

## ISG USB3 Camera API

### API Interface Overview

The following is an overview of how an application would interface to an USB3Vision camera through the ISGApi library ISGU3VApi.DLL. As with all camera interface software, the API can be thought of in two parts. A control interface for setting camera features (i.e. gain, exposure...) and a stream interface for capturing video.

In order to obtain the required files, visit the ISG website link under support/Allegro USB3 cameras <http://isgcameras.com/support-allegro-usb3.php>. There you will find the ISG GUI/API file installation link [ISG\\_USB3\\_UI\\_64-bit\\_Compressed-Folder.zip](#). After installing this project you will have the ISG GUI that can be used to run the camera. In addition, a sub folder called "library files" will be created which contains the .lib and .h files that make up the ISGApi library. This library provides the stream channel interface, along with a path to the control channel via the GenICam library.

The GenICam library requires a separate installation operation. Download the file [GenICam\\_v2\\_4\\_0\\_public\\_data.zip](#) from the same page as above on the ISG website. Install the host appropriate version of the GenICam project, and use the default selections which include the files that are needed for software development. When building your software project you will want to setup your library include paths to point to this GenICam installation, as well as the ISG library files of the previous paragraph.

**Control Interface:** The ISGApi library is built upon the GenICam reference library implementation which provides a path for all control operations in the camera. The GenICam reference library can be downloaded and installed from the GenICam website at: <http://www.emva.org/cms/index.php?idcat=47&lang=1>. The ISGApi library makes use of V2.4.0 of the GenApi which is the latest version released in 2/2014.

**Stream Interface:** The Drivers and DLL used for capturing video from the camera are the Microsoft WinUSB driver for USB3 device control and the libusb DLL for handling asynchronous bulk video data.

In order to support the GenAPI interface, the library is built as a CDevPort class which inherits from GenApi::IPort.

This allows a GUI to connect to the GenICam reference library in order to perform all control operations defined in the camera XML.

CDevPort is currently defined as a C++ Class, as a method of defining the virtual Read/Write functions of the GenApi IPort.

From the ISGU3VApi.h file, here are the CDevPort functions that provide video from the device via ISGU3VApi.DLL.

```
int CDevPort::initDevice();
int CDevPort::InitStream(uint32_t frameSize);
void CDevPort::HaltStream();
int CDevPort::GetNextFrame(uint8_t **imagePtr,uint32_t timeout);
int CDevPort::ReleaseFrame();
```

## ISG USB3 Camera API

A sample GUI project is provided in order to demonstrate camera control via a Visual Studio C++ project. Note that in the `GetNextFrame` interface, the application holds an image buffer that was allocated as part of a circular queue within the `ISGU3VApi.DLL`. The application is therefore not responsible for any image allocation. The ISG sample GUI makes the following calls in order to receive video:

Initialization:

```
CDevPort * theU3VPort; // create a device port for control functions
theU3VPort = new(CDevPort);
retVal = theU3VPort->InitDevice();
```

```
ISGCamera = new(CNodeMapRef);
retVal = ISGCamera->_Connect(theU3VPort,"Device"); // attach GeniCam API to the specific device
```

// Connecting to Video Interface:

```
theFrameSize = theImageHeight * theImageWidth * bytesPerPixel;
```

```
status = theU3VPort->InitStream(theFrameSize);
```

// while grabbing images, Get and Release frames

```
status = dialogPtr->theU3VPort->GetNextFrame(&pdata,1000); // 1 second timeout
dialogPtr->theU3VPort->ReleaseFrame(); // done with frame, return to circular buffer
```

```
theU3VPort->HaltStream(); // stop video when all done getting images
```

The default startup video mode of the camera is continuous acquisition. In this mode, the sensor is streaming frames across the USB3 bus. If the host GUI application will be provided with the most recently captured image. Alternatively, the camera can be used in single frame acquisition mode. Using this mode along with the software trigger command gives the user control over the image pipeline. In this mode the host software will use the same video interface, starting with an `InitStream` function call. A `GetNextFrame` call can then be made to prepare for the software trigger of a frame.

// Connecting to Control Interface:

Refer to section 2.9 of the GenApi standard document

[http://www.emva.org/cms/upload/Standards/GeniCam\\_Downloads/GeniCam\\_Standard\\_v2\\_0.pdf](http://www.emva.org/cms/upload/Standards/GeniCam_Downloads/GeniCam_Standard_v2_0.pdf) for a complete description of the control interface of the ISG camera. In that document are details on interface functions such `GetMin`, `GetMax`, `SetValue`, `GetValue` ..., that can be used for each of the `Integer/Float/Boolean/String` ... types that are available in the camera.