

Novitec Camera API

Generated by Doxygen 1.9.3

1 Introduction to the C++ interface reference	1
1.1 Introduction	1
1.2 Image acquisition	1
1.3 Feature control	2
1.3.1 Feature error handling	4
2 Hierarchical Index	5
2.1 Class Hierarchy	5
3 Class Index	7
3.1 Class List	7
4 Class Documentation	9
4.1 novitec::CameraAPI::Camera Class Reference	9
4.1.1 Detailed Description	10
4.1.2 Member Function Documentation	10
4.1.2.1 Connect()	10
4.1.2.2 Disconnect()	11
4.1.2.3 GetBufferMode()	11
4.1.2.4 GetImage()	11
4.1.2.5 GetXML()	12
4.1.2.6 IsConnected()	12
4.1.2.7 ReadMemory()	12
4.1.2.8 ReadRegister()	13
4.1.2.9 SetBufferMode()	13
4.1.2.10 SetDeviceEventCallback()	14
4.1.2.11 SetImageCallback()	14
4.1.2.12 Start()	14
4.1.2.13 Stop()	15
4.1.2.14 UpdateFirmware()	15
4.1.2.15 WriteMemory()	15
4.1.2.16 WriteRegister()	16
4.2 novitec::CameraAPI::CameraInfo Class Reference	16
4.2.1 Detailed Description	17
4.2.2 Constructor & Destructor Documentation	17
4.2.2.1 CameraInfo()	17
4.2.3 Member Function Documentation	17
4.2.3.1 GetAccessStatus()	17
4.2.3.2 GetDeviceVersion()	17
4.2.3.3 GetDisplayName()	18
4.2.3.4 GetID()	18
4.2.3.5 GetManufacturer()	18
4.2.3.6 GetModelName()	18

4.2.3.7 GetSerialNumber()	19
4.2.3.8 GetTimestampFrequency()	19
4.2.3.9 GetTransportLayer()	19
4.2.3.10 GetUserDefinedName()	19
4.3 novitec::CameraAPI::GenICam::CameraInterface Class Reference	20
4.3.1 Detailed Description	21
4.3.2 Constructor & Destructor Documentation	21
4.3.2.1 CameraInterface()	21
4.3.3 Member Function Documentation	21
4.3.3.1 GetFeature()	21
4.3.3.2 GetFeatureType() [1/2]	21
4.3.3.3 GetFeatureType() [2/2]	22
4.3.3.4 GetFirstChildFeature()	22
4.3.3.5 GetNumberOfInvalidator()	22
4.3.3.6 Open()	23
4.3.3.7 OpenFromMem()	23
4.4 novitec::CameraAPI::DeviceBase Class Reference	23
4.4.1 Detailed Description	24
4.4.2 Member Function Documentation	24
4.4.2.1 Connect()	24
4.4.2.2 Disconnect()	25
4.4.2.3 GetImage()	25
4.4.2.4 GetInterfaceType()	25
4.4.2.5 IsConnected()	26
4.4.2.6 SetImageCallback()	26
4.4.2.7 Start()	26
4.4.2.8 Stop()	26
4.5 novitec::CameraAPI::DeviceManager Class Reference	27
4.5.1 Detailed Description	27
4.5.2 Member Function Documentation	27
4.5.2.1 GetCameraHandle()	27
4.5.2.2 GetCameraHandleByIPAddress()	28
4.5.2.3 GetCameraHandleBySerial()	28
4.5.2.4 GetCameraInfo() [1/4]	28
4.5.2.5 GetCameraInfo() [2/4]	29
4.5.2.6 GetCameraInfo() [3/4]	29
4.5.2.7 GetCameraInfo() [4/4]	29
4.5.2.8 GetNumberOfCameras()	30
4.5.2.9 SendForceIP()	30
4.6 novitec::CameraAPI::Error Class Reference	30
4.6.1 Detailed Description	31
4.6.2 Constructor & Destructor Documentation	31

4.6.2.1 Error()	31
4.6.3 Member Function Documentation	31
4.6.3.1 GetDescription()	31
4.7 novitec::CameraAPI::Exception Class Reference	32
4.7.1 Detailed Description	32
4.7.2 Constructor & Destructor Documentation	32
4.7.2.1 Exception()	32
4.8 novitec::CameraAPI::GATEWAY Struct Reference	33
4.8.1 Detailed Description	33
4.9 novitec::CameraAPI::GEVCameraInfo Class Reference	33
4.9.1 Detailed Description	35
4.10 novitec::CameraAPI::Handle Class Reference	35
4.10.1 Detailed Description	35
4.10.2 Member Function Documentation	35
4.10.2.1 GetDescriptor()	35
4.11 novitec::CameraAPI::HostControllerInfo Class Reference	36
4.11.1 Detailed Description	36
4.11.2 Constructor & Destructor Documentation	36
4.11.2.1 HostControllerInfo()	36
4.12 novitec::CameraAPI::GenICam::IBoolean Class Reference	36
4.12.1 Detailed Description	37
4.13 novitec::CameraAPI::GenICam::ICategory Class Reference	37
4.13.1 Detailed Description	37
4.14 novitec::CameraAPI::GenICam::ICommand Class Reference	38
4.14.1 Detailed Description	38
4.15 novitec::CameraAPI::GenICam::IEnumEntry Class Reference	38
4.15.1 Detailed Description	39
4.16 novitec::CameraAPI::GenICam::IEnumeration Class Reference	39
4.16.1 Detailed Description	39
4.17 novitec::CameraAPI::GenICam::IFloat Class Reference	40
4.17.1 Detailed Description	40
4.17.2 Member Function Documentation	40
4.17.2.1 GetInc()	41
4.17.2.2 GetMax()	41
4.17.2.3 GetMin()	41
4.17.2.4 GetNumberOfValidValues()	41
4.18 novitec::CameraAPI::GenICam::Integer Class Reference	42
4.18.1 Detailed Description	42
4.18.2 Member Function Documentation	42
4.18.2.1 GetCacheValue()	43
4.18.2.2 GetInc()	43
4.18.2.3 GetMax()	43

4.18.2.4	GetMin()	43
4.18.2.5	GetNumberOfValidValues()	44
4.18.2.6	GetValue()	44
4.18.2.7	SetValue()	44
4.19	novitec::CameraAPI::Image Class Reference	44
4.19.1	Detailed Description	46
4.19.2	Member Function Documentation	46
4.19.2.1	Convert()	46
4.19.2.2	Copy() [1/2]	46
4.19.2.3	Copy() [2/2]	46
4.19.2.4	Create()	47
4.19.2.5	CreateJPEG()	47
4.19.2.6	Load()	48
4.19.2.7	Save()	48
4.20	novitec::CameraAPI::IPConfiguration Struct Reference	48
4.20.1	Detailed Description	49
4.21	novitec::CameraAPI::GenlCam::IString Class Reference	49
4.21.1	Detailed Description	49
4.22	novitec::CameraAPI::GenlCam::ITypeBase Class Reference	50
4.22.1	Detailed Description	51
4.22.2	Constructor & Destructor Documentation	51
4.22.2.1	ITypeBase()	51
4.22.3	Member Function Documentation	52
4.22.3.1	GetElement()	52
4.23	novitec::CameraAPI::MAC_ADDRESS Struct Reference	52
4.23.1	Detailed Description	52
4.24	novitec::CameraAPI::NetworkAdapterInfo Class Reference	52
4.24.1	Detailed Description	53
4.24.2	Member Function Documentation	53
4.24.2.1	GetAdapterName()	53
4.24.2.2	GetDescription()	53
4.24.2.3	GetFriendlyName()	54
4.24.2.4	GetGateway()	54
4.24.2.5	GetMACAddress()	54
4.24.2.6	GetNumberOfGateways()	54
4.24.2.7	GetNumberOfSubnets()	55
4.24.2.8	GetSubnet()	55
4.25	novitec::CameraAPI::NU3CameraInfo Class Reference	55
4.25.1	Detailed Description	56
4.26	novitec::CameraAPI::PixelFormatUtil Class Reference	56
4.26.1	Detailed Description	56
4.26.2	Member Function Documentation	56

4.26.2.1 GetBitsPerPixel()	56
4.26.2.2 GetBytesPerPixel()	57
4.26.2.3 GetPixelFormatDescription()	57
4.26.2.4 GetPixelFormatName()	57
4.26.2.5 Is3DFormat()	58
4.27 novitec::CameraAPI::SUBNET Struct Reference	58
4.27.1 Detailed Description	58
4.28 novitec::CameraAPI::U3VCameraInfo Class Reference	59
4.28.1 Detailed Description	59
Index	61

Chapter 1

Introduction to the C++ interface reference

1.1 Introduction

This is the documentation for developers who want to work with the C++ interface of Novitec Camera API.

1.2 Image acquisition

Every program written using this interface will start in one or the other form with creating an instance of the class [novitec::CameraAPI::DeviceManager](#). Each application needs at least one instance of this class while devices shall be accessed. To find out how to gain access to a certain device look at the detailed description of this class.

Once a pointer to the desired device represented by an instance of the class [novitec::CameraAPI::Handle](#) has been obtained every other device related properties or functions can be accessed.

Some source code samples how to locate a certain [novitec::CameraAPI::Handle](#) also can be found in the detailed description of the class [novitec::CameraAPI::DeviceManager](#).

To capture images, [novitec::CameraAPI::Camera](#) instance must be created. This class can be constructed by passing a pointer to the [novitec::Handle](#) object obtained from the [novitec::CameraAPI::Devicemanager](#) to the class constructor.

The [novitec::CameraAPI::Camera](#) class provides basically control function for image acquisition. Getting the first image might e.g. look like that:

```
using namespace novitec::CameraAPI;
#define CHK_ERR(function) \
    err = function; \
    if(err != NVT_OK){ \
        printf("%s [%s:%d]\n", err.GetDescription(), __FILE__, __LINE__); \
        return -1; \
    } \
int main(int argc, char* argv[])
{
    DeviceManager manager;
    Handle deviceHandle;
    Camera camera;
    Error err;
    unsigned int numCameras;
    CHK_ERR(manager.Update()); // Update device list. If this function is called, it will search devices on
        whole interfaces.
    CHK_ERR(manager.GetCameraHandle(0, &deviceHandle)); // Get index 0 camera handle from device manager.
    CHK_ERR(camera.Connect(&deviceHandle)); // Connect camera
    CHK_ERR(camera.Start()); // Start image grab process.
    int grabbedImages = 0;
    for (int i = 0; i < 100; i++)
    {
        Image image;
        err = camera.GetImage(&image, 1000); // Get image from frame buffer.
        if (err != NVT_OK)
```

```

    {
        printf("Error - %s\n", err.GetDescription());
        continue;
    }
    printf("image %d | size = %d, data = %02x %02x %02x %02x ... \n",
        grabbedImages++, image.dataSize,
        image.data[0], image.data[1], image.data[2], image.data[3]);
}
CHK_ERR(camera.Stop()); // Stop the image grab process.
CHK_ERR(camera.Disconnect()); // Disconnect camera.
return 0;
}

```

This sample contains everything the user needs to do to capture one image including all initialization work and error handling for every source of error one can think of. Several example codes will provide an even better understanding of the interface.

1.3 Feature control

To control camera features, [novitec::CameraAPI::GenICam::CameraInterface](#) instance must be created. This class can be constructed from pointer of the [novitec::CameraAPI::Camera](#) object.

API follows GenICam standard, and functions are controlled by descriptor in XML format defined as GenAPI.

When creating a [novitec::CameraAPI::GenICam::CameraInterface](#) instance, the XML descriptor is loaded from the device and create a feature tree. User can search with the provided functions.

To control each features, It can be created by specifying the name and type of the feature by [novitec::CameraAPI::GenICam::CameraInterface](#) function.

```

#include "NovitecCameraAPI.h"
#include <deque>
using namespace novitec::CameraAPI;
using namespace novitec::CameraAPI::GenICam;
#define CHK_ERR(function) \
    err = function; \
    if(err != NVT_OK){ \
        printf("%s [%s:%d]\n", err.GetDescription(), __FILE__, __LINE__); \
        return -1; \
    } \
void printFeatureTree(CameraInterface* camIf, FeatureHandle handle, int level);
int main(int argc, char* argv[])
{
    DeviceManager manager;
    Handle deviceHandle;
    Camera camera;
    Error err;
    unsigned int numCameras;
    CHK_ERR(manager.Update()); // Update device list. If this function is called, it will search devices on
    whole inferfaces.
#ifdef 1
    CHK_ERR(manager.GetNumberOfCameras(&numCameras)); // Get number of cameras.
    for (int i = 0; i < numCameras; i++)
    {
        CameraInfo info;
        manager.GetCameraInfo(i, &info);
        printf("%d | %s - %s\n", i, info.GetModelName(), info.GetSerialNumber());
    }
    printf("type camera index > ");
    int camIdx = -1;
    scanf("%d", &camIdx);
    CHK_ERR(manager.GetCameraHandle(camIdx, &deviceHandle)); // Get camera handle from device manager.
#else
    // get camera handle by serial number
    CHK_ERR(manager.GetCameraHandleBySerial("ESB20049", &deviceHandle)); // Get camera handle by serial
    number.
#endif
    CHK_ERR(camera.Connect(&deviceHandle)); // Connect camera

    // feature control
    // 0. Get camera interface
    CameraInterface camIf(&camera);
    // 1. Basic instruction
    {
        // - exposure time read
        IInteger exposureTime = camIf.GetFeature<IInteger>("ExposureTime");
        int exposureTimeVal = exposureTime.GetValue();
        printf("current exposure time = %d us\n", exposureTimeVal);
        // - exposure time wrtie
    }
}

```

```

    exposureTime.SetValue(5000);
    const char* unit = exposureTime.GetUnit();
    printf("unit = %s\n", unit);
}
// 2. Feature finding
{
    // - get available features, refer to printFeatureTree() function
    printf("Camera feature list - \n");
    printFeatureTree(&camIf, FEATURE_HANDLE_ROOT, 0);
}
// 3. Get feature information
{
    // - description

    IInteger exposureTime = camIf.GetFeature<IInteger>("ExposureTime");
    printf("\n\n" \
        "feature name = %s\n" \
        "display name = %s\n" \
        "description = %s\n" \
        ,
        exposureTime.GetFeatureName(),
        exposureTime.GetDisplayName(),
        exposureTime.GetDescription());

    // - accessibility
    novitec::CameraAPI::GenICam::AccessMode accessMode = exposureTime.GetAccessMode(); // ReadWrite,
    ReadOnly or WriteOnly
    // - range
    if(exposureTime.HasMin() && exposureTime.HasMax())
    {
        double min = exposureTime.GetMin();
        double max = exposureTime.GetMax();
        printf("value range = %f - %f\n\n", min, max);
    }
}
// 4. Exception processing
{
    // - type mismatching
    try
    {
        IInteger exposureTime = camIf.GetFeature<IInteger>("ExposureTime");
    }
    catch (Exception e)
    {
        printf(e.errorMessage);
    }
}
// 5. Feature types
{
    try
    {
        // - IInteger
        IInteger triggerDebounceActive = camIf.GetFeature<IInteger>("TriggerDebounceActive");
        triggerDebounceActive.SetValue(10);
        int triggerDebounceActiveValue = triggerDebounceActive.GetValue();
        // - IBoolean
        IBoolean lineInverter = camIf.GetFeature<IBoolean>("LineInverter");
        lineInverter.SetValue(true);
        bool lineInverterValue = lineInverter.GetValue();
        // - IFloat
        IFloat gain = camIf.GetFeature<IFloat>("Gain");
        gain.SetValue(1.0);
        double gainValue = gain.GetValue();
        // - IString
        IString serialNumber = camIf.GetFeature<IString>("DeviceSerialNumber");
        const char* serialNumberValue = serialNumber.GetValue();
        // - IEnumeration
        IEnumeration filterSwitcher = camIf.GetFeature<IEnumeration>("FilterSwitcher");
        const char* filterSwitcherValue = filterSwitcher.GetSymbolicValue();
        int numEntries = filterSwitcher.GetNumberOfEntries();
        for (int i = 0; i < numEntries; i++) // get available entry
        {
            auto entry = filterSwitcher.GetEntry(i);
            const char* symbolic = entry.GetFeatureName();
            int value = entry.GetValue();
            printf("entry %d - %s\n", value, symbolic);
        }
        filterSwitcher.SetSymbolicValue("IRCutFilter"); // by symbolic value
        filterSwitcher.SetIntValue(0); // by entry value
        // - ICommand
        ICommand triggerSoftware = camIf.GetFeature<ICommand>("TriggerSoftware");
        triggerSoftware.Execute();
    }
    catch (Exception e)
    {
        printf(e.errorMessage);
    }
}

```

```
    }
    CHK_ERR(camera.Disconnect()); // Disconnect camera
}
void printFeatureTree(CameraInterface* camIf, FeatureHandle handle, int level)
{
    FeatureHandle _hs = camIf->GetFirstChildFeature(handle);
    while (_hs)
    {
        for (int i = 0; i < level; i++)
            printf(" ");
        if (level > 0)
            printf("- ");
        printf("%s [%s]\n", camIf->GetFeatureName(_hs), camIf->GetFeatureType(_hs));
        printFeatureTree(camIf, _hs, level + 1);
        _hs = camIf->GetNextSiblingFeature(_hs);
    }
}
```

1.3.1 Feature error handling

When handling features, different errors can occur: feature read/write fail, feature handles with incorrect type or no such feature exists. When an error occurs, API will throw an [novitec::CameraAPI::Exception](#). You should handle exceptions by try, catch keyword.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

novitec::CameraAPI::CameraInfo	16
novitec::CameraAPI::GenlCam::CameraInterface	20
novitec::CameraAPI::DeviceBase	23
novitec::CameraAPI::Camera	9
novitec::CameraAPI::DeviceManager	27
novitec::CameraAPI::Error	30
novitec::CameraAPI::Exception	32
novitec::CameraAPI::GATEWAY	33
novitec::CameraAPI::GEVCameraInfo	33
novitec::CameraAPI::Handle	35
novitec::CameraAPI::HostControllerInfo	36
novitec::CameraAPI::Image	44
novitec::CameraAPI::IPConfiguration	48
novitec::CameraAPI::GenlCam::ITypeBase	50
novitec::CameraAPI::GenlCam::IBoolean	36
novitec::CameraAPI::GenlCam::ICategory	37
novitec::CameraAPI::GenlCam::ICommand	38
novitec::CameraAPI::GenlCam::IEnumEntry	38
novitec::CameraAPI::GenlCam::IEnumeration	39
novitec::CameraAPI::GenlCam::IFloat	40
novitec::CameraAPI::GenlCam::IInteger	42
novitec::CameraAPI::GenlCam::IString	49
novitec::CameraAPI::MAC_ADDRESS	52
novitec::CameraAPI::NetworkAdapterInfo	52
novitec::CameraAPI::NU3CameraInfo	55
novitec::CameraAPI::PixelFormatUtil	56
novitec::CameraAPI::SUBNET	58
novitec::CameraAPI::U3VCameraInfo	59

Chapter 3

Class Index

3.1 Class List

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

novitec::CameraAPI::Camera	
The class which provides the camera control	9
novitec::CameraAPI::CameraInfo	
GenICam Compliant Information	16
novitec::CameraAPI::GenICam::CameraInterface	
The class that contains information about CameraInterface	20
novitec::CameraAPI::DeviceBase	
The class which provides the device control	23
novitec::CameraAPI::DeviceManager	
The class which provides the device management	27
novitec::CameraAPI::Error	
The class for errors in API functions	30
novitec::CameraAPI::Exception	
The class that throws exception for Novitec Camera API	32
novitec::CameraAPI::GATEWAY	
Gateway Structure	33
novitec::CameraAPI::GEVCameraInfo	
GigE Vision camera Information	33
novitec::CameraAPI::Handle	
The class which provides handle of device	35
novitec::CameraAPI::HostControllerInfo	
Host Controller Information	36
novitec::CameraAPI::GenICam::IBoolean	
IBoolean class	36
novitec::CameraAPI::GenICam::ICategory	
ICategory class	37
novitec::CameraAPI::GenICam::ICommand	
ICommand class	38
novitec::CameraAPI::GenICam::IEnumEntry	
IEnumEntry class	38
novitec::CameraAPI::GenICam::IEnumeration	
IEnumeration class	39
novitec::CameraAPI::GenICam::IFloat	
IFloat class	40
novitec::CameraAPI::GenICam::IInteger	
IInteger class	42

novitec::CameraAPI::Image	
The class for image	44
novitec::CameraAPI::IPConfiguration	
GigE Vision Device IP Configurations	48
novitec::CameraAPI::GenICam::IString	
IString class	49
novitec::CameraAPI::GenICam::ITypeBase	
The class on which the supported types are based	50
novitec::CameraAPI::MAC_ADDRESS	
MAC Address Structure	52
novitec::CameraAPI::NetworkAdapterInfo	
Network Adapter Information	52
novitec::CameraAPI::NU3CameraInfo	
Novitec USB3.0 Camera Information	55
novitec::CameraAPI::PixelFormatUtil	
The class which provides the utils for PixelFormat	56
novitec::CameraAPI::SUBNET	
Subnet Structure	58
novitec::CameraAPI::U3VCameraInfo	
USB3.0 Vision Camera Information	59

Chapter 4

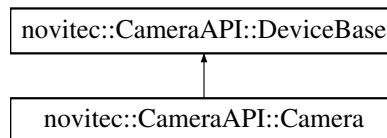
Class Documentation

4.1 novitec::CameraAPI::Camera Class Reference

The class which provides the camera control.

```
#include <Camera.h>
```

Inheritance diagram for novitec::CameraAPI::Camera:



Public Member Functions

- **Camera** (void)
Constructor.
- virtual **~Camera** (void)
Destructor.
- virtual **novitec::CameraAPI::Error Connect** (novitec::CameraAPI::Handle *pHandle, novitec::CameraAPI::AccessMode accessMode=novitec::CameraAPI::AM_CONTROL)
Connect to a camera.
- virtual **novitec::CameraAPI::Error Disconnect** (void)
Disconnect a camera.
- virtual bool **IsConnected** (void)
Get the connection status to a camera.
- virtual **novitec::CameraAPI::Error Start** (void)
Start grabbing images from a camera.
- virtual **novitec::CameraAPI::Error Stop** (void)
Stop grabbing images from a camera.
- virtual **novitec::CameraAPI::Error GetImage** (Image *pImage, unsigned int uiTimeout=500)
Get an image from the frame buffer.
- virtual **novitec::CameraAPI::Error SetImageCallback** (ImageCallbackFunc func, void *param)
Set an image callback function.

- char * [GetXML](#) (bool forceUpdate=false)
Get XML Register Descriptor.
- virtual [novitec::CameraAPI::Error ReadRegister](#) (unsigned int address, unsigned int *value)
Read register from device.
- virtual [novitec::CameraAPI::Error WriteRegister](#) (unsigned int address, unsigned int value)
Write register to device.
- virtual [novitec::CameraAPI::Error ReadMemory](#) (unsigned int address, void *value, unsigned int length)
Read memory from device.
- virtual [novitec::CameraAPI::Error WriteMemory](#) (unsigned int address, void *value, unsigned int length)
Write memory to device.
- virtual [novitec::CameraAPI::Error SetDeviceEventCallback](#) (DeviceEventCallbackFunc func, void *param)
Set device event callback function.
- virtual [novitec::CameraAPI::Error SetBufferMode](#) (novitec::CameraAPI::BufferMode bufferMode, int num↔ Buffers)
Set frame buffer mode.
- virtual [novitec::CameraAPI::Error GetBufferMode](#) (novitec::CameraAPI::BufferMode &bufferMode, int &numBuffers)
Get frame buffer mode.
- virtual [novitec::CameraAPI::Error UpdateFirmware](#) (unsigned char *data, int length)
Update firmware.

Additional Inherited Members

4.1.1 Detailed Description

The class which provides the camera control.

4.1.2 Member Function Documentation

4.1.2.1 Connect()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::Connect (
    novitec::CameraAPI::Handle * pHandle,
    novitec::CameraAPI::AccessMode accessMode = novitec::CameraAPI::AM\_CONTROL )
[virtual]
```

Connect to a camera.

Parameters

in	<i>pHandle</i>	Pointer to the camera handle.
in	<i>accessMode</i>	Set access mode of camera.

See also

[DeviceManager::GetCameraHandle\(\)](#)

Returns

[novitec::CameraAPI::Error](#)

Implements [novitec::CameraAPI::DeviceBase](#).

4.1.2.2 Disconnect()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::Disconnect (
    void ) [virtual]
```

Disconnect a camera.

Returns

[novitec::CameraAPI::Error](#)

Implements [novitec::CameraAPI::DeviceBase](#).

4.1.2.3 GetBufferMode()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::GetBufferMode (
    novitec::CameraAPI::BufferMode & bufferMode,
    int & numBuffers ) [virtual]
```

Get frame buffer mode.

Parameters

out	<i>bufferMode</i>	
out	<i>numBuffers</i>	- number of frame buffers

4.1.2.4 GetImage()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::GetImage (
    Image * pImage,
    unsigned int uiTimeout = 500 ) [virtual]
```

Get an image from the frame buffer.

Parameters

out	<i>pImage</i>	Image buffer to store the image.
in	<i>uiTimeout</i>	Timeout (ms)

Returns

[novitec::CameraAPI::Error](#)

Implements [novitec::CameraAPI::DeviceBase](#).

4.1.2.5 GetXML()

```
char * novitec::CameraAPI::Camera::GetXML (
    bool forceUpdate = false )
```

Get XML Register Descriptor.

Returns

Return character pointer about XML Register Descriptor. If it fails, return NULL.
This pointer validates before destructing class or calling next [GetXML\(\)](#).

4.1.2.6 IsConnected()

```
virtual bool novitec::CameraAPI::Camera::IsConnected (
    void ) [virtual]
```

Get the connection status to a camera.

Returns

Returns true when camera is conneted or false.

Implements [novitec::CameraAPI::DeviceBase](#).

4.1.2.7 ReadMemory()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::ReadMemory (
    unsigned int address,
    void * value,
    unsigned int length ) [virtual]
```

Read memory from device.

Parameters

in	<i>address</i>	- register address
out	<i>value</i>	- register value
in	<i>length</i>	- memory length

Returns

[novitec::CameraAPI::Error](#)

4.1.2.8 ReadRegister()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::ReadRegister (
    unsigned int address,
    unsigned int * value ) [virtual]
```

Read register from device.

Parameters

in	<i>address</i>	- register address
out	<i>value</i>	- register value

Returns

[novitec::CameraAPI::Error](#)

4.1.2.9 SetBufferMode()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::SetBufferMode (
    novitec::CameraAPI::BufferMode bufferMode,
    int numBuffers ) [virtual]
```

Set frame buffer mode.

Parameters

in	<i>bufferMode</i>	Buffer mode
in	<i>numBuffers</i>	number of frame buffers

4.1.2.10 SetDeviceEventCallback()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::SetDeviceEventCallback (
    DeviceEventCallbackFunc func,
    void * param ) [virtual]
```

Set device event callback function.

Returns

[novitec::CameraAPI::Error](#)

4.1.2.11 SetImageCallback()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::SetImageCallback (
    ImageCallbackFunc func,
    void * param ) [virtual]
```

Set an image callback function.

Parameters

<i>func</i>	Pointer to a function which is called by internal acquisition thread when new image is acquired.
-------------	--

Warning

It must be called before [Camera::Start\(\)](#)

Returns

[novitec::CameraAPI::Error](#)

Implements [novitec::CameraAPI::DeviceBase](#).

4.1.2.12 Start()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::Start (
    void ) [virtual]
```

Start grabbing images from a camera.

Returns

[novitec::CameraAPI::Error](#)

Implements [novitec::CameraAPI::DeviceBase](#).

4.1.2.13 Stop()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::Stop (
    void ) [virtual]
```

Stop grabbing images from a camera.

Returns

[novitec::CameraAPI::Error](#)

Implements [novitec::CameraAPI::DeviceBase](#).

4.1.2.14 UpdateFirmware()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::UpdateFirmware (
    unsigned char * data,
    int length ) [virtual]
```

Update firmware.

Parameters

in	<i>data</i>	- binary firmware data
in	<i>length</i>	- firmware data length

4.1.2.15 WriteMemory()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::WriteMemory (
    unsigned int address,
    void * value,
    unsigned int length ) [virtual]
```

Write memory to device.

Parameters

in	<i>address</i>	- register address
in	<i>value</i>	- register value
in	<i>length</i>	- memory length

Returns

[novitec::CameraAPI::Error](#)

4.1.2.16 WriteRegister()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::Camera::WriteRegister (
    unsigned int address,
    unsigned int value ) [virtual]
```

Write register to device.

Parameters

in	<i>address</i>	- register address
in	<i>value</i>	- register value

Returns

[novitec::CameraAPI::Error](#)

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

- Camera.h

4.2 novitec::CameraAPI::CameraInfo Class Reference

GenlCam Compliant Information.

```
#include <Types.h>
```

Public Member Functions

- **CameraInfo** ()
Constructor.
- [CameraInfo](#) (const [CameraInfo](#) &obj)
Constructor.
- **~CameraInfo** ()
Destructor.
- const char * [GetID](#) ()
Get unique ID of the device.
- const char * [GetManufacturer](#) ()
Get device vendor name.
- const char * [GetModelName](#) ()
Get device model name.
- const char * [GetTransportLayer](#) ()
Get transport layer technology that is supported.
- const char * [GetDisplayName](#) ()
Get display name for the device (including a unique ID).
- int [GetAccessStatus](#) ()
- const char * [GetUserDefinedName](#) ()
Get user defined name. (GenTL v1.4)
- const char * [GetSerialNumber](#) ()
Get the device's serial number. (GenTL v1.4)
- const char * [GetDeviceVersion](#) ()
Get the device version. (GenTL v1.4)
- unsigned long long int [GetTimestampFrequency](#) ()
Get tick-frequency of the timestamp clock. (GenTL v1.4)

4.2.1 Detailed Description

GenICam Compliant Information.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 CameraInfo()

```
novitec::CameraAPI::CameraInfo::CameraInfo (
    const CameraInfo & obj )
```

Constructor.

Parameters

<i>obj</i>	The CameraInfo from which to create the new CameraInfo .
------------	--

4.2.3 Member Function Documentation

4.2.3.1 GetAccessStatus()

```
int novitec::CameraAPI::CameraInfo::GetAccessStatus ( )
```

Gets the access mode the GenTL Producer has on the opened device. (DEVICE_ACCESS_STATUS enumeration value).

Returns

Int that represents access mode the GenTL Producer has on the opened device.

4.2.3.2 GetDeviceVersion()

```
const char * novitec::CameraAPI::CameraInfo::GetDeviceVersion ( )
```

Get the device version. (GenTL v1.4)

Returns

String that represents the device version. (GenTL v1.4)

4.2.3.3 GetDisplayName()

```
const char * novitec::CameraAPI::CameraInfo::GetDisplayName ( )
```

Get display name for the device (including a unique ID).

Returns

String that represents display name for the device (including a unique ID).

4.2.3.4 GetID()

```
const char * novitec::CameraAPI::CameraInfo::GetID ( )
```

Get unique ID of the device.

Returns

String that represents unique ID of the device.

4.2.3.5 GetManufacturer()

```
const char * novitec::CameraAPI::CameraInfo::GetManufacturer ( )
```

Get device vendor name.

Returns

String that represents device vendor name.

4.2.3.6 GetModelName()

```
const char * novitec::CameraAPI::CameraInfo::GetModelName ( )
```

Get device model name.

Returns

String that represents device model name.

4.2.3.7 GetSerialNumber()

```
const char * novitec::CameraAPI::CameraInfo::GetSerialNumber ( )
```

Get the device's serial number. (GenTL v1.4)

Returns

String that represents the device's serial number. (GenTL v1.4)

4.2.3.8 GetTimestampFrequency()

```
unsigned long long int novitec::CameraAPI::CameraInfo::GetTimestampFrequency ( )
```

Get tick-frequency of the timestamp clock. (GenTL v1.4)

Returns

Long long that represents the tick-frequency of the timestamp clock. (GenTL v1.4)

4.2.3.9 GetTransportLayer()

```
const char * novitec::CameraAPI::CameraInfo::GetTransportLayer ( )
```

Get transport layer technology that is supported.

Returns

String that represents transport layer technology that is supported.

4.2.3.10 GetUserDefinedName()

```
const char * novitec::CameraAPI::CameraInfo::GetUserDefinedName ( )
```

Get user defined name. (GenTL v1.4)

Returns

String that represents user defined name. (GenTL v1.4)

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

- Types.h

4.3 novitec::CameraAPI::GenlCam::CameraInterface Class Reference

The class that contains information about [CameraInterface](#).

```
#include <CameraInterface.h>
```

Public Member Functions

- **CameraInterface** ()
Constructs a new [novitec::CameraAPI::GenlCam::CameraInterface](#) object.
- **CameraInterface** ([novitec::CameraAPI::Camera](#) *device)
Constructs a new [novitec::CameraAPI::GenlCam::CameraInterface](#) object.
- virtual **CameraInterface** & **operator=** (const **CameraInterface** &rhs)
- int **Open** (const char *XMLFilePath)
Open camera interface from an XML file.
- int **OpenFromMem** (const char *xml)
Open camera interface from an XML on the memory.
- int **Close** ()
Close camera interface.
- FeatureHandle **GetFirstChildFeature** (FeatureHandle handle=FEATURE_HANDLE_ROOT)
Get first child feature handle.
- FeatureHandle **GetNextSiblingFeature** (FeatureHandle handle)
Get next sibling feature handle.
- const char * **GetFeatureName** (FeatureHandle handle)
Get the interface type of the specified feature.
- const char * **GetFeatureType** (FeatureHandle handle)
Get the interface type of the specified feature.
- const char * **GetFeatureType** (const char *featureName)
Get the interface type of the specified feature by feature name.
- template<typename TFeature >
TFeature **GetFeature** (FeatureHandle handle)
Get a feature by handle.
- template<typename TFeature >
TFeature **GetFeature** (const char *featureName)
Get a feature by feature name.
- unsigned int **GetNumberOfInvalidator** (const char *featureName)
Get invalidator for update linked features.
- const char * **GetInvalidator** (const char *featureName, int idx)
- unsigned int **GetNumberOfLocker** (const char *featureName)
Get locker for update linked features.
- const char * **GetLocker** (const char *featureName, int idx)

Friends

- class **ITypeBase**
- class **Integer**
- class **IFloat**
- class **IString**
- class **IEnumeration**
- class **IBoolean**
- class **ICommand**
- class **IEnumEntry**

4.3.1 Detailed Description

The class that contains information about [CameraInterface](#).

4.3.2 Constructor & Destructor Documentation

4.3.2.1 CameraInterface()

```
novitec::CameraAPI::GenICam::CameraInterface::CameraInterface (
    novitec::CameraAPI::Camera * device )
```

Constructs a new [novitec::CameraAPI::GenICam::CameraInterface](#) object.

This function will retrieve XML descriptor from specified device.

Parameters

in	<i>device</i>	- A pointer of novitec::CameraAPI::Camera
----	---------------	---

4.3.3 Member Function Documentation

4.3.3.1 GetFeature()

```
template<typename TFeature >
TFeature novitec::CameraAPI::GenICam::CameraInterface::GetFeature (
    const char * featureName )
```

Get a feature by feature name.

Exceptions

novitec::CameraAPI::Exception	
---	--

4.3.3.2 GetFeatureType() [1/2]

```
const char * novitec::CameraAPI::GenICam::CameraInterface::GetFeatureType (
    const char * featureName )
```

Get the interface type of the specified feature by feature name.

Exceptions

novitec::CameraAPI::Exception	
---	--

4.3.3.3 GetFeatureType() [2/2]

```
const char * novitec::CameraAPI::GenICam::CameraInterface::GetFeatureType (
    FeatureHandle handle )
```

Get the interface type of the specified feature.

Exceptions

novitec::CameraAPI::Exception	
---	--

4.3.3.4 GetFirstChildFeature()

```
FeatureHandle novitec::CameraAPI::GenICam::CameraInterface::GetFirstChildFeature (
    FeatureHandle handle = FEATURE_HANDLE_ROOT )
```

Get first child feature handle.

If input argument handle is 0 (FEATURE_HANDLE_ROOT), it return first category.

Parameters

in	<i>handle</i>	- Parent feature handle.
----	---------------	--------------------------

Returns

A FeatureHandle If had child feature.
a null pointer or reference otherwise.

4.3.3.5 GetNumberOfInvalidator()

```
unsigned int novitec::CameraAPI::GenICam::CameraInterface::GetNumberOfInvalidator (
    const char * featureName )
```

Get invalidator for update linked features.

Exceptions

novitec::CameraAPI::Exception	
---	--

4.3.3.6 Open()

```
int novitec::CameraAPI::GenICam::CameraInterface::Open (
    const char * XMLFilePath )
```

Open camera interface from an XML file.

Warning

When using the function [novitec::CameraAPI::GenICam::CameraInterface\(novitec::CameraAPI::BaseDevice* device\)](#) to create [CameraInterface](#), the XML file is automatically loaded so calling [Open\(\)](#) is not necessary.

4.3.3.7 OpenFromMem()

```
int novitec::CameraAPI::GenICam::CameraInterface::OpenFromMem (
    const char * xml )
```

Open camera interface from an XML on the memory.

Warning

When using the function [novitec::CameraAPI::GenICam::CameraInterface\(novitec::CameraAPI::BaseDevice* device\)](#) to create [CameraInterface](#), the XML file is automatically loaded so calling [Open\(\)](#) is not necessary.

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

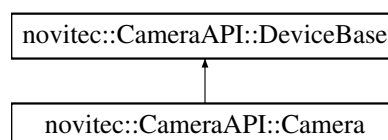
- GenAPI/CameraInterface.h

4.4 novitec::CameraAPI::DeviceBase Class Reference

The class which provides the device control.

```
#include <DeviceBase.h>
```

Inheritance diagram for novitec::CameraAPI::DeviceBase:



Public Member Functions

- **DeviceBase** (void)
Constructor.
- virtual **~DeviceBase** (void)
Destructor.
- virtual **novitec::CameraAPI::Error Connect** (novitec::CameraAPI::Handle *pHandle, novitec::CameraAPI::↔ AccessMode accessMode)=0
- virtual **novitec::CameraAPI::Error Disconnect** (void)=0
Disconnect a camera.
- virtual bool **IsConnected** (void)=0
Get the connection status to a camera.
- virtual **novitec::CameraAPI::Error Start** (void)=0
Start grabbing images from a camera.
- virtual **novitec::CameraAPI::Error Stop** (void)=0
Stop grabbing images from a camera.
- virtual **novitec::CameraAPI::Error GetImage** (novitec::CameraAPI::Image *pImage, unsigned int ui↔ Timeout=500)=0
Get an image from the frame buffer.
- virtual **novitec::CameraAPI::Error SetImageCallback** (novitec::CameraAPI::ImageCallbackFunc func, void *param)=0
*Set an image callback function.
It must be called before [Camera::Start\(\)](#)*
- virtual novitec::CameraAPI::InterfaceType **GetInterfaceType** ()
Get interface type.

Protected Attributes

- InterfaceType **m_interfaceType**

4.4.1 Detailed Description

The class which provides the device control.

4.4.2 Member Function Documentation

4.4.2.1 Connect()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::DeviceBase::Connect (
    novitec::CameraAPI::Handle * pHandle,
    novitec::CameraAPI::AccessMode accessMode ) [pure virtual]
```

Connect to a camera.

Parameters

<i>pHandle</i>	Pointer to the camera handle.
<i>accessMode</i>	Camera access mode.

Implemented in [novitec::CameraAPI::Camera](#).

4.4.2.2 Disconnect()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::DeviceBase::Disconnect (
    void ) [pure virtual]
```

Disconnect a camera.

Implemented in [novitec::CameraAPI::Camera](#).

4.4.2.3 GetImage()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::DeviceBase::GetImage (
    novitec::CameraAPI::Image * pImage,
    unsigned int uiTimeout = 500 ) [pure virtual]
```

Get an image from the frame buffer.

Parameters

<i>pImage</i>	Image buffer to store the image.
<i>uiTimeout</i>	Timeout (ms)

Implemented in [novitec::CameraAPI::Camera](#).

4.4.2.4 GetInterfaceType()

```
novitec::CameraAPI::InterfaceType novitec::CameraAPI::DeviceBase::GetInterfaceType ( ) [virtual]
```

Get interface type.

Returns

4.4.2.5 IsConnected()

```
virtual bool novitec::CameraAPI::DeviceBase::IsConnected (
    void ) [pure virtual]
```

Get the connection status to a camera.

Implemented in [novitec::CameraAPI::Camera](#).

4.4.2.6 SetImageCallback()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::DeviceBase::SetImageCallback (
    novitec::CameraAPI::ImageCallbackFunc func,
    void * param ) [pure virtual]
```

Set an image callback function.

It must be called before [Camera::Start\(\)](#)

Parameters

<i>func</i>	Pointer to a function which is called by internal acquisition thread when new image is acquired.
-------------	--

Implemented in [novitec::CameraAPI::Camera](#).

4.4.2.7 Start()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::DeviceBase::Start (
    void ) [pure virtual]
```

Start grabbing images from a camera.

Implemented in [novitec::CameraAPI::Camera](#).

4.4.2.8 Stop()

```
virtual novitec::CameraAPI::Error novitec::CameraAPI::DeviceBase::Stop (
    void ) [pure virtual]
```

Stop grabbing images from a camera.

Implemented in [novitec::CameraAPI::Camera](#).

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

- DeviceBase.h
- DeviceBase.cpp

4.5 novitec::CameraAPI::DeviceManager Class Reference

The class which provides the device management.

```
#include <DeviceManager.h>
```

Public Member Functions

- **DeviceManager** (void)
Constructor.
- **~DeviceManager** (void)
Destructor.
- **Error Update** ()
Updates the internal list of available devices.
- **Error GetNumberOfCameras** (unsigned int *pNumCameras)
Get the number of available cameras.
- **Error GetCameraHandle** (unsigned int uiIndex, **Handle** *pHandle)
Get the camera handle from a given index.
- **Error GetCameraHandleBySerial** (const char *pSerialNumber, **Handle** *pHandle)
Get the camera handle from a given serial number.
- **Error GetCameraHandleByIPAddress** (const char *pIPAddress, **Handle** *pHandle)
Get the camera handle from a given IP address.
- **Error GetCameraInfo** (unsigned int uiIndex, **novitec::CameraAPI::CameraInfo** *pCameraInfo)
Get the camera information.
- **Error GetCameraInfo** (unsigned int uiIndex, **novitec::CameraAPI::GEVCameraInfo** *pGEVCameraInfo, **novitec::CameraAPI::NetworkAdapterInfo** *pAdapterInfo=NULL)
Get the GigE Vision Camera information.
- **Error GetCameraInfo** (unsigned int uiIndex, **novitec::CameraAPI::U3VCameraInfo** *pU3VCameraInfo, **novitec::CameraAPI::HostControllerInfo** *pHostControllerInfo=NULL)
Get the USB3 Vision Camera information.
- **Error GetCameraInfo** (unsigned int uiIndex, **novitec::CameraAPI::NU3CameraInfo** *pNU3CameraInfo, **novitec::CameraAPI::HostControllerInfo** *pHostControllerInfo=NULL)
Get the Novitec USB3 Camera information.
- **Error SendForceIP** (unsigned int uiIndex, const char *ip=NULL, const char *subnetMask=NULL, const char *gateway=NULL)
Send Force IP command. (IP restoration)

4.5.1 Detailed Description

The class which provides the device management.

4.5.2 Member Function Documentation

4.5.2.1 GetCameraHandle()

```
Error novitec::CameraAPI::DeviceManager::GetCameraHandle (
    unsigned int uiIndex,
    Handle * pHandle )
```

Get the camera handle from a given index.

Parameters

in	<i>uiIndex</i>	Index of camera.
out	<i>pHandle</i>	Handle of camera.

4.5.2.2 GetCameraHandleByIPAddress()

```
Error novitec::CameraAPI::DeviceManager::GetCameraHandleByIPAddress (
    const char * pIPAddress,
    Handle * pHandle )
```

Get the camera handle from a given IP address.

Parameters

in	<i>pIPAddress</i>	IP address.
out	<i>pHandle</i>	Handle of camera.

4.5.2.3 GetCameraHandleBySerial()

```
Error novitec::CameraAPI::DeviceManager::GetCameraHandleBySerial (
    const char * pSerialNumber,
    Handle * pHandle )
```

Get the camera handle from a given serial number.

Parameters

in	<i>pSerialNumber</i>	Serial number.
out	<i>pHandle</i>	Handle of camera.

4.5.2.4 GetCameraInfo() [1/4]

```
Error novitec::CameraAPI::DeviceManager::GetCameraInfo (
    unsigned int uiIndex,
    novitec::CameraAPI::CameraInfo * pCameraInfo )
```

Get the camera information.

Parameters

in	<i>uiIndex</i>	Index of camera.
out	<i>pCameraInfo</i>	CameraInfo of camera.

4.5.2.5 GetCameraInfo() [2/4]

```
Error novitec::CameraAPI::DeviceManager::GetCameraInfo (
    unsigned int uiIndex,
    novitec::CameraAPI::GEVCameraInfo * pGEVCameraInfo,
    novitec::CameraAPI::NetworkAdapterInfo * pAdapterInfo = NULL )
```

Get the GigE Vision [Camera](#) information.

Parameters

in	<i>uiIndex</i>	Index of camera.
out	<i>pGEVCameraInfo</i>	GEVCameraInfo of camera.
out	<i>pAdapterInfo</i>	NetworkAdapterInfo

4.5.2.6 GetCameraInfo() [3/4]

```
Error novitec::CameraAPI::DeviceManager::GetCameraInfo (
    unsigned int uiIndex,
    novitec::CameraAPI::NU3CameraInfo * pNU3CameraInfo,
    novitec::CameraAPI::HostControllerInfo * pHostControllerInfo = NULL )
```

Get the Novitec USB3 [Camera](#) information.

Parameters

in	<i>uiIndex</i>	Index of camera.
out	<i>pNU3CameraInfo</i>	NU3CameraInfo of camera.
out	<i>pHostControllerInfo</i>	HostControllerInfo

4.5.2.7 GetCameraInfo() [4/4]

```
Error novitec::CameraAPI::DeviceManager::GetCameraInfo (
    unsigned int uiIndex,
    novitec::CameraAPI::U3VCameraInfo * pU3VCameraInfo,
    novitec::CameraAPI::HostControllerInfo * pHostControllerInfo = NULL )
```

Get the USB3 Vision [Camera](#) information.

Parameters

in	<i>uiIndex</i>	Index of camera.
out	<i>pU3VCameraInfo</i>	U3VCameraInfo of camera.
out	<i>pHostControllerInfo</i>	HostControllerInfo

4.5.2.8 GetNumberOfCameras()

```
Error novitec::CameraAPI::DeviceManager::GetNumberOfCameras (
    unsigned int * pNumCameras )
```

Get the number of available cameras.

Parameters

out	<i>pNumCameras</i>	number of cameras.
-----	--------------------	--------------------

4.5.2.9 SendForceIP()

```
Error novitec::CameraAPI::DeviceManager::SendForceIP (
    unsigned int uiIndex,
    const char * ip = NULL,
    const char * subnetMask = NULL,
    const char * gateway = NULL )
```

Send Force IP command. (IP restoration)

If camera and PC can't be communicated because network band of camera is different and PC, you can use this function to recover the camera.

If IP is not specified, it is set to a random IP in the same band.

This function is only available for GigE Vision camera.

Parameters

in	<i>uiIndex</i>	Index of camera.
in	<i>ip</i>	String of camera IP address to change.
in	<i>subnetMask</i>	String of subnet mask to change.
in	<i>gateway</i>	String of gateway to change.

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

- DeviceManager.h

4.6 novitec::CameraAPI::Error Class Reference

The class for errors in API functions.

```
#include <Error.h>
```

Public Member Functions

- **Error** (void)
Constructor.
- **Error** (const ErrorType &errorType)
Initialize [Error](#) with error type.
- virtual ~**Error** (void)
Destructor.
- virtual **Error** & **operator=** (const **Error** &error)
- virtual **Error** & **operator=** (const ErrorType &errorType)
- virtual bool **operator==** (const **Error** &error) const
- virtual bool **operator==** (const ErrorType &errorType) const
- virtual bool **operator!=** (const **Error** &error) const
- virtual bool **operator!=** (const ErrorType &errorType) const
- virtual int **GetType** (void)
Returns the error code.
- const char * **GetDescription** (void)
Returns the explanatory string.

4.6.1 Detailed Description

The class for errors in API functions.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Error()

```
novitec::CameraAPI::Error::Error (
    const ErrorType & errorType )
```

Initialize [Error](#) with error type.

Parameters

<i>errorType</i>	Set error type.
------------------	-----------------

4.6.3 Member Function Documentation

4.6.3.1 GetDescription()

```
const char * novitec::CameraAPI::Error::GetDescription (
    void )
```

Returns the explanatory string.

Returns

Pointer to a null-terminated string with explanatory information.
 The pointer is guaranteed to be valid at least until the exception object from which it is obtained is destroyed, or until a non-const member function (e.g. copy assignment operator) on the exception object is called.

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

- Error.h

4.7 novitec::CameraAPI::Exception Class Reference

The class that throws exception for Novitec [Camera](#) API.

```
#include <Exception.h>
```

Public Member Functions

- **Exception** ()
Initializer.
- **Exception** (ErrorType code, const char *message=NULL)
Initialize exception with error type and message.
- **~Exception** ()
Destructor.
- char * **What** ()
Get error message.

Public Attributes

- ErrorType **errorCode**
Get error code.
- char **errorMessage** [NVT_EXCEPTION_MESSAGE_LENGTH]
Get error message.

4.7.1 Detailed Description

The class that throws exception for Novitec [Camera](#) API.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 Exception()

```
novitec::CameraAPI::Exception::Exception (
    ErrorType code,
    const char * message = NULL )
```

Initialize exception with error type and message.

Parameters

<i>code</i>	Error type.
<i>message</i>	Error message.

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

- Exception.h
- Exception.cpp

4.8 novitec::CameraAPI::GATEWAY Struct Reference

Gateway Structure.

```
#include <Types.h>
```

Public Attributes

- unsigned char **address** [4]
Address of Gateway.

4.8.1 Detailed Description

Gateway Structure.

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

- Types.h

4.9 novitec::CameraAPI::GEVCameraInfo Class Reference

GigE Vision camera Information.

```
#include <Types.h>
```

Public Attributes

- unsigned short **spec_version_major**
Major version of the GigE standard with which the device is compliant.
- unsigned short **spec_version_minor**
Major version of the GigE standard with which the device is compliant..
- unsigned int **device_mode**
Device mode.
- unsigned short **reserved_0**
Reserved.
- unsigned char **device_MAC_address_h** [2]
The 2 high-order bytes of the MAC address. (16 bits)
- unsigned char **device_MAC_address_l** [4]
The 4 low-order bytes of the MAC address. (32 bits)
- unsigned int **IP_config_options**
Network interface capability.
- unsigned int **IP_config_current**
Network interface configuration.
- unsigned int **reserved_1**
Reserved.
- unsigned int **reserved_2**
Reserved.
- unsigned int **reserved_3**
Reserved.
- unsigned char **current_IP** [4]
Current IP address.
- unsigned int **reserved_4**
Reserved.
- unsigned int **reserved_5**
Reserved.
- unsigned int **reserved_6**
Reserved.
- unsigned char **current_subnet_mask** [4]
Current subnet mask.
- unsigned int **reserved_7**
Reserved.
- unsigned int **reserved_8**
Reserved.
- unsigned int **reserved_9**
Reserved.
- unsigned char **default_gateway** [4]
Current gateway.
- char **manufacturer_name** [32]
Device vendor name.
- char **model_name** [32]
Device model name.
- char **device_version** [32]
Device verison.
- char **manufacturer_specific_information** [48]
Manufacturer specific information.
- char **serial_number** [16]
Serial number.
- char **user_defined_name** [16]
User defined name.

4.9.1 Detailed Description

GigE Vision camera Information.

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

- Types.h

4.10 novitec::CameraAPI::Handle Class Reference

The class which provides handle of device.

```
#include <Handle.h>
```

Public Member Functions

- **Handle** (void)
Constructor.
- **~Handle** (void)
Destructor.
- void * **GetDescriptor** (void)
Get descriptor of handle.
- novitec::CameraAPI::InterfaceType **GetInterfaceType** (void)
Get interface type.
- void * **GetInterfaceHandle** (void)
Get interface handle.

4.10.1 Detailed Description

The class which provides handle of device.

4.10.2 Member Function Documentation

4.10.2.1 GetDescriptor()

```
void * novitec::CameraAPI::Handle::GetDescriptor (
    void )
```

Get descriptor of handle.

Returns

Descriptor of handle.

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

- Handle.h

4.11 novitec::CameraAPI::HostControllerInfo Class Reference

Host Controller Information.

```
#include <Types.h>
```

Public Member Functions

- **HostControllerInfo** ()
Constructor.
- **HostControllerInfo** (const [HostControllerInfo](#) &obj)
Constructor.
- **~HostControllerInfo** ()
Destructor.

4.11.1 Detailed Description

Host Controller Information.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 HostControllerInfo()

```
novitec::CameraAPI::HostControllerInfo::HostControllerInfo (
    const HostControllerInfo & obj )
```

Constructor.

Parameters

<i>obj</i>	Set HostControllerInfo
------------	--

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

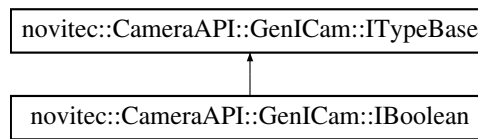
- Types.h

4.12 novitec::CameraAPI::GenlCam::IBoolean Class Reference

[IBoolean](#) class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::IBoolean:



Public Member Functions

- bool **GetValue** ()
Get the boolean value.
- bool **GetCacheValue** ()
Get the cached boolean value.
- void **SetValue** (bool value)
Set the boolean value.

4.12.1 Detailed Description

[IBoolean](#) class.

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

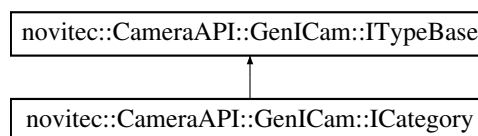
- GenAPI/ITypes.h

4.13 novitec::CameraAPI::GenICam::IBoolean Class Reference

[IBoolean](#) class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::IBoolean:



Additional Inherited Members

4.13.1 Detailed Description

[IBoolean](#) class.

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

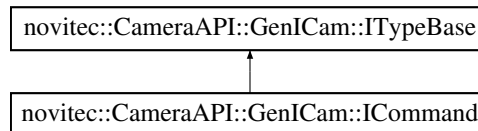
- GenAPI/ITypes.h

4.14 novitec::CameraAPI::GenICam::ICommand Class Reference

[ICommand](#) class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::ICommand:



Public Member Functions

- void **Execute** ()
Execute the command.
- bool **IsDone** ()
Check if the command execution is done.

4.14.1 Detailed Description

[ICommand](#) class.

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

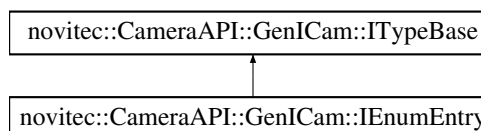
- GenAPI/ITypes.h

4.15 novitec::CameraAPI::GenICam::IEnumEntry Class Reference

[IEnumEntry](#) class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::IEnumEntry:



Public Member Functions

- int64_t **GetValue** ()
Get the value.

Friends

- class **IEnumeration**

4.15.1 Detailed Description

[IEnumEntry](#) class.

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

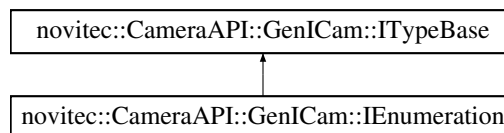
- GenAPI/ITypes.h

4.16 novitec::CameraAPI::GenICam::IEnumeration Class Reference

[IEnumeration](#) class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::IEnumeration:



Public Member Functions

- int **GetIntValue** ()
Get the integer value.
- int **GetCacheIntValue** ()
Get the cached integer value.
- void **SetIntValue** (int64_t value)
Set the integer value.
- const char * **GetSymbolicValue** ()
Get the symbolic value.
- void **SetSymbolicValue** (const char *value)
Set the symbolic value.
- int **GetNumberOfEntries** ()
Get number of entries.
- [IEnumEntry](#) **GetEntry** (int index)
Get entry.

4.16.1 Detailed Description

[IEnumeration](#) class.

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

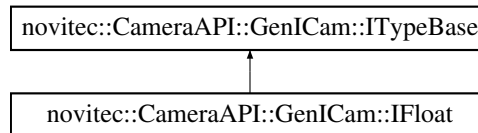
- GenAPI/ITypes.h

4.17 novitec::CameraAPI::GenICam::IFloat Class Reference

IFloat class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::IFloat:



Public Member Functions

- double **GetValue** ()
Get the value.
- double **GetCacheValue** ()
Get cached value.
- void **SetValue** (double value)
Set the value.
- bool **HasMin** ()
- bool **HasMax** ()
- bool **HasInc** ()
- double **GetMin** ()
Get the minimum value.
- double **GetMax** ()
Get the maximum value.
- IncMode **GetIncMode** ()
Get the increment mode.
- double **GetInc** ()
Get the increment value.
- int **GetNumberOfValidValues** ()
Get the list of valid values.
- double **GetValidValue** (int index)
- FloatRepresentation **GetRepresentation** ()
Get the representation.
- const char * **GetUnit** ()
Get the unit.

4.17.1 Detailed Description

IFloat class.

4.17.2 Member Function Documentation

4.17.2.1 GetInc()

```
double novitec::CameraAPI::GenICam::IFloat::GetInc ( )
```

Get the increment value.

Warning

An exception will be raised if the increment value is not specified.

4.17.2.2 GetMax()

```
double novitec::CameraAPI::GenICam::IFloat::GetMax ( )
```

Get the maximum value.

Warning

An exception will be raised if the maximum value is not specified.

4.17.2.3 GetMin()

```
double novitec::CameraAPI::GenICam::IFloat::GetMin ( )
```

Get the minimum value.

Warning

An exception will be raised if the minimum value is not specified.

4.17.2.4 GetNumberOfValidValues()

```
int novitec::CameraAPI::GenICam::IFloat::GetNumberOfValidValues ( )
```

Get the list of valid values.

Warning

Returns values if the IncMode is IM_LIst.

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

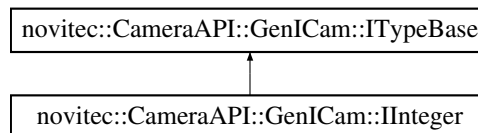
- GenAPI/Types.h

4.18 novitec::CameraAPI::GenICam::Integer Class Reference

[Integer](#) class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::Integer:



Public Member Functions

- `int64_t` [GetValue](#) ()
Get value of integer type feature.
- `int64_t` [GetCacheValue](#) ()
Get cached feature value. Returns the stored feature value without attempting to read register.
- `void` [SetValue](#) (`int64_t` value)
Set value of integer type feature.
- `bool` **HasMin** ()
- `bool` **HasMax** ()
- `bool` **HasInc** ()
- `int64_t` [GetMin](#) ()
Get the minimum value of integer type feature.
- `int64_t` [GetMax](#) ()
Get the maximum value of integer type feature.
- `IncMode` **GetIncMode** ()
Get the increment mode.
- `int64_t` [GetInc](#) ()
Get the increment value.
- `int` [GetNumberOfValidValues](#) ()
Get the list of valid values.
- `int64_t` **GetValidValue** (`int` index)
- `IntRepresentation` **GetRepresentation** ()
Get the representation.
- `const char *` **GetUnit** ()
Get the unit.

4.18.1 Detailed Description

[Integer](#) class.

4.18.2 Member Function Documentation

4.18.2.1 GetCacheValue()

```
int64_t novitec::CameraAPI::GenICam::Integer::GetCacheValue ( )
```

Get cached feature value. Returns the stored feature value without attempting to read register.

Returns

cached feature value.

4.18.2.2 GetInc()

```
int64_t novitec::CameraAPI::GenICam::Integer::GetInc ( )
```

Get the increment value.

Exceptions

novitec::CameraAPI::Exception	An exception will be raised if the increment value is not specified.
---	--

4.18.2.3 GetMax()

```
int64_t novitec::CameraAPI::GenICam::Integer::GetMax ( )
```

Get the maximum value of integer type feature.

Exceptions

novitec::CameraAPI::Exception	An exception will be raised if the maximum value is not specified.
---	--

4.18.2.4 GetMin()

```
int64_t novitec::CameraAPI::GenICam::Integer::GetMin ( )
```

Get the minimum value of integer type feature.

Exceptions

novitec::CameraAPI::Exception	An exception will be raised if the minimum value is not specified.
---	--

4.18.2.5 GetNumberOfValidValues()

```
int novitec::CameraAPI::GenICam::IInteger::GetNumberOfValidValues ( )
```

Get the list of valid values.

Warning

Returns values if the IncMode is IM_List.

4.18.2.6 GetValue()

```
int64_t novitec::CameraAPI::GenICam::IInteger::GetValue ( )
```

Get value of integer type feature.

Returns

feature value.

Exceptions

novitec::CameraAPI::Exception	An exception will be rised if the feature register read fail.
---	---

4.18.2.7 SetValue()

```
void novitec::CameraAPI::GenICam::IInteger::SetValue (
    int64_t value )
```

Set value of integer type feature.

Exceptions

novitec::CameraAPI::Exception	An exception will be rised if the feature register write fail.
---	--

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

- GenAPI/ITypes.h

4.19 novitec::CameraAPI::Image Class Reference

The class for image.

```
#include <Image.h>
```

Public Types

- typedef std::shared_ptr< [Image](#) > **Ptr**

Public Member Functions

- **Image** ()
Constructor.
- **~Image** ()
Destructor.
- void **Create** (int iWidth, int iHeight, novitec::CameraAPI::PixelFormat ePixelFormat, unsigned int uiChunkSize=0)
Create new image.
- void **CreateJPEG** (int iWidth, int iHeight, novitec::CameraAPI::PixelFormat ePixelFormat, unsigned int uiDataSize, unsigned int uiChunkSize=0)
Create new JPEG image.
- void **Copy** (unsigned char *pData, int iWidth, int iHeight, novitec::CameraAPI::PixelFormat ePixelFormat)
Copy the image from pixel data.
- void **Copy** (unsigned char *pData, unsigned int uiDataSize)
Copy the image from pixel data.
- void **Release** ()
Releases all resources used by this Image.
- novitec::CameraAPI::Error **Convert** (novitec::CameraAPI::Image *dst, novitec::CameraAPI::PixelFormat ePixelFormat, novitec::CameraAPI::DebayerMethod eDebayerMethod=HQ_LINEAR)
Convert image.
- novitec::CameraAPI::Error **Save** (const char *fileName, novitec::CameraAPI::FileFormat fileFormat)
Save this image to the specified file in the specified pixel format.
- novitec::CameraAPI::Error **Load** (const char *fileName)
Load image from file.

Public Attributes

- int **width**
Width of Image.
- int **height**
Height of Image.
- int **bpp**
bpp(Bits Per Pixel) of Image.
- unsigned long long **timeStamp**
Timestamp of Image.
- unsigned long long **frameNum**
Frame Number of Image.
- unsigned char * **chunkData**
Chunk Data of Image.
- unsigned char * **data**
Image Data.
- unsigned int **dataSize**
Size of Image Data.
- unsigned int **chunkSize**
Size of Chunk Data.
- PayloadType **payloadType**
Payload Type.
- novitec::CameraAPI::PixelFormat **pixelFormat**
Pixel Format.

4.19.1 Detailed Description

The class for image.

4.19.2 Member Function Documentation

4.19.2.1 Convert()

```
novitec::CameraAPI::Error novitec::CameraAPI::Image::Convert (
    novitec::CameraAPI::Image * dst,
    novitec::CameraAPI::PixelFormat ePixelFormat,
    novitec::CameraAPI::DebayerMethod eDebayerMethod = HQ_LINEAR )
```

Convert image.

Parameters

<i>dst</i>	Destinaton Image
<i>ePixelFormat</i>	Pixel Format
<i>eDebayerMethod</i>	Debayer Method

4.19.2.2 Copy() [1/2]

```
void novitec::CameraAPI::Image::Copy (
    unsigned char * pData,
    int iWidth,
    int iHeight,
    novitec::CameraAPI::PixelFormat ePixelFormat )
```

Copy the image from pixel data.

Parameters

<i>pData</i>	Image Data
<i>iWidth</i>	Image Width
<i>iHeight</i>	Image Height
<i>ePixelFormat</i>	Pixel Format

4.19.2.3 Copy() [2/2]

```
void novitec::CameraAPI::Image::Copy (
```

```

    unsigned char * pData,
    unsigned int uiDataSize )

```

Copy the image from pixel data.

Parameters

<i>pData</i>	Image Data
<i>uiDataSize</i>	Image Data Size

4.19.2.4 Create()

```

void novitec::CameraAPI::Image::Create (
    int iWidth,
    int iHeight,
    novitec::CameraAPI::PixelFormat ePixelFormat,
    unsigned int uiChunkSize = 0 )

```

Create new image.

Parameters

<i>iWidth</i>	Image Width
<i>iHeight</i>	Image Height
<i>ePixelFormat</i>	Pixel Format
<i>uiChunkSize</i>	Image Chunk Size

4.19.2.5 CreateJPEG()

```

void novitec::CameraAPI::Image::CreateJPEG (
    int iWidth,
    int iHeight,
    novitec::CameraAPI::PixelFormat ePixelFormat,
    unsigned int uiDataSize,
    unsigned int uiChunkSize = 0 )

```

Create new JPEG image.

Parameters

<i>iWidth</i>	Image Width
<i>iHeight</i>	Image Height
<i>ePixelFormat</i>	Pixel Format
<i>uiDataSize</i>	Image Data Size
<i>uiChunkSize</i>	Image Chunk Size

4.19.2.6 Load()

```
novitec::CameraAPI::Error novitec::CameraAPI::Image::Load (
    const char * fileName )
```

Load image from file.

Parameters

<i>fileName</i>	A file name in which to load image.
-----------------	-------------------------------------

4.19.2.7 Save()

```
novitec::CameraAPI::Error novitec::CameraAPI::Image::Save (
    const char * fileName,
    novitec::CameraAPI::FileFormat fileFormat )
```

Save this image to the specified file in the specified pixel format.

Parameters

<i>fileName</i>	A file name in which to save this Image .
<i>fileFormat</i>	File format.

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

- Image.h

4.20 novitec::CameraAPI::IPConfiguration Struct Reference

GigE Vision Device IP Configurations.

```
#include <Types.h>
```

Public Attributes

- **bool use_lls**
Use Whether or not to use an LLA.
- **bool use_dhcp**
Use Whether or not to use an DHCP.
- **bool use_persistent_ip**
Use Whether or not to use an persistent IP.

- unsigned char **persistent_ip_address** [4]
IP Address of persistent IP.
- unsigned char **persistent_netmask** [4]
Subnet mask of persistent IP.
- unsigned char **persistent_gateway** [4]
Gateway of persistent IP.

4.20.1 Detailed Description

GigE Vision Device IP Configurations.

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

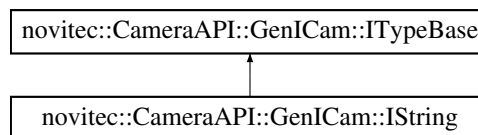
- Types.h

4.21 novitec::CameraAPI::GenICam::IString Class Reference

[IString](#) class.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::IString:



Public Member Functions

- const char * **GetValue** ()
Get the current value.
- const char * **GetCacheValue** ()
Get the current value.
- void **SetValue** (const char *value)
Set the value.
- int **GetMaxLength** ()
Get the maximum length of the string.

4.21.1 Detailed Description

[IString](#) class.

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

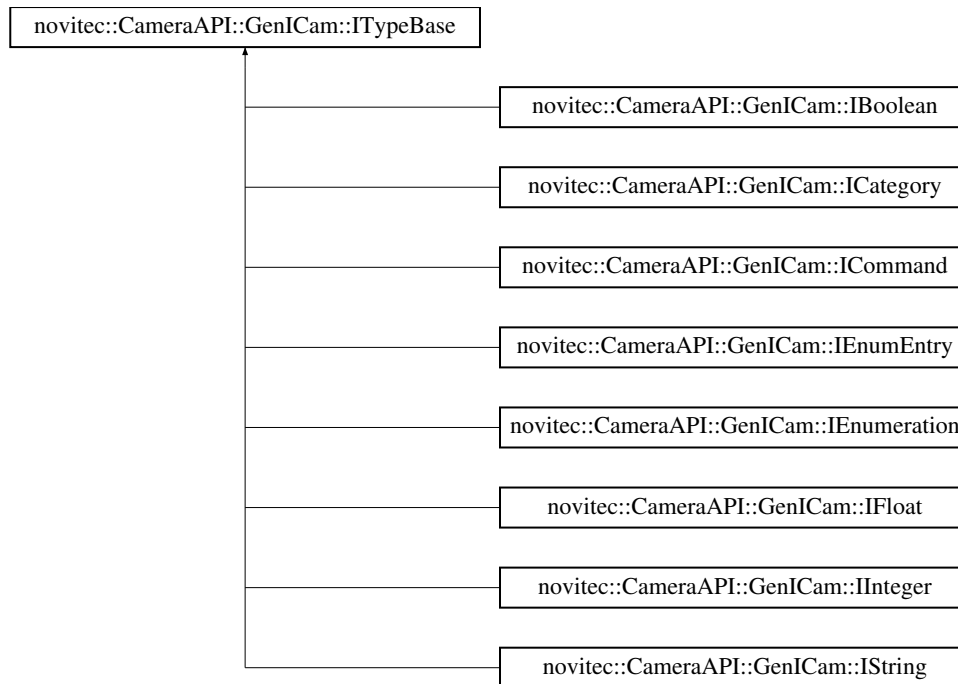
- GenAPI/ITypes.h

4.22 novitec::CameraAPI::GenICam::ITypeBase Class Reference

The class on which the supported types are based.

```
#include <ITypes.h>
```

Inheritance diagram for novitec::CameraAPI::GenICam::ITypeBase:



Public Member Functions

- **ITypeBase** ()
Initialize a new instance of the [ITypeBase](#) class.
- **ITypeBase** (const [ITypeBase](#) &obj)
Initialize a new instance of the [ITypeBase](#) class with another [ITypeBase](#) instance.
- virtual **~ITypeBase** ()
Desturctor.
- **ITypeBase** & **operator=** (const [ITypeBase](#) &rhs)
- const char * **GetType** ()
Get feature type.
- const char * **GetType** **Name** ()
Get feature type name.
- const char * **GetCategory** ()
Get the category of the feature.
- const char * **GetFeatureName** ()
Get the name of the feature.
- bool **IsValid** ()
Test if the feature is valid.
- bool **IsImplemented** ()
Test if the feature is implemented.
- bool **IsLocked** ()

- Test if the feature is locked.*

 - AccessMode **GetAccessMode** ()
 - Get the access mode.*
 - Visibility **GetVisibility** ()
 - Get the visibility.*
- const char * **GetDisplayName** ()
 - Get the display name.*
- const char * **GetToolTip** ()
 - Get the tool tip.*
- const char * **GetDescription** ()
 - Get the description.*
- const char * **GetElement** (const char *elementName)
 - Get the specified element.*

Friends

- class **CameraInterface_impl**
- class **ICategory**
- class **IInteger**
- class **IFloat**
- class **IString**
- class **IEnumeration**
- class **ICommand**
- class **IBoolean**
- class **IEnumEntry**

4.22.1 Detailed Description

The class on which the supported types are based.

4.22.2 Constructor & Destructor Documentation

4.22.2.1 ITypeBase()

```
novitec::CameraAPI::GenICam::ITypeBase::ITypeBase (
    const ITypeBase & obj )
```

Initialize a new instance of the [ITypeBase](#) class with another [ITypeBase](#) instance.

Parameters

<i>obj</i>	A instance of the ITypeBase class.
------------	--

4.22.3 Member Function Documentation

4.22.3.1 GetElement()

```
const char * novitec::CameraAPI::GenICam::ITypeBase::GetElement (
    const char * elementName )
```

Get the specified element.

Returns

Pointer to a null-terminated string with value of specified element. If feature has not specified element, It will return NULL. The pointer is guaranteed to be valid at least until the exception object from which it is obtained is destroyed, or until a non-const member function (e.g. copy assignment operator) on the exception object is called.

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

- GenAPI/ITypes.h

4.23 novitec::CameraAPI::MAC_ADDRESS Struct Reference

MAC Address Structure.

```
#include <Types.h>
```

Public Attributes

- unsigned char **address** [6]
MAC Address.

4.23.1 Detailed Description

MAC Address Structure.

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

- Types.h

4.24 novitec::CameraAPI::NetworkAdapterInfo Class Reference

Network Adapter Information.

```
#include <Types.h>
```

Public Member Functions

- **NetworkAdapterInfo** (const [NetworkAdapterInfo](#) &obj)
- char * [GetAdapterName](#) ()
Get adapter name.
- char * [GetFriendlyName](#) ()
Get friendly name.
- char * [GetDescription](#) ()
Get description.
- [MACAddress](#) [GetMACAddress](#) ()
Get mac address.
- int [GetNumberOfSubnets](#) ()
Get number of subnets.
- int [GetNumberOfGateways](#) ()
Get number of gateways.
- [Subnet](#) [GetSubnet](#) (int idx)
Get subnet of a specific index.
- [Gateway](#) [GetGateway](#) (int idx)
Get gateway of a specific index.

4.24.1 Detailed Description

Network Adapter Information.

4.24.2 Member Function Documentation

4.24.2.1 GetAdapterName()

```
char * novitec::CameraAPI::NetworkAdapterInfo::GetAdapterName ( )
```

Get adapter name.

Returns

String that represents the adapter name.

4.24.2.2 GetDescription()

```
char * novitec::CameraAPI::NetworkAdapterInfo::GetDescription ( )
```

Get description.

Returns

String that represents the description.

4.24.2.3 GetFriendlyName()

```
char * novitec::CameraAPI::NetworkAdapterInfo::GetFriendlyName ( )
```

Get friendly name.

Returns

String that represents the friendly name.

4.24.2.4 GetGateway()

```
Gateway novitec::CameraAPI::NetworkAdapterInfo::GetGateway (
    int idx )
```

Get gateway of a specific index.

Parameters

<i>idx</i>	Index of gateway.
------------	-------------------

Returns

Gateway class that represents gateway of a specific index.

4.24.2.5 GetMACAddress()

```
MACAddress novitec::CameraAPI::NetworkAdapterInfo::GetMACAddress ( )
```

Get mac address.

Returns

MACAddress struct that represents the mac address.

4.24.2.6 GetNumberOfGateways()

```
int novitec::CameraAPI::NetworkAdapterInfo::GetNumberOfGateways ( )
```

Get number of gateways.

Returns

Integer that represents the number of gateways.

4.24.2.7 GetNumberOfSubnets()

```
int novitec::CameraAPI::NetworkAdapterInfo::GetNumberOfSubnets ( )
```

Get number of subnets.

Returns

Integer that represents the number of subnets.

4.24.2.8 GetSubnet()

```
Subnet novitec::CameraAPI::NetworkAdapterInfo::GetSubnet (
    int idx )
```

Get subnet of a specific index.

Parameters

<i>idx</i>	Index of subnet.
------------	------------------

Returns

Subnet class that represents subnet of a specific index.

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

- Types.h

4.25 novitec::CameraAPI::NU3CameraInfo Class Reference

Novitec USB3.0 [Camera](#) Information.

```
#include <Types.h>
```

Public Attributes

- char **manufacturer** [32]
Device vendor name.
- char **model_name** [32]
Device model name.
- char **serial_number** [16]
Serial number.
- char **device_version** [32]
Device version.

4.25.1 Detailed Description

Novitec USB3.0 [Camera](#) Information.

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

- [Types.h](#)

4.26 novitec::CameraAPI::PixelFormatUtil Class Reference

The class which provides the utils for PixelFormat.

```
#include <PixelFormatUtil.h>
```

Static Public Member Functions

- static int [GetBitsPerPixel](#) (PixelFormat pixelFormat)
Get bits-per-pixel from PixelFormat.
- static int [GetBytesPerPixel](#) (PixelFormat pixelFormat)
Get bytes-per-pixel from PixelFormat.
- static const char * [GetPixelFormatName](#) (PixelFormat pixelFormat)
Get name of PixelFormat.
- static const char * [GetPixelFormatDescription](#) (PixelFormat pixelFormat)
Get description of PixelFormat.
- static bool [Is3DFormat](#) (PixelFormat pixelFormat)
Returns a value that indicates whether the pixel format is type of 3D.

4.26.1 Detailed Description

The class which provides the utils for PixelFormat.

4.26.2 Member Function Documentation

4.26.2.1 GetBitsPerPixel()

```
int novitec::CameraAPI::PixelFormatUtil::GetBitsPerPixel (
    PixelFormat pixelFormat ) [static]
```

Get bits-per-pixel from PixelFormat.

Parameters

<i>pixelFormat</i>	Pixel Format
--------------------	--------------

Returns

Return bits-per-pixel value.

4.26.2.2 GetBytesPerPixel()

```
int novitec::CameraAPI::PixelFormatUtil::getBytesPerPixel (
    PixelFormat pixelFormat ) [static]
```

Get bytes-per-pixel from PixelFormat.

Parameters

<i>pixelFormat</i>	PixelFormat
--------------------	-------------

Returns

Return bytes-per-pixel value.

4.26.2.3 GetPixelFormatDescription()

```
const char * novitec::CameraAPI::PixelFormatUtil::getPixelFormatDescription (
    PixelFormat pixelFormat ) [static]
```

Get description of PixelFormat.

Parameters

<i>pixelFormat</i>	PixelFormat
--------------------	-------------

Returns

A character pointer that contains description of PixelFormat.

4.26.2.4 GetPixelFormatName()

```
const char * novitec::CameraAPI::PixelFormatUtil::getPixelFormatName (
    PixelFormat pixelFormat ) [static]
```

Get name of PixelFormat.

Parameters

<i>pixelFormat</i>	Pixel Format
--------------------	--------------

Returns

A character pointer that contains name of PixelFormat.

4.26.2.5 Is3DFormat()

```
bool novitec::CameraAPI::PixelFormatUtil::Is3DFormat (
    PixelFormat pixelFormat ) [static]
```

Returns a value that indicates whether the pixel format is type of 3D.

Parameters

<i>pixelFormat</i>	Pixel Format
--------------------	--------------

Returns

Return true if pixelFormat is type of 3D; otherwise, return false.

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

- PixelFormatUtil.h
- PixelFormatUtil.cpp

4.27 novitec::CameraAPI::SUBNET Struct Reference

Subnet Structure.

```
#include <Types.h>
```

Public Attributes

- unsigned char **address** [4]
- unsigned char **mask** [4]

4.27.1 Detailed Description

Subnet Structure.

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

- Types.h

4.28 novitec::CameraAPI::U3VCameraInfo Class Reference

USB3.0 Vision [Camera](#) Information.

```
#include <Types.h>
```

Public Attributes

- unsigned short **spec_version_major**
Major Version of USB3 Vision.
- unsigned short **spec_version_minor**
Minor Version of USB3 Vision.
- char **manufacturer_name** [32]
Device vendor name.
- char **model_name** [32]
Device model name.
- char **device_version** [32]
Device version.
- char **manufacturer_specific_information** [48]
Manufacturer specific information.
- char **serial_number** [16]
Serial number.
- char **user_defined_name** [16]
User defined name.

4.28.1 Detailed Description

USB3.0 Vision [Camera](#) Information.

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

- Types.h

Index

CameraInfo
 novitec::CameraAPI::CameraInfo, 17

CameraInterface
 novitec::CameraAPI::GenICam::CameraInterface, 21

Connect
 novitec::CameraAPI::Camera, 10
 novitec::CameraAPI::DeviceBase, 24

Convert
 novitec::CameraAPI::Image, 46

Copy
 novitec::CameraAPI::Image, 46

Create
 novitec::CameraAPI::Image, 47

CreateJPEG
 novitec::CameraAPI::Image, 47

Disconnect
 novitec::CameraAPI::Camera, 11
 novitec::CameraAPI::DeviceBase, 25

Error
 novitec::CameraAPI::Error, 31

Exception
 novitec::CameraAPI::Exception, 32

GetAccessStatus
 novitec::CameraAPI::CameraInfo, 17

GetAdapterName
 novitec::CameraAPI::NetworkAdapterInfo, 53

GetBitsPerPixel
 novitec::CameraAPI::PixelFormatUtil, 56

GetBufferMode
 novitec::CameraAPI::Camera, 11

GetBytesPerPixel
 novitec::CameraAPI::PixelFormatUtil, 57

GetCacheValue
 novitec::CameraAPI::GenICam::Integer, 42

GetCameraHandle
 novitec::CameraAPI::DeviceManager, 27

GetCameraHandleByIPAddress
 novitec::CameraAPI::DeviceManager, 28

GetCameraHandleBySerial
 novitec::CameraAPI::DeviceManager, 28

GetCameraInfo
 novitec::CameraAPI::DeviceManager, 28, 29

GetDescription
 novitec::CameraAPI::Error, 31
 novitec::CameraAPI::NetworkAdapterInfo, 53

GetDescriptor
 novitec::CameraAPI::Handle, 35

GetDeviceVersion
 novitec::CameraAPI::CameraInfo, 17

GetDisplayName
 novitec::CameraAPI::CameraInfo, 17

GetElement
 novitec::CameraAPI::GenICam::ITypeBase, 52

GetFeature
 novitec::CameraAPI::GenICam::CameraInterface, 21

GetFeatureType
 novitec::CameraAPI::GenICam::CameraInterface, 21, 22

GetFirstChildFeature
 novitec::CameraAPI::GenICam::CameraInterface, 22

GetFriendlyName
 novitec::CameraAPI::NetworkAdapterInfo, 53

GetGateway
 novitec::CameraAPI::NetworkAdapterInfo, 54

GetID
 novitec::CameraAPI::CameraInfo, 18

GetImage
 novitec::CameraAPI::Camera, 11
 novitec::CameraAPI::DeviceBase, 25

GetInc
 novitec::CameraAPI::GenICam::IFloat, 40
 novitec::CameraAPI::GenICam::Integer, 43

GetInterfaceType
 novitec::CameraAPI::DeviceBase, 25

GetMACAddress
 novitec::CameraAPI::NetworkAdapterInfo, 54

GetManufacturer
 novitec::CameraAPI::CameraInfo, 18

GetMax
 novitec::CameraAPI::GenICam::IFloat, 41
 novitec::CameraAPI::GenICam::Integer, 43

GetMin
 novitec::CameraAPI::GenICam::IFloat, 41
 novitec::CameraAPI::GenICam::Integer, 43

GetModelName
 novitec::CameraAPI::CameraInfo, 18

GetNumberOfCameras
 novitec::CameraAPI::DeviceManager, 30

GetNumberOfGateways
 novitec::CameraAPI::NetworkAdapterInfo, 54

GetNumberOfInvalidator
 novitec::CameraAPI::GenICam::CameraInterface, 22

- GetNumberOfSubnets
 - novitec::CameraAPI::NetworkAdapterInfo, 54
- GetNumberOfValidValues
 - novitec::CameraAPI::GenICam::IFloat, 41
 - novitec::CameraAPI::GenICam::IInteger, 43
- GetPixelFormatDescription
 - novitec::CameraAPI::PixelFormatUtil, 57
- GetPixelFormatName
 - novitec::CameraAPI::PixelFormatUtil, 57
- GetSerialNumber
 - novitec::CameraAPI::CameraInfo, 18
- GetSubnet
 - novitec::CameraAPI::NetworkAdapterInfo, 55
- GetTimestampFrequency
 - novitec::CameraAPI::CameraInfo, 19
- GetTransportLayer
 - novitec::CameraAPI::CameraInfo, 19
- GetUserDefinedName
 - novitec::CameraAPI::CameraInfo, 19
- GetValue
 - novitec::CameraAPI::GenICam::IInteger, 44
- GetXML
 - novitec::CameraAPI::Camera, 12
- HostControllerInfo
 - novitec::CameraAPI::HostControllerInfo, 36
- Is3DFormat
 - novitec::CameraAPI::PixelFormatUtil, 58
- IsConnected
 - novitec::CameraAPI::Camera, 12
 - novitec::CameraAPI::DeviceBase, 25
- ITypeBase
 - novitec::CameraAPI::GenICam::ITypeBase, 51
- Load
 - novitec::CameraAPI::Image, 48
- novitec::CameraAPI::Camera, 9
 - Connect, 10
 - Disconnect, 11
 - GetBufferMode, 11
 - GetImage, 11
 - GetXML, 12
 - IsConnected, 12
 - ReadMemory, 12
 - ReadRegister, 13
 - SetBufferMode, 13
 - SetDeviceEventCallback, 13
 - SetImageCallback, 14
 - Start, 14
 - Stop, 14
 - UpdateFirmware, 15
 - WriteMemory, 15
 - WriteRegister, 15
- novitec::CameraAPI::CameraInfo, 16
 - CameraInfo, 17
 - GetAccessStatus, 17
 - GetDeviceVersion, 17
 - GetDisplayName, 17
 - GetID, 18
 - GetManufacturer, 18
 - GetModelName, 18
 - GetSerialNumber, 18
 - GetTimestampFrequency, 19
 - GetTransportLayer, 19
 - GetUserDefinedName, 19
- novitec::CameraAPI::DeviceBase, 23
 - Connect, 24
 - Disconnect, 25
 - GetImage, 25
 - GetInterfaceType, 25
 - IsConnected, 25
 - SetImageCallback, 26
 - Start, 26
 - Stop, 26
- novitec::CameraAPI::DeviceManager, 27
 - GetCameraHandle, 27
 - GetCameraHandleByIPAddress, 28
 - GetCameraHandleBySerial, 28
 - GetCameraInfo, 28, 29
 - GetNumberOfCameras, 30
 - SendForceIP, 30
- novitec::CameraAPI::Error, 30
 - Error, 31
 - GetDescription, 31
- novitec::CameraAPI::Exception, 32
 - Exception, 32
- novitec::CameraAPI::GATEWAY, 33
- novitec::CameraAPI::GenICam::CameraInterface, 20
 - CameraInterface, 21
 - GetFeature, 21
 - GetFeatureType, 21, 22
 - GetFirstChildFeature, 22
 - GetNumberOfInvalidator, 22
 - Open, 23
 - OpenFromMem, 23
- novitec::CameraAPI::GenICam::IBoolean, 36
- novitec::CameraAPI::GenICam::ICategory, 37
- novitec::CameraAPI::GenICam::ICommand, 38
- novitec::CameraAPI::GenICam::IEnumEntry, 38
- novitec::CameraAPI::GenICam::IEnumeration, 39
- novitec::CameraAPI::GenICam::IFloat, 40
 - GetInc, 40
 - GetMax, 41
 - GetMin, 41
 - GetNumberOfValidValues, 41
- novitec::CameraAPI::GenICam::IInteger, 42
 - GetCacheValue, 42
 - GetInc, 43
 - GetMax, 43
 - GetMin, 43
 - GetNumberOfValidValues, 43
 - GetValue, 44
 - SetValue, 44
- novitec::CameraAPI::GenICam::IString, 49
- novitec::CameraAPI::GenICam::ITypeBase, 50

- GetElement, [52](#)
- ITypeBase, [51](#)
- novitec::CameraAPI::GEVCameraInfo, [33](#)
- novitec::CameraAPI::Handle, [35](#)
 - GetDescriptor, [35](#)
- novitec::CameraAPI::HostControllerInfo, [36](#)
 - HostControllerInfo, [36](#)
- novitec::CameraAPI::Image, [44](#)
 - Convert, [46](#)
 - Copy, [46](#)
 - Create, [47](#)
 - CreateJPEG, [47](#)
 - Load, [48](#)
 - Save, [48](#)
- novitec::CameraAPI::IPConfiguration, [48](#)
- novitec::CameraAPI::MAC_ADDRESS, [52](#)
- novitec::CameraAPI::NetworkAdapterInfo, [52](#)
 - GetAdapterName, [53](#)
 - GetDescription, [53](#)
 - GetFriendlyName, [53](#)
 - GetGateway, [54](#)
 - GetMACAddress, [54](#)
 - GetNumberOfGateways, [54](#)
 - GetNumberOfSubnets, [54](#)
 - GetSubnet, [55](#)
- novitec::CameraAPI::NU3CameraInfo, [55](#)
- novitec::CameraAPI::PixelFormatUtil, [56](#)
 - GetBitsPerPixel, [56](#)
 - GetBytesPerPixel, [57](#)
 - GetPixelFormatDescription, [57](#)
 - GetPixelFormatName, [57](#)
 - Is3DFormat, [58](#)
- novitec::CameraAPI::SUBNET, [58](#)
- novitec::CameraAPI::U3VCameraInfo, [59](#)
- Open
 - novitec::CameraAPI::GenICam::CameraInterface, [23](#)
- OpenFromMem
 - novitec::CameraAPI::GenICam::CameraInterface, [23](#)
- ReadMemory
 - novitec::CameraAPI::Camera, [12](#)
- ReadRegister
 - novitec::CameraAPI::Camera, [13](#)
- Save
 - novitec::CameraAPI::Image, [48](#)
- SendForceIP
 - novitec::CameraAPI::DeviceManager, [30](#)
- SetBufferMode
 - novitec::CameraAPI::Camera, [13](#)
- SetDeviceEventCallback
 - novitec::CameraAPI::Camera, [13](#)
- SetImageCallback
 - novitec::CameraAPI::Camera, [14](#)
 - novitec::CameraAPI::DeviceBase, [26](#)
- SetValue
 - novitec::CameraAPI::GenICam::Integer, [44](#)
- Start
 - novitec::CameraAPI::Camera, [14](#)
 - novitec::CameraAPI::DeviceBase, [26](#)
- Stop
 - novitec::CameraAPI::Camera, [14](#)
 - novitec::CameraAPI::DeviceBase, [26](#)
- UpdateFirmware
 - novitec::CameraAPI::Camera, [15](#)
- WriteMemory
 - novitec::CameraAPI::Camera, [15](#)
- WriteRegister
 - novitec::CameraAPI::Camera, [15](#)