Imperx Camera Python SDK

1.5.0.54

Generated by Doxygen 1.8.13

Contents

Chapter 1

Imperx Camera Python SDK

1.1 General Information

The Imperx Camera Python SDK is designed to provide software developers with API methods for ease of integrating Imperx cameras into their software application, created with Python programming language The Python API mainly repeats C++ API structure and functionality. It includes three namespaces: IpxCamPy, IpxGuiPy and IpxGenParam↩Py. The API implemented in two libraries: IpxCameraApiPy (implementing IpxCamPy and IpxGenParamPy) and Ipx↩CameraGuiApiPy (implementing IpxGuiPy).

The IpxCamPy namespace provides the scope to the API of GenICam GenTL transport layer to acquire images with an Imperx Camera. The IpxGenParamPy namespace provides the scope to the API to control the GenICam camera parameters, like image Width, Height, Pixel Format, Gain, Exposure, Trigger settings, etc. The IpxGuiPy namespace provides the scope for the user interface functionality, like windows and panels.

1.2 IpxCamPy namespace

The IpxCam namespace consist of several main classes that represent the GenTL modules and GUI objects. The main classes are

- IpxCamPy.PyIpxSystem - The System class is the entry point to the GenTL Producer software driver.
- IpxCamPy.PyDeviceInterface - The Interface class provides method to represents an individual physical interface, like GigE or USB3
- IpxCamPy.PyDevice - The Device class provides methods to enable the communication with the camera device and enumerate/instantiate the video data streams.
- IpxCamPy.PyDataStream - The Stream class purpose is to access the image buffer data acquirement from the Acquisition engine.
- IpxCamPy.PyBuffer - The Buffer class contains the methods to access the image data and parameters of the acquired image buffer.

Example of GenTL System Hierarchy

1.3 IpxGenParamPy namespace

The IpxGenParamPy namespace consist of the following main classes to access the GenICam parameters features. The main classes are

- IpxGenParamPy.PyGenParam - General class for accessing the GenICam feature node of the Camera parameters.
- IpxGenParamPy.PyGenParamBoolean - Class representing the Boolean GenICam camera parameter.
- IpxGenParamPy.PyGenParamCommand - Class representing the Command GenICam camera parameter.
- IpxGenParamPy.PyGenParamEnum - Class representing the Enumeration GenICam camera parameter.
- IpxGenParamPy.PyGenParamFloat - Class representing the Float GenICam camera parameter.
- IpxGenParamPy.PyGenParamInt - Class representing the Integer GenICam camera parameter.
- IpxGenParamPy.PyGenParamString - Class representing the String GenICam camera parameter.

1.4 IpxGuiPy namespace

The IpxCam namespace consist of several main classes that represent the GenTL modules and GUI objects. The main classes are

- IpxGuiPy.PyGenParamView - The class representing node tree view of GenICam parameters panel
- IpxGuiPy.PyIpxSystemGui - The class extending the IpxCamPy.PyIpxSystem to provide the GUI functionality
- IpxGuiPy.PyIpxCameraApiGui - The class extending the IpxCamPy.PyIpxCameraApi to provide the GUI functionality

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

Namespace Documentation

5.1 IpxCamPy Namespace Reference

Python Wrapper for IpxCameraApi.dll.

Classes

- class PyBuffer

  The PyBuffer class represents the buffer module in the GenTL module hierarchy.
- class PyDataStream

  The PyDataStream class represents the data stream module in the GenTL module hierarchy.
- class PyDevice

  The Device class represents the device module in the GenTL module hierarchy.
- class PyDeviceInfo

  DeviceInfo class provides the information about the device.
- class PyDeviceInterface

  The PyDeviceInterface class represents a interface module in the GenTL module hierarchy.
- class PyImage
- class PyIpxSystem

  The PyIpxSystem class represents an abstraction of the System module of the GenTL module hierarchy.

5.1.1 Detailed Description

Python Wrapper for IpxCameraApi.dll.

A namespace providing scope to the GenICam GenTL transport layer interface to acquire images with an Imperx Camera.

Python Wrapper for IpxCameraApi.dll. Python 3 API providing scope to the GenICam GenTL transport layer interface to acquire images and control the Imperx camera.

IpxCam namespace includes classes that represent the base GenTL transport layer modules: System, Interface, Device, Stream, Buffer. These modules can be used to enumerate the interfaces in the system, enumerate the cameras, connected to each interface, connect to necessary camera, allocate the memory buffers for images, and run the video acquisition.

5.2 IpxGenParamPy Namespace Reference

A namespace provides the scope to the API to access the GenICam parameters.

Classes

- class PyGenParam

  General class for GenICam parameter.
- class PyGenParamBoolean

  PyGenParamBoolean class represents the Boolean GenICam camera parameter.
- class PyGenParamCategory

  *A class containing methods that the user can access the categories GenICam features. It will access the node object's of an ICategory interface. Each feature of a device will be placed in a **Category**. The Category feature is used to present the user with a group of features for the named category.*
- class PyGenParamCommand

  PyGenParamCommand class represents Command GenICam camera parameter.
- class PyGenParamEnum

  PyGenParamEnum class represents Enumeration GenICam camera parameter.
- class PyGenParamEnumEntry

  PyGenParamEnumEntry class represents the entry of GenICam Enum parameter.
- class PyGenParamFloat

  PyGenParamFloat class represents Float GenICam camera parameter.
- class PyGenParamInt

  PyGenParamInt class represents Integer GenICam camera parameter.
- class PyGenParamNode

  PyGenParamNode class represents GenICam INode class.
- class PyGenParamNodeMap
- class PyGenParams

  An PyGenParams class contains methods to access all GenICam camera parameters.
- class PyGenParamString

  PyGenParamString class represents String GenICam camera parameter.

5.2.1 Detailed Description

A namespace provides the scope to the API to access the GenICam parameters.

The IpxGenParamPy namespace provides the scope to the API to control the GenICam camera parameters of types: Boolean, Enumeration, String, Float, Integer, Commands and Categories. Such parameters may include image Width, Height, Pixel Format, Gain, Exposure, Trigger, I/O settings, etc. Parameters are described in camera GenICam XML file, and documented in appropriate camera user's manual.

5.3 IpxGuiPy Namespace Reference

Classes

- class PyGenParamView
- class PyIpxCameraApiGui

  Extension of IpxCamPy.PyIpxCameraApi class to provide the GUI functionality.
- class PyIpxSystemGui

  Extension of IpxCamPy.PyIpxSystem class to provide the GUI functionality.

5.3.1 Detailed Description

The IpxGuiPy namespace is a declarative region that provides a scope to the Imperx Camera GUI API classes and functions.

Chapter 6

Class Documentation

6.1 IpxCamPy::PyBuffer Class Reference

The PyBuffer class represents the buffer module in the GenTL module hierarchy.

`#include <IpxCameraApiPy.h>`

Inherits PythonExtension< PyBuffer >.

Public Member Functions

- Py::Object GetWidth ()
  Returns the image width.
- Py::Object GetHeight ()
  Returns the image height.
- Py::Object GetBufferPtr ()
  Returns the image data array.
- Py::Object GetImage ()
  This method returns the Image object, associated with the Buffer object.
- Py::Object IsIncomplete ()
  This method returns a flag indicating if the buffer data has been fully transferred or incompleted.
- Py::Object GetPixelFormat ()
  This method returns the pixel format of the buffer object.
- Py::Object GetFrameID ()
  This method returns the identificator of the image stream block of the buffer object.
- Py::Object GetXOffset ()
  Returns the horizontal offset of the image data in the buffer.
- Py::Object GetYOffset ()
  Returns the vertical offset of the image data in the buffer.
- Py::Object GetXPadding ()
  This method returns the number of extra bytes padded in the horizontal direction.
- Py::Object GetYPadding ()
  This method returns the number of extra bytes padded in the vertical direction.
- Py::Object GetTimestamp ()
  This method returns the timestamp of the acquired buffer.

6.1.1 Detailed Description

The PyBuffer class represents the buffer module in the GenTL module hierarchy.

The Buffer class contains the methods that can be used to get the pointer to the acquired image data memory and / or retrieve the information about the received image data such as timestamp, image size, pixel format, etc

IpxCamPy.PyBuffer Python class is wrapper around C++ class **IpxCam::Buffer**

6.1.2 Member Function Documentation

6.1.2.1 GetWidth()

```
Py::Object IpxCamPy::PyBuffer::GetWidth ( )
```

Returns the image width.

This method returns the image width of the buffer data in number of pixels. Usually the return value equals to **Width** GenICam parameter value

Returns

Returns the image width. Value type: Integer

6.1.2.2 GetHeight()

```
Py::Object IpxCamPy::PyBuffer::GetHeight ( )
```

Returns the image height.

This method returns the image height of the buffer data in number of lines. Usually the return value equals to **Height** GenICam parameter value

Returns

Returns the image height. Value type: Integer

6.1.2.3 GetBufferPtr()

```
Py::Object IpxCamPy::PyBuffer::GetBufferPtr ( )
```

Returns the image data array.

This method returns the array of the image data of the Buffer object.

Returns

Returns the image data array. Value type: bytearray

6.1.2.4 GetImage()

```
Py::Object IpxCamPy::PyBuffer::GetImage ( )
```

This method returns the Image object, associated with the Buffer object.

Returns

Returns the Image object, associated with the Buffer object. Value type: IpxCamPy.PyImage

6.1.2.5 IsIncomplete()

```
Py::Object IpxCamPy::PyBuffer::IsIncomplete ( )
```

This method returns a flag indicating if the buffer data has been fully transferred or incompleted.

Returns

Returns True, if buffer transfer was incompleted, False, if transfer was successful. Value type: Boolean

6.1.2.6 GetPixelFormat()

```
Py::Object IpxCamPy::PyBuffer::GetPixelFormat ( )
```

This method returns the pixel format of the buffer object.

Returns

Returns the pixel format of the image in the buffer object. Value type: Integer This value equals to **PixeFormat** GenICam parameter

6.1.2.7 GetFrameID()

```
Py::Object IpxCamPy::PyBuffer::GetFrameID ( )
```

This method returns the identificator of the image stream block of the buffer object.

Returns

Returns the identificator of the image stream block of the buffer object. Value type: Integer

6.1.2.8 GetXOffset()

```
Py::Object IpxCamPy::PyBuffer::GetXOffset ( )
```

Returns the horizontal offset of the image data in the buffer.

This method returns the horizontal offset of the image data in the buffer in number of pixels from the image origin. Usually the return value equals to **OffsetX** GenICam parameter value

Returns

Returns the horizontal offset in number of pixels. Value type: Integer

6.1.2.9 GetYOffset()

```
Py::Object IpxCamPy::PyBuffer::GetYOffset ( )
```

Returns the vertical offset of the image data in the buffer.

This method returns the vertical offset of the image data in the buffer in number of lines from the image origin. Usually the return value equals to **OffsetY** GenICam parameter value

Returns

Returns the vertical offset of the data in the buffer in number of lines from the image origin. Value type: Integer

6.1.2.10 GetXPadding()

```
Py::Object IpxCamPy::PyBuffer::GetXPadding ( )
```

This method returns the number of extra bytes padded in the horizontal direction.

Returns

Returns the XPadding of the data in the buffer in number of bytes. Value type: Integer

6.1.2.11 GetYPadding()

```
Py::Object IpxCamPy::PyBuffer::GetYPadding ( )
```

This method returns the number of extra bytes padded in the vertical direction.

Returns

Returns the YPadding of the data in the buffer in number of bytes. Value type: Integer

6.1.2.12 GetTimestamp()

```
Py::Object IpxCamPy::PyBuffer::GetTimestamp ( )
```

This method returns the timestamp of the acquired buffer.

This method returns the timestamp of the acquired buffer. Imperx USB3 and GEV cameras have 10ns timestamp granularity. GEV cameras timestamp clock frequency can be obtained from **GevTimestampTickFrequency** GenICam parameter

Returns

Returns the timestamp of the acquired buffer. Value type: Integer

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

- IpxCameraApiPy.h

6.2 IpxCamPy::PyDataStream Class Reference

The PyDataStream class represents the data stream module in the GenTL module hierarchy.

`#include <IpxCameraApiPy.h>`

Inherits PythonExtension< PyDataStream >.

Public Member Functions

- Py::Object GetBufferSize ()
  This method returns the buffer size of the data stream object.
- Py::Object GetBufferAlignment ()
  This method returns the alignment size of the stream object.
- Py::Object GetMinNumBuffers ()
  This method returns the minimum number of buffers.
- Py::Object StartAcquisition ()
  Starts the Acquisition Engine.
- Py::Object StopAcquisition (const Py::Tuple &args)
  Stops the stream's acquisition engine.
- Py::Object CreateBuffer (const Py::Tuple &args)
  Creates the buffer in the data stream object.
- Py::Object GetBufferQueueSize ()
  Retrieves the Buffer Queue size.
- Py::Object GetBuffer (const Py::Tuple &args)
  This method retrieves the buffer object.
- Py::Object QueueBuffer (const Py::Tuple &args)
  This method queues the specified buffers.
- Py::Object FlushBuffers (const Py::Tuple &args)
  This method flushes the buffers of the data stream object.
- Py::Object RevokeBuffer (const Py::Tuple &args)
  Revokes any announced buffer.
- Py::Object AllocBufferQueue (const Py::Tuple &args)
  Allocates the Buffer Queue.
- Py::Object ReleaseBufferQueue ()
  Releases the Buffer Queue.
- Py::Object Release ()
- Py::Object CancelBuffer ()
  Cancels the buffer events waiting.

6.2.1 Detailed Description

The PyDataStream class represents the data stream module in the GenTL module hierarchy.

This data stream class provides buffer methods. This data stream class purpose is to access the buffer data acquirement from the Acquisition engine.
IpxCamPy::PyDataStream Python class is wrapper around C++ class **IpxCam::Stream**

6.2.2 Member Function Documentation

6.2.2.1 GetBufferSize()

```
Py::Object IpxCamPy::PyDataStream::GetBufferSize ( )
```

This method returns the buffer size of the data stream object.

Returns

Returns the buffer size. Value type: Integer

6.2.2.2 GetBufferAlignment()

```
Py::Object IpxCamPy::PyDataStream::GetBufferAlignment ( )
```

This method returns the alignment size of the stream object.

Returns

Returns the alignment size in bytes of the buffer passed. Value type: Integer

6.2.2.3 GetMinNumBuffers()

```
Py::Object IpxCamPy::PyDataStream::GetMinNumBuffers ( )
```

This method returns the minimum number of buffers.

Returns

Returns the minimum number of buffers to announce. Value type: Integer

6.2.2.4 StartAcquisition()

```
Py::Object IpxCamPy::PyDataStream::StartAcquisition ( )
```

Starts the Acquisition Engine.

This method starts the acquisition engine of the stream to acquire the unlimited number of image data frames to the queued buffers

Returns

True, if acquisition successfully started, False otherwise. Value type: Boolean

6.2.2.5 StopAcquisition()

```
Py::Object IpxCamPy::PyDataStream::StopAcquisition (
            const Py::Tuple & args )
```

Stops the stream's acquisition engine.

This method stops the acquisition engine of the stream and terminates the image data frames acquisition

Parameters

in	*args[0]*	- Value type: Integer. Flags: • ACQ_STOP_FLAGS_DEFAULT=0, Stop the acquisition engine when the currently running tasks like filling a buffer are completed (default behavior). • ACQ_STOP_FLAGS_KILL=1, Stop the acquisition engine immediately and leave buffers currently being filled in the Input Buffer Pool.

Returns

True, if acquisition successfully stopped, False otherwise. Value type: Boolean

6.2.2.6 CreateBuffer()

```
Py::Object IpxCamPy::PyDataStream::CreateBuffer (
            const Py::Tuple & args )
```

Creates the buffer in the data stream object.

This method allocates the memory for a buffer and announces this buffer to the data stream

Parameters

in	*args[0]*	Buffer size in bytes. Value type: Integer

Returns

Returns IpxCamPy.PyBuffer object created

6.2.2.7 GetBufferQueueSize()

```
Py::Object IpxCamPy::PyDataStream::GetBufferQueueSize ( )
```

Retrieves the Buffer Queue size.

This functions returns the buffer queue size of the data stream object.

Returns

Returns the Buffer Queue size. Value type: Integer

6.2.2.8 GetBuffer()

```
Py::Object IpxCamPy::PyDataStream::GetBuffer (
            const Py::Tuple & args )
```

This method retrieves the buffer object.

Retrieves the next acquired buffer entry from the acquisition engine's queue and returns the acquired Buffer object

Parameters

in	*args[0]*	timeout in milliseconds

Returns

Returns the IpxCamPy.PyBuffer object for acquired buffer

6.2.2.9 QueueBuffer()

```
Py::Object IpxCamPy::PyDataStream::QueueBuffer (
            const Py::Tuple & args )
```

This method queues the specified buffers.

During the acquisition, this method is used to return the specified buffer to the acquisition engine's queue

Parameters

in	*args[0]*	IpxCamPy.PyBuffer object

Returns

True, if buffer was successfully queued, False otherwise. Value type: Boolean

6.2.2.10 FlushBuffers()

```
Py::Object IpxCamPy::PyDataStream::FlushBuffers (
            const Py::Tuple & args )
```

This method flushes the buffers of the data stream object.

Performs the specified Flush Operation on the acquisition engine's queue. Operations type is defined in FlushOperations enum.

Parameters

in	*args[0]*	operation FlushOperation. Value type: Integer. Possible values are: • PyDataStream.Flush_OutputDiscard (1) Discards all buffers in the output queue and if necessary remove the entries from the event data queue. • PyDataStream.Flush_AllToInput (2) Puts all buffers in the input pool. Even those in the output queue and discard entries in the event data queue. • PyDataStream.Flush_UnqueuedToInput (3) Puts all buffers that are not in the input pool or the output queue in the input pool. • PyDataStream.Flush_AllDiscard (4) Discards all buffers in the input pool and output queue.

Returns

True, if operation completed OK, False otherwise. Value type: Boolean

6.2.2.11 RevokeBuffer()

```
Py::Object IpxCamPy::PyDataStream::RevokeBuffer (
            const Py::Tuple & args )
```

Revokes any announced buffer.

This method removes the specified announced Buffer from the acquisition engine's queue

Parameters

in	*args[0]*	IpxCamPy.PyBuffer object

Returns

True, if operation completed OK, False otherwise. Value type: Boolean

6.2.2.12 AllocBufferQueue()

```
Py::Object IpxCamPy::PyDataStream::AllocBufferQueue (
            const Py::Tuple & args )
```

Allocates the Buffer Queue.

This method allocates the buffers in the queue of the acquisition engine of the data stream object.

Parameters

in	*args[0]*	number of Buffers to allocate. Value type: Integer.

Returns

True, if operation completed OK, False otherwise. Value type: Boolean

6.2.2.13 ReleaseBufferQueue()

```
Py::Object IpxCamPy::PyDataStream::ReleaseBufferQueue ( )
```

Releases the Buffer Queue.

This method releases the buffer queue of the data stream object.

Returns

True, if operation completed OK, False otherwise. Value type: Boolean

6.2.2.14 Release()

```
Py::Object IpxCamPy::PyDataStream::Release ( )
```

This method releases the instance of the IpxCamPy.PyDataStream object.

6.2.2.15 CancelBuffer()

```
Py::Object IpxCamPy::PyDataStream::CancelBuffer ( )
```

Cancels the buffer events waiting.

This method cancels any previously registered buffer events that have been waiting to be performed.

Returns

True, if operation completed OK, False otherwise. Value type: Boolean

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

- IpxCameraApiPy.h

6.3 IpxCamPy::PyDevice Class Reference

The Device class represents the device module in the GenTL module hierarchy.

```
#include <IpxCameraApiPy.h>
```

Inherits PythonExtension< PyDevice >.

Public Member Functions

- Py::Object GetInfo ()
  This method returns IpxCamPy.PyDeviceInfo object, associated with the PyDevice object.
- Py::Object GetDisplayName ()
  This method returns the user readable display name of the Camera device object.
- Py::Object GetNumStreams ()
  This method retrieves the number of the data streams, provided by the Device.
- Py::Object Release ()
  This method releases the IpxCamPy.PyDevice object.
- Py::Object ReadMem (const Py::Tuple &args)
  This method reads a number of bytes from a given address.
- Py::Object WriteMem (const Py::Tuple &args)
  This method writes a number of bytes at a given address.
- Py::Object GetStreamByIndex (const Py::Tuple &args)

This method retrieves the PyDataStream object by stream index.

- Py::Object GetStreamById (const Py::Tuple &args)

  This method retrieves the PyDataStream object by stream identifier.
- Py::Object SaveConfiguration (const Py::Tuple &args)

  This method saves the camera device configuration to the file.
- Py::Object LoadConfiguration (const Py::Tuple &args)

  This method loads the camera device configuration from the file.
- Py::Object GetCameraParameters ()

  This method returns the camera parameters of the device object.
- Py::Object GetTransportParameters ()

  This method returns the transport parameters of the device object.
- Py::Object RegisterEvent (const Py::Tuple &args)

  This method registers the Device class method as a callback method to be called when event of the specified type occurs.
- Py::Object UnRegisterEvent (const Py::Tuple &args)

  This method unregisters the Device class callback method for the event type.

6.3.1 Detailed Description

The Device class represents the device module in the GenTL module hierarchy.

This Device class provides methods to enable the communication and control of the Imperx device and enumerate/instantiate data stream objects. The methods can be used to enumerate and instantiate the Data Stream module objects. The device must must correspond to the interface transport layer technology. For example,the device could be an Imperx GEV Camera and the transport layer technology would be GEV. The Device class can be used to retrieve data information about the device by returning the pointer to the DeviceInfo class. It can be used to retrieve the pointer to the Stream object and save / load the camera configurations to / from file.

IpxCamPy.PyDevice class is wrapper around C++ class **IpxCam::Device**

6.3.2 Member Function Documentation

6.3.2.1 GetInfo()

```
Py::Object IpxCamPy::PyDevice::GetInfo ( )
```

This method returns IpxCamPy.PyDeviceInfo object, associated with the PyDevice object.

Returns

Returns IpxCamPy.PyDeviceInfo class object

6.3.2.2 GetDisplayName()

```
Py::Object IpxCamPy::PyDevice::GetDisplayName ( )
```

This method returns the user readable display name of the Camera device object.

Returns

Returns the user readable name of the Camera device. Value type: String

6.3.2.3 GetNumStreams()

```
Py::Object IpxCamPy::PyDevice::GetNumStreams ( )
```

This method retrieves the number of the data streams, provided by the Device.

Returns

returns the number of the data streams. Value type: Integer

6.3.2.4 Release()

```
Py::Object IpxCamPy::PyDevice::Release ( )
```

This method releases the IpxCamPy.PyDevice object.

Returns

none

6.3.2.5 ReadMem()

```
Py::Object IpxCamPy::PyDevice::ReadMem (
            const Py::Tuple & args )
```

This method reads a number of bytes from a given address.

Parameters

in	*args[0]*	Byte address to read from. Value type: Integer
in	*args[1]*	Amount of bytes to read from the register map address. Value type: Integer

Returns

Returns Tuple of:

- [0] Boolean value: Success of operation
- [1] Py::Bytes array of readout data

6.3.2.6 WriteMem()

```
Py::Object IpxCamPy::PyDevice::WriteMem (
            const Py::Tuple & args )
```

This method writes a number of bytes at a given address.

Parameters

in	*args[0]*	Byte address to write to
in	*args[1]*	Py::Bytes array

Returns

Returns Tuple of:

- [0] Boolean value: Success of operation
- [1] Integer value: number of bytes written

6.3.2.7 GetStreamByIndex()

```
Py::Object IpxCamPy::PyDevice::GetStreamByIndex (
            const Py::Tuple & args )
```

This method retrieves the PyDataStream object by stream index.

Parameters

in	*args[0]*	stream index. Value type: Integer

Returns

Returns the IpxCamPy.PyDataStream object or None

6.3.2.8 GetStreamById()

```
Py::Object IpxCamPy::PyDevice::GetStreamById (
            const Py::Tuple & args )
```

This method retrieves the PyDataStream object by stream identifier.

Parameters

in	*args[0]*	stream identifier. Value type: String

Returns

Returns the IpxCamPy.PyDataStream object or None

6.3.2.9 SaveConfiguration()

```
Py::Object IpxCamPy::PyDevice::SaveConfiguration (
            const Py::Tuple & args )
```

This method saves the camera device configuration to the file.

Parameters

in	*args[0]*	Configuration file name. Value type: String

Returns

Returns True, if operation completed OK, False otherwise. Value type: Boolean

6.3.2.10 LoadConfiguration()

```
Py::Object IpxCamPy::PyDevice::LoadConfiguration (
            const Py::Tuple & args )
```

This method loads the camera device configuration from the file.

Parameters

in	*args[0]*	Configuration file name. Value type: String

Returns

Returns True, if operation completed OK, False otherwise. Value type: Boolean

6.3.2.11 GetCameraParameters()

```
Py::Object IpxCamPy::PyDevice::GetCameraParameters ( )
```

This method returns the camera parameters of the device object.

Returns

Returns IpxGenParamPy.PyGenParams object

6.3.2.12 GetTransportParameters()

```
Py::Object IpxCamPy::PyDevice::GetTransportParameters ( )
```

This method returns the transport parameters of the device object.

Returns

Returns IpxGenParamPy.PyGenParams object

6.3.2.13 RegisterEvent()

```
Py::Object IpxCamPy::PyDevice::RegisterEvent (
            const Py::Tuple & args )
```

This method registers the Device class method as a callback method to be called when event of the specified type occurs.

Parameters

in	*args[0]*	Event Type. Value type: Integer. It can receive one of the following values: • **GenICamEvent** [1002] - this event occurs, if GenICam event was triggered by the camera • **CameraConnected** [1003] - this event occurs, when camera was connected to the computer • **CameraDisconnected** [1004] - this event occurs, when camera was disconnected from the computer
in	*args[1]*	event handler callback function.

Returns

Returns True, if operation completed OK, False otherwise. Value type: Boolean

6.3.2.14 UnRegisterEvent()

```
Py::Object IpxCamPy::PyDevice::UnRegisterEvent (
            const Py::Tuple & args )
```

This method unregisters the Device class callback method for the event type.

Parameters

in	*args[0]*	Event Type. Value type: Integer. It can receive one of the following values: • **GenICamEvent** [1002] - this event occurs, if GenICam event was triggered by the camera • **CameraConnected** [1003] - this event occurs, when camera was connected to the computer • **CameraDisconnected** [1004] - this event occurs, when camera was disconnected from the computer
in	*args[1]*	event handler callback function.

Returns

Returns True, if operation completed OK, False otherwise. Value type: Boolean

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

- IpxCameraApiPy.h

6.4 IpxCamPy::PyDeviceInfo Class Reference

DeviceInfo class provides the information about the device.

`#include <IpxCameraApiPy.h>`

Inherits PythonExtension< PyDeviceInfo >.

Public Member Functions

- Py::Object GetInterface ()

  This method returns the interface of the device object.
- Py::Object GetModel ()

  This method returns the Camera device model name of the device object.
- Py::Object GetVendor ()

  This method returns the device vendor name of the device object.
- Py::Object GetVersion ()

  This method returns the version of the Camera device.
- Py::Object GetSerialNumber ()

  This method returns the serial number of the Camera device.
- Py::Object GetDisplayName ()

  This method returns the user readable display name of the Camera device object.
- Py::Object GetUserDefinedName ()

  This method returns the user defined name.
- Py::Object GetAccessStatus ()

  Returns the device access status.
- Py::Object GetIPAddress ()

  Returns the IP address of the GEV camera.
- Py::Object GetIPMask ()

  Returns the IP subnet mask of the GEV camera.
- Py::Object GetIPGateway ()

  Returns the IP gateway of GEV camera.
- Py::Object ForceIP (const Py::Tuple &args)

  Set the IP address to GEV camera.

6.4.1 Detailed Description

DeviceInfo class provides the information about the device.

The device info class can be used to retrieve information about the device.DeviceInfo class provides the information about the camera device.

The DeviceInfo class can be used to retrieve the information about the device, and to create the IpxCamPy.PyDevice object by PyIpxCameraApi.PyIpxCreateDevice() method call IpxCamPy.PyDeviceInfo Python class is wrapper around C++ class **IpxCam::DeviceInfo**

6.4.2 Member Function Documentation

6.4.2.1 GetInterface()

```
Py::Object IpxCamPy::PyDeviceInfo::GetInterface ( )
```

This method returns the interface of the device object.

Returns the IpxCamPy.PyDeviceInterface object for the camera device, associated with the PyDeviceInfo object

Returns

Returns the IpxCamPy.PyDeviceInterface class object

6.4.2.2 GetModel()

```
Py::Object IpxCamPy::PyDeviceInfo::GetModel ( )
```

This method returns the Camera device model name of the device object.

Returns

Returns the Camera device model name. Value type: String

6.4.2.3 GetVendor()

```
Py::Object IpxCamPy::PyDeviceInfo::GetVendor ( )
```

This method returns the device vendor name of the device object.

Returns

Returns the Imperx Camera device vendor name. Value type: String

6.4.2.4 GetVersion()

```
Py::Object IpxCamPy::PyDeviceInfo::GetVersion ( )
```

This method returns the version of the Camera device.

Returns

Returns the Camera Device version. Value type: String

6.4.2.5 GetSerialNumber()

```
Py::Object IpxCamPy::PyDeviceInfo::GetSerialNumber ( )
```

This method returns the serial number of the Camera device.

Returns

Returns the serial number of the Camera device. Value type: String

6.4.2.6 GetDisplayName()

```
Py::Object IpxCamPy::PyDeviceInfo::GetDisplayName ( )
```

This method returns the user readable display name of the Camera device object.

Returns

Returns the user readable name of the Camera device. Value type: String

6.4.2.7 GetUserDefinedName()

```
Py::Object IpxCamPy::PyDeviceInfo::GetUserDefinedName ( )
```

This method returns the user defined name.

Returns

Returns the user defined name of the device. Value type: String

6.4.2.8 GetAccessStatus()

```
Py::Object IpxCamPy::PyDeviceInfo::GetAccessStatus ( )
```

Returns the device access status.

This method returns the information about the current access status of the Camera device

Returns

Status Access Code. Value type: Integer. It can receive one of the following values:

- `0` (AccessStatusUnknown) - The current availability of the device is unknown.
- `1` (AccessStatusReadWrite) - The device is available for Read/Write access
- `2` (AccessStatusReadOnly) - The device is available for Read only access
- `3` (AccessStatusNoAccess) - The device is not available either because it is already open or because it is not reachable.
- `1001` (IpSubnetMismatch) - The device is available, but IP address does not match to the host subnet mask.

6.4.2.9 GetIPAddress()

```
Py::Object IpxCamPy::PyDeviceInfo::GetIPAddress ( )
```

Returns the IP address of the GEV camera.

This method returns the IP address of the GEV camera, retrieved from DISCOVERY_ACK packet, received from the camera device

Returns

Returns IP Address string or empty string for non-GEV camera. Value type: String

6.4.2.10 GetIPMask()

```
Py::Object IpxCamPy::PyDeviceInfo::GetIPMask ( )
```

Returns the IP subnet mask of the GEV camera.

This method returns the IP subnet mask of the GEV camera, retrieved from DISCOVERY_ACK packet, received from the camera device

Returns

Returns IP subnet mask string or empty string for non-GEV camera. Value type: String

6.4.2.11 GetIPGateway()

```
Py::Object IpxCamPy::PyDeviceInfo::GetIPGateway ( )
```

Returns the IP gateway of GEV camera.

This method returns the IP gateway of the GEV camera, retrieved from DISCOVERY_ACK packet, received from the camera device

Returns

Returns IP gateway string or empty string for non-GEV camera. Value type: String

6.4.2.12 ForceIP()

```
Py::Object IpxCamPy::PyDeviceInfo::ForceIP (
            const Py::Tuple & args )
```

Set the IP address to GEV camera.

This method sets the specified IP address to the GEV camera, using ForceIP GVCP command

Parameters

in	*args[0]*	IP Address string to set
in	*args[1]*	IP Address subnet mask string
in	*args[2]*	Gateway address string

Returns

Returns True, if operation completed OK, False otherwise. Value type: Boolean

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

- IpxCameraApiPy.h

6.5 IpxCamPy::PyDeviceInterface Class Reference

The PyDeviceInterface class represents a interface module in the GenTL module hierarchy.

```
#include <IpxCameraApiPy.h>
```

Inherits PythonExtension< PyDeviceInterface >.

Public Member Functions

- Py::Object GetDescription ()

  This method returns the description of the interface.
- Py::Object GetFirstDeviceInfo ()

  This method retrieves the DeviceInfo object for the first device available on this Interface.
- Py::Object ReEnumerateDevices ()

  This method re-enumerates the devices.
- Py::Object GetDeviceInfoList ()

  This method retrieves the list of PyDeviceInfo objects for the camera devices, available on this Interface.
- Py::Object GetType ()

  This method gets the type of interface.

6.5.1 Detailed Description

The PyDeviceInterface class represents a interface module in the GenTL module hierarchy.

This class represents an individual physical interface in the System. For example, a network interface card (NIC) for GigE Vision connection, CXP or Camera Link frame grabber board, or USB3 Vision driver in the GenTL system. The PyDeviceInterface class includes methods to enumerate the available devices on the physical interface in the system.

IpxCamPy.PyDeviceInterface Python class is wrapper around C++ class **IpxCam::Interface**

6.5.2 Member Function Documentation

6.5.2.1 GetDescription()

```
Py::Object IpxCamPy::PyDeviceInterface::GetDescription ( )
```

This method returns the description of the interface.

The GetDescription method gets the user readable string description of the interface.

Returns

Returns the Description of the interface. Value type: String

6.5.2.2 GetFirstDeviceInfo()

```
Py::Object IpxCamPy::PyDeviceInterface::GetFirstDeviceInfo ( )
```

This method retrieves the DeviceInfo object for the first device available on this Interface.

Returns

Returns the IpxCamPy.PyDeviceInfo object

6.5.2.3 ReEnumerateDevices()

```
Py::Object IpxCamPy::PyDeviceInterface::ReEnumerateDevices ( )
```

This method re-enumerates the devices.

The ReEnumerateDevices method allows the user to re-enumerate the devices connected to the Interface and update the list of PyDeviceInfo objects returned by subsequent GetDeviceInfoList() method calls.

Returns

returns Tuple of:

- [0] Boolean value: Success of operation
- [1] Boolean value: True - if list of devices was changed, False - no changes

6.5.2.4 GetDeviceInfoList()

```
Py::Object IpxCamPy::PyDeviceInterface::GetDeviceInfoList ( )
```

This method retrieves the list of PyDeviceInfo objects for the camera devices, available on this Interface.

Returns

Returns the list of IpxCamPy.PyDeviceInfo objects

6.5.2.5 GetType()

```
Py::Object IpxCamPy::PyDeviceInterface::GetType ( )
```

This method gets the type of interface.

The GetType method returns the Interface Type (Transport Layer Technology) of this interface object

Returns

Returns Interface Type. Value type: Integer. Possible values:

- 1 - USB3Vision
- 2 - GigEVision
- 3 - CameraLink
- 4 - CoaxPress
- 5 - HdSdi

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

- IpxCameraApiPy.h

6.6 IpxGenParamPy::PyGenParam Class Reference

General class for GenICam parameter.

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParam >.

Public Member Functions

- Py::Object GetType ()
  This method returns the Parameter Node Type.
- Py::Object GetName ()
  This method returns the parameter node name.
- Py::Object GetToolTip ()
  This method returns a short description of the parameter node.
- Py::Object GetDescription ()
  This method returns a long description of the parameter node.
- Py::Object GetDisplayName ()
  This method returns the string to be used for the parameter displaying.
- Py::Object GetVisibility ()
  This method returns the visibility of the node.
- Py::Object IsVisible (const Py::Tuple &args)
  This method checks if the Param is visible.

- Py::Object IsValueCached ()

  This method checks if the parameter node is cached.
- Py::Object IsAvailable ()

  This method checks if parameter node is available.
- Py::Object IsWritable ()

  This method checks if parameter node is writable.
- Py::Object IsReadable ()

  This method checks if the parameter node is readable.
- Py::Object IsStreamable ()

  This method checks if the parameter node is streamable.

6.6.1 Detailed Description

General class for GenICam parameter.

PyGenParam Class for accessing the GenICam feature node of the Camera parameters

IpxGenParamPy.PyGenParam is wrapper around IpxGenParam::Param C++ class

6.6.2 Member Function Documentation

6.6.2.1 GetType()

```
Py::Object IpxGenParamPy::PyGenParam::GetType ( )
```

This method returns the Parameter Node Type.

Returns

Returns the parameter type or 0 if failed. Value type: Integer.

6.6.2.2 GetName()

```
Py::Object IpxGenParamPy::PyGenParam::GetName ( )
```

This method returns the parameter node name.

Returns

If the method succeeds, it will return the parameter node name. Otherwise, it will return None. Value type: String.

6.6.2.3 GetToolTip()

```
Py::Object IpxGenParamPy::PyGenParam::GetToolTip ( )
```

This method returns a short description of the parameter node.

Returns

If the method succeeds, it will return a short description of the parameter node. Otherwise, it will return None. Value type: String.

6.6.2.4 GetDescription()

```
Py::Object IpxGenParamPy::PyGenParam::GetDescription ( )
```

This method returns a long description of the parameter node.

Returns

If the method succeeds, it will return a long description of the parameter node. Otherwise, it will return None. Value type: String.

6.6.2.5 GetDisplayName()

```
Py::Object IpxGenParamPy::PyGenParam::GetDisplayName ( )
```

This method returns the string to be used for the parameter displaying.

Returns

If the method succeeds, it will return the parameter display name. Otherwise, it will return None. Value type: String.

6.6.2.6 GetVisibility()

```
Py::Object IpxGenParamPy::PyGenParam::GetVisibility ( )
```

This method returns the visibility of the node.

Returns

It will return the visibility setting of the parameter node. Possible values are: Basic (0), Expert (1), Guru (2), Invisible (3) or Undefined (99). Value type: Integer

6.6.2.7 IsVisible()

```
Py::Object IpxGenParamPy::PyGenParam::IsVisible (
            const Py::Tuple & args )
```

This method checks if the Param is visible.

Parameters

in	*args[0]*	Visibility of the parameter node

Returns

True if the parameter node is visible. Otherwise, it is not visible.

6.6.2.8 IsValueCached()

```
Py::Object IpxGenParamPy::PyGenParam::IsValueCached ( )
```

This method checks if the parameter node is cached.

Returns

True if the value is cached. False if the value is not cached. Value type: Boolean

6.6.2.9 IsAvailable()

```
Py::Object IpxGenParamPy::PyGenParam::IsAvailable ( )
```

This method checks if parameter node is available.

Returns

True if the parameter node is available. False if it is not available. Value type: Boolean

6.6.2.10 IsWritable()

```
Py::Object IpxGenParamPy::PyGenParam::IsWritable ( )
```

This method checks if parameter node is writable.

Returns

True if the parameter node is writable. False if is not writable. Value type: Boolean

6.6.2.11 IsReadable()

```
Py::Object IpxGenParamPy::PyGenParam::IsReadable ( )
```

This method checks if the parameter node is readable.

Returns

True if the parameter node is readable. False if it is not readable. Value type: Boolean

6.6.2.12 IsStreamable()

```
Py::Object IpxGenParamPy::PyGenParam::IsStreamable ( )
```

This method checks if the parameter node is streamable.

Returns

True if the parameter node is streamable. False if it is not streamable. Value type: Boolean

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

- IpxGenParamApiPy.h

6.7 IpxGenParamPy::PyGenParamBoolean Class Reference

PyGenParamBoolean class represents the Boolean GenICam camera parameter.

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParamBoolean >.

Public Member Functions

- Py::Object GetType ()
  This method returns the node object Boolean type.
- Py::Object GetValue ()
  This method gets the Boolean node value.
- Py::Object SetValue (const Py::Tuple &args)
  This method sets the Boolean node value.
- Py::Object IsWritable ()
  This method checks if parameter node is writable.
- Py::Object IsReadable ()
  This method checks if parameter node is readable.

6.7.1 Detailed Description

PyGenParamBoolean class represents the Boolean GenICam camera parameter.

PyGenParamBoolean class contains methods that map the integer element value of a GenICam IBoolean interface feature to true or false.

For example, the mapping below will illustrate the IBoolean interfaces of a **LUTEnable** feature.

IpxGenParamPy.PyGenParamBoolean class is a wrapper around IpxGenParam::Boolean C++ class

6.7.2 Member Function Documentation

6.7.2.1 GetType()

```
Py::Object IpxGenParamPy::PyGenParamBoolean::GetType ( )
```

This method returns the node object Boolean type.

Returns

Returns the node object Boolean type. Value type: Integer.

6.7.2.2 GetValue()

```
Py::Object IpxGenParamPy::PyGenParamBoolean::GetValue ( )
```

This method gets the Boolean node value.

Returns

Returns tuple of Success of the operation ant the Boolean value. Value type: Boolean.

6.7.2.3 SetValue()

```
Py::Object IpxGenParamPy::PyGenParamBoolean::SetValue (
            const Py::Tuple & args )
```

This method sets the Boolean node value.

Parameters

in	*args[0]*	Boolean node value to set. Value type: Boolean.

Returns

Returns Success of the operation. Value type: Boolean.

6.7.2.4 IsWritable()

```
Py::Object IpxGenParamPy::PyGenParamBoolean::IsWritable ( )
```

This method checks if parameter node is writable.

Returns

True if the parameter node is writable, False - otherwise. Value type: Boolean.

6.7.2.5 IsReadable()

```
Py::Object IpxGenParamPy::PyGenParamBoolean::IsReadable ( )
```

This method checks if parameter node is readable.

Returns

True if the parameter node is readable, False - otherwise. Value type: Boolean.

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

- IpxGenParamApiPy.h

6.8 IpxGenParamPy::PyGenParamCategory Class Reference

A class containing methods that the user can access the categories GenICam features. It will access the node object's of an ICategory interface. Each feature of a device will be placed in a **Category**. The Category feature is used to present the user with a group of features for the named category.

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParamCategory >.

Public Member Functions

- Py::Object GetType ()

  This method returns the node object Category type.
- Py::Object GetCount ()

  This method returns the number of parameters in the category.
- Py::Object GetParamByIndex (const Py::Tuple &args)

  This method returns the Parameter by Index.
- Py::Object GetNode ()

  This method returns the PyGenParamNode object.

6.8.1 Detailed Description

A class containing methods that the user can access the categories GenICam features. It will access the node object's of an ICategory interface. Each feature of a device will be placed in a **Category**. The Category feature is used to present the user with a group of features for the named category.

A class containing methods for GenICam Category Properties. For example, the mapping below will illustrate the I↩ Category interfaces categories such as DeviceControl and EventControl.

PyGenParamCategory class represents the GenICam Category.

PyGenParamCategory class contains methods that the user can access the categories of GenICam features. It will access the node object's of an ICategory interface. Each feature of a device will be placed in a **Category**. The Category feature is used to present the user with a group of features for the named category.

For example, the mapping below will illustrate the ICategory interfaces categories such as DeviceControl and Event↩ Control.

IpxGenParamPy.PyGenParamCategory class is a wrapper around IpxGenParam::Category C++ class

6.8.2 Member Function Documentation

6.8.2.1 GetType()

```
Py::Object IpxGenParamPy::PyGenParamCategory::GetType ( )
```

This method returns the node object Category type.

Returns

Returns the node object Category type. Value type: Integer.

6.8.2.2 GetCount()

```
Py::Object IpxGenParamPy::PyGenParamCategory::GetCount ( )
```

This method returns the number of parameters in the category.

Returns

Returns the number of parameters in the category. Value type: Integer.

6.8.2.3 GetParamByIndex()

```
Py::Object IpxGenParamPy::PyGenParamCategory::GetParamByIndex (
            const Py::Tuple & args )
```

This method returns the Parameter by Index.

Parameters

in	*args[0]*	parameter index

Returns

Returns the parameter object

6.8.2.4 GetNode()

```
Py::Object IpxGenParamPy::PyGenParamCategory::GetNode ( )
```

This method returns the PyGenParamNode object.

Returns

If the method succeeds, it will return IpxGenParamPy.PyGenParamNode object for this Parameter. Otherwise, it will return a value of None.

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

- IpxGenParamApiPy.h

6.9 IpxGenParamPy::PyGenParamCommand Class Reference

PyGenParamCommand class represents Command GenICam camera parameter.

`#include <IpxGenParamApiPy.h>`

Inherits PythonExtension< PyGenParamCommand >.

Public Member Functions

- Py::Object Execute ()

  This method executes the command.
- Py::Object IsDone ()

  This method queries whether the command is executed and completed.

6.9.1 Detailed Description

PyGenParamCommand class represents Command GenICam camera parameter.

A class for GenICam Command contains methods that allow the user submit a command for execution as well as poll the command status.

For example, the mapping below will illustrate the ICommand interface for AcquisitionStart. This feature starts the Acquisition of the device.

IpxGenParamPy.PyGenParamCommand class is a wrapper around IpxGenParam::Command C++ class

6.9.2 Member Function Documentation

6.9.2.1 Execute()

```
Py::Object IpxGenParamPy::PyGenParamCommand::Execute ( )
```

This method executes the command.

Returns

Success of the operation. Value type: Boolean.

6.9.2.2 IsDone()

```
Py::Object IpxGenParamPy::PyGenParamCommand::IsDone ( )
```

This method queries whether the command is executed and completed.

Returns

Returns True, if the command has finished. Otherwise, it returns False. Value type: Boolean.

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

- IpxGenParamApiPy.h

6.10 IpxGenParamPy::PyGenParamEnum Class Reference

PyGenParamEnum class represents Enumeration GenICam camera parameter.

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParamEnum >.

Public Member Functions

- Py::Object GetCount ()
  This method gets count of the entry items the Enum .
- Py::Object GetValue ()
  This method gets the Enum Entry node value.
- Py::Object GetValueStr ()
  This method gets the Enum Entry node String value.
- Py::Object SetValue (const Py::Tuple &args)
  This method Sets the Enum Entry node numerical value.
- Py::Object SetValueStr (const Py::Tuple &args)
  This method Sets the Enum Entry node String value.
- Py::Object GetType ()
  This method returns the node object Enum type.
- Py::Object GetEnumEntryByIndex (const Py::Tuple &args)
  This method gets the Enum Entry node by the Index number.

6.10.1 Detailed Description

PyGenParamEnum class represents Enumeration GenICam camera parameter.

A class containing methods to access the Enumeration GenICam camera parameter, using Integer or String value.

For example, the picture below illustrates the enumeration "WhiteBalanceMode".

IpxGenParamPy.PyGenParamEnum class is a wrapper around IpxGenParam::Enum C++ class

6.10.2 Member Function Documentation

6.10.2.1 GetCount()

```
Py::Object IpxGenParamPy::PyGenParamEnum::GetCount ( )
```

This method gets count of the entry items the Enum .

Returns

Returns Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Integer value: Number of entries in the Enum

6.10.2.2 GetValue()

```
Py::Object IpxGenParamPy::PyGenParamEnum::GetValue ( )
```

This method gets the Enum Entry node value.

Returns

Returns Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Integer value: Enum numerical value

6.10.2.3 GetValueStr()

```
Py::Object IpxGenParamPy::PyGenParamEnum::GetValueStr ( )
```

This method gets the Enum Entry node String value.

Returns

Returns Tuple of values:

- [0] Boolean value: Success of the operation
- [1] String value: Enum string value

6.10.2.4 SetValue()

```
Py::Object IpxGenParamPy::PyGenParamEnum::SetValue (
            const Py::Tuple & args )
```

This method Sets the Enum Entry node numerical value.

Parameters

in	*args[0]*	Enum numerical value. Value type: Integer

Returns

Returns True, if operation succeeded. False, if failed. Value type: Boolean

6.10.2.5 SetValueStr()

```
Py::Object IpxGenParamPy::PyGenParamEnum::SetValueStr (
            const Py::Tuple & args )
```

This method Sets the Enum Entry node String value.

Parameters

in	*args[0]*	Enum string value. Value type: String

Returns

Returns True, if operation succeeded. False, if failed. Value type: Boolean

6.10.2.6 GetType()

```
Py::Object IpxGenParamPy::PyGenParamEnum::GetType ( )
```

This method returns the node object Enum type.

Returns

Returns the parameter type. Value type: Integer

6.10.2.7 GetEnumEntryByIndex()

```
Py::Object IpxGenParamPy::PyGenParamEnum::GetEnumEntryByIndex (
            const Py::Tuple & args )
```

This method gets the Enum Entry node by the Index number.

Parameters

in	*args[0]*	Index of the Enum Entry. Value type: Integer

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamEnumEntry object, otherwise returns None.

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

- IpxGenParamApiPy.h

6.11 IpxGenParamPy::PyGenParamEnumEntry Class Reference

PyGenParamEnumEntry class represents the entry of GenICam Enum parameter.

`#include <IpxGenParamApiPy.h>`

Inherits PythonExtension< PyGenParamEnumEntry >.

Public Member Functions

- Py::Object GetValue ()

  This method gets the EnumEntry numerical value.
- Py::Object GetValueStr ()

  This method gets the EnumEntry String value.
- Py::Object GetType ()

  This method returns the node object EnumEntry type.
- Py::Object IsAvailable ()

  This method checks if EnumEntry is available.

6.11.1 Detailed Description

PyGenParamEnumEntry class represents the entry of GenICam Enum parameter.

A Class for GenICam Enum Entries has methods to access the Enumeration GenICam parameter entry.

For example, the mapping below illustrates entries of the IEnumeration interface for the AOI2_Select feature. This feature can select the mode of operation for Slave AOI #2. The enumeration entries could be "Off", "Include", and "Exclude".

IpxGenParamPy.PyGenParamEnumEntry class is a wrapper around IpxGenParam::EnumEntry C++ class

6.11.2 Member Function Documentation

6.11.2.1 GetValue()

```
Py::Object IpxGenParamPy::PyGenParamEnumEntry::GetValue ( )
```

This method gets the EnumEntry numerical value.

Returns

Returns Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Integer EnumEntry value

6.11.2.2 GetValueStr()

```
Py::Object IpxGenParamPy::PyGenParamEnumEntry::GetValueStr ( )
```

This method gets the EnumEntry String value.

Returns

Returns Tuple of values:

- [0] Boolean value: Success of the operation
- [1] String EnumEntry value

6.11.2.3 GetType()

```
Py::Object IpxGenParamPy::PyGenParamEnumEntry::GetType ( )
```

This method returns the node object EnumEntry type.

Returns

Returns the node type. Value type: Integer

6.11.2.4 IsAvailable()

```
Py::Object IpxGenParamPy::PyGenParamEnumEntry::IsAvailable ( )
```

This method checks if EnumEntry is available.

Returns

True, if EnumEntry is available. False, if it is not available. Value type: Boolean

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

- IpxGenParamApiPy.h

6.12 IpxGenParamPy::PyGenParamFloat Class Reference

PyGenParamFloat class represents Float GenICam camera parameter.

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParamFloat >.

Public Member Functions

- Py::Object GetType ()

  This method returns the node object Float type.
- Py::Object IsWritable ()

  This method checks if parameter node is writable.
- Py::Object IsReadable ()

  This method checks if parameter node is readable.
- Py::Object GetValue ()

  This method gets the Float node value.
- Py::Object SetValue (const Py::Tuple &args)

  This method sets the Float node value.
- Py::Object GetMin ()

  This method gets the minimum value.
- Py::Object GetMax ()

  This method gets the maximum value.
- Py::Object GetUnit ()

  This method gets the Unit string.

6.12.1 Detailed Description

PyGenParamFloat class represents Float GenICam camera parameter.

A class containing methods to access the Float GenICam camera parameter as floating point value.

For example, the picture below illustrates the float "ExposureTime".

IpxGenParamPy.PyGenParamFloat class is a wrapper around IpxGenParam::Float C++ class

6.12.2 Member Function Documentation

6.12.2.1 GetType()

```
Py::Object IpxGenParamPy::PyGenParamFloat::GetType ( )
```

This method returns the node object Float type.

Returns

Returns the parameter type. Value type: Integer.

6.12.2.2 IsWritable()

```
Py::Object IpxGenParamPy::PyGenParamFloat::IsWritable ( )
```

This method checks if parameter node is writable.

Returns

True if the parameter node is writable, False - otherwise. Value type: Boolean.

6.12.2.3 IsReadable()

```
Py::Object IpxGenParamPy::PyGenParamFloat::IsReadable ( )
```

This method checks if parameter node is readable.

Returns

True if the parameter node is readable, False - otherwise. Value type: Boolean.

6.12.2.4 GetValue()

```
Py::Object IpxGenParamPy::PyGenParamFloat::GetValue ( )
```

This method gets the Float node value.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] Integer: Float parameter value

6.12.2.5 SetValue()

```
Py::Object IpxGenParamPy::PyGenParamFloat::SetValue (
            const Py::Tuple & args )
```

This method sets the Float node value.

Parameters

in	*args[0]*	Float node value to set. Value type: Integer.

Returns

Returns Success of the operation. Value type: Boolean.

6.12.2.6 GetMin()

```
Py::Object IpxGenParamPy::PyGenParamFloat::GetMin ( )
```

This method gets the minimum value.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] Integer: Float parameter minimum value

6.12.2.7 GetMax()

```
Py::Object IpxGenParamPy::PyGenParamFloat::GetMax ( )
```

This method gets the maximum value.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] Integer: Float parameter maximum value

6.12.2.8 GetUnit()

```
Py::Object IpxGenParamPy::PyGenParamFloat::GetUnit ( )
```

This method gets the Unit string.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] String: Float parameter measurement unit string

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

- IpxGenParamApiPy.h

6.13 IpxGenParamPy::PyGenParamInt Class Reference

PyGenParamInt class represents Integer GenICam camera parameter.

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParamInt >.

Public Member Functions

- Py::Object GetType ()

  This method returns the node object Int type.
- Py::Object IsWritable ()

  This method checks if parameter node is writable.
- Py::Object IsReadable ()

  This method checks if parameter node is readable.
- Py::Object GetValue ()

  This method gets the Int node value.
- Py::Object SetValue (const Py::Tuple &args)

  This method sets the Int node value.
- Py::Object GetMin ()

  This method gets the minimum value.
- Py::Object GetMax ()

  This method gets the maximum value.
- Py::Object GetIncrement ()

  This method gets the Increment value.

6.13.1 Detailed Description

PyGenParamInt class represents Integer GenICam camera parameter.

A class containing methods to access the Integer GenICam camera parameter as integer value.

For example, the mapping below illustrates "Width" Integer parameter.

IpxGenParamPy.PyGenParamInt class is a wrapper around IpxGenParam::Int C++ class

6.13.2 Member Function Documentation

6.13.2.1 GetType()

```
Py::Object IpxGenParamPy::PyGenParamInt::GetType ( )
```

This method returns the node object Int type.

Returns

Returns the parameter type. Value type: Integer.

6.13.2.2 IsWritable()

```
Py::Object IpxGenParamPy::PyGenParamInt::IsWritable ( )
```

This method checks if parameter node is writable.

Returns

True if the parameter node is writable, False - otherwise. Value type: Boolean.

6.13.2.3 IsReadable()

```
Py::Object IpxGenParamPy::PyGenParamInt::IsReadable ( )
```

This method checks if parameter node is readable.

Returns

True if the parameter node is readable, False - otherwise. Value type: Boolean.

6.13.2.4 GetValue()

```
Py::Object IpxGenParamPy::PyGenParamInt::GetValue ( )
```

This method gets the Int node value.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] Integer: Int parameter value

6.13.2.5 SetValue()

```
Py::Object IpxGenParamPy::PyGenParamInt::SetValue (
            const Py::Tuple & args )
```

This method sets the Int node value.

Parameters

in	*args[0]*	Int node value to set. Value type: Integer.

Returns

Returns Success of the operation. Value type: Boolean.

6.13.2.6 GetMin()

```
Py::Object IpxGenParamPy::PyGenParamInt::GetMin ( )
```

This method gets the minimum value.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] Integer: Int parameter minimum value

6.13.2.7 GetMax()

```
Py::Object IpxGenParamPy::PyGenParamInt::GetMax ( )
```

This method gets the maximum value.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] Integer: Int parameter maximum value

6.13.2.8 GetIncrement()

```
Py::Object IpxGenParamPy::PyGenParamInt::GetIncrement ( )
```

This method gets the Increment value.

Returns

Returns Tuple of values:

- [0] Boolean: Success of the operation
- [1] Integer: Int parameter increment value

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

- IpxGenParamApiPy.h

6.14 IpxGenParamPy::PyGenParamNode Class Reference

PyGenParamNode class represents GenICam INode class.

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParamNode >.

Public Member Functions

- Py::Object IsFeature ()

  Returns True if the node can be reached via category nodes from a category node named "Root". Otherwise it returns False.

6.14.1 Detailed Description

PyGenParamNode class represents GenICam INode class.

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

- IpxGenParamApiPy.h

6.15 IpxGenParamPy::PyGenParamNodeMap Class Reference

```
#include <IpxGenParamApiPy.h>
```

Inherits PythonExtension< PyGenParamNodeMap >.

6.15.1 Detailed Description

Interface to access the node map

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

- IpxGenParamApiPy.h

6.16 IpxGenParamPy::PyGenParams Class Reference

An PyGenParams class contains methods to access all GenICam camera parameters.

`#include <IpxGenParamApiPy.h>`

Inherits PythonExtension< PyGenParams >.

Public Member Functions

- Py::Object SetIntegerValue (const Py::Tuple &args)
  This method sets the Integer value of the Integer node.
- Py::Object GetIntegerValue (const Py::Tuple &args)
  This method gets the Integer value of the Integer node.
- Py::Object SetFloatValue (const Py::Tuple &args)
  This method sets the Float value of the Float node.
- Py::Object GetFloatValue (const Py::Tuple &args)
  This method gets the Float value of the Float node.
- Py::Object ExecuteCommand (const Py::Tuple &args)
  This method executes/submits the command.
- Py::Object GetStringValue (const Py::Tuple &args)
  This method gets the String value of the String node.
- Py::Object SetStringValue (const Py::Tuple &args)
  This method sets the String value of the String node.
- Py::Object GetEnumValueStr (const Py::Tuple &args)
  This method gets the String value of the Enum node.
- Py::Object GetEnumValue (const Py::Tuple &args)
  This method gets the Integer value of the Enum node.
- Py::Object SetEnumValueStr (const Py::Tuple &args)
  This method sets the String value of the Enum node.
- Py::Object SetEnumValue (const Py::Tuple &args)
  This method sets the Integer value of the Enum node.
- Py::Object GetEnum (const Py::Tuple &args)
  This method gets link to the Enum object for the specified node name of the camera descriptor XML file.
- Py::Object GetParamByIndex (const Py::Tuple &args)
  This method gets the parameter object by index.
- Py::Object GetParam (const Py::Tuple &args)

This method gets the parameter object by name.

- Py::Object GetCount ()

  This method gets the number of nodes.

- Py::Object GetRootCategory ()

  This method gets the object of the root category node. The Root node is considered a special node. It has no parent node. In the topology graph, it is the top node which connects to at least one child node. The child node may connect to the device node that provides the connection to the transport layer.

- Py::Object GetBooleanValue (const Py::Tuple &args)

  This method gets the value of the Boolean node.

- Py::Object SetBooleanValue (const Py::Tuple &args)

  This method sets the value of the Boolean node.

- Py::Object GetBoolean (const Py::Tuple &args)

  This method gets the PyGenParamBoolean class object for the specified node name of the camera descriptor XML file.

- Py::Object GetInt (const Py::Tuple &args)

  This method gets link to the Int object for the specified node name of the camera descriptor XML file.

- Py::Object GetString (const Py::Tuple &args)

  This method gets the PyGenParamString class object for the specified node name of the camera descriptor XML file.

- Py::Object GetFloat (const Py::Tuple &args)

  This method gets the PyGenParamFloat class object for the specified node name of the camera descriptor XML file.

- Py::Object GetCommand (const Py::Tuple &args)

  This method gets the PyGenParamCommand class object for the specified node name of the camera descriptor XML file.

- Py::Object IsCommandDone (const Py::Tuple &args)

  This method polls the corresponding executed command to see if the executed command is done or not.

- Py::Object GetNodeMap ()

  This method gets the pointer to the NodeMap interface. The NodeMap interface will provide methods to retrieves all nodes in the node map.

6.16.1 Detailed Description

An PyGenParams class contains methods to access all GenICam camera parameters.

This class contains methods that can access each node from the GenICam camera description XML file by parameters type and name.

6.16.2 Member Function Documentation

6.16.2.1 SetIntegerValue()

```
Py::Object IpxGenParamPy::PyGenParams::SetIntegerValue (
            const Py::Tuple & args )
```

This method sets the Integer value of the Integer node.

Parameters

in	*args[0]*	String value: Unique name of Integer parameter to set
in	*args[1]*	Integer value: value to set

Returns

Returns Success of the operation. Value type: Boolean

6.16.2.2 GetIntegerValue()

```
Py::Object IpxGenParamPy::PyGenParams::GetIntegerValue (
            const Py::Tuple & args )
```

This method gets the Integer value of the Integer node.

Parameters

in	*args[0]*	Unique name of Integer parameter to get. Value type: String

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Integer value: Parameter value

6.16.2.3 SetFloatValue()

```
Py::Object IpxGenParamPy::PyGenParams::SetFloatValue (
            const Py::Tuple & args )
```

This method sets the Float value of the Float node.

Parameters

in	*args[0]*	String value: Unique name of Float parameter to set
in	*args[1]*	Float value: value to set

Returns

Returns Success of the operation. Value type: Boolean

6.16.2.4 GetFloatValue()

```
Py::Object IpxGenParamPy::PyGenParams::GetFloatValue (
            const Py::Tuple & args )
```

This method gets the Float value of the Float node.

Parameters

in	*args[0]*	Unique name of Float parameter to get. Value type: String

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Float value: Parameter value

6.16.2.5 ExecuteCommand()

```
Py::Object IpxGenParamPy::PyGenParams::ExecuteCommand (
            const Py::Tuple & args )
```

This method executes/submits the command.

Parameters

in	*args[0]*	Unique name of Command type node in the camera descriptor XML file. Value type: String

Returns

Returns True, if operation succeeded. False, if failed. Value type: Boolean

6.16.2.6 GetStringValue()

```
Py::Object IpxGenParamPy::PyGenParams::GetStringValue (
            const Py::Tuple & args )
```

This method gets the String value of the String node.

Parameters

in	*args[0]*	Unique name of String parameter to get. Value type: String

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] String value: Parameter value

6.16.2.7 SetStringValue()

```
Py::Object IpxGenParamPy::PyGenParams::SetStringValue (
            const Py::Tuple & args )
```

This method sets the String value of the String node.

Parameters

in	*args[0]*	String value: Unique name of String parameter to set
in	*args[1]*	String value: value to set

Returns

Returns Success of the operation. Value type: Boolean

6.16.2.8 GetEnumValueStr()

```
Py::Object IpxGenParamPy::PyGenParams::GetEnumValueStr (
            const Py::Tuple & args )
```

This method gets the String value of the Enum node.

Parameters

in	*args[0]*	Unique name of Enum parameter to get. Value type: String

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] String value: Enum parameter value

6.16.2.9 GetEnumValue()

```
Py::Object IpxGenParamPy::PyGenParams::GetEnumValue (
            const Py::Tuple & args )
```

This method gets the Integer value of the Enum node.

Parameters

in	*args[0]*	Unique name of Enum parameter to get. Value type: String

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Integer value: Enum parameter value

6.16.2.10 SetEnumValueStr()

```
Py::Object IpxGenParamPy::PyGenParams::SetEnumValueStr (
            const Py::Tuple & args )
```

This method sets the String value of the Enum node.

Parameters

in	*args[0]*	String value: Unique name of Enum parameter to set
in	*args[1]*	String value: value to set

Returns

Returns Success of the operation. Value type: Boolean

6.16.2.11 SetEnumValue()

```
Py::Object IpxGenParamPy::PyGenParams::SetEnumValue (
            const Py::Tuple & args )
```

This method sets the Integer value of the Enum node.

Parameters

in	*args[0]*	String value: Unique name of Enum parameter to set
in	*args[1]*	Integer value: value to set

Returns

Returns Success of the operation. Value type: Boolean

6.16.2.12 GetEnum()

```
Py::Object IpxGenParamPy::PyGenParams::GetEnum (
            const Py::Tuple & args )
```

This method gets link to the Enum object for the specified node name of the camera descriptor XML file.

Parameters

in	*name*	Unique name of a node in node map.

Returns

If the method succeeds, it returns link to the Enum object. Otherwise, it returns a None.This method gets the PyGenParamEnum class object for the specified node name of the camera descriptor XML file

Parameters

in	*args[0]*	A unique name of Enum type node in the camera descriptor XML file. Value type: String

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamEnum class object for the specific node name. Returns None if failed

6.16.2.13 GetParamByIndex()

```
Py::Object IpxGenParamPy::PyGenParams::GetParamByIndex (
            const Py::Tuple & args )
```

This method gets the parameter object by index.

Parameters

in	*args[0]*	Parameter index. Value type: Integer

Returns

If the method succeeds, it returns IpxGenParamPy.PyGenParam or IpxGenParamPy.PyGenParamCategory object of the node, referenced by the index value. Returns None if failed.

6.16.2.14 GetParam()

```
Py::Object IpxGenParamPy::PyGenParams::GetParam (
            const Py::Tuple & args )
```

This method gets the parameter object by name.

Parameters

in	*args[0]*	Parameter name. Value type: String

Returns

If the method succeeds, it returns IpxGenParamPy.PyGenParam or IpxGenParamPy.PyGenParamCategory object of the node, referenced by the index value. Returns None if failed.

6.16.2.15 GetCount()

```
Py::Object IpxGenParamPy::PyGenParams::GetCount ( )
```

This method gets the number of nodes.

Returns

Returns the number of nodes. This number should be greater than 0. Value type: Integer

6.16.2.16 GetRootCategory()

```
Py::Object IpxGenParamPy::PyGenParams::GetRootCategory ( )
```

This method gets the object of the root category node. The Root node is considered a special node. It has no parent node. In the topology graph, it is the top node which connects to at least one child node. The child node may connect to the device node that provides the connection to the transport layer.

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamCategory object of the root category node. Returns None if failed

6.16.2.17 GetBooleanValue()

```
Py::Object IpxGenParamPy::PyGenParams::GetBooleanValue (
            const Py::Tuple & args )
```

This method gets the value of the Boolean node.

Parameters

in	*args[0]*	Unique name of Boolean parameter to get. Value type: String

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Boolean value: Parameter value

6.16.2.18 SetBooleanValue()

```
Py::Object IpxGenParamPy::PyGenParams::SetBooleanValue (
            const Py::Tuple & args )
```

This method sets the value of the Boolean node.

Parameters

in	*args[0]*	String value: Unique name of Boolean parameter to set
in	*args[1]*	Boolean value: value to set

Returns

Returns Success of the operation. Value type: Boolean

6.16.2.19 GetBoolean()

```
Py::Object IpxGenParamPy::PyGenParams::GetBoolean (
            const Py::Tuple & args )
```

This method gets the PyGenParamBoolean class object for the specified node name of the camera descriptor XML file.

Parameters

in	*args[0]*	A unique name of Boolean type node in the camera descriptor XML file. Value type: String

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamBoolean class object for the specific node name. Returns None if failed

6.16.2.20 GetInt()

```
Py::Object IpxGenParamPy::PyGenParams::GetInt (
            const Py::Tuple & args )
```

This method gets link to the Int object for the specified node name of the camera descriptor XML file.

Parameters

in	*name*	Unique name of a node in node map.

Returns

If the method succeeds, it returns link to the Int object. Otherwise, it returns a None.This method gets the PyGenParamInt class object for the specified node name of the camera descriptor XML file

Parameters

in	*args[0]*	A unique name of Int type node in the camera descriptor XML file. Value type: String

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamInt class object for the specific node name. Returns None if failed

6.16.2.21 GetString()

```
Py::Object IpxGenParamPy::PyGenParams::GetString (
            const Py::Tuple & args )
```

This method gets the PyGenParamString class object for the specified node name of the camera descriptor XML file.

Parameters

in	*args[0]*	A unique name of String type node in the camera descriptor XML file. Value type: String

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamString class object for the specific node name. Returns None if failed

6.16.2.22 GetFloat()

```
Py::Object IpxGenParamPy::PyGenParams::GetFloat (
            const Py::Tuple & args )
```

This method gets the PyGenParamFloat class object for the specified node name of the camera descriptor XML file.

Parameters

in	*args[0]*	A unique name of Float type node in the camera descriptor XML file. Value type: String

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamFloat class object for the specific node name. Returns None if failed

6.16.2.23 GetCommand()

```
Py::Object IpxGenParamPy::PyGenParams::GetCommand (
            const Py::Tuple & args )
```

This method gets the PyGenParamCommand class object for the specified node name of the camera descriptor XML file.

Parameters

in	*args[0]*	A unique name of Command type node in the camera descriptor XML file. Value type: String

Returns

If the method succeeds, it returns the IpxGenParamPy.PyGenParamCommand class object for the specific node name. Returns None if failed

6.16.2.24 IsCommandDone()

```
Py::Object IpxGenParamPy::PyGenParams::IsCommandDone (
            const Py::Tuple & args )
```

This method polls the corresponding executed command to see if the executed command is done or not.

Parameters

in	*args[0]*	A unique name of Command type node in the camera descriptor XML file.

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] Boolean value: True, if command done, False - otherwise

6.16.2.25 GetNodeMap()

```
Py::Object IpxGenParamPy::PyGenParams::GetNodeMap ( )
```

This method gets the pointer to the NodeMap interface. The NodeMap interface will provide methods to retrieves all nodes in the node map.

Returns

Tuple of values:

- [0] Boolean value: Success of the operation
- [1] PyGenParamNodeMap object

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

- IpxGenParamApiPy.h

6.17 IpxGenParamPy::PyGenParamString Class Reference

PyGenParamString class represents String GenICam camera parameter.

`#include <IpxGenParamApiPy.h>`

Inherits PythonExtension< PyGenParamString >.

Public Member Functions

- Py::Object GetType ()
  This method returns the node object Param type.
- Py::Object IsWritable ()
  This method checks if parameter node is writable.
- Py::Object IsReadable ()
  This method checks if parameter node is readable.
- Py::Object GetValue ()
  This method gets the value of the string node.
- Py::Object GetMaxLength ()
  This method gets the Maximum Length of the string. Value type: Integer.
- Py::Object SetValue (const Py::Tuple &args)
  This method sets the value of the string node.

6.17.1 Detailed Description

PyGenParamString class represents String GenICam camera parameter.

A class containing methods to access the String GenICam camera parameter as zero-terminated array of characters

For example, the image below illustrates "DeviceModelName" parameter.

IpxGenParamPy.PyGenParamString class is a wrapper around IpxGenParam::String C++ class

6.17.2 Member Function Documentation

6.17.2.1 GetType()

```
Py::Object IpxGenParamPy::PyGenParamString::GetType ( )
```

This method returns the node object Param type.

Returns

The parameter type

6.17.2.2 IsWritable()

```
Py::Object IpxGenParamPy::PyGenParamString::IsWritable ( )
```

This method checks if parameter node is writable.

Returns

True if the parameter node is writable. False, it is not writable. Value type: Boolean.

6.17.2.3 IsReadable()

```
Py::Object IpxGenParamPy::PyGenParamString::IsReadable ( )
```

This method checks if parameter node is readable.

Returns

True if the parameter node is readable. False, it is not readable. Value type: Boolean.

6.17.2.4 GetValue()

```
Py::Object IpxGenParamPy::PyGenParamString::GetValue ( )
```

This method gets the value of the string node.

Returns

Returns Tuple of values:

- [0] Boolean value: Success of the operation
- [1] String parameter value

6.17.2.5 SetValue()

```
Py::Object IpxGenParamPy::PyGenParamString::SetValue (
            const Py::Tuple & args )
```

This method sets the value of the string node.

Parameters

in	*args[0]*	String value to set. Value type: String.

Returns

Returns True if operation succeeded, False - otherwise. Value type: Boolean.

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

- IpxGenParamApiPy.h

6.18 IpxGuiPy::PyGenParamView Class Reference

`#include <IpxCameraGuiApiPy.h>`

Inherits PythonExtension< PyGenParamView >.

Public Member Functions

- void clearParams ()

  This method clears the parameters node tree view of the panel.
- void release ()

  This method destroys the IIpxGenParamTreeView class previously created.

6.18.1 Detailed Description

The PyGenParamView class represents node tree view of GenICam parameters panel

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

- IpxCameraGuiApiPy.h

6.19 IpxCamPy::PyImage Class Reference

`#include <IpxCameraApiPy.h>`

Inherits PythonExtension< PyImage >.

Public Member Functions

- IpxImage * getImage ()

  Returns IpxImage object.

6.19.1 Detailed Description

! The Image class represents the image, acquired from the camera /** Image object can be created from IpxCam↩ Py::Buffer object, and can be used to display the image on the screen or convert it to Bitmap and save it to the hard drive.

IpxCamPy::Image Python class is wrapper around C++ structure **IpxImage**

6.19.2 Member Function Documentation

6.19.2.1 getImage()

```
IpxImage* IpxCamPy::PyImage::getImage ( )
```

Returns IpxImage object.

Returns

IpxImage object.

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

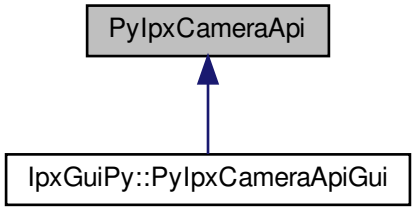
- IpxCameraApiPy.h

6.20 PyIpxCameraApi Class Reference

Python wrapper for General API.

```
#include <IpxCameraApiPy.h>
```

Inheritance diagram for PyIpxCameraApi:



Public Member Functions

- Py::Object PyIpxCreateDevice (const Py::Tuple &args)

  Creates IpxCamPy.PyDevice object.

6.20.1 Detailed Description

Python wrapper for General API.

6.20.2 Member Function Documentation

6.20.2.1 PyIpxCreateDevice()

```
Py::Object PyIpxCameraApi::PyIpxCreateDevice (
            const Py::Tuple & args )
```

Creates IpxCamPy.PyDevice object.

This method creates IpxCamPy.PyDevice object, representing the Camera

Parameters

in	*IpxCamPy.PyDeviceInfo*	object

Returns

Returns IpxCamPy::PyDevice object or None

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

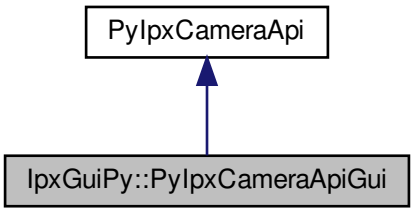
- IpxCameraApiPy.h

6.21 IpxGuiPy::PyIpxCameraApiGui Class Reference

Extension of IpxCamPy.PyIpxCameraApi class to provide the GUI functionality.

```
#include <IpxCameraGuiApiPy.h>
```

Inheritance diagram for IpxGuiPy::PyIpxCameraApiGui:



Public Member Functions

- Py::Object PyCreateDisplay (const Py::Tuple &args)
  Creates the display window.
- Py::Object PyShowImageOnDisplay (const Py::Tuple &args)
  Displays the image.
- Py::Object PyCreateGenParamTreeViewForArray (const Py::Tuple &args)
  Creates the GenICam parameters panel.
- Py::Object PyDestroyGenParamTreeView (const Py::Tuple &args)
  Destroys the GenICam parameters panel.

6.21.1 Detailed Description

Extension of IpxCamPy.PyIpxCameraApi class to provide the GUI functionality.

6.21.2 Member Function Documentation

6.21.2.1 PyCreateDisplay()

```
Py::Object IpxGuiPy::PyIpxCameraApiGui::PyCreateDisplay (
            const Py::Tuple & args )  [virtual]
```

Creates the display window.

Creates the child window to display the images acquired from the camera

Parameters

in	*args[0]*	Handle of parent window

Returns

Returns True, if window created OK, False - otherwise. Value type: Boolean

Reimplemented from PyIpxCameraApi.

6.21.2.2 PyShowImageOnDisplay()

```
Py::Object IpxGuiPy::PyIpxCameraApiGui::PyShowImageOnDisplay (
            const Py::Tuple & args )  [virtual]
```

Displays the image.

Shows the image on display window created by SystemGui::CreateDisplay method

Parameters

in	*args[0]*	IpxCamPy.PyImage object to display

Returns

None

Reimplemented from PyIpxCameraApi.

6.21.2.3 PyCreateGenParamTreeViewForArray()

```
Py::Object IpxGuiPy::PyIpxCameraApiGui::PyCreateGenParamTreeViewForArray (
            const Py::Tuple & args )  [virtual]
```

Creates the GenICam parameters panel.

Creates the panel of the camera GenICam parameters for specified IpxGenParamPy.PyGenParams object

Parameters

in	*args[0]*	IpxGenParamPy.PyGenParams object
in	*args[1]*	Handle of parent window

Returns

IpxGuiPy.PyGenParamView class object, representing the panel created

Reimplemented from PyIpxCameraApi.

6.21.2.4 PyDestroyGenParamTreeView()

```
Py::Object IpxGuiPy::PyIpxCameraApiGui::PyDestroyGenParamTreeView (
            const Py::Tuple & args )  [virtual]
```

Destroys the GenICam parameters panel.

Closes and destroys the GenICam parameters panel, previously created with PyCreateGenParamTreeViewForArray method

Parameters

in	*args[0]*	IpxGenParamPy.PyGenParams object

Returns

Returns True, operation completed successfully, False - otherwise. Value type: Boolean

Reimplemented from PyIpxCameraApi.

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

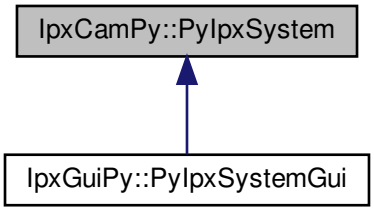
- IpxCameraGuiApiPy.h

6.22 IpxCamPy::PyIpxSystem Class Reference

The PyIpxSystem class represents an abstraction of the System module of the GenTL module hierarchy.

```
#include <IpxCameraApiPy.h>
```

Inheritance diagram for IpxCamPy::PyIpxSystem:



Public Member Functions

- Py::Object GetInterfaceList ()

  This method returns the list of all the interfaces of the system object.
- Py::Object GetDisplayName ()

  Returns the name of the GenTL Producer.
- Py::Object GetVersion ()

  Returns the GenTL Producer version.
- Py::Object CreateDeviceFromConfig (const Py::Tuple &args)

  Creates the Device object from configuration file.

6.22.1 Detailed Description

The PyIpxSystem class represents an abstraction of the System module of the GenTL module hierarchy.

This class provides member functions to enumerate and instantiate the available interfaces reachable. It also provides a method for the configuration of the device module. This system module is the root of the GenTL Module hierarchy. IpxCamPy.PyIpxSystem class has member functions to find all the interfaces, display the user readable name and producer version of the GenTL system. The IpxCamPy.PyIpxSystem class can be used to obtain the list of PyDevice↩ Interface objects, then get the list of IpxCamPy.PyDeviceInfo objects for the devices connected to the Interface, and create IpxCamPy.PyDevice object, representing the camera device.

IpxCamPy.PyIpxSystem Python class is wrapper around C++ class **IpxCam::System**

6.22.2 Member Function Documentation

6.22.2.1 GetInterfaceList()

```
Py::Object IpxCamPy::PyIpxSystem::GetInterfaceList ( )
```

This method returns the list of all the interfaces of the system object.

GetInterfaceList method lists all the available hardware interfaces with the transport layers technologies, supported by GenTL producer library

Returns

Returns the list of IpxCamPy.PyDeviceInterface objects

6.22.2.2 GetDisplayName()

```
Py::Object IpxCamPy::PyIpxSystem::GetDisplayName ( )
```

Returns the name of the GenTL Producer.

This method returns the User readable name of the GenTL Producer of the system object.

Returns

Returns the system display name. Value type: String

6.22.2.3 GetVersion()

```
Py::Object IpxCamPy::PyIpxSystem::GetVersion ( )
```

Returns the GenTL Producer version.

This method returns the version of the GenTL Producer of the system object.

Returns

Returns the System version. Value type: String

6.22.2.4 CreateDeviceFromConfig()

```
Py::Object IpxCamPy::PyIpxSystem::CreateDeviceFromConfig (
            const Py::Tuple & args )
```

Creates the Device object from configuration file.

This method creates, configures and sets up the device using the information retrieved from the specified configuration file

Parameters

in	*args[0]*	Configuration file to open. Value type: String

Returns

Returns IpxCamPy.PyDevice object or None if device cannot be instantiated

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

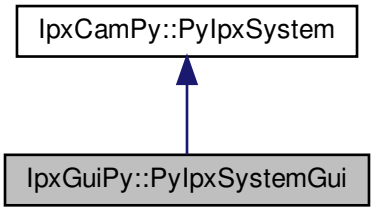
- IpxCameraApiPy.h

6.23 IpxGuiPy::PyIpxSystemGui Class Reference

Extension of IpxCamPy.PyIpxSystem class to provide the GUI functionality.

`#include <IpxCameraGuiApiPy.h>`

Inheritance diagram for IpxGuiPy::PyIpxSystemGui:



Public Member Functions

- virtual Py::Object SelectCamera (const Py::Tuple &args)

  Creates the camera selection dialog.

6.23.1 Detailed Description

Extension of IpxCamPy.PyIpxSystem class to provide the GUI functionality.

6.23.2 Member Function Documentation

6.23.2.1 SelectCamera()

```
virtual Py::Object IpxGuiPy::PyIpxSystemGui::SelectCamera (
            const Py::Tuple & args )  [virtual]
```

Creates the camera selection dialog.

This method creates the modal device discovery dialog to select the camera and obtain the PyDeviceInfo object. PyDeviceInfo object can be used to create PyDevice object, representing the camera device

Parameters

in	*args[0]*	- Handle of parent window
in	*args[1]*	- True if device discovery polling is ON by default, False - polling is OFF. Value type: Boolean.

Returns

Returns IpxCamPy.PyDeviceInfo object or None, if nothing selected

Reimplemented from IpxCamPy::PyIpxSystem.

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

- IpxCameraGuiApiPy.h

Index