LightWise®

Imaging Solutions Group

Software Installation Instructions

No MVTec Halcon Install – ISG GUI only

ISG LightWise IQ
GigE Vision Cameras

IMAGING SOLUTIONS GROUP

Imaging Solutions Group
1387 Fairport Road, Suite 890
Fairport, NY 14450 USA
585-388-5220 x101
http://www.isgcameras.com

Software Install Instructions Overview

The ISG LightWise™ IQ Cameras are provided with a set of software that you can use to operate the cameras for the first time. These can be found on the ISG CD that is shipped with our cameras or on our website.

1) **ISG LightWise™ IQ Camera Control Graphical User Interface (IQ-GUI)**

ISG LightWise™ Camera Control Graphical User Interface (IQ-GUI)

Please note that the ISG Camera Control GUI (Graphical User Interface) can run on a Host PC running either 32-bit or 64-bit Operating systems. This software requires the Reference Library from the GenICam™ Standard. An executable utility provides an interface to the reference library of GenICam standard. This must be installed prior to using the camera if you want to use the ISG Camera Control GUI Software. This utility software can also run under either 32-bit or 64-bit Operating systems. You need to load the proper software for whichever operating system your Host PC is using.
This software provides an interface to the reference library of the GenICam standard.

The software is located in a zip file by the name of **GenICam_v2_4_0__public_data.zip**
This zip folder will contain two executable files that loads the software.

The ISG GUI uses this library to set the control parameters on the camera.

Windows 32-bit OS = use =	**GenICam_VC80_Win32_i86_v2_4_0.exe**
Windows 64-bit OS = use =	**GenICam_VC80_Win64_x64_v2_4_0.exe**

Please install the GenICam™ Software Utility prior to installing the ISG IQ-GUI

IQ-GUI for 32 bit OS	**Call if needed**
IQ-GUI for 64 bit OS	**ISG_Gige_UI_64-bit_v1.21_Compressed-Folder.zip**

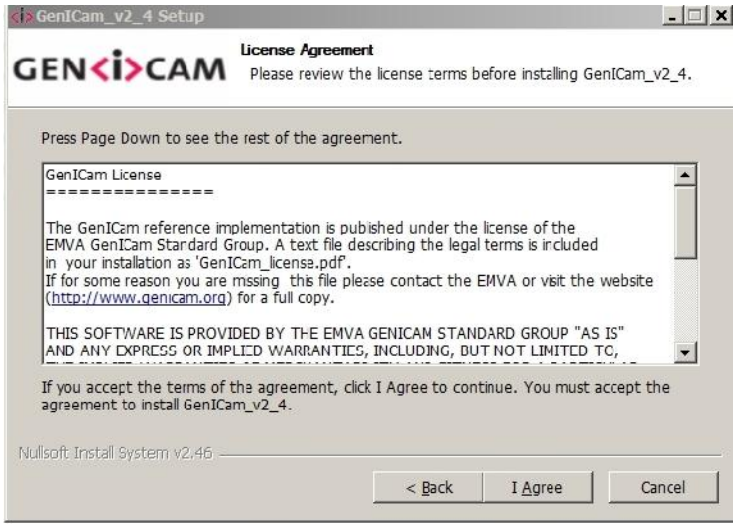
Install GenICam Software:

This software provides an interface to the reference library of GenICam standard.
The ISG GUI uses this library to set the control parameters on the camera.

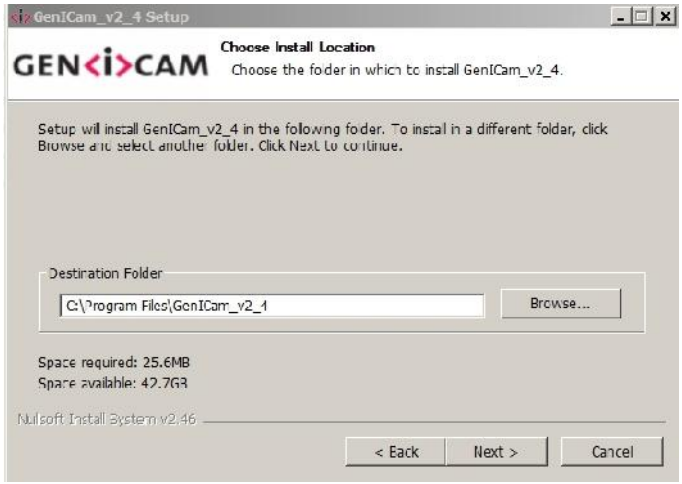
Windows 32-bit OS = use = GenICam_VC80_Win32_i86_v2_4_0.exe

Windows 64-bit OS = use = GenICam_VC80_Win64_x64_v2_4_0.exe

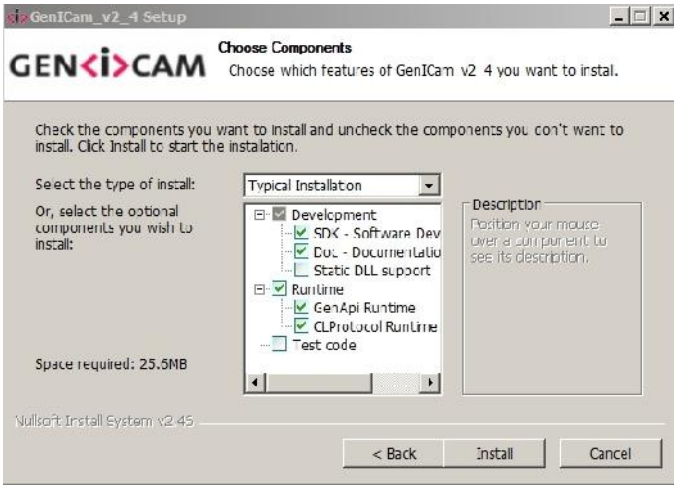
Next: Agree to License Agreement.



Then: Choose Install Location and the Start Menu Folder (creates shortcut).



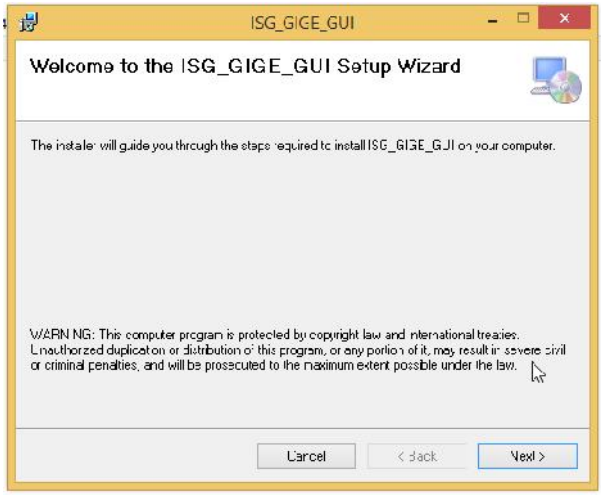
Next: Choose Start Menu Folder (creates shortcut).
Then: Choose Components to install. (Typical is fine)

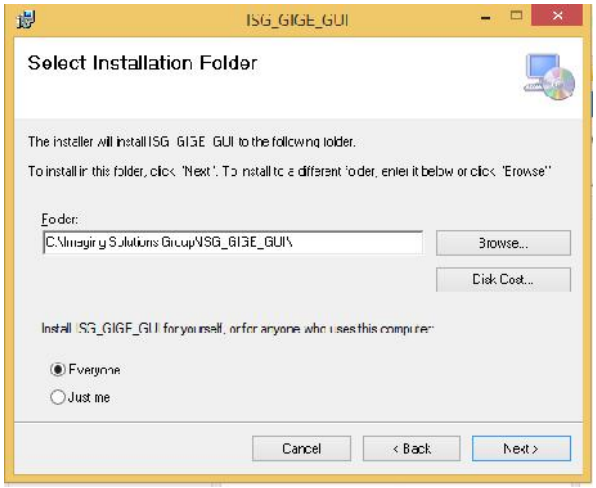


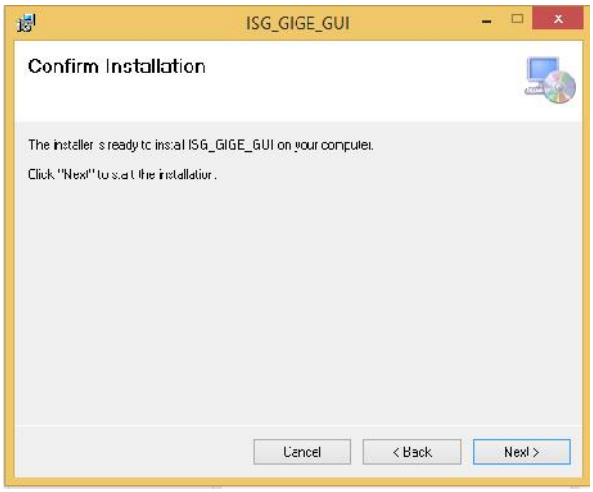
Next: This installs the software. = Click to finish. Install Finished!

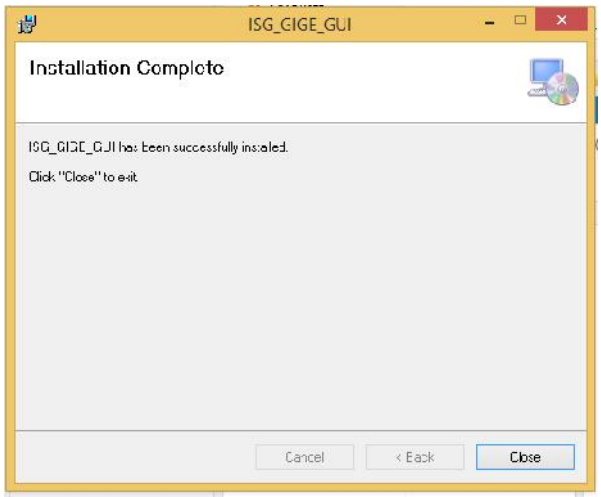
Next: Install the ISG GUI (Graphical User Interface).
This software is located in one of two folders.
ISG LightWise IQ GUI Software - 32-Bit Call if needed.
ISG LightWise IQ GUI Software - 64-Bit ISG_Gige_UI_64-bit_v1.21_Compressed-Folder.zip
For Windows 64-bit OS = look in release folder = Release_121_x64 = Click setup.exe

Next: Select installation folder & confirm installation.
This installs the ISG GigE GUI. The OS may require permission – click allow.
This will put a shortcut onto the desktop.









"C:\Imaging Solutions Group\ISG_GIGE_GUI_SETUP\"
ISGUserInterface.exe

GigE Install Notes:

1) Host GUI expects to match a known named string for the host card.
    a. Host Cards using Intel chips are preferred.
2) Host card should be set to fixed IP address 192.168.2.1
3) Host application must be added in the firewall settings as an "allowed" program.
    a. ** Make sure both private and public networks are checked.
4) Host card must be set to "gigabit" instead of "auto-negotiate"
5) Jumbo packets should be enabled for best performance

Troubleshooting:

A) Red X mark on Network status with camera connected -- see item #4 above

B) Host GUI talks to the camera, but no video is seen -- see item #3 above

C) Host GUI shows fast frame rate, but video is blank -- see item #5 above (packet size mismatch)

D) Host GUI cannot discover camera -- see item #2

6) On most Windows systems, it is necessary to setup the GigeUI application as an approved program within the firewall settings.

7) Installing in the program files path requires administrator privledge in order to write files (such as the local xml file written by the ISG UI).

8) The GUI/Camera will allow packet sizes of up to the Jumbo max of 9K. Some cards still do not support this, and therefore will not ship video.

Error Condition: The camera is powered on, but we cannot connect to, or ping to, the camera.

Error Cause: Looking in device manager revealed that the network adapter devices were both in a warning, “not working properly” state

Error Resolution: Performing a disable followed by enable sequence on the device inside device manager resolved the issue.

Additional references

ISG LightWise™ IQ webpage

Imaging Solutions Group
1387 Fairport Road, Suite 890
Fairport, NY 14450 USA
585-388-5220 x101

http://www.ISGCameras.com

ISG LightWise™ IQ webpage
http://www.isgcameras.com/gigE-CoaXpress-cameras.php

ISG LightWise™ IQ Documentation
http://www.isgcameras.com/support.php
This manual can be downloaded from this webpage.

ISG GigE API SDK
Application Programming Interface and Software Development Kit
http://www.isgcameras.com/SDKs-APIs.php

ISG LightWise™ Firmware
http://www.isgcameras.com/support.php
Firmware updates will be provided from this webpage.

GigE Vision®

This industry standard is managed by the Automated Imaging Association (AIA) at:
http://www.visiononline.org/vision-standards-details.cfm?id=36&type=5

GigE Vision® is a global camera interface standard developed using the Gigabit Ethernet communication protocol. GigE Vision allows for fast image transfer using low cost standard cables over very long lengths. With GigE Vision, hardware and software from different vendors can interoperate seamlessly over GigE connections

The GigE Vision specification relies on GenICam™ to describe the features supported by the camera. This description takes the form of an XML device description file respecting the syntax defined by the GenApi module of the GenICam specification. GenICam is a generic programming interface for all kinds of cameras. For more information visit:

www.genicam.org

The GenICam™ standard group is hosted by the European Machine Vision Association (EMVA).

GenICam™ overview

Today's digital cameras are packed with much more functionality than just delivering an image. Processing the image and appending the results to the image data stream, controlling external hardware, and doing the real-time part of the application have become common tasks for machine vision cameras. As a result, the programming interface for cameras has become more and more complex.

The goal of GenICam™ is to provide a generic programming interface for all kinds of cameras. No matter what interface technology (GigE Vision, Camera Link, 1394 DCAM, USB3 Vision, etc.) they are using or what features they are implementing, the application programming interface (API) should be always the same.

The GenICam™ standard consists of multiple modules according to the main tasks to be solved:

- GenApi: configuring the camera
- Standard Feature Naming Convention (SFNC): recommended names and types for common camera features
- GenTL: transport layer interface, grabbing images
- CLProtocol: GenICam™ for Camera Link
- GenCP: generic control protocol
- GenTL SFNC: recommended names and types for transport layer interface

The GenICam™ group also provides GenICam™ reference implementation. The reference implementation (not part of the standard) comes with a BSD-like license and can be freely used by associated members of the GenICam™ group.

GenICam™ Documentation:

http://www.emva.org/cms/index.php?idcatart=76&client=1&lang=1

GenICam™ GenApi Standard Document 2.0 GenICam_Standard_v2_0.pdf
This document provides a description of the interfaces that are available for each data type defined by the specification (see section 2.9).

GenICam™ Standard Features Naming Convention 2.0 **GenICam_SFNC_2_0_0.pdf**
This document defines all of the features that are common to a digital video camera, and therefore most cameras such as the ISG device, conform to these functions, along with some additional custom controls.

Official GenICam™ Flyer GENiCAM_Flyer.pdf
GenICam™, generic interface for cameras: interconnection made easy

GenICam™ Introduction genicam_introduction.pdf
Brief introduction to GenICam™ standard

GenICam™ for Developer's genicam_for_developers.pdf
More detailed presentation designed for developers and other technically oriented audience

These documents can be found on our CD or from the GenICam Website.

GigE Vision Documentation:

GigE Vision Standard Specification **GigE_Vision_Specification_2-0-02.pdf**

Automated Imaging Association **AIA_Pixel_Format_Naming_Convention_1-1-00.pdf**

ISG GigeVision Host Application Interface Developer's Guide:

ISG GigeVision Host Application Interface Developer's Guide **ISG_Genapi_Interface.pdf**

The purpose of this document is to describe the way in which a user can create a Windows application to control and acquire video from an ISG GigeVision™ camera.

This document is found below and on the CD shipped with the camera and on our website.

The purpose of this document is to describe the way in which a user can create a Windows application to control and acquire video from an ISG GigeVision™ camera. Two independent libraries are used for this purpose. The first is the GenICam™ library which is a reference implementation that is supplied by the EMVA standards organization. This provides an interface to the full feature set of the camera. A suite of header files which are installed in the Program Files\GeniCam_ver_xx include path provide all of the class and type definitions that are required. A pair of GenICam™ standard documents have been included within the ISG software disk that provides details on the GenICam interface. The first one is the GenICam_Standard_vxx.pdf document. Among other things, it provides a description of the interfaces that are available for each data type defined by the specification (see section 2.9). The second GenICam™ standard document in the software packet is the GenICam_SFNC_xx.pdf file. This document defines all of the features that are common to a digital video camera, and therefore most cameras such as the ISG device, conform to these functions, along with some additional custom controls.

Here is the quick start guide to a set of commands that can be used to control an ISG GigeVision™ camera via the GenICam/GigeVisionLib library interface.

The ISG sample GUI is referenced as a source code example of these operations.

1) Discovery and Connection to the camera

In order to find a camera on the host network, a CDeviceDiscovery object is created, and a discovery operation is started. This is shown in the "OnBnClickedButtonDiscovery" button in the GUI. When a device is discovered, it is added to the device list (m_pDeviceList variable), and its MAC address is displayed in the discovery list box.

The discovery response from the camera is stored to a GigEDevice variable called m_pCurrentDevice in the GUI, which is copied to the active device called m_pDevice. This is the variable that is used to connect to the camera device port. The statically compiled GigeVision.lib library, which provides the software API to the camera, defines a CControlChannel object in order to create a linkage between the GigEDevice and the camera.

The device is "connected" to the camera as shown in the "OnBnClickedButtonConnect" function of the GUI. Once this path has been established, the CControlChannel can then be used to access resources in the camera. As defined in the GigeVision™ specification, the XML data file of the camera can be located by reading first URL data at address 0x200. This is shown in the GetXmlFileData function. As shown there, a generically named "HostXMLFile.xml" is generated. This file is then used in order to attach the GenApi defined CNodeMapRef object to the camera's supported resources via a LoadXMLFromFile member function.
Refer to the GenICam™ standard document for a description of the variable type categories that are defined which allow the full functionality of the camera to be controlled. Also reference the GenICam SFNC (standard function naming convention) document for a complete listing of all of the common

features that are available for control via the GenApi. Note that subsets of these, as listed in the XML file of the camera, are available in the ISG device.

2) Using the control interface of the GenApi to set camera features

The ISG GUI provides the complete list of features that are supported in the camera via a CTreeCtrl variable as defined by the Microsoft Foundation Class (MFC) library. The DumpFeatures function of the GUI allows a user to traverse the entire list of features, which always begin at the "Root" node of a camera. When the root node is opened, the top level feature categories are then displayed. As a subsequent control set is opened, the next level of feature detail becomes available. Eventually the user will be given a set of selections that map to register control of the camera.

Each of these controls has a specific, GenICam™ defined, data type associated with it. Examples of these data types include IInteger, IFoat, IString, and ICommand, each of which provides their own specific methods of getting and setting data. There are also some common member functions among the various data types. For instance, most provide both a SetValue and GetValue method for writing integer or floating point data to the camera. Other functionality such as "execute", is specific to a single data type, in this case the ICommand type.

Following the CTreeCtrl model of the GUI provides a very convenient way to display the data types associated for each of the camera features automatically. It also enables a user to interact with all of the control data as shown in the edit boxes, drop list boxes and command buttons on the right hand side of the Camera Feature Control group box.

3) Camera video setup routines and data interfacing

The previous section described the way in which an application writer can display and interface to the entire camera feature set. There are also examples in the GUI of setting up a minimal number of features that will allow a custom image frame size to be delivered. The OnBnClickedButtonStartCamera function can be referenced for this purpose. In that function, the height and width resources of the camera are accessed in order to read their state. This access is available using the GetValue method, via an integer pointer variable which was assigned using the GetNode and the respective resource name. In the same way, the SetValue method could have instead been used in order to setup a desired custom frame. The StartCamera function then proceeds to go through the process of setting up for video transmission. This is done by opening and starting a data stream via the GigeVision.lib API (OpenStreamChannel, StartStream methods) and then executing the "AcquisitionStart" command. The ViewThreadMethod process is kicked off in the GUI in order to receive the video data. Inside this thread, the "WaitForBuffer" function will block until the image becomes available within the data buffer pointer.