.Camera property), 28, 135

LensShadingCorrectionCalibrationSetup
(PySpin.Camera property), 28, 135

LensShadingCorrectionCalibrationStatus
(PySpin.Camera property), 28, 135

LensShadingCorrectionMode (PySpin.Camera prop-
erty), 28, 135

LensShadingCorrectionStepSize (PySpin.Camera
property), 28, 135

LensShadingCorrectionVersion (PySpin.Camera
property), 28, 135

LibraryVersion (class in PySpin), 385

LineFilterWidth (PySpin.Camera property), 28, 135

LineFormat (PySpin.Camera property), 28, 135

LineInputFilterSelector (PySpin.Camera prop-
erty), 28, 135

LineInverter (PySpin.Camera property), 28, 135

LineMode (PySpin.Camera property), 28, 135

LinePitch (PySpin.Camera property), 28, 135

LineSelector (PySpin.Camera property), 28, 135

LineSource (PySpin.Camera property), 28, 135

LineStatus (PySpin.Camera property), 28, 135

LineStatusAll (PySpin.Camera property), 28, 135

LinkErrorCount (PySpin.Camera property), 28, 136

LinkRecoveryCount (PySpin.Camera property), 28,
136

LinkUptime (PySpin.Camera property), 28, 136

Load() (PySpin.Image static method), 52, 350

Load() (PySpin.ImageList static method), 55, 354

LoadFromBag() (PySpin.CFeatureBag method), 99

LoadPointCloudFromPly() (PySpin.IPointCloud
method), 340

LoadPointCloudFromPly() (PySpin.PointCloud
method), 69, 398

LoadXMLFromFile() (PySpin.CNodeMapDynPtr
method), 104

LoadXMLFromFile() (PySpin.INodeMapDyn method),
338

LoadXMLFromFile() (PySpin.NodeMap method), 395

LoadXMLFromFileInject()
(PySpin.CNodeMapDynPtr method), 104

LoadXMLFromFileInject() (PySpin.INodeMapDyn
method), 338

LoadXMLFromFileInject() (PySpin.NodeMap
method), 395

LoadXMLFromString() (PySpin.CNodeMapDynPtr
method), 104

LoadXMLFromString() (PySpin.INodeMapDyn
method), 338

LoadXMLFromString() (PySpin.NodeMap method), 396

LoadXMLFromStringInject()
(PySpin.CNodeMapDynPtr method), 104

LoadXMLFromStringInject() (PySpin.INodeMapDyn
method), 338

LoadXMLFromStringInject() (PySpin.NodeMap
method), 396

LoadXMLFromZIPData() (PySpin.CNodeMapDynPtr
method), 104

LoadXMLFromZIPData() (PySpin.INodeMapDyn
method), 339

LoadXMLFromZIPData() (PySpin.NodeMap method),
396

LoadXMLFromZIPFile() (PySpin.CNodeMapDynPtr
method), 104

LoadXMLFromZIPFile() (PySpin.INodeMapDyn
method), 339

LoadXMLFromZIPFile() (PySpin.NodeMap method),
396

LoggingEventData (class in PySpin), 385

LoggingEventDataPtr (class in PySpin), 8, 386

LoggingEventHandler (class in PySpin), 8, 386

LogicBlockLUTInputActivation (PySpin.Camera
property), 28, 136

LogicBlockLUTInputSelector (PySpin.Camera prop-
erty), 28, 136

LogicBlockLUTInputSource (PySpin.Camera prop-
erty), 28, 136

LogicBlockLUTOutputValue (PySpin.Camera prop-
erty), 28, 136

LogicBlockLUTOutputValueAll (PySpin.Camera
property), 28, 136

LogicBlockLUTRowIndex (PySpin.Camera property),
28, 136

LogicBlockLUTSelector (PySpin.Camera property),
29, 136

LogicBlockSelector (PySpin.Camera property), 29,
136

LUTEnable (PySpin.Camera property), 27, 135

LUTIndex (PySpin.Camera property), 27, 135

LUTSelector (PySpin.Camera property), 28, 135

LUTValue (PySpin.Camera property), 28, 135

LUTValueAll (PySpin.Camera property), 28, 135

M

major (*PySpin.LibraryVersion* property), 385
 Major (*PySpin.Version_t* property), 418
 max_size() (*PySpin.gcstring* method), 421
 max_size() (*PySpin.node_vector* method), 423
 max_size() (*PySpin.value_vector* method), 425
 MaxDatarateThreshold (*PySpin.Camera* property), 29, 136
 maxDepthThresholdInMeter (*PySpin.ImageUtilityStereo* property), 66, 369
 maxDepthThresholdInMm (*PySpin.ImageUtilityStereo* property), 66, 369
 MaxDeviceResetTime (*PySpin.Camera* property), 29, 136
 MergeXMLFiles() (*PySpin.CNodeMapDynPtr* method), 105
 MergeXMLFiles() (*PySpin.INodeMapDyn* method), 339
 message (*PySpin.SpinnakerException* attribute), 69
 minor (*PySpin.LibraryVersion* property), 385
 Minor (*PySpin.Version_t* property), 418
 MJPGOption (class in *PySpin*), 387
 module
 PySpin, 83
 MultiRoiConfigurationInvalidReason (*PySpin.Camera* property), 29, 136
 MultiRoiConfigurationInvalidReasonAll (*PySpin.Camera* property), 29, 136
 MultiRoiEnable (*PySpin.Camera* property), 29, 136
 MultiRoiFeatureEnable (*PySpin.Camera* property), 29, 136
 MultiRoiHeight (*PySpin.Camera* property), 29, 136
 MultiRoiOffsetX (*PySpin.Camera* property), 29, 136
 MultiRoiOffsetY (*PySpin.Camera* property), 29, 136
 MultiRoiSelector (*PySpin.Camera* property), 29, 136
 MultiRoiWidth (*PySpin.Camera* property), 29, 136
 MultiRoiWindows (*PySpin.Camera* property), 29, 136

N

Node (class in *PySpin*), 387
 node_vector (class in *PySpin*), 422
 NodeCallback (class in *PySpin*), 392
 NodeMap (class in *PySpin*), 392
 NodeMap_ClearXMLCache() (in module *PySpin*), 397
 npos (*PySpin.gcstring* attribute), 421
 num_pixel_values (*PySpin.ChannelStatistics* property), 42, 150
 NumDirections (*PySpin.Camera* property), 29, 136

O

OffsetX (*PySpin.Camera* property), 29, 136
 OffsetY (*PySpin.Camera* property), 29, 136
 OnDeviceArrival() (*PySpin.DeviceArrivalEventHandler* method), 5, 155

OnDeviceArrival() (*PySpin.IDeviceArrivalEventHandler* method), 180
 OnDeviceArrival() (*PySpin.IInterfaceEventHandler* method), 334
 OnDeviceArrival() (*PySpin.InterfaceEventHandler* method), 7, 378
 OnDeviceEvent() (*PySpin.DeviceEventHandler* method), 6, 155
 OnDeviceEvent() (*PySpin.IDeviceEventHandler* method), 180
 OnDeviceRemoval() (*PySpin.DeviceRemovalEventHandler* method), 6, 156
 OnDeviceRemoval() (*PySpin.IDeviceRemovalEventHandler* method), 181
 OnDeviceRemoval() (*PySpin.IInterfaceEventHandler* method), 334
 OnDeviceRemoval() (*PySpin.InterfaceEventHandler* method), 7, 378
 OnImageEvent() (*PySpin.ImageEventHandler* method), 6, 353
 OnImageListEvent() (*PySpin.ImageListEventHandler* method), 7, 355
 OnInterfaceArrival() (*PySpin.IInterfaceArrivalEventHandler* method), 334
 OnInterfaceArrival() (*PySpin.InterfaceArrivalEventHandler* method), 7, 378
 OnInterfaceArrival() (*PySpin.ISystemEventHandler* method), 344
 OnInterfaceArrival() (*PySpin.SystemEventHandler* method), 8, 409
 OnInterfaceRemoval() (*PySpin.IInterfaceRemovalEventHandler* method), 335
 OnInterfaceRemoval() (*PySpin.InterfaceRemovalEventHandler* method), 7, 380
 OnInterfaceRemoval() (*PySpin.ISystemEventHandler* method), 344
 OnInterfaceRemoval() (*PySpin.SystemEventHandler* method), 8, 409
 OnLogEvent() (*PySpin.ILoggingEventHandler* method), 335
 OnLogEvent() (*PySpin.LoggingEventHandler* method), 8, 387
 Open() (*PySpin.SpinVideo* method), 70, 402

P

PacketResendRequestCount (*PySpin.Camera* property), 29, 136
 PacketResendRequestsDroppedCount (*PySpin.Camera* property), 29, 136
 PauseFrameCount (*PySpin.Camera* property), 29, 136

PayloadSize (*PySpin.Camera* property), 29, 136
PersistFeature() (*PySpin.CFeatureBag* method), 99
PersistFeature() (*PySpin.IPersistScript* method), 339
PGMOption (class in *PySpin*), 397
pixel (*PySpin.Stereo3DPoint* property), 403
pixel_value_max (*PySpin.ChannelStatistics* property), 42, 150
pixel_value_mean (*PySpin.ChannelStatistics* property), 42, 150
pixel_value_min (*PySpin.ChannelStatistics* property), 42, 150
PixelColorFilter (*PySpin.Camera* property), 29, 136
PixelDynamicRangeMax (*PySpin.Camera* property), 29, 137
PixelDynamicRangeMin (*PySpin.Camera* property), 29, 137
PixelFormat (*PySpin.Camera* property), 29, 137
PixelFormatInfoID (*PySpin.Camera* property), 29, 137
PixelFormatInfoSelector (*PySpin.Camera* property), 29, 137
PixelSize (*PySpin.Camera* property), 29, 137
PNGOption (class in *PySpin*), 397
POEStatus (*PySpin.TransportLayerInterface* property), 80, 414
PointCloud (class in *PySpin*), 68, 397
PointCloudParameters (class in *PySpin*), 398
Poll() (*PySpin.CNodeMapDynPtr* method), 105
Poll() (*PySpin.CNodeMapPtr* method), 106
Poll() (*PySpin.INodeMap* method), 338
Poll() (*PySpin.NodeMap* method), 396
pop_back() (*PySpin.node_vector* method), 423
pop_back() (*PySpin.value_vector* method), 425
PowerSupplyCurrent (*PySpin.Camera* property), 29, 137
PowerSupplyVoltage (*PySpin.Camera* property), 29, 137
PPMOption (class in *PySpin*), 397
PreprocessXMLFromFile() (*PySpin.CNodeMapDynPtr* method), 105
PreprocessXMLFromFile() (*PySpin.INodeMapDyn* method), 339
PreprocessXMLFromZIPFile() (*PySpin.CNodeMapDynPtr* method), 105
PreprocessXMLFromZIPFile() (*PySpin.INodeMapDyn* method), 339
principalPointU (*PySpin.StereoCameraParameters* property), 403
principalPointV (*PySpin.StereoCameraParameters* property), 404
PrintPoints() (*PySpin.IPointCloud* method), 340
PrintPoints() (*PySpin.PointCloud* method), 69, 398
progressive (*PySpin.JPEGOption* property), 385
push_back() (*PySpin.node_vector* method), 423

push_back() (*PySpin.value_vector* method), 425
PySpin
 module, 83

Q

quality (*PySpin.JPEGOption* property), 385
quality (*PySpin.JPG2Option* property), 385
quality (*PySpin.MJPGOption* property), 387

R

r (*PySpin.Stereo3DPoint* property), 403
radius (*PySpin.InferenceBoxCircle* property), 375
range_max (*PySpin.ChannelStatistics* property), 42, 150
range_min (*PySpin.ChannelStatistics* property), 42, 150
ReadPort() (*PySpin.ICameraBase* method), 175
rect (*PySpin.InferenceBoundingBox* property), 374
RegionDestination (*PySpin.Camera* property), 29, 137
RegionMode (*PySpin.Camera* property), 30, 137
RegionSelector (*PySpin.Camera* property), 30, 137
RegisterCallback() (*PySpin.CBooleanPtr* method), 87
RegisterCallback() (*PySpin.CCategoryPtr* method), 90
RegisterCallback() (*PySpin.CCommandPtr* method), 92
RegisterCallback() (*PySpin.CEnumEntryPtr* method), 95
RegisterCallback() (*PySpin.CEnumerationPtr* method), 98
RegisterCallback() (*PySpin.CIntegerPtr* method), 103
RegisterCallback() (*PySpin.CNodePtr* method), 108
RegisterCallback() (*PySpin.CRegisterPtr* method), 110
RegisterCallback() (*PySpin.CStringPtr* method), 114
RegisterCallback() (*PySpin.CValuePtr* method), 117
RegisterCallback() (*PySpin.INode* method), 337
RegisterCallback() (*PySpin.Node* method), 391
RegisterEventHandler() (*PySpin.CameraBase* method), 38, 146
RegisterEventHandler() (*PySpin.ICameraBase* method), 175
RegisterEventHandler() (*PySpin.IInterface* method), 66, 334
RegisterEventHandler() (*PySpin.ISystem* method), 343
RegisterEventHandler() (*PySpin.System* method), 72, 407
RegisterLoggingEventHandler() (*PySpin.ISystem* method), 343
RegisterLoggingEventHandler() (*PySpin.System* method), 72, 407
RegisterNode (class in *PySpin*), 399

- RegisterNodeCallback() (in module *PySpin*), 400
 Release() (*PySpin.IImage* method), 327
 Release() (*PySpin.IImageList* method), 331
 Release() (*PySpin.Image* method), 52, 350
 Release() (*PySpin.ImageList* method), 56, 354
 ReleaseInstance() (*PySpin.ISystem* method), 343
 ReleaseInstance() (*PySpin.System* method), 73, 407
 Remove() (*PySpin.CameraList* method), 40, 148
 Remove() (*PySpin.ICameraList* method), 177
 Remove() (*PySpin.IInterfaceList* method), 335
 Remove() (*PySpin.InterfaceList* method), 68, 379
 RemoveByDeviceID() (*PySpin.CameraList* method), 41, 148
 RemoveByDeviceID() (*PySpin.ICameraList* method), 177
 RemoveByIndex() (*PySpin.CameraList* method), 41, 148
 RemoveByIndex() (*PySpin.ICameraList* method), 177
 RemoveByIndex() (*PySpin.IImageList* method), 331
 RemoveByIndex() (*PySpin.ImageList* method), 56, 354
 RemoveByPayloadType() (*PySpin.IImageList* method), 331
 RemoveByPayloadType() (*PySpin.ImageList* method), 56, 354
 RemoveByPixelFormat() (*PySpin.IImageList* method), 331
 RemoveByPixelFormat() (*PySpin.ImageList* method), 56, 354
 RemoveBySerial() (*PySpin.CameraList* method), 41, 148
 RemoveBySerial() (*PySpin.ICameraList* method), 177
 RemoveByStreamIndex() (*PySpin.IImageList* method), 331
 RemoveByStreamIndex() (*PySpin.ImageList* method), 56, 355
 ReplaceEnvironmentVariables() (in module *PySpin*), 400
 reserve() (*PySpin.node_vector* method), 424
 reserve() (*PySpin.value_vector* method), 425
 reserved (*PySpin.AVIOption* property), 83
 reserved (*PySpin.BMPOption* property), 83
 reserved (*PySpin.H264Option* property), 173
 reserved (*PySpin.JPEGOption* property), 385
 reserved (*PySpin.JPG2Option* property), 385
 reserved (*PySpin.MJPGOption* property), 387
 reserved (*PySpin.PGMOption* property), 397
 reserved (*PySpin.PNGOption* property), 397
 reserved (*PySpin.PPMOption* property), 397
 reserved (*PySpin.SIOption* property), 400
 reserved (*PySpin.TIFFOption* property), 410
 ResetImage() (*PySpin.IImage* method), 327
 ResetImage() (*PySpin.Image* method), 52, 350
 resize() (*PySpin.gcstring* method), 421
 resize() (*PySpin.node_vector* method), 424
 resize() (*PySpin.value_vector* method), 425
 Restore() (*PySpin.CSelectorSet* method), 112
 Restore() (*PySpin.ISelectorDigit* method), 342
 result (*PySpin.DeviceEventInferenceData* property), 156
 ReverseX (*PySpin.Camera* property), 30, 137
 ReverseY (*PySpin.Camera* property), 30, 137
 RgbTransformLightSource (*PySpin.Camera* property), 30, 137
 ROIImageBottom (*PySpin.PointCloudParameters* property), 398
 ROIImageLeft (*PySpin.PointCloudParameters* property), 398
 ROIImageRight (*PySpin.PointCloudParameters* property), 398
 ROIImageTop (*PySpin.PointCloudParameters* property), 398
 ROIWorldCoordinatesXMax (*PySpin.PointCloudParameters* property), 398
 ROIWorldCoordinatesXMin (*PySpin.PointCloudParameters* property), 398
 ROIWorldCoordinatesYMax (*PySpin.PointCloudParameters* property), 398
 ROIWorldCoordinatesYMin (*PySpin.PointCloudParameters* property), 398
 ROIWorldCoordinatesZMax (*PySpin.PointCloudParameters* property), 398
 ROIWorldCoordinatesZMin (*PySpin.PointCloudParameters* property), 398
 rotatedRect (*PySpin.InferenceBoundingBox* property), 374
 rotationAngle (*PySpin.InferenceBoxRotatedRect* property), 375
- ## S
- Saturation (*PySpin.Camera* property), 30, 137
 SaturationEnable (*PySpin.Camera* property), 30, 137
 Save() (*PySpin.IImage* method), 329
 Save() (*PySpin.IImageList* method), 331
 Save() (*PySpin.Image* method), 53, 352
 Save() (*PySpin.ImageList* method), 56, 355
 SavePointCloudAsPly() (*PySpin.IPointCloud* method), 340
 SavePointCloudAsPly() (*PySpin.PointCloud* method), 69, 398
 Scan3dAxisMax (*PySpin.Camera* property), 30, 137
 Scan3dAxisMin (*PySpin.Camera* property), 30, 137
 Scan3dBaseline (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateOffset` (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateReferenceSelector` (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateReferenceValue` (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateScale` (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateSelector` (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateSystem` (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateSystemReference` (*PySpin.Camera* property), 30, 137

`Scan3dCoordinateTransformSelector` (*PySpin.Camera* property), 30, 137

`Scan3dDistanceUnit` (*PySpin.Camera* property), 30, 137

`Scan3dFocallLength` (*PySpin.Camera* property), 30, 137

`Scan3dInvalidDataFlag` (*PySpin.Camera* property), 30, 137

`Scan3dInvalidDataValue` (*PySpin.Camera* property), 30, 137

`Scan3dOutputMode` (*PySpin.Camera* property), 30, 138

`Scan3dPrincipalPointU` (*PySpin.Camera* property), 30, 138

`Scan3dPrincipalPointV` (*PySpin.Camera* property), 30, 138

`Scan3dTransformValue` (*PySpin.Camera* property), 30, 138

`SendActionCommand()` (*PySpin.IInterface* method), 67, 334

`SendActionCommand()` (*PySpin.ISystem* method), 343

`SendActionCommand()` (*PySpin.System* method), 73, 407

`Sensor` (*PySpin.CCMSettings* property), 88

`SensorDescription` (*PySpin.Camera* property), 30, 138

`SensorDigitizationTaps` (*PySpin.Camera* property), 30, 138

`SensorHeight` (*PySpin.Camera* property), 30, 138

`SensorShutterMode` (*PySpin.Camera* property), 30, 138

`SensorTaps` (*PySpin.Camera* property), 30, 138

`SensorToString()` (*PySpin.ImageUtilityCCM* static method), 60, 359

`SensorWidth` (*PySpin.Camera* property), 31, 138

`SequencerConfigurationMode` (*PySpin.Camera* property), 31, 138

`SequencerConfigurationReset` (*PySpin.Camera* property), 31, 138

`SequencerConfigurationValid` (*PySpin.Camera* property), 31, 138

`SequencerFeatureEnable` (*PySpin.Camera* property), 31, 138

`SequencerMode` (*PySpin.Camera* property), 31, 138

`SequencerPathSelector` (*PySpin.Camera* property), 31, 138

`SequencerSetActive` (*PySpin.Camera* property), 31, 138

`SequencerSetLoad` (*PySpin.Camera* property), 31, 138

`SequencerSetNext` (*PySpin.Camera* property), 31, 138

`SequencerSetSave` (*PySpin.Camera* property), 31, 138

`SequencerSetSelector` (*PySpin.Camera* property), 31, 138

`SequencerSetStart` (*PySpin.Camera* property), 31, 138

`SequencerSetValid` (*PySpin.Camera* property), 31, 138

`SequencerTriggerActivation` (*PySpin.Camera* property), 31, 138

`SequencerTriggerSource` (*PySpin.Camera* property), 31, 138

`SerialPortBaudRate` (*PySpin.Camera* property), 31, 138

`SerialPortDataBits` (*PySpin.Camera* property), 31, 138

`SerialPortParity` (*PySpin.Camera* property), 31, 138

`SerialPortSelector` (*PySpin.Camera* property), 31, 138

`SerialPortSource` (*PySpin.Camera* property), 31, 138

`SerialPortStopBits` (*PySpin.Camera* property), 31, 138

`SerialReceiveFramingErrorCount` (*PySpin.Camera* property), 31, 139

`SerialReceiveParityErrorCount` (*PySpin.Camera* property), 31, 139

`SerialReceiveQueueClear` (*PySpin.Camera* property), 31, 139

`SerialReceiveQueueCurrentCharacterCount` (*PySpin.Camera* property), 31, 139

`SerialReceiveQueueMaxCharacterCount` (*PySpin.Camera* property), 31, 139

`SerialTransmitQueueCurrentCharacterCount` (*PySpin.Camera* property), 31, 139

`SerialTransmitQueueMaxCharacterCount` (*PySpin.Camera* property), 31, 139

`Set()` (*PySpin.CRegisterPtr* method), 110

`Set()` (*PySpin.IRegister* method), 341

`Set()` (*PySpin.RegisterNode* method), 399

`SetBufferOwnership()` (*PySpin.CameraBase* method), 39, 146

`SetBufferOwnership()` (*PySpin.ICameraBase* method), 176

`SetChunks()` (*PySpin.ChunkData* method), 46, 153

`SetChunks()` (*PySpin.IChunkData* method), 179

`SetColorProcessing()` (*PySpin.IImageProcessor*

- method*), 332
- SetColorProcessing() (*PySpin.ImageProcessor method*), 57, 356
- SetEnumReference() (*PySpin.IEnumReference method*), 181
- SetEventType() (*PySpin.EventHandler method*), 6, 169
- SetFirst() (*PySpin.CSelectorSet method*), 112
- SetFirst() (*PySpin.ISelectorDigit method*), 342
- SetGenICamCacheFolder() (*in module PySpin*), 401
- SetGenICamCLProtocolFolder() (*in module PySpin*), 401
- SetGenICamLogConfig() (*in module PySpin*), 401
- SetHeatmapColorGradient() (*PySpin.ImageUtilityHeatmap static method*), 61, 361
- SetHeatmapRange() (*PySpin.ImageUtilityHeatmap static method*), 61, 361
- SetInfo() (*PySpin.CFeatureBag method*), 99
- SetInfo() (*PySpin.IPersistScript method*), 340
- SetIntValue() (*PySpin.CEnumerationPtr method*), 98
- SetIntValue() (*PySpin.EnumNode method*), 168
- SetIntValue() (*PySpin.IEnumeration method*), 182
- SetLoggingEventPriorityLevel() (*PySpin.ISystem method*), 343
- SetLoggingEventPriorityLevel() (*PySpin.System method*), 73, 408
- SetMaximumFileSize() (*PySpin.SpinVideo method*), 70, 403
- SetMessageCallback() (*in module PySpin*), 401
- SetNext() (*PySpin.CSelectorSet method*), 112
- SetNext() (*PySpin.ISelectorDigit method*), 342
- SetNodeHandle() (*PySpin.Node method*), 391
- SetNodeMap() (*PySpin.Node method*), 391
- SetNumDecompressionThreads() (*PySpin.IImageProcessor method*), 332
- SetNumDecompressionThreads() (*PySpin.ImageProcessor method*), 57, 357
- SetNumEnums() (*PySpin.IEnumReference method*), 181
- SetProgressCallback() (*in module PySpin*), 401
- SetReference() (*PySpin.BooleanNode method*), 84
- SetReference() (*PySpin.CategoryNode method*), 149
- SetReference() (*PySpin.CBooleanPtr method*), 87
- SetReference() (*PySpin.CCategoryPtr method*), 90
- SetReference() (*PySpin.CCommandPtr method*), 92
- SetReference() (*PySpin.CEnumEntryPtr method*), 96
- SetReference() (*PySpin.CEnumerationPtr method*), 98
- SetReference() (*PySpin.CIntegerPtr method*), 103
- SetReference() (*PySpin.CNodePtr method*), 108
- SetReference() (*PySpin.CommandNode method*), 155
- SetReference() (*PySpin.CRegisterPtr method*), 111
- SetReference() (*PySpin.CStringPtr method*), 114
- SetReference() (*PySpin.CValuePtr method*), 117
- SetReference() (*PySpin.EnumEntryNode method*), 166
- SetReference() (*PySpin.EnumNode method*), 168
- SetReference() (*PySpin.FloatNode method*), 171
- SetReference() (*PySpin.FloatRegNode method*), 171
- SetReference() (*PySpin.IntegerNode method*), 377
- SetReference() (*PySpin.IntRegNode method*), 376
- SetReference() (*PySpin.IReference method*), 340
- SetReference() (*PySpin.Node method*), 392
- SetReference() (*PySpin.RegisterNode method*), 400
- SetReference() (*PySpin.StringNode method*), 404
- SetReference() (*PySpin.StringRegNode method*), 405
- SetReference() (*PySpin.ValueNode method*), 417
- SetUserBuffers() (*PySpin.CameraBase method*), 39, 146
- SetUserBuffers() (*PySpin.ICameraBase method*), 176
- SetValue() (*PySpin.BooleanNode method*), 84
- SetValue() (*PySpin.CBooleanPtr method*), 87
- SetValue() (*PySpin.CIntegerPtr method*), 103
- SetValue() (*PySpin.CStringPtr method*), 114
- SetValue() (*PySpin.FloatNode method*), 171
- SetValue() (*PySpin.IBoolean method*), 174
- SetValue() (*PySpin.IEnumerationT_AcquisitionModeEnums method*), 183
- SetValue() (*PySpin.IEnumerationT_AcquisitionStatusSelectorEnums method*), 183
- SetValue() (*PySpin.IEnumerationT_ActionSelectorEnums method*), 184
- SetValue() (*PySpin.IEnumerationT_ActionUnconditionalModeEnums method*), 185
- SetValue() (*PySpin.IEnumerationT_AdcBitDepthEnums method*), 185
- SetValue() (*PySpin.IEnumerationT_AutoAlgorithmSelectorEnums method*), 186
- SetValue() (*PySpin.IEnumerationT_AutoExposureControlPriorityEnums method*), 187
- SetValue() (*PySpin.IEnumerationT_AutoExposureLightingModeEnums method*), 187
- SetValue() (*PySpin.IEnumerationT_AutoExposureMeteringModeEnums method*), 188
- SetValue() (*PySpin.IEnumerationT_AutoExposureTargetGreyValueAutoE method*), 189
- SetValue() (*PySpin.IEnumerationT_BalanceRatioSelectorEnums method*), 189
- SetValue() (*PySpin.IEnumerationT_BalanceWhiteAutoEnums method*), 190
- SetValue() (*PySpin.IEnumerationT_BalanceWhiteAutoProfileEnums method*), 191
- SetValue() (*PySpin.IEnumerationT_BinningHorizontalModeEnums method*), 191
- SetValue() (*PySpin.IEnumerationT_BinningSelectorEnums method*), 192
- SetValue() (*PySpin.IEnumerationT_BinningVerticalModeEnums method*), 193
- SetValue() (*PySpin.IEnumerationT_BlackLevelAutoBalanceEnums method*), 193

SetValue() (PySpin.IEnumerationT_BlackLevelAutoEnum method), 194	SetValue() (PySpin.IEnumerationT_ColorTransformationValueSelectorEn method), 212
SetValue() (PySpin.IEnumerationT_BlackLevelSelectorEnum method), 195	SetValue() (PySpin.IEnumerationT_ComponentDestinationEnums method), 213
SetValue() (PySpin.IEnumerationT_BsiFlatFieldCorrection method), 195	SetValue() (PySpin.IEnumerationT_ComponentSelectorEnums method), 213
SetValue() (PySpin.IEnumerationT_BsiFlatFieldCorrection method), 196	SetValue() (PySpin.IEnumerationT_CompressionSaturationPriorityEnum method), 214
SetValue() (PySpin.IEnumerationT_ChunkBlackLevelSele method), 197	SetValue() (PySpin.IEnumerationT_CounterEventActivationEnums method), 215
SetValue() (PySpin.IEnumerationT_ChunkCounterSelecto method), 197	SetValue() (PySpin.IEnumerationT_CounterEventSourceEnums method), 215
SetValue() (PySpin.IEnumerationT_ChunkEncoderSelecto method), 198	SetValue() (PySpin.IEnumerationT_CounterResetActivationEnums method), 216
SetValue() (PySpin.IEnumerationT_ChunkEncoderStatus method), 199	SetValue() (PySpin.IEnumerationT_CounterResetSourceEnums method), 217
SetValue() (PySpin.IEnumerationT_ChunkExposureTime method), 199	SetValue() (PySpin.IEnumerationT_CounterSelectorEnums method), 217
SetValue() (PySpin.IEnumerationT_ChunkGainSelectorEn method), 200	SetValue() (PySpin.IEnumerationT_CounterStatusEnums method), 218
SetValue() (PySpin.IEnumerationT_ChunkImageCompon method), 201	SetValue() (PySpin.IEnumerationT_CounterTriggerActivationEnums method), 219
SetValue() (PySpin.IEnumerationT_ChunkPixelFormatEn method), 201	SetValue() (PySpin.IEnumerationT_CounterTriggerSourceEnums method), 219
SetValue() (PySpin.IEnumerationT_ChunkRegionIDEnum method), 202	SetValue() (PySpin.IEnumerationT_CxpConnectionTestModeEnums method), 220
SetValue() (PySpin.IEnumerationT_ChunkScan3dCoordin method), 203	SetValue() (PySpin.IEnumerationT_CxpLinkConfigurationEnums method), 221
SetValue() (PySpin.IEnumerationT_ChunkScan3dCoordin method), 203	SetValue() (PySpin.IEnumerationT_CxpLinkConfigurationPreferredEnum method), 221
SetValue() (PySpin.IEnumerationT_ChunkScan3dCoordin method), 204	SetValue() (PySpin.IEnumerationT_CxpLinkConfigurationStatusEnums method), 222
SetValue() (PySpin.IEnumerationT_ChunkScan3dCoordin method), 205	SetValue() (PySpin.IEnumerationT_CxpPoCxpStatusEnums method), 223
SetValue() (PySpin.IEnumerationT_ChunkScan3dCoordin method), 206	SetValue() (PySpin.IEnumerationT_DecimationHorizontalModeEnums method), 223
SetValue() (PySpin.IEnumerationT_ChunkScan3dDistanc method), 206	SetValue() (PySpin.IEnumerationT_DecimationSelectorEnums method), 224
SetValue() (PySpin.IEnumerationT_ChunkScan3dOutput method), 207	SetValue() (PySpin.IEnumerationT_DecimationVerticalModeEnums method), 225
SetValue() (PySpin.IEnumerationT_ChunkSelectorEnums method), 207	SetValue() (PySpin.IEnumerationT_DefectCorrectionModeEnums method), 225
SetValue() (PySpin.IEnumerationT_ChunkSourceIDEnum method), 208	SetValue() (PySpin.IEnumerationT_DeinterlacingEnums method), 226
SetValue() (PySpin.IEnumerationT_ChunkTimerSelector method), 209	SetValue() (PySpin.IEnumerationT_DeviceAccessStatusEnum method), 227
SetValue() (PySpin.IEnumerationT_ChunkTransferStream method), 209	SetValue() (PySpin.IEnumerationT_DeviceCharacterSetEnums method), 227
SetValue() (PySpin.IEnumerationT_ClConfigurationEnum method), 210	SetValue() (PySpin.IEnumerationT_DeviceClockSelectorEnums method), 228
SetValue() (PySpin.IEnumerationT_ClTimeSlotsCountEn method), 211	SetValue() (PySpin.IEnumerationT_DeviceConnectionStatusEnums method), 229
SetValue() (PySpin.IEnumerationT_ColorTransformation method), 211	SetValue() (PySpin.IEnumerationT_DeviceCurrentSpeedEnum method), 229

SetValue() (PySpin.IEnumerationT_DeviceEndiannessMechanismEnums method), 230	SetValue() (PySpin.IEnumerationT_ExposureActiveModeEnums method), 248
SetValue() (PySpin.IEnumerationT_DeviceIndicatorModeEnums method), 231	SetValue() (PySpin.IEnumerationT_ExposureAutoEnums method), 249
SetValue() (PySpin.IEnumerationT_DeviceLinkHeartbeatEnums method), 231	SetValue() (PySpin.IEnumerationT_ExposureModeEnums method), 249
SetValue() (PySpin.IEnumerationT_DeviceLinkThroughputEnums method), 232	SetValue() (PySpin.IEnumerationT_ExposureTimeModeEnums method), 250
SetValue() (PySpin.IEnumerationT_DevicePowerSupplySelectorEnums method), 233	SetValue() (PySpin.IEnumerationT_ExposureTimeSelectorEnums method), 251
SetValue() (PySpin.IEnumerationT_DeviceRegistersEndiannessEnums method), 233	SetValue() (PySpin.IEnumerationT_ExternalVoltageSelectorEnums method), 251
SetValue() (PySpin.IEnumerationT_DeviceScanTypeEnum method), 234	SetValue() (PySpin.IEnumerationT_FfcModeEnums method), 253
SetValue() (PySpin.IEnumerationT_DeviceSensorChromaEnums method), 235	SetValue() (PySpin.IEnumerationT_FileOpenModeEnums method), 253
SetValue() (PySpin.IEnumerationT_DeviceSerialPortBaudRateEnums method), 235	SetValue() (PySpin.IEnumerationT_FileOperationSelectorEnums method), 254
SetValue() (PySpin.IEnumerationT_DeviceSerialPortSelectorEnums method), 236	SetValue() (PySpin.IEnumerationT_FileOperationStatusEnums method), 255
SetValue() (PySpin.IEnumerationT_DeviceStreamChannelTypeEnums method), 237	SetValue() (PySpin.IEnumerationT_FileSelectorEnums method), 255
SetValue() (PySpin.IEnumerationT_DeviceStreamChannelTypeEnums method), 237	SetValue() (PySpin.IEnumerationT_FLIRFilterDriverStatusEnum method), 252
SetValue() (PySpin.IEnumerationT_DeviceTapGeometryEnums method), 239	SetValue() (PySpin.IEnumerationT_GainAutoBalanceEnums method), 257
SetValue() (PySpin.IEnumerationT_DeviceTemperatureSelectorEnums method), 239	SetValue() (PySpin.IEnumerationT_GainAutoEnums method), 257
SetValue() (PySpin.IEnumerationT_DeviceTLTypeEnum method), 238	SetValue() (PySpin.IEnumerationT_GainConversionEnums method), 258
SetValue() (PySpin.IEnumerationT_DeviceTypeEnum method), 240	SetValue() (PySpin.IEnumerationT_GainSelectorEnums method), 259
SetValue() (PySpin.IEnumerationT_DeviceTypeEnum method), 241	SetValue() (PySpin.IEnumerationT_GenICamXMLLocationEnum method), 259
SetValue() (PySpin.IEnumerationT_EncoderModeEnums method), 241	SetValue() (PySpin.IEnumerationT_GevCCPEnum method), 260
SetValue() (PySpin.IEnumerationT_EncoderOutputModeEnums method), 242	SetValue() (PySpin.IEnumerationT_GevCCPEnums method), 261
SetValue() (PySpin.IEnumerationT_EncoderResetActivationEnums method), 243	SetValue() (PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums method), 261
SetValue() (PySpin.IEnumerationT_EncoderResetSourceEnums method), 243	SetValue() (PySpin.IEnumerationT_GevGVCPExtendedStatusCodesSelectorEnums method), 262
SetValue() (PySpin.IEnumerationT_EncoderSelectorEnums method), 244	SetValue() (PySpin.IEnumerationT_GevGVSPExtendedIDModeEnums method), 263
SetValue() (PySpin.IEnumerationT_EncoderSourceAEnums method), 245	SetValue() (PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnums method), 263
SetValue() (PySpin.IEnumerationT_EncoderSourceBEnums method), 245	SetValue() (PySpin.IEnumerationT_GevIEEE1588ModeEnums method), 264
SetValue() (PySpin.IEnumerationT_EncoderStatusEnums method), 246	SetValue() (PySpin.IEnumerationT_GevIEEE1588StatusEnums method), 265
SetValue() (PySpin.IEnumerationT_EventNotificationEnums method), 247	SetValue() (PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnums method), 265
SetValue() (PySpin.IEnumerationT_EventSelectorEnums method), 247	SetValue() (PySpin.IEnumerationT_GevIPConfigurationStatusEnums method), 266

<code>SetValue()</code> (PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnums (PySpin.IEnumerationT_PixelSizeEnums method), 267	<code>SetValue()</code> (PySpin.IEnumerationT_PixelSizeEnums (PySpin.IEnumerationT_PixelSizeEnums method), 285
<code>SetValue()</code> (PySpin.IEnumerationT_GevSCPDDirectionEnums (PySpin.IEnumerationT_POEStatusEnums method), 267	<code>SetValue()</code> (PySpin.IEnumerationT_POEStatusEnums (PySpin.IEnumerationT_POEStatusEnums method), 282
<code>SetValue()</code> (PySpin.IEnumerationT_GevSupportedOptionsEnums (PySpin.IEnumerationT_RegionDestinationEnums method), 268	<code>SetValue()</code> (PySpin.IEnumerationT_RegionDestinationEnums (PySpin.IEnumerationT_RegionDestinationEnums method), 285
<code>SetValue()</code> (PySpin.IEnumerationT_GUIXMLLocationEnums (PySpin.IEnumerationT_RegionModeEnums method), 256	<code>SetValue()</code> (PySpin.IEnumerationT_RegionModeEnums (PySpin.IEnumerationT_RegionModeEnums method), 286
<code>SetValue()</code> (PySpin.IEnumerationT_ImageComponentSelectorsEnums (PySpin.IEnumerationT_RegionSelectorEnums method), 269	<code>SetValue()</code> (PySpin.IEnumerationT_RegionSelectorEnums (PySpin.IEnumerationT_RegionSelectorEnums method), 287
<code>SetValue()</code> (PySpin.IEnumerationT_ImageCompressionJPEGEnums (PySpin.IEnumerationT_RgbTransformLightSourceEnums method), 269	<code>SetValue()</code> (PySpin.IEnumerationT_RgbTransformLightSourceEnums (PySpin.IEnumerationT_RgbTransformLightSourceEnums method), 287
<code>SetValue()</code> (PySpin.IEnumerationT_ImageCompressionMJPEGEnums (PySpin.IEnumerationT_Scan3dCoordinateReferenceSelectorsEnums method), 270	<code>SetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateReferenceSelectorsEnums (PySpin.IEnumerationT_Scan3dCoordinateReferenceSelectorsEnums method), 288
<code>SetValue()</code> (PySpin.IEnumerationT_ImageCompressionRawEnums (PySpin.IEnumerationT_Scan3dCoordinateSelectorsEnums method), 271	<code>SetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateSelectorsEnums (PySpin.IEnumerationT_Scan3dCoordinateSelectorsEnums method), 289
<code>SetValue()</code> (PySpin.IEnumerationT_InterfaceTypeEnum (PySpin.IEnumerationT_Scan3dCoordinateSystemEnums method), 271	<code>SetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateSystemEnums (PySpin.IEnumerationT_Scan3dCoordinateSystemEnums method), 289
<code>SetValue()</code> (PySpin.IEnumerationT_LensShadingCoefficientsEnums (PySpin.IEnumerationT_Scan3dCoordinateSystemReferenceSelectorsEnums method), 273	<code>SetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateSystemReferenceSelectorsEnums (PySpin.IEnumerationT_Scan3dCoordinateSystemReferenceSelectorsEnums method), 290
<code>SetValue()</code> (PySpin.IEnumerationT_LensShadingCorrectionEnums (PySpin.IEnumerationT_Scan3dCoordinateTransformSelectorsEnums method), 273	<code>SetValue()</code> (PySpin.IEnumerationT_Scan3dCoordinateTransformSelectorsEnums (PySpin.IEnumerationT_Scan3dCoordinateTransformSelectorsEnums method), 291
<code>SetValue()</code> (PySpin.IEnumerationT_LineFormatEnums (PySpin.IEnumerationT_Scan3dDistanceUnitEnums method), 274	<code>SetValue()</code> (PySpin.IEnumerationT_Scan3dDistanceUnitEnums (PySpin.IEnumerationT_Scan3dDistanceUnitEnums method), 291
<code>SetValue()</code> (PySpin.IEnumerationT_LineInputFilterSelectorsEnums (PySpin.IEnumerationT_Scan3dOutputModeEnums method), 275	<code>SetValue()</code> (PySpin.IEnumerationT_Scan3dOutputModeEnums (PySpin.IEnumerationT_Scan3dOutputModeEnums method), 292
<code>SetValue()</code> (PySpin.IEnumerationT_LineModeEnums (PySpin.IEnumerationT_SensorDigitizationTapsEnums method), 275	<code>SetValue()</code> (PySpin.IEnumerationT_SensorDigitizationTapsEnums (PySpin.IEnumerationT_SensorDigitizationTapsEnums method), 293
<code>SetValue()</code> (PySpin.IEnumerationT_LineSelectorEnums (PySpin.IEnumerationT_SensorShutterModeEnums method), 276	<code>SetValue()</code> (PySpin.IEnumerationT_SensorShutterModeEnums (PySpin.IEnumerationT_SensorShutterModeEnums method), 293
<code>SetValue()</code> (PySpin.IEnumerationT_LineSourceEnums (PySpin.IEnumerationT_SensorTapsEnums method), 277	<code>SetValue()</code> (PySpin.IEnumerationT_SensorTapsEnums (PySpin.IEnumerationT_SensorTapsEnums method), 294
<code>SetValue()</code> (PySpin.IEnumerationT_LogicBlockLUTInputEnums (PySpin.IEnumerationT_SequencerConfigurationModeEnums method), 277	<code>SetValue()</code> (PySpin.IEnumerationT_SequencerConfigurationModeEnums (PySpin.IEnumerationT_SequencerConfigurationModeEnums method), 295
<code>SetValue()</code> (PySpin.IEnumerationT_LogicBlockLUTInputEnums (PySpin.IEnumerationT_SequencerConfigurationValidEnums method), 278	<code>SetValue()</code> (PySpin.IEnumerationT_SequencerConfigurationValidEnums (PySpin.IEnumerationT_SequencerConfigurationValidEnums method), 295
<code>SetValue()</code> (PySpin.IEnumerationT_LogicBlockLUTInputEnums (PySpin.IEnumerationT_SequencerModeEnums method), 279	<code>SetValue()</code> (PySpin.IEnumerationT_SequencerModeEnums (PySpin.IEnumerationT_SequencerModeEnums method), 296
<code>SetValue()</code> (PySpin.IEnumerationT_LogicBlockLUTSelectorsEnums (PySpin.IEnumerationT_SequencerSetValidEnums method), 279	<code>SetValue()</code> (PySpin.IEnumerationT_SequencerSetValidEnums (PySpin.IEnumerationT_SequencerSetValidEnums method), 297
<code>SetValue()</code> (PySpin.IEnumerationT_LogicBlockSelectorEnums (PySpin.IEnumerationT_SequencerTriggerActivationEnums method), 280	<code>SetValue()</code> (PySpin.IEnumerationT_SequencerTriggerActivationEnums (PySpin.IEnumerationT_SequencerTriggerActivationEnums method), 297
<code>SetValue()</code> (PySpin.IEnumerationT_LUTSelectorEnums (PySpin.IEnumerationT_SequencerTriggerSourceEnums method), 272	<code>SetValue()</code> (PySpin.IEnumerationT_SequencerTriggerSourceEnums (PySpin.IEnumerationT_SequencerTriggerSourceEnums method), 298
<code>SetValue()</code> (PySpin.IEnumerationT_MultiRoiConfigurationEnums (PySpin.IEnumerationT_SerialPortBaudRateEnums method), 281	<code>SetValue()</code> (PySpin.IEnumerationT_SerialPortBaudRateEnums (PySpin.IEnumerationT_SerialPortBaudRateEnums method), 299
<code>SetValue()</code> (PySpin.IEnumerationT_MultiRoiSelectorEnums (PySpin.IEnumerationT_SerialPortParityEnums method), 281	<code>SetValue()</code> (PySpin.IEnumerationT_SerialPortParityEnums (PySpin.IEnumerationT_SerialPortParityEnums method), 299
<code>SetValue()</code> (PySpin.IEnumerationT_PixelColorFilterEnums (PySpin.IEnumerationT_SerialPortSelectorEnums method), 283	<code>SetValue()</code> (PySpin.IEnumerationT_SerialPortSelectorEnums (PySpin.IEnumerationT_SerialPortSelectorEnums method), 300
<code>SetValue()</code> (PySpin.IEnumerationT_PixelFormatEnums (PySpin.IEnumerationT_SerialPortSourceEnums method), 283	<code>SetValue()</code> (PySpin.IEnumerationT_SerialPortSourceEnums (PySpin.IEnumerationT_SerialPortSourceEnums method), 301
<code>SetValue()</code> (PySpin.IEnumerationT_PixelFormatInfoSelectorsEnums (PySpin.IEnumerationT_SerialPortStopBitsEnums method), 284	<code>SetValue()</code> (PySpin.IEnumerationT_SerialPortStopBitsEnums (PySpin.IEnumerationT_SerialPortStopBitsEnums method), 301

SetValue() (PySpin.IEnumerationT_SoftwareSignalSelectorEnums method), 302
 SetValue() (PySpin.IEnumerationT_SourceSelectorEnums method), 303
 SetValue() (PySpin.IEnumerationT_StereoResolutionEnums method), 303
 SetValue() (PySpin.IEnumerationT_StreamBufferCountModeEnums method), 304
 SetValue() (PySpin.IEnumerationT_StreamBufferHandlingModeEnums method), 305
 SetValue() (PySpin.IEnumerationT_StreamModeEnum method), 305
 SetValue() (PySpin.IEnumerationT_StreamTypeEnum method), 306
 SetValue() (PySpin.IEnumerationT_TeledyneGigeVisionFireRateEnums method), 307
 SetValue() (PySpin.IEnumerationT_TestPatternEnums method), 308
 SetValue() (PySpin.IEnumerationT_TestPatternGeneratorEnums method), 309
 SetValue() (PySpin.IEnumerationT_TimerSelectorEnums method), 309
 SetValue() (PySpin.IEnumerationT_TimerStatusEnums method), 310
 SetValue() (PySpin.IEnumerationT_TimerTriggerActivationEnums method), 311
 SetValue() (PySpin.IEnumerationT_TimerTriggerSourceEnums method), 311
 SetValue() (PySpin.IEnumerationT_TLTypeEnum method), 307
 SetValue() (PySpin.IEnumerationT_TransferComponentSelectorEnums method), 312
 SetValue() (PySpin.IEnumerationT_TransferControlModeEnums method), 313
 SetValue() (PySpin.IEnumerationT_TransferOperationModeEnums method), 313
 SetValue() (PySpin.IEnumerationT_TransferQueueModeEnums method), 314
 SetValue() (PySpin.IEnumerationT_TransferSelectorEnums method), 315
 SetValue() (PySpin.IEnumerationT_TransferStatusSelectorEnums method), 315
 SetValue() (PySpin.IEnumerationT_TransferTriggerActivationEnums method), 316
 SetValue() (PySpin.IEnumerationT_TransferTriggerModeEnums method), 317
 SetValue() (PySpin.IEnumerationT_TransferTriggerSelectorEnums method), 317
 SetValue() (PySpin.IEnumerationT_TransferTriggerSourceEnums method), 318
 SetValue() (PySpin.IEnumerationT_TriggerActivationEnums method), 319
 SetValue() (PySpin.IEnumerationT_TriggerModeEnums method), 319
 SetValue() (PySpin.IEnumerationT_TriggerOverlapEnums method), 320
 SetValue() (PySpin.IEnumerationT_TriggerSelectorEnums method), 321
 SetValue() (PySpin.IEnumerationT_TriggerSourceEnums method), 321
 SetValue() (PySpin.IEnumerationT_U3VCurrentSpeedEnums method), 322
 SetValue() (PySpin.IEnumerationT_UserOutputSelectorEnums method), 323
 SetValue() (PySpin.IEnumerationT_UserSetDefaultEnums method), 323
 SetValue() (PySpin.IEnumerationT_UserSetSelectorEnums method), 324
 SetValue() (PySpin.IEnumerationT_WhiteClipSelectorEnums method), 325
 SetValue() (PySpin.IFloat method), 326
 SetValue() (PySpin.IInteger method), 333
 SetValue() (PySpin.IntegerNode method), 378
 SetValue() (PySpin.IString method), 342
 SetValue() (PySpin.StringNode method), 404
 Sharpening (PySpin.Camera property), 31, 139
 SharpeningAuto (PySpin.Camera property), 31, 139
 SharpeningEnable (PySpin.Camera property), 32, 139
 SharpeningThreshold (PySpin.Camera property), 32, 139
 SIOption (class in PySpin), 400
 size() (PySpin.double_autovector_t method), 419
 size() (PySpin.gcstring method), 422
 size() (PySpin.int64_autovector_t method), 422
 size() (PySpin.node_vector method), 424
 size() (PySpin.value_vector method), 425
 SmallPenalty (PySpin.Camera property), 32, 139
 SoftwareSignalPulse (PySpin.Camera property), 32, 139
 SoftwareSignalSelector (PySpin.Camera property), 32, 139
 SourceCount (PySpin.Camera property), 32, 139
 SourceSelector (PySpin.Camera property), 32, 139
 SpinnakerException (class in PySpin), 69
 SpinnakerUpdate_SetMsgCallback() (in module PySpin), 401
 SpinnakerUpdate_SetProgCallback() (in module PySpin), 401
 SpinVideo (class in PySpin), 69, 401
 Status (PySpin.ActionCommandResult property), 83
 Stereo3DPoint (class in PySpin), 403
 StereoCameraParameters (class in PySpin), 403
 StereoHeight (PySpin.Camera property), 32, 139
 StereoResolution (PySpin.Camera property), 32, 139
 StereoWidth (PySpin.Camera property), 32, 139
 StoreToBag() (PySpin.CFeatureBag method), 100
 StreamAnnounceBufferMinimum (PySpin.TransportLayerStream property),

81, 414		81, 414	
StreamAnnouncedBufferCount	(PySpin.TransportLayerStream property), 81, 414	StreamID (PySpin.TransportLayerDevice property), 78, 412	
StreamBlocksProcessingTimeLast	(PySpin.TransportLayerStream property), 81, 414	StreamID (PySpin.TransportLayerStream property), 81, 414	
StreamBlocksProcessingTimeMax	(PySpin.TransportLayerStream property), 81, 414	StreamIncompleteFrameCount	(PySpin.TransportLayerStream property), 81, 414
StreamBlocksProcessingTimeMin	(PySpin.TransportLayerStream property), 81, 414	StreamInputBufferCount	(PySpin.TransportLayerStream property), 81, 414
StreamBlocksReceptionTimeLast	(PySpin.TransportLayerStream property), 81, 414	StreamIsGrabbing	(PySpin.TransportLayerStream property), 81, 414
StreamBlocksReceptionTimeMax	(PySpin.TransportLayerStream property), 81, 414	StreamLostFrameCount	(PySpin.TransportLayerStream property), 81, 414
StreamBlocksReceptionTimeMin	(PySpin.TransportLayerStream property), 81, 414	StreamMissedPacketCount	(PySpin.TransportLayerStream property), 81, 415
StreamBlockTransferSize	(PySpin.TransportLayerStream property), 81, 414	StreamMode	(PySpin.TransportLayerStream property), 81, 415
StreamBufferAlignment	(PySpin.TransportLayerStream property), 81, 414	StreamOutputBufferCount	(PySpin.TransportLayerStream property), 81, 415
StreamBufferCountManual	(PySpin.TransportLayerStream property), 81, 414	StreamPacketResendEnable	(PySpin.TransportLayerStream property), 81, 415
StreamBufferCountMax	(PySpin.TransportLayerStream property), 81, 414	StreamPacketResendMaxRequests	(PySpin.TransportLayerStream property), 82, 415
StreamBufferCountMode	(PySpin.TransportLayerStream property), 81, 414	StreamPacketResendReceivedPacketCount	(PySpin.TransportLayerStream property), 82, 415
StreamBufferCountResult	(PySpin.TransportLayerStream property), 81, 414	StreamPacketResendRequestCount	(PySpin.TransportLayerStream property), 82, 415
StreamBufferHandlingMode	(PySpin.TransportLayerStream property), 81, 414	StreamPacketResendRequestedPacketCount	(PySpin.TransportLayerStream property), 82, 415
StreamChunkCountMaximum	(PySpin.TransportLayerStream property), 81, 414	StreamPacketResendRequestTimeoutCount	(PySpin.TransportLayerStream property), 82, 415
StreamCRCCheckEnable	(PySpin.TransportLayerStream property), 81, 414	StreamPacketResendTimeout	(PySpin.TransportLayerStream property), 82, 415
StreamDeliveredFrameCount	(PySpin.TransportLayerStream property), 81, 414	StreamPacketsDuplicatedCount	(PySpin.TransportLayerStream property), 82, 415
StreamDroppedFrameCount	(PySpin.TransportLayerStream property), 81, 414	StreamPacketsNotYetAvailableCount	(PySpin.TransportLayerStream property), 82, 415
		StreamPacketsPerFrameCount	(PySpin.TransportLayerStream property), 82, 415

StreamPacketsTemporarilyUnavailableCount
 (PySpin.TransportLayerStream property), 82, 415
 StreamPacketsTimeoutCount
 (PySpin.TransportLayerStream property), 82, 415
 StreamPacketsUnavailableCount
 (PySpin.TransportLayerStream property), 82, 415
 StreamReceivedFrameCount
 (PySpin.TransportLayerStream property), 82, 415
 StreamReceivedPacketCount
 (PySpin.TransportLayerStream property), 82, 415
 StreamSelector (PySpin.TransportLayerDevice property), 78, 412
 StreamStartedFrameCount
 (PySpin.TransportLayerStream property), 82, 415
 StreamType (PySpin.TransportLayerStream property), 82, 415
 StringNode (class in PySpin), 404
 StringRegNode (class in PySpin), 405
 SubMinor (PySpin.Version_t property), 418
 substr() (PySpin.gcstring method), 422
 swap() (PySpin.gcstring method), 422
 System (class in PySpin), 71, 405
 System_GetInstance() (in module PySpin), 409
 SystemEventHandler (class in PySpin), 8, 409
 SystemPtr (class in PySpin), 75, 409
T
 TeledyneGigeVisionFilterDriverStatus
 (PySpin.TransportLayerInterface property), 80, 414
 Test0001 (PySpin.Camera property), 32, 139
 TestEventGenerate (PySpin.Camera property), 32, 139
 TestPattern (PySpin.Camera property), 32, 139
 TestPatternGeneratorSelector (PySpin.Camera property), 32, 139
 TestPendingAck (PySpin.Camera property), 32, 139
 thisown (PySpin.ActionCommandResult property), 83
 thisown (PySpin.AVIOption property), 83
 thisown (PySpin.BMPOption property), 83
 thisown (PySpin.BooleanNode property), 85
 thisown (PySpin.Camera property), 35, 142
 thisown (PySpin.CameraBase property), 39, 146
 thisown (PySpin.CameraList property), 41, 149
 thisown (PySpin.CameraPtr property), 41, 149
 thisown (PySpin.CategoryNode property), 149
 thisown (PySpin.CBasePtr property), 10, 85
 thisown (PySpin.CBooleanPtr property), 87
 thisown (PySpin.CCategoryPtr property), 90
 thisown (PySpin.CCMSettings property), 88
 thisown (PySpin.CCommandPtr property), 93
 thisown (PySpin.CDeviceInfoPtr property), 93
 thisown (PySpin.CEnumEntryPtr property), 96
 thisown (PySpin.CEnumerationPtr property), 99
 thisown (PySpin.CFeatureBag property), 100
 thisown (PySpin.CFloatPtr property), 100
 thisown (PySpin.ChannelStatistics property), 42, 150
 thisown (PySpin.ChunkData property), 46, 153
 thisown (PySpin.CIntegerPtr property), 103
 thisown (PySpin.CNodeMapDynPtr property), 105
 thisown (PySpin.CNodeMapPtr property), 106
 thisown (PySpin.CNodePtr property), 108
 thisown (PySpin.CommandNode property), 155
 thisown (PySpin.CRegisterPtr property), 111
 thisown (PySpin.CSelectorPtr property), 111
 thisown (PySpin.CSelectorSet property), 112
 thisown (PySpin.CStringPtr property), 115
 thisown (PySpin.CValuePtr property), 117
 thisown (PySpin.DeviceArrivalEventHandler property), 5, 155
 thisown (PySpin.DeviceEventExposureEndData property), 155
 thisown (PySpin.DeviceEventHandler property), 6, 155
 thisown (PySpin.DeviceEventInferenceData property), 156
 thisown (PySpin.DeviceRemovalEventHandler property), 6, 156
 thisown (PySpin.double_autovector_t property), 419
 thisown (PySpin.EAccessModeClass property), 157
 thisown (PySpin.ECachingModeClass property), 157
 thisown (PySpin.EDisplayNotationClass property), 158
 thisown (PySpin.EEndianessClass property), 159
 thisown (PySpin.EGenApiSchemaVersionClass property), 160
 thisown (PySpin.EInputDirectionClass property), 160
 thisown (PySpin.ENamespaceClass property), 161
 thisown (PySpin.EnumEntryNode property), 167
 thisown (PySpin.EnumNode property), 168
 thisown (PySpin.ERepresentationClass property), 162
 thisown (PySpin.ESignClass property), 162
 thisown (PySpin.ESlopeClass property), 163
 thisown (PySpin.EStandardNameSpaceClass property), 164
 thisown (PySpin.EventHandler property), 6, 169
 thisown (PySpin.EVisibilityClass property), 165
 thisown (PySpin.EYesNoClass property), 165
 thisown (PySpin.FloatNode property), 171
 thisown (PySpin.FloatRegNode property), 172
 thisown (PySpin.gcstring property), 422
 thisown (PySpin.H264Option property), 173
 thisown (PySpin.IBase property), 174
 thisown (PySpin.IBoolean property), 174

`thisown` (`PySpin.ICameraBase` property), 176

`thisown` (`PySpin.ICameraList` property), 177

`thisown` (`PySpin.ICategory` property), 177

`thisown` (`PySpin.IChunkData` property), 179

`thisown` (`PySpin.ICommand` property), 179

`thisown` (`PySpin.IDestroy` property), 180

`thisown` (`PySpin.IDeviceArrivalEventHandler` property), 180

`thisown` (`PySpin.IDeviceEventHandler` property), 180

`thisown` (`PySpin.IDeviceInfo` property), 181

`thisown` (`PySpin.IDeviceRemovalEventHandler` property), 181

`thisown` (`PySpin.IEnumEntry` property), 181

`thisown` (`PySpin.IEnumeration` property), 182

`thisown` (`PySpin.IEnumerationT_AcquisitionModeEnums` property), 183

`thisown` (`PySpin.IEnumerationT_AcquisitionStatusSelectorEnums` property), 184

`thisown` (`PySpin.IEnumerationT_ActionSelectorEnums` property), 184

`thisown` (`PySpin.IEnumerationT_ActionUnconditionalModeEnums` property), 185

`thisown` (`PySpin.IEnumerationT_AdcBitDepthEnums` property), 186

`thisown` (`PySpin.IEnumerationT_AutoAlgorithmSelectorEnums` property), 186

`thisown` (`PySpin.IEnumerationT_AutoExposureControlPriorityEnums` property), 187

`thisown` (`PySpin.IEnumerationT_AutoExposureLightingModeEnums` property), 188

`thisown` (`PySpin.IEnumerationT_AutoExposureMeteringModeEnums` property), 188

`thisown` (`PySpin.IEnumerationT_AutoExposureTargetGreyValuesEnums` property), 189

`thisown` (`PySpin.IEnumerationT_BalanceRatioSelectorEnums` property), 190

`thisown` (`PySpin.IEnumerationT_BalanceWhiteAutoEnums` property), 190

`thisown` (`PySpin.IEnumerationT_BalanceWhiteAutoProfileEnums` property), 191

`thisown` (`PySpin.IEnumerationT_BinningHorizontalModeEnums` property), 192

`thisown` (`PySpin.IEnumerationT_BinningSelectorEnums` property), 192

`thisown` (`PySpin.IEnumerationT_BinningVerticalModeEnums` property), 193

`thisown` (`PySpin.IEnumerationT_BlackLevelAutoBalanceEnums` property), 194

`thisown` (`PySpin.IEnumerationT_BlackLevelAutoEnums` property), 194

`thisown` (`PySpin.IEnumerationT_BlackLevelSelectorEnums` property), 195

`thisown` (`PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums` property), 196

`thisown` (`PySpin.IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums` property), 196

`thisown` (`PySpin.IEnumerationT_ChunkBlackLevelSelectorEnums` property), 197

`thisown` (`PySpin.IEnumerationT_ChunkCounterSelectorEnums` property), 198

`thisown` (`PySpin.IEnumerationT_ChunkEncoderSelectorEnums` property), 198

`thisown` (`PySpin.IEnumerationT_ChunkEncoderStatusEnums` property), 199

`thisown` (`PySpin.IEnumerationT_ChunkExposureTimeSelectorEnums` property), 200

`thisown` (`PySpin.IEnumerationT_ChunkGainSelectorEnums` property), 200

`thisown` (`PySpin.IEnumerationT_ChunkImageComponentEnums` property), 201

`thisown` (`PySpin.IEnumerationT_ChunkPixelFormatEnums` property), 202

`thisown` (`PySpin.IEnumerationT_ChunkRegionIDEnums` property), 202

`thisown` (`PySpin.IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums` property), 203

`thisown` (`PySpin.IEnumerationT_ChunkScan3dCoordinateSelectorEnums` property), 204

`thisown` (`PySpin.IEnumerationT_ChunkScan3dCoordinateSystemEnums` property), 204

`thisown` (`PySpin.IEnumerationT_ChunkScan3dCoordinateSystemReferenceSelectorEnums` property), 205

`thisown` (`PySpin.IEnumerationT_ChunkScan3dCoordinateTransformSelectorEnums` property), 206

`thisown` (`PySpin.IEnumerationT_ChunkScan3dDistanceUnitEnums` property), 206

`thisown` (`PySpin.IEnumerationT_ChunkScan3dOutputModeEnums` property), 207

`thisown` (`PySpin.IEnumerationT_ChunkSelectorEnums` property), 208

`thisown` (`PySpin.IEnumerationT_ChunkSourceIDEnums` property), 208

`thisown` (`PySpin.IEnumerationT_ChunkTimerSelectorEnums` property), 209

`thisown` (`PySpin.IEnumerationT_ChunkTransferStreamIDEnums` property), 210

`thisown` (`PySpin.IEnumerationT_CiConfigurationEnums` property), 210

`thisown` (`PySpin.IEnumerationT_CiTimeSlotsCountEnums` property), 211

`thisown` (`PySpin.IEnumerationT_ColorTransformationSelectorEnums` property), 212

`thisown` (`PySpin.IEnumerationT_ColorTransformationValueSelectorEnums` property), 212

`thisown` (`PySpin.IEnumerationT_ComponentDestinationEnums` property), 213

`thisown` (`PySpin.IEnumerationT_ComponentSelectorEnums` property), 214

[thisown\(PySpin.IEnumerationT_CompressionSaturationProperty\), 214](#)
[thisown\(PySpin.IEnumerationT_CounterEventActivationEnum\), 215](#)
[thisown\(PySpin.IEnumerationT_CounterEventSourceEnum\), 216](#)
[thisown\(PySpin.IEnumerationT_CounterResetActivationEnum\), 216](#)
[thisown\(PySpin.IEnumerationT_CounterResetSourceEnum\), 217](#)
[thisown\(PySpin.IEnumerationT_CounterSelectorEnums\), 218](#)
[thisown\(PySpin.IEnumerationT_CounterStatusEnums\), 218](#)
[thisown\(PySpin.IEnumerationT_CounterTriggerActivationEnum\), 219](#)
[thisown\(PySpin.IEnumerationT_CounterTriggerSourceEnum\), 220](#)
[thisown\(PySpin.IEnumerationT_CxpConnectionTestModeEnum\), 220](#)
[thisown\(PySpin.IEnumerationT_CxpLinkConfigurationEnum\), 221](#)
[thisown\(PySpin.IEnumerationT_CxpLinkConfigurationPropertyEnum\), 222](#)
[thisown\(PySpin.IEnumerationT_CxpLinkConfigurationStatusEnum\), 222](#)
[thisown\(PySpin.IEnumerationT_CxpPoCxpStatusEnums\), 223](#)
[thisown\(PySpin.IEnumerationT_DecimationHorizontalModeEnum\), 224](#)
[thisown\(PySpin.IEnumerationT_DecimationSelectorEnum\), 224](#)
[thisown\(PySpin.IEnumerationT_DecimationVerticalModeEnum\), 225](#)
[thisown\(PySpin.IEnumerationT_DefectCorrectionModeEnum\), 226](#)
[thisown\(PySpin.IEnumerationT_DeinterlacingEnums\), 226](#)
[thisown\(PySpin.IEnumerationT_DeviceAccessStatusEnum\), 227](#)
[thisown\(PySpin.IEnumerationT_DeviceCharacterSetEnum\), 228](#)
[thisown\(PySpin.IEnumerationT_DeviceClockSelectorEnum\), 228](#)
[thisown\(PySpin.IEnumerationT_DeviceConnectionStatusEnum\), 229](#)
[thisown\(PySpin.IEnumerationT_DeviceCurrentSpeedEnum\), 230](#)
[thisown\(PySpin.IEnumerationT_DeviceEndianessMechanismEnum\), 230](#)
[thisown\(PySpin.IEnumerationT_DeviceIndicatorModeEnum\), 231](#)
[thisown\(PySpin.IEnumerationT_DeviceLinkHeartbeatModeEnum\), 232](#)
[thisown\(PySpin.IEnumerationT_DeviceLinkThroughputLimitModeEnum\), 232](#)
[thisown\(PySpin.IEnumerationT_DevicePowerSupplySelectorEnums\), 233](#)
[thisown\(PySpin.IEnumerationT_DeviceRegistersEndiannessEnums\), 234](#)
[thisown\(PySpin.IEnumerationT_DeviceScanTypeEnum\), 234](#)
[thisown\(PySpin.IEnumerationT_DeviceSensorChromaEnums\), 235](#)
[thisown\(PySpin.IEnumerationT_DeviceSerialPortBaudRateEnums\), 236](#)
[thisown\(PySpin.IEnumerationT_DeviceSerialPortSelectorEnums\), 236](#)
[thisown\(PySpin.IEnumerationT_DeviceStreamChannelEndiannessEnums\), 237](#)
[thisown\(PySpin.IEnumerationT_DeviceStreamChannelTypeEnum\), 238](#)
[thisown\(PySpin.IEnumerationT_DeviceTapGeometryEnums\), 239](#)
[thisown\(PySpin.IEnumerationT_DeviceTemperatureSelectorEnums\), 240](#)
[thisown\(PySpin.IEnumerationT_DeviceTLTypeEnum\), 238](#)
[thisown\(PySpin.IEnumerationT_DeviceTypeEnum\), 240](#)
[thisown\(PySpin.IEnumerationT_DeviceTypeEnum\), 241](#)
[thisown\(PySpin.IEnumerationT_EncoderModeEnums\), 242](#)
[thisown\(PySpin.IEnumerationT_EncoderOutputModeEnums\), 242](#)
[thisown\(PySpin.IEnumerationT_EncoderResetActivationEnums\), 243](#)
[thisown\(PySpin.IEnumerationT_EncoderResetSourceEnums\), 244](#)
[thisown\(PySpin.IEnumerationT_EncoderSelectorEnums\), 244](#)
[thisown\(PySpin.IEnumerationT_EncoderSourceAEnums\), 245](#)
[thisown\(PySpin.IEnumerationT_EncoderSourceBEnums\), 246](#)
[thisown\(PySpin.IEnumerationT_EncoderStatusEnums\), 246](#)
[thisown\(PySpin.IEnumerationT_EventNotificationEnums\), 247](#)
[thisown\(PySpin.IEnumerationT_EventSelectorEnums\), 248](#)
[thisown\(PySpin.IEnumerationT_ExposureActiveModeEnums\), 248](#)
[thisown\(PySpin.IEnumerationT_ExposureAutoEnums\), 249](#)
[thisown\(PySpin.IEnumerationT_ExposureModeEnums\), 250](#)

`thisown(PySpin.IEnumerationT_ExposureTimeModeEnums property)`, 250

`thisown(PySpin.IEnumerationT_ExposureTimeSelectorEnums property)`, 251

`thisown(PySpin.IEnumerationT_ExternalVoltageSelectorEnums property)`, 252

`thisown(PySpin.IEnumerationT_FfcModeEnums property)`, 253

`thisown(PySpin.IEnumerationT_FileOpenModeEnums property)`, 254

`thisown(PySpin.IEnumerationT_FileOperationSelectorEnums property)`, 254

`thisown(PySpin.IEnumerationT_FileOperationStatusEnums property)`, 255

`thisown(PySpin.IEnumerationT_FileSelectorEnums property)`, 256

`thisown(PySpin.IEnumerationT_FLIRFilterDriverStatusEnums property)`, 252

`thisown(PySpin.IEnumerationT_GainAutoBalanceEnums property)`, 257

`thisown(PySpin.IEnumerationT_GainAutoEnums property)`, 258

`thisown(PySpin.IEnumerationT_GainConversionEnums property)`, 258

`thisown(PySpin.IEnumerationT_GainSelectorEnums property)`, 259

`thisown(PySpin.IEnumerationT_GenICamXMLLocationEnums property)`, 260

`thisown(PySpin.IEnumerationT_GevCCPEnum property)`, 260

`thisown(PySpin.IEnumerationT_GevCCPEnums property)`, 261

`thisown(PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums property)`, 262

`thisown(PySpin.IEnumerationT_GevGVCPExtendedStatusEnums property)`, 262

`thisown(PySpin.IEnumerationT_GevGVSPExtendedIDModes property)`, 263

`thisown(PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnums property)`, 264

`thisown(PySpin.IEnumerationT_GevIEEE1588ModeEnums property)`, 264

`thisown(PySpin.IEnumerationT_GevIEEE1588StatusEnums property)`, 265

`thisown(PySpin.IEnumerationT_GevIEEE1588StatusLatchEnums property)`, 266

`thisown(PySpin.IEnumerationT_GevIPConfigurationStatusEnums property)`, 266

`thisown(PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnums property)`, 267

`thisown(PySpin.IEnumerationT_GevSCPDDirectionEnums property)`, 268

`thisown(PySpin.IEnumerationT_GevSupportedOptionSelectorEnums property)`, 268

`thisown(PySpin.IEnumerationT_GUIXMLLocationEnums property)`, 256

`thisown(PySpin.IEnumerationT_ImageComponentSelectorEnums property)`, 269

`thisown(PySpin.IEnumerationT_ImageCompressionJPEGFormatOptions property)`, 270

`thisown(PySpin.IEnumerationT_ImageCompressionModeEnums property)`, 270

`thisown(PySpin.IEnumerationT_ImageCompressionRateOptions property)`, 271

`thisown(PySpin.IEnumerationT_InterfaceTypeEnum property)`, 272

`thisown(PySpin.IEnumerationT_LensShadingCoefficientActiveSetEnums property)`, 273

`thisown(PySpin.IEnumerationT_LensShadingCorrectionModeEnums property)`, 274

`thisown(PySpin.IEnumerationT_LineFormatEnums property)`, 274

`thisown(PySpin.IEnumerationT_LineInputFilterSelectorEnums property)`, 275

`thisown(PySpin.IEnumerationT_LineModeEnums property)`, 276

`thisown(PySpin.IEnumerationT_LineSelectorEnums property)`, 276

`thisown(PySpin.IEnumerationT_LineSourceEnums property)`, 277

`thisown(PySpin.IEnumerationT_LogicBlockLUTInputActivationEnums property)`, 278

`thisown(PySpin.IEnumerationT_LogicBlockLUTInputSelectorEnums property)`, 278

`thisown(PySpin.IEnumerationT_LogicBlockLUTInputSourceEnums property)`, 279

`thisown(PySpin.IEnumerationT_LogicBlockLUTSelectorEnums property)`, 280

`thisown(PySpin.IEnumerationT_LogicBlockSelectorEnums property)`, 280

`thisown(PySpin.IEnumerationT_LUTSelectorEnums property)`, 272

`thisown(PySpin.IEnumerationT_MultiRoiConfigurationInvalidReasonEnums property)`, 281

`thisown(PySpin.IEnumerationT_MultiRoiSelectorEnums property)`, 282

`thisown(PySpin.IEnumerationT_PixelColorFilterEnums property)`, 283

`thisown(PySpin.IEnumerationT_PixelFormatEnums property)`, 284

`thisown(PySpin.IEnumerationT_PixelFormatInfoSelectorEnums property)`, 284

`thisown(PySpin.IEnumerationT_PixelSizeEnums property)`, 285

`thisown(PySpin.IEnumerationT_POEStatusEnum property)`, 282

`thisown(PySpin.IEnumerationT_RegionDestinationEnums property)`, 286

<code>thisown (PySpin.IEnumerationT_RegionModeEnums property), 286</code>	<code>thisown (PySpin.IEnumerationT_StreamBufferCountModeEnum property), 304</code>
<code>thisown (PySpin.IEnumerationT_RegionSelectorEnums property), 287</code>	<code>thisown (PySpin.IEnumerationT_StreamBufferHandlingModeEnum property), 305</code>
<code>thisown (PySpin.IEnumerationT_RgbTransformLightSource property), 288</code>	<code>thisown (PySpin.IEnumerationT_StreamModeEnum property), 306</code>
<code>thisown (PySpin.IEnumerationT_Scan3dCoordinateReferenceSelectorEnums property), 288</code>	<code>thisown (PySpin.IEnumerationT_StreamTypeEnum property), 306</code>
<code>thisown (PySpin.IEnumerationT_Scan3dCoordinateSelectorEnums property), 289</code>	<code>thisown (PySpin.IEnumerationT_TeledyneGigeVisionFilterDriverStatusEnum property), 308</code>
<code>thisown (PySpin.IEnumerationT_Scan3dCoordinateSystem property), 290</code>	<code>thisown (PySpin.IEnumerationT_TestPatternEnums property), 308</code>
<code>thisown (PySpin.IEnumerationT_Scan3dCoordinateSystemReferenceSelectorEnums property), 290</code>	<code>thisown (PySpin.IEnumerationT_TestPatternGeneratorSelectorEnums property), 309</code>
<code>thisown (PySpin.IEnumerationT_Scan3dCoordinateTransform property), 291</code>	<code>thisown (PySpin.IEnumerationT_TimerSelectorEnums property), 310</code>
<code>thisown (PySpin.IEnumerationT_Scan3dDistanceUnitEnum property), 292</code>	<code>thisown (PySpin.IEnumerationT_TimerStatusEnums property), 310</code>
<code>thisown (PySpin.IEnumerationT_Scan3dOutputModeEnums property), 292</code>	<code>thisown (PySpin.IEnumerationT_TimerTriggerActivationEnums property), 311</code>
<code>thisown (PySpin.IEnumerationT_SensorDigitizationTapsEnum property), 293</code>	<code>thisown (PySpin.IEnumerationT_TimerTriggerSourceEnums property), 312</code>
<code>thisown (PySpin.IEnumerationT_SensorShutterModeEnum property), 294</code>	<code>thisown (PySpin.IEnumerationT_TLTypeEnum property), 307</code>
<code>thisown (PySpin.IEnumerationT_SensorTapsEnums property), 294</code>	<code>thisown (PySpin.IEnumerationT_TransferComponentSelectorEnums property), 312</code>
<code>thisown (PySpin.IEnumerationT_SequencerConfigurationMode property), 295</code>	<code>thisown (PySpin.IEnumerationT_TransferControlModeEnums property), 313</code>
<code>thisown (PySpin.IEnumerationT_SequencerConfigurationValid property), 296</code>	<code>thisown (PySpin.IEnumerationT_TransferOperationModeEnums property), 314</code>
<code>thisown (PySpin.IEnumerationT_SequencerModeEnums property), 296</code>	<code>thisown (PySpin.IEnumerationT_TransferQueueModeEnums property), 314</code>
<code>thisown (PySpin.IEnumerationT_SequencerSetValidEnum property), 297</code>	<code>thisown (PySpin.IEnumerationT_TransferSelectorEnums property), 315</code>
<code>thisown (PySpin.IEnumerationT_SequencerTriggerActivation property), 298</code>	<code>thisown (PySpin.IEnumerationT_TransferStatusSelectorEnums property), 316</code>
<code>thisown (PySpin.IEnumerationT_SequencerTriggerSourceEnum property), 298</code>	<code>thisown (PySpin.IEnumerationT_TransferTriggerActivationEnums property), 316</code>
<code>thisown (PySpin.IEnumerationT_SerialPortBaudRateEnums property), 299</code>	<code>thisown (PySpin.IEnumerationT_TransferTriggerModeEnums property), 317</code>
<code>thisown (PySpin.IEnumerationT_SerialPortParityEnums property), 300</code>	<code>thisown (PySpin.IEnumerationT_TransferTriggerSelectorEnums property), 318</code>
<code>thisown (PySpin.IEnumerationT_SerialPortSelectorEnums property), 300</code>	<code>thisown (PySpin.IEnumerationT_TransferTriggerSourceEnums property), 318</code>
<code>thisown (PySpin.IEnumerationT_SerialPortSourceEnums property), 301</code>	<code>thisown (PySpin.IEnumerationT_TriggerActivationEnums property), 319</code>
<code>thisown (PySpin.IEnumerationT_SerialPortStopBitsEnum property), 302</code>	<code>thisown (PySpin.IEnumerationT_TriggerModeEnums property), 320</code>
<code>thisown (PySpin.IEnumerationT_SoftwareSignalSelectorEnum property), 302</code>	<code>thisown (PySpin.IEnumerationT_TriggerOverlapEnums property), 320</code>
<code>thisown (PySpin.IEnumerationT_SourceSelectorEnums property), 303</code>	<code>thisown (PySpin.IEnumerationT_TriggerSelectorEnums property), 321</code>
<code>thisown (PySpin.IEnumerationT_StereoResolutionEnums property), 304</code>	<code>thisown (PySpin.IEnumerationT_TriggerSourceEnums property), 322</code>

- `thisown (PySpin.IEnumerationT_U3VCurrentSpeedEnums property)`, 322
- `thisown (PySpin.IEnumerationT_UserOutputSelectorEnums property)`, 323
- `thisown (PySpin.IEnumerationT_UserSetDefaultEnums property)`, 324
- `thisown (PySpin.IEnumerationT_UserSetSelectorEnums property)`, 324
- `thisown (PySpin.IEnumerationT_WhiteClipSelectorEnums property)`, 325
- `thisown (PySpin.IEnumReference property)`, 181
- `thisown (PySpin.IFloat property)`, 326
- `thisown (PySpin.IImage property)`, 330
- `thisown (PySpin.IImageEventHandler property)`, 330
- `thisown (PySpin.IImageList property)`, 331
- `thisown (PySpin.IImageListEventHandler property)`, 331
- `thisown (PySpin.IImageProcessor property)`, 332
- `thisown (PySpin.IInteger property)`, 333
- `thisown (PySpin.IInterface property)`, 67, 334
- `thisown (PySpin.IInterfaceArrivalEventHandler property)`, 334
- `thisown (PySpin.IInterfaceEventHandler property)`, 335
- `thisown (PySpin.IInterfaceList property)`, 335
- `thisown (PySpin.IInterfaceRemovalEventHandler property)`, 335
- `thisown (PySpin.ILoggingEventHandler property)`, 335
- `thisown (PySpin.Image property)`, 55, 353
- `thisown (PySpin.ImageEventHandler property)`, 6, 354
- `thisown (PySpin.ImageList property)`, 56, 355
- `thisown (PySpin.ImageListEventHandler property)`, 7, 355
- `thisown (PySpin.ImagePixel property)`, 355
- `thisown (PySpin.ImageProcessor property)`, 57, 357
- `thisown (PySpin.ImagePtr property)`, 58, 357
- `thisown (PySpin.ImageUtility property)`, 59, 358
- `thisown (PySpin.ImageUtilityCCM property)`, 60, 359
- `thisown (PySpin.ImageUtilityHeatmap property)`, 61, 361
- `thisown (PySpin.ImageUtilityPolarization property)`, 64, 365
- `thisown (PySpin.ImageUtilityStereo property)`, 66, 369
- `thisown (PySpin.InferenceBoundingBox property)`, 374
- `thisown (PySpin.InferenceBoundingBoxResult property)`, 375
- `thisown (PySpin.InferenceBoxCircle property)`, 375
- `thisown (PySpin.InferenceBoxRect property)`, 375
- `thisown (PySpin.InferenceBoxRotatedRect property)`, 375
- `thisown (PySpin.INode property)`, 337
- `thisown (PySpin.INodeMap property)`, 338
- `thisown (PySpin.INodeMapDyn property)`, 339
- `thisown (PySpin.int64_autovector_t property)`, 422
- `thisown (PySpin.IntegerNode property)`, 378
- `thisown (PySpin.InterfaceArrivalEventHandler property)`, 7, 378
- `thisown (PySpin.InterfaceEventHandler property)`, 7, 378
- `thisown (PySpin.InterfaceList property)`, 68, 379
- `thisown (PySpin.InterfacePtr property)`, 68, 380
- `thisown (PySpin.InterfaceRemovalEventHandler property)`, 7, 380
- `thisown (PySpin.IntRegNode property)`, 376
- `thisown (PySpin.IPersistScript property)`, 340
- `thisown (PySpin.IPointCloud property)`, 340
- `thisown (PySpin.IReference property)`, 340
- `thisown (PySpin.IRegister property)`, 341
- `thisown (PySpin.ISelector property)`, 342
- `thisown (PySpin.ISelectorDigit property)`, 342
- `thisown (PySpin.IString property)`, 342
- `thisown (PySpin.ISystem property)`, 344
- `thisown (PySpin.ISystemEventHandler property)`, 344
- `thisown (PySpin.IValue property)`, 344
- `thisown (PySpin.JPEGOption property)`, 385
- `thisown (PySpin.JPG2Option property)`, 385
- `thisown (PySpin.LibraryVersion property)`, 385
- `thisown (PySpin.LoggingEventData property)`, 386
- `thisown (PySpin.LoggingEventDataPtr property)`, 8, 386
- `thisown (PySpin.LoggingEventHandler property)`, 8, 387
- `thisown (PySpin.MJPGOption property)`, 387
- `thisown (PySpin.Node property)`, 392
- `thisown (PySpin.node_vector property)`, 424
- `thisown (PySpin.NodeCallback property)`, 392
- `thisown (PySpin.NodeMap property)`, 397
- `thisown (PySpin.PGMOption property)`, 397
- `thisown (PySpin.PNGOption property)`, 397
- `thisown (PySpin.PointCloud property)`, 69, 398
- `thisown (PySpin.PointCloudParameters property)`, 399
- `thisown (PySpin.PPMOption property)`, 397
- `thisown (PySpin.RegisterNode property)`, 400
- `thisown (PySpin.SIOption property)`, 400
- `thisown (PySpin.SpinVideo property)`, 70, 403
- `thisown (PySpin.Stereo3DPoint property)`, 403
- `thisown (PySpin.StereoCameraParameters property)`, 404
- `thisown (PySpin.StringNode property)`, 405
- `thisown (PySpin.StringRegNode property)`, 405
- `thisown (PySpin.System property)`, 75, 409
- `thisown (PySpin.SystemEventHandler property)`, 8, 409
- `thisown (PySpin.SystemPtr property)`, 75, 409
- `thisown (PySpin.TIFFOption property)`, 410
- `thisown (PySpin.TransportLayerDevice property)`, 78, 412
- `thisown (PySpin.TransportLayerInterface property)`, 80, 414
- `thisown (PySpin.TransportLayerStream property)`, 82, 415
- `thisown (PySpin.TransportLayerSystem property)`, 416

- thisown (*PySpin.value_vector* property), 425
 thisown (*PySpin.ValueNode* property), 418
 thisown (*PySpin.Version_t* property), 418
 ThrowBadAlloc() (*in module PySpin*), 410
 TIFFOption (*class in PySpin*), 410
 TimerDelay (*PySpin.Camera* property), 32, 139
 TimerDuration (*PySpin.Camera* property), 32, 139
 TimerReset (*PySpin.Camera* property), 32, 139
 TimerSelector (*PySpin.Camera* property), 32, 139
 TimerStatus (*PySpin.Camera* property), 32, 139
 TimerTriggerActivation (*PySpin.Camera* property), 32, 139
 TimerTriggerSource (*PySpin.Camera* property), 32, 140
 TimerValue (*PySpin.Camera* property), 32, 140
 Timestamp (*PySpin.Camera* property), 32, 140
 TimestampIncrement (*PySpin.Camera* property), 32, 140
 TimestampLatch (*PySpin.Camera* property), 32, 140
 TimestampLatchValue (*PySpin.Camera* property), 32, 140
 TimestampReset (*PySpin.Camera* property), 32, 140
 TLDevice (*PySpin.ICameraBase* property), 176
 TLDisplayName (*PySpin.TransportLayerSystem* property), 416
 TLFileName (*PySpin.TransportLayerSystem* property), 416
 TLID (*PySpin.TransportLayerSystem* property), 416
 TLInterface (*PySpin.IInterface* property), 67, 334
 TLModelName (*PySpin.TransportLayerSystem* property), 416
 TLParamsLocked (*PySpin.Camera* property), 32, 139
 TLPath (*PySpin.TransportLayerSystem* property), 416
 TLStream (*PySpin.ICameraBase* property), 176
 TLSystem (*PySpin.ISystem* property), 343
 TLType (*PySpin.TransportLayerSystem* property), 416
 TLVendorName (*PySpin.TransportLayerSystem* property), 416
 TLVersion (*PySpin.TransportLayerSystem* property), 416
 Tokenize() (*in module PySpin*), 410
 topLeftXCoord (*PySpin.InferenceBoxRect* property), 375
 topLeftXCoord (*PySpin.InferenceBoxRotatedRect* property), 375
 topLeftYCoord (*PySpin.InferenceBoxRect* property), 375
 topLeftYCoord (*PySpin.InferenceBoxRotatedRect* property), 375
 ToString() (*PySpin.CBooleanPtr* method), 87
 ToString() (*PySpin.CCategoryPtr* method), 90
 ToString() (*PySpin.CCommandPtr* method), 92
 ToString() (*PySpin.CEnumEntryPtr* method), 96
 ToString() (*PySpin.CEnumerationPtr* method), 99
 ToString() (*PySpin.CIntegerPtr* method), 103
 ToString() (*PySpin.CRegisterPtr* method), 111
 ToString() (*PySpin.CSelectorSet* method), 112
 ToString() (*PySpin.CStringPtr* method), 115
 ToString() (*PySpin.CValuePtr* method), 117
 ToString() (*PySpin.EAccessModeClass* static method), 156
 ToString() (*PySpin.ECachingModeClass* static method), 157
 ToString() (*PySpin.EDisplayNotationClass* static method), 158
 ToString() (*PySpin.EEndianessClass* static method), 159
 ToString() (*PySpin.ESchemaVersionClass* static method), 159
 ToString() (*PySpin.EInputDirectionClass* static method), 160
 ToString() (*PySpin.ENamespaceClass* static method), 161
 ToString() (*PySpin.ERepresentationClass* static method), 161
 ToString() (*PySpin.ESignClass* static method), 162
 ToString() (*PySpin.ESlopeClass* static method), 163
 ToString() (*PySpin.EStandardNameSpaceClass* static method), 164
 ToString() (*PySpin.EVisibilityClass* static method), 164
 ToString() (*PySpin.EYesNoClass* static method), 165
 ToString() (*PySpin.ISelectorDigit* method), 342
 ToString() (*PySpin.IValue* method), 344
 ToString() (*PySpin.ValueNode* method), 418
 TotalDisparity (*PySpin.Camera* property), 32, 140
 TransferAbort (*PySpin.Camera* property), 32, 140
 TransferBlockCount (*PySpin.Camera* property), 33, 140
 TransferBurstCount (*PySpin.Camera* property), 33, 140
 TransferComponentSelector (*PySpin.Camera* property), 33, 140
 TransferControlMode (*PySpin.Camera* property), 33, 140
 TransferOperationMode (*PySpin.Camera* property), 33, 140
 TransferPause (*PySpin.Camera* property), 33, 140
 TransferQueueCurrentBlockCount (*PySpin.Camera* property), 33, 140
 TransferQueueMaxBlockCount (*PySpin.Camera* property), 33, 140
 TransferQueueMode (*PySpin.Camera* property), 33, 140
 TransferQueueOverflowCount (*PySpin.Camera* property), 33, 140
 TransferResume (*PySpin.Camera* property), 33, 140
 TransferSelector (*PySpin.Camera* property), 33, 140

- TransferStart (*PySpin.Camera* property), 33, 140
 TransferStatus (*PySpin.Camera* property), 33, 140
 TransferStatusSelector (*PySpin.Camera* property), 33, 140
 TransferStop (*PySpin.Camera* property), 33, 140
 TransferStreamChannel (*PySpin.Camera* property), 33, 140
 TransferTriggerActivation (*PySpin.Camera* property), 33, 140
 TransferTriggerMode (*PySpin.Camera* property), 33, 140
 TransferTriggerSelector (*PySpin.Camera* property), 33, 140
 TransferTriggerSource (*PySpin.Camera* property), 33, 140
 TransmissionDelay (*PySpin.Camera* property), 33, 140
 TransmissionDelayAverage (*PySpin.Camera* property), 33, 141
 TransmissionDelayMax (*PySpin.Camera* property), 33, 141
 TransportLayerDevice (class in *PySpin*), 77, 410
 TransportLayerInterface (class in *PySpin*), 79, 412
 TransportLayerStream (class in *PySpin*), 81, 414
 TransportLayerSystem (class in *PySpin*), 415
 TriggerActivation (*PySpin.Camera* property), 33, 141
 TriggerDelay (*PySpin.Camera* property), 33, 141
 TriggerDivider (*PySpin.Camera* property), 33, 141
 TriggerEventTest (*PySpin.Camera* property), 33, 141
 TriggerMode (*PySpin.Camera* property), 33, 141
 TriggerMultiplier (*PySpin.Camera* property), 33, 141
 TriggerOverlap (*PySpin.Camera* property), 33, 141
 TriggerSelector (*PySpin.Camera* property), 34, 141
 TriggerSoftware (*PySpin.Camera* property), 34, 141
 TriggerSource (*PySpin.Camera* property), 34, 141
 Type (*PySpin.CCMSettings* property), 88
 type (*PySpin.LibraryVersion* property), 385
 TypeToString() (*PySpin.ImageUtilityCCM* static method), 60, 359
- ## U
- u (*PySpin.ImagePixel* property), 355
 U3VAccessPrivilege (*PySpin.Camera* property), 34, 141
 U3VCPCapability (*PySpin.Camera* property), 34, 141
 U3VCPEIRMAvailable (*PySpin.Camera* property), 34, 141
 U3VCPIIDC2Available (*PySpin.Camera* property), 34, 141
 U3VCPSIRMAvailable (*PySpin.Camera* property), 34, 141
 U3VCurrentSpeed (*PySpin.Camera* property), 34, 141
 U3VMaxAcknowledgeTransferLength (*PySpin.Camera* property), 34, 141
 U3VMaxCommandTransferLength (*PySpin.Camera* property), 34, 141
 U3VMaxDeviceResponseTime (*PySpin.Camera* property), 34, 141
 U3VMessageChannelID (*PySpin.Camera* property), 34, 141
 U3VNumberOfStreamChannels (*PySpin.Camera* property), 34, 141
 U3VVersionMajor (*PySpin.Camera* property), 34, 141
 U3VVersionMinor (*PySpin.Camera* property), 34, 141
 UniquenessRatio (*PySpin.Camera* property), 34, 141
 UnregisterAllLoggingEventHandlers() (*PySpin.ISystem* method), 343
 UnregisterAllLoggingEventHandlers() (*PySpin.System* method), 74, 408
 UnregisterEventHandler() (*PySpin.CameraBase* method), 39, 146
 UnregisterEventHandler() (*PySpin.ICameraBase* method), 176
 UnregisterEventHandler() (*PySpin.IInterface* method), 67, 334
 UnregisterEventHandler() (*PySpin.ISystem* method), 343
 UnregisterEventHandler() (*PySpin.System* method), 74, 408
 UnregisterLoggingEventHandler() (*PySpin.ISystem* method), 344
 UnregisterLoggingEventHandler() (*PySpin.System* method), 74, 408
 UpdateCameras() (*PySpin.IInterface* method), 67, 334
 UpdateCameras() (*PySpin.ISystem* method), 344
 UpdateCameras() (*PySpin.System* method), 74, 409
 UpdateFirmware() (in module *PySpin*), 416
 UpdateFirmwareConsole() (in module *PySpin*), 416
 UpdateFirmwareGUI() (in module *PySpin*), 416
 UpdateInterfaceList() (*PySpin.ISystem* method), 344
 UpdateInterfaceList() (*PySpin.System* method), 75, 409
 UrlDecode() (in module *PySpin*), 416
 UrlEncode() (in module *PySpin*), 417
 useMP4 (*PySpin.H264Option* property), 173
 UserOutputSelector (*PySpin.Camera* property), 34, 141
 UserOutputValue (*PySpin.Camera* property), 34, 141
 UserOutputValueAll (*PySpin.Camera* property), 34, 141
 UserOutputValueAllMask (*PySpin.Camera* property), 34, 141
 UserSetDefault (*PySpin.Camera* property), 34, 141
 UserSetFeatureEnable (*PySpin.Camera* property), 34, 142

UserSetLoad (*PySpin.Camera* property), 34, 142
UserSetSave (*PySpin.Camera* property), 34, 142
UserSetSelector (*PySpin.Camera* property), 34, 142

V

v (*PySpin.ImagePixel* property), 355
V3_3Enable (*PySpin.Camera* property), 34, 142
value_vector (class in *PySpin*), 424
ValueNode (class in *PySpin*), 417
Version_t (class in *PySpin*), 418

W

WhiteClip (*PySpin.Camera* property), 34, 142
WhiteClipSelector (*PySpin.Camera* property), 34, 142
width (*PySpin.AVIOption* property), 83
Width (*PySpin.Camera* property), 34, 142
width (*PySpin.H264Option* property), 173
width (*PySpin.MJPGOption* property), 387
WidthMax (*PySpin.Camera* property), 34, 142
WindowSizeH (*PySpin.Camera* property), 35, 142
WindowSizeW (*PySpin.Camera* property), 35, 142
WritePort() (*PySpin.ICameraBase* method), 176

X

x (*PySpin.Stereo3DPoint* property), 403

Y

y (*PySpin.Stereo3DPoint* property), 403

Z

z (*PySpin.Stereo3DPoint* property), 403