NEW ELECTRONIC TECHNOLOGY

Quick Start Guide

SynView

NET's Software Development Kit

Table of Contents

List of Figures

List of Tables

General information

Scope of the manual

This manual walks you through the SynView software installation process (**Installation details** p.46) on various operating systems, suggests how to test the installation and provides information about various configuration (**Connecting and configuring the camera(s)** p.7) and troubleshooting options for SynView and its components (**Troubleshooting & Support** p.73). It also provides high-level overview of the package and its relationship with industry standards (Vision Standards p.70).

Related documents

SynView programmers guide — overview and tutorial of the SynView API library.
GigEPRO user manual — user manual for the NET GigEPRO camera series.
GimagoEasy user manual — user manual for the NET GimagoEasy camera series.
CorSight user manual — user manual for the NET CorSight smart camera series.

Quick start

This chapter provides instructions how to quickly set up and test a system using the SynView package with Net GmbH hardware. Individual sections discuss following tasks:

- Connecting the camera — install, connect and configure the cameras
- Installation — quick guide through SynView installation process under Windows
- Basic camera configuration — prepare the camera for the first acquisition
- Test the camera — instructions how to verify camera functionality in SynView Explorer

Connecting and configuring the camera(s)

Before installing the software and testing the functionality, let's prepare the camera(s) for use. The range of Net GmbH camera selection consists of several main families:

- GimagoEasy — CCD based GigE Vision cameras (Gigabit Ethernet interface); hardware manual: ***GimagoEasy user-manual***
- GigEPRO — advanced version of GigE Vision cameras supporting various CMOS sensors including image processing capabilities on embedded FPGA; hardware manual: ***GigEPRO user-manual***
- CorSight — true PC based camera; hardware manual: ***CorSight user-manual***

GigEPRO & GimagoEasy

1. Consult precautions listed in ***GigEPRO /GimagoEasy user manual***. Be sure to not violate instructions in the manual
2. Connect the camera to proper power supply
3. Connect the camera to the network, or directly to the PC using a suitable Ethernet cable. We recommend using shielded (S/STP) category 6 cables or better.
4. If you have a DHCP server running in the network, the camera will get a proper IP address automatically. Otherwise you will need to adjust the IP configuration of the camera using tools from the SynView package; once it is installed (other software packages might also offer IP configuration tools).
5. It is highly recommended and best common practice to install a dedicated network card, which only interfaces to camera(s) and will be dedicated to image acquisition only. In such case the firewall can be fully disabled for this network interface (windows XP).

For Windows 7 the Firewall settings (Control Panel-> System and Security) can be changed after selecting “Allow a program through Windows Firewall”.

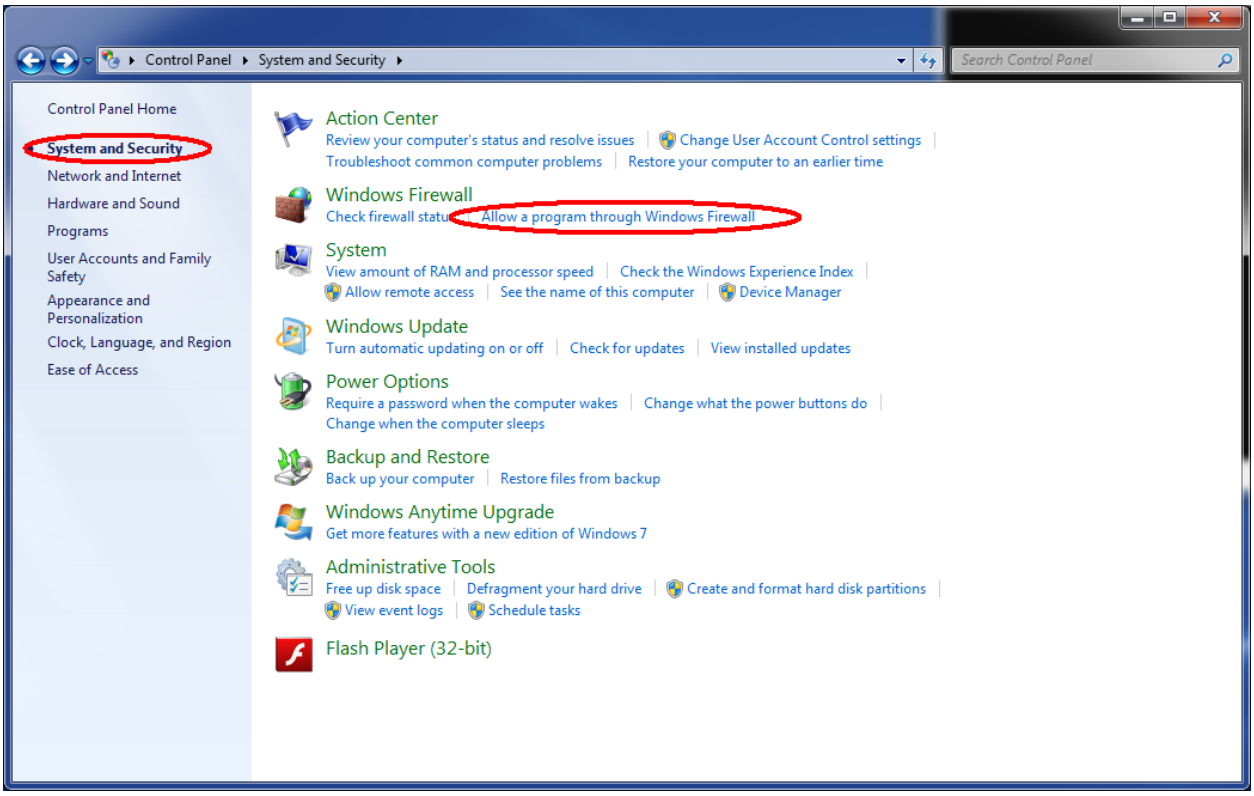


Figure 1: Windows 7 Firewall: Allow Program Through Firewall

In the following menu check “Change settings” first and then select “Allow another program…..”.

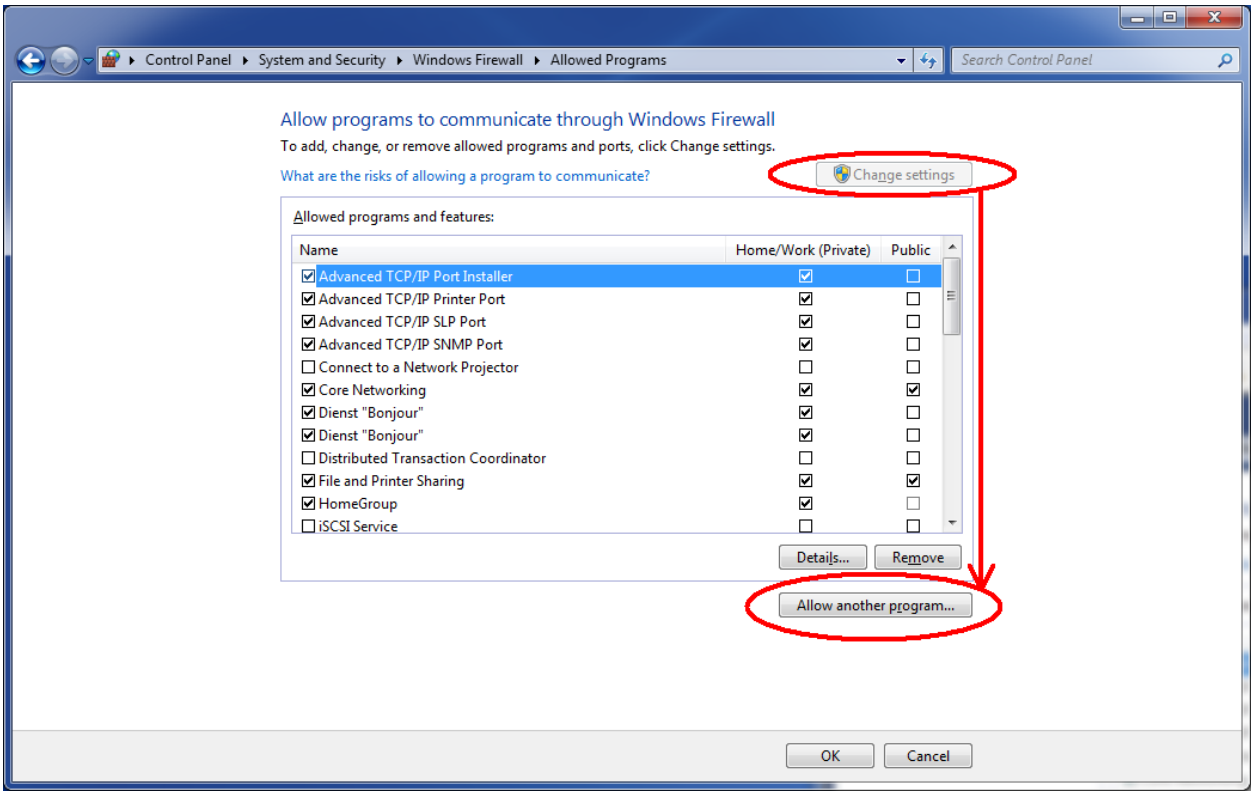


Figure 2: Windows 7 Firewall: Allow another Program Through Firewall

That allows to select the SynView Explorer executable from the program list to become an exception for the firewall.

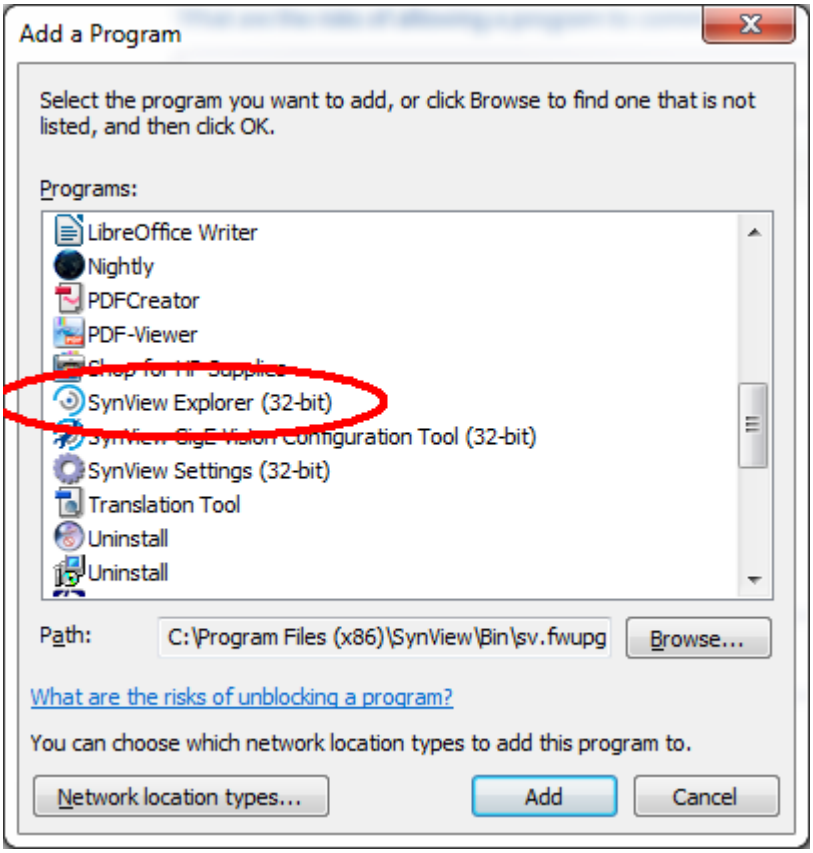


Figure 3: Windows 7 Firewall: Allow SynView Explorer

After selecting the SynView Explorer in the Program list double-check the option “Network location types….” And make sure both options for the network types are selected.

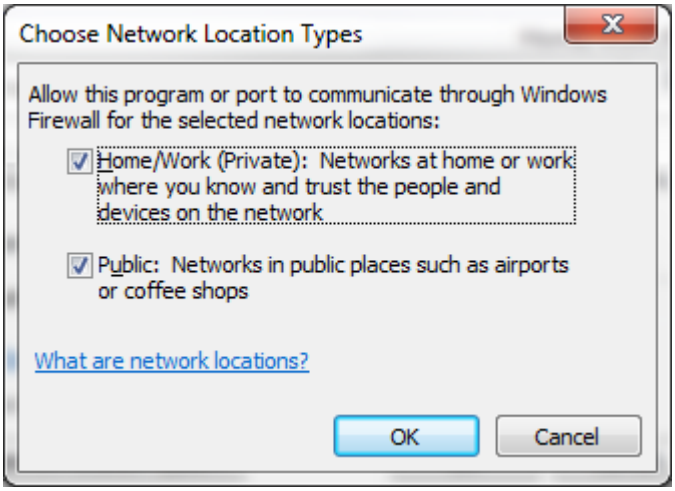


Figure 4: Windows 7 Firewall: Network Location Types

Finally double check in the Advanced Security subtab of the Windows firewall menu that with all “Inbound Rules” of the SynviewExplorer the UDP Protocol selected.

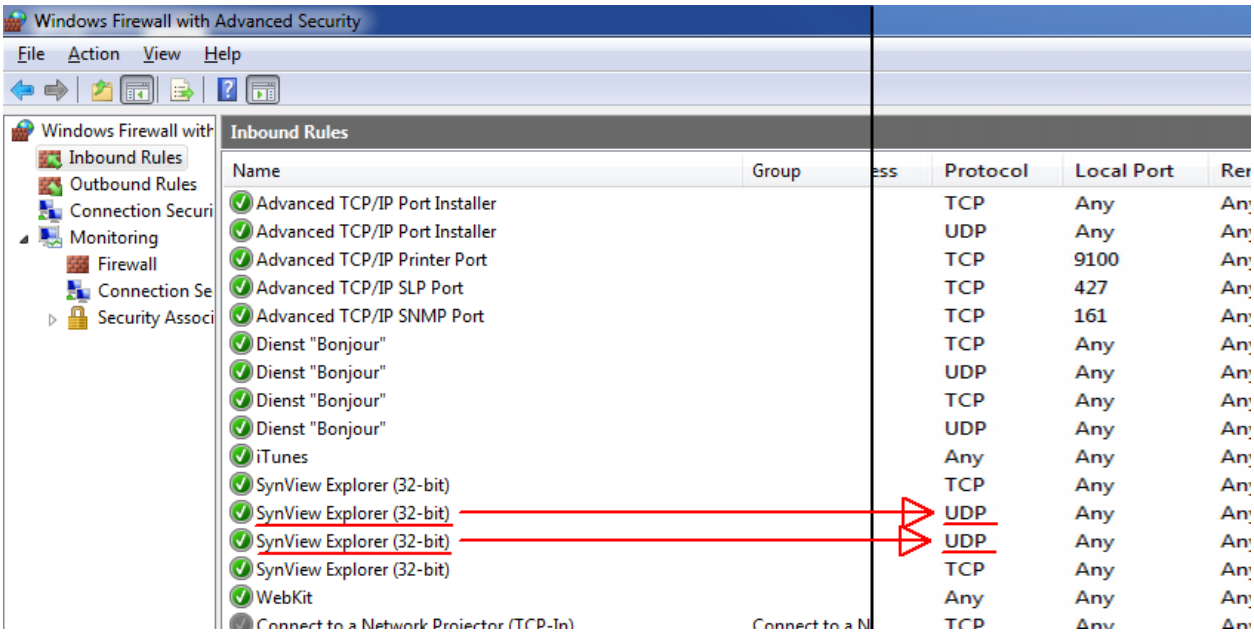


Figure 5: Windows 7 Firewall: Advanced Security Settings

- If your network configuration, however, requires sharing internet and other network access with the image acquisition over the same media (which is not recommended practice), you will need to configure your firewall so that it does not block the communication with the camera. In Windows Firewall this can be done by creating an exception for each application that will be accessing the camera — especially for the SynView Explorer that we will use for the first tests (possibly limiting the exception's scope to the local subnet only). Note that the applications to be unblocked (including SynView Explorer) are not usually present yet in the system at the time of installation, therefore this step might have to be repeated after the installation itself and again for each new application. Therefore we again recommend having a network connection dedicated solely for camera connection — and disabling firewall fully for this connection.

CorSight

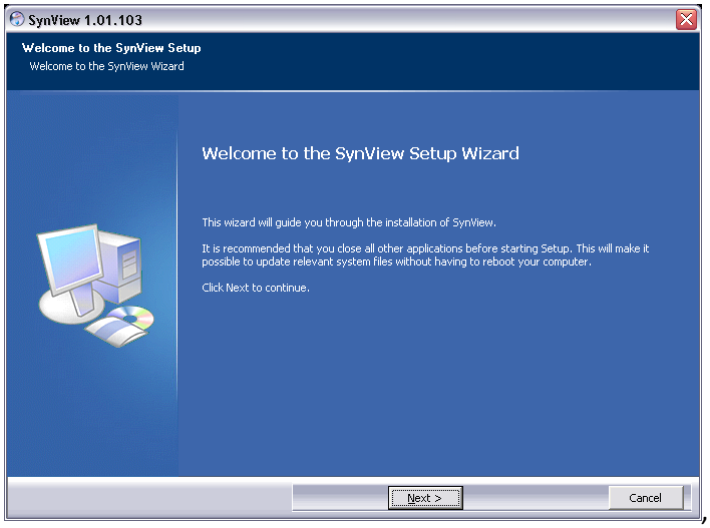
1. Consult precautions listed in CorSight user manual. Be sure to not violate instructions in the manual.
2. Unpack the camera, connect necessary peripherals and power it up.
3. Assuming that the desired operating system is already pre-installed on the camera, no additional steps are needed and the camera is ready for SynView installation and test.

SynView installation — quick guide for Windows

Following points guide you quickly through SynView installation under Windows. If you need detailed instructions or if installing in another operating system, refer to full SynView "**Installation details**" **p.46**.
Now let's walk through the installation procedure:

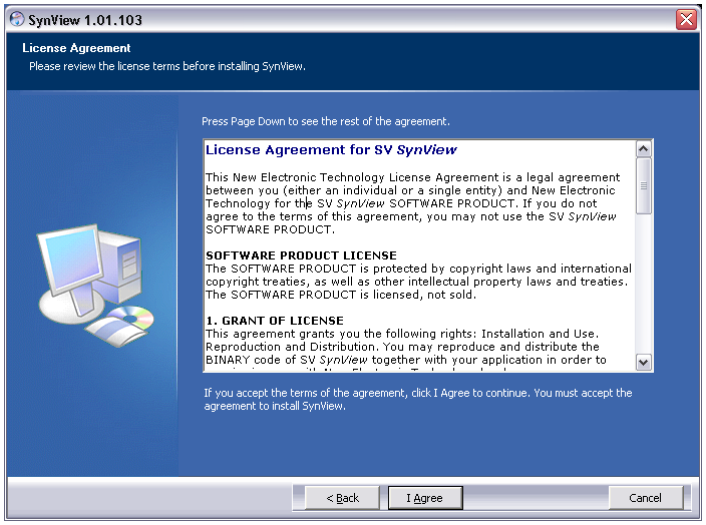
1. Prepare the installation media. The software should be installed by a user with administrator rights.
    a. If you have the installation CD, insert it in the CD-ROM drive and wait until installation starts.
    b. If you have downloaded the installer from our website, execute it.
2. Depending on security adjustments of your system and whether you are running the installer from a local disk or a network share, you might get an "Open File - Security Warning". Simply proceed by clicking Run.

   The installer welcomes you with following screen:



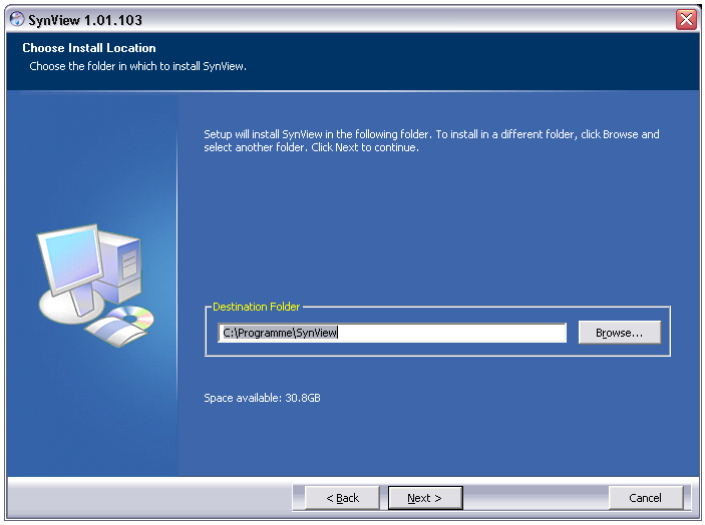
   click Next.

3. Next screen is the license agreement. Please read carefully the license text and if you agree.



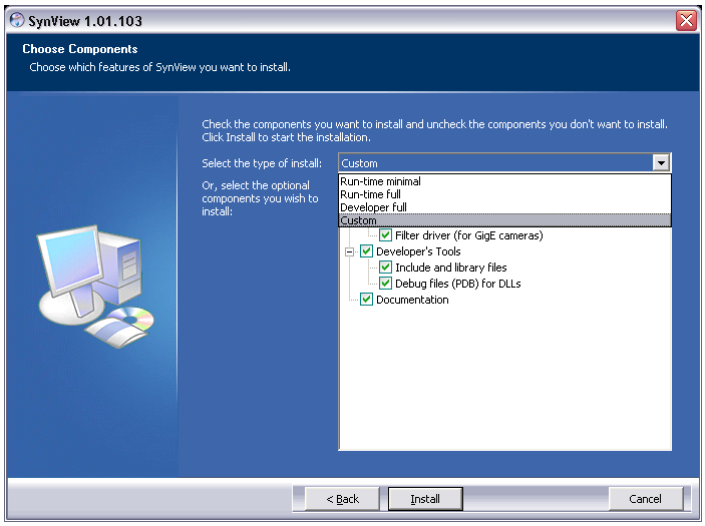
   Click the I Agree button to proceed with installation.

4. Next the installer asks you about the location where to install SynView. The default location will do well in most cases.



Keep it or select another preferred location and click Next.

5. If running Windows 7 or newer Windows version, the default software installation directory (C:\Program Files) is read only. The installer will ask you where to put the application data requiring write access (the dialog will not appear on Windows XP). Select a desired location and click Next.
6. In the next step you can select the SynView components to install. For testing cameras the installation of the “runtime” components is sufficient. For programming with SynView you need to make sure the Developer's tools are installed as well.



When finished with the selection, click Install.

7. After starting the installation, the selected components will be installed to your computer. The installer will inform you about the progress.
8. When completed, the installer displays the final dialog. Keep the default option (Reboot now) and click Finish. After rebooting, SynView is ready to use.

Basic camera configuration

For CorSight cameras no additional actions are needed, you can directly proceed with testing the camera. When using GigEPRO /GimagoEasy cameras, you might need to adjust their IP configuration so that it fits your network settings.
Start the SynView GigE Vision Configuration Tool: Start→Programs→SynView→SynView GigE Vision Configuration Tool (under Linux, start /opt/synview/bin/sv.ipconf).
The tool will scan for the GigE Vision cameras connected to the system. If the camera(s) is directly accessible from your PC, they will be displayed with a green "OK" icon. In such case the camera(s) is(are) ready for acquisition, no further actions are necessary.

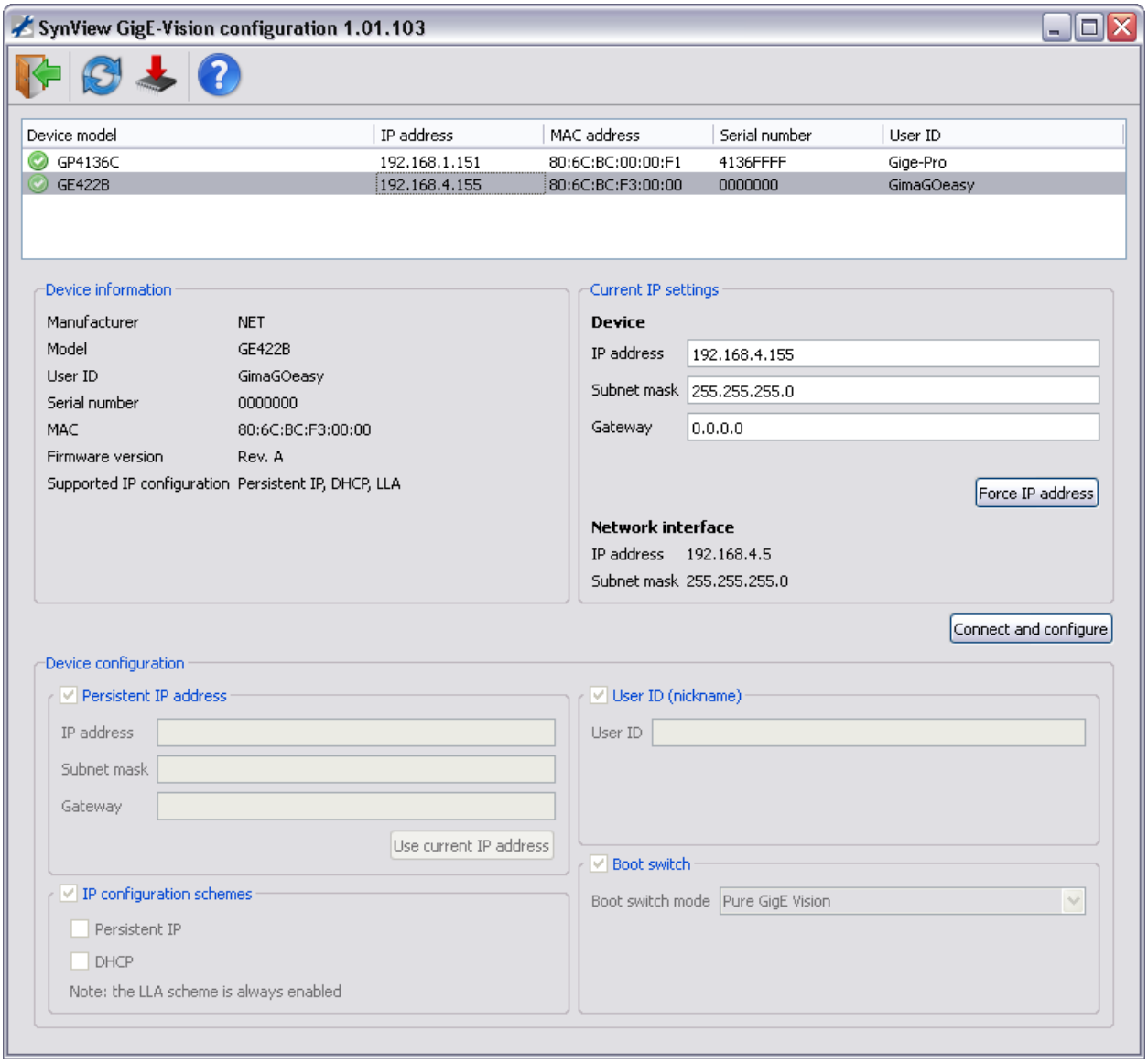


Figure 6: SynView: GigE-Vision camera configuration

If the camera is displayed using a yellow icon, it is "visible" from your PC, but cannot be directly accessed, because it is configured with different subnet than your network card. In such case, select the

camera in the list, adjust its IP configuration so that it matches settings of your network card, select the Fix IP checkbox and click the *Set persistent* Conf button. When finished, the camera's IP configuration should match the network card. It will display with the green icon in the list and will be now directly accessible from the PC. It is ready for test.
If desired, the camera's “nickname” (user name) can be adjusted as well, together with the IP configuration. The nickname can be used to identify individual cameras connected to the network.

Testing the camera(s)

The SynView package contains the SynView Explorer tool, which allows to enumerate, connect and configure cameras, acquire images or generate sample source code for SynView API. It is a useful tool for testing both the SynView and camera functionality.

1. Start the tool from system menu: Start →Programs → SynView →SynView Explorer (under Linux, start /opt/synview/bin/sv.explorer).
2. Connect the camera to be tested — select it in the list of found cameras and press the Connect camera button. Of course, the camera must be connected and powered.

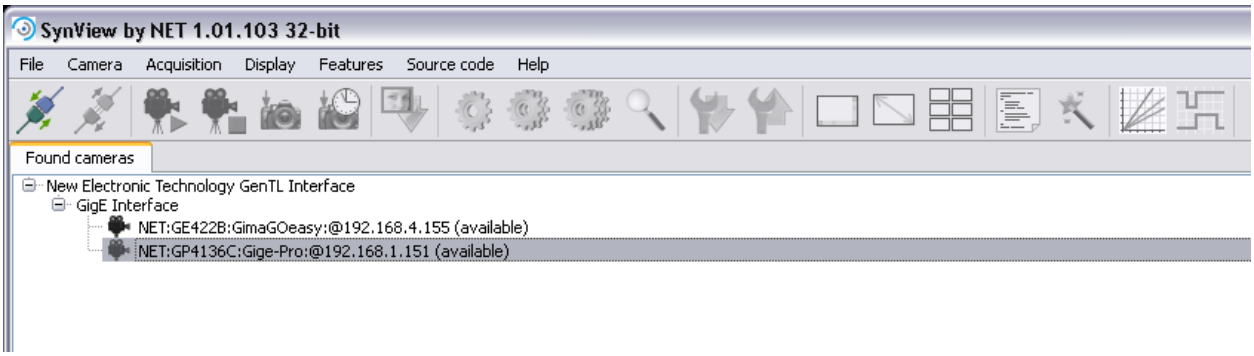


Figure 7: SynView Explorer: GigE camera enumeration

3. Configure the camera features as desired. Pay attention especially to features in Image Format Control and Acquisition Control categories.

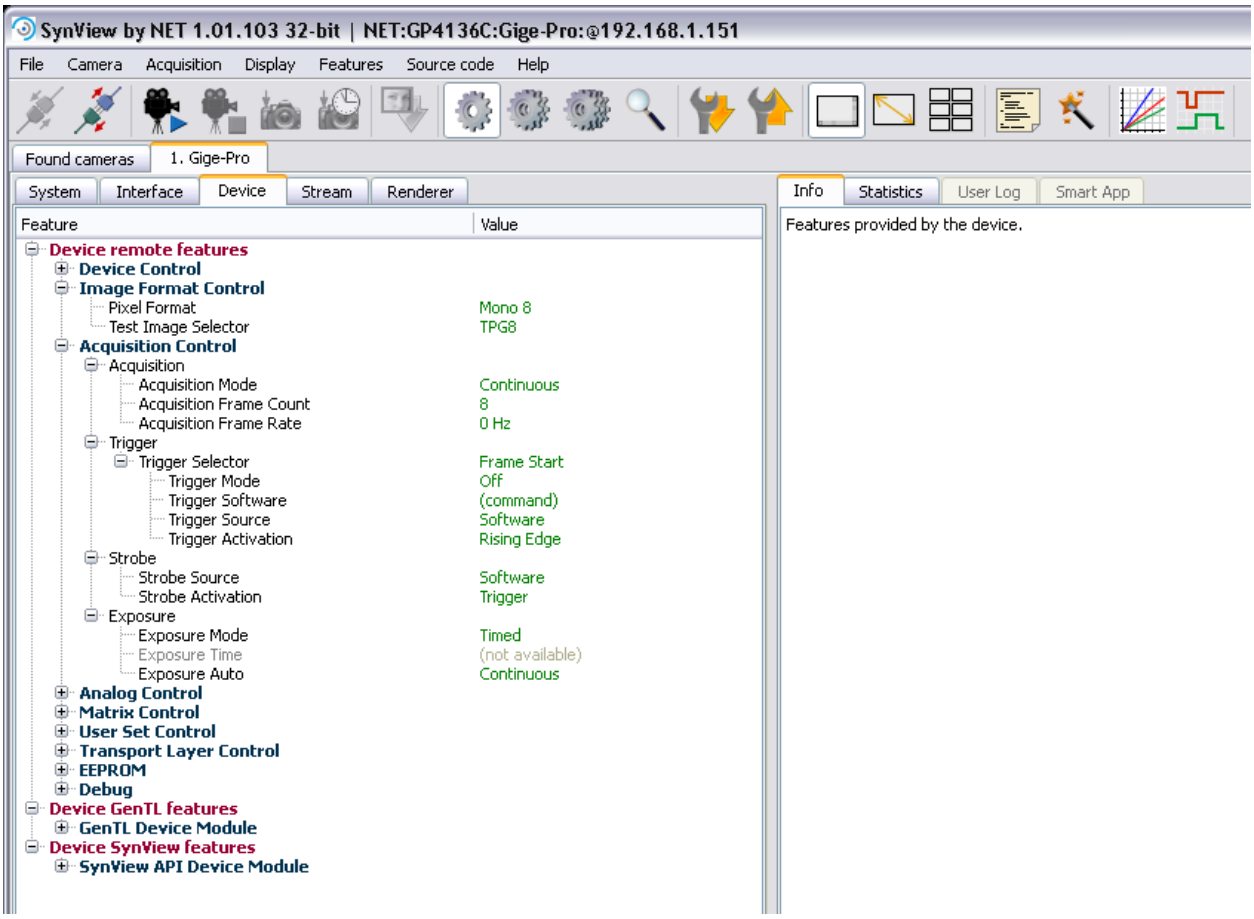


Figure 8: SynView Explorer: GigEPRO Acquisition Control

4. When prepared, click the Start acquisition button — the camera should start acquiring. If not, verify again, if the camera is properly connected and running, if the system was properly configured (remember the notes above about firewall, network configuration, etc.) or if the camera was not set to triggered mode, while no trigger was attached.

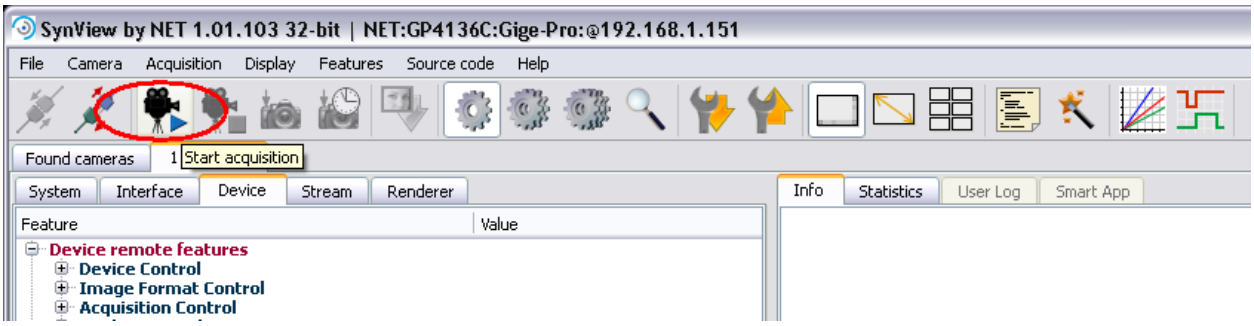


Figure 9: SynView Explorer: Start Acquisition

5. While the acquisition is active, you can still adjust the runtime parameters, such as exposure time or gain. Basic acquisition parameters, such as pixel format or trigger mode become locked when the acquisition starts.

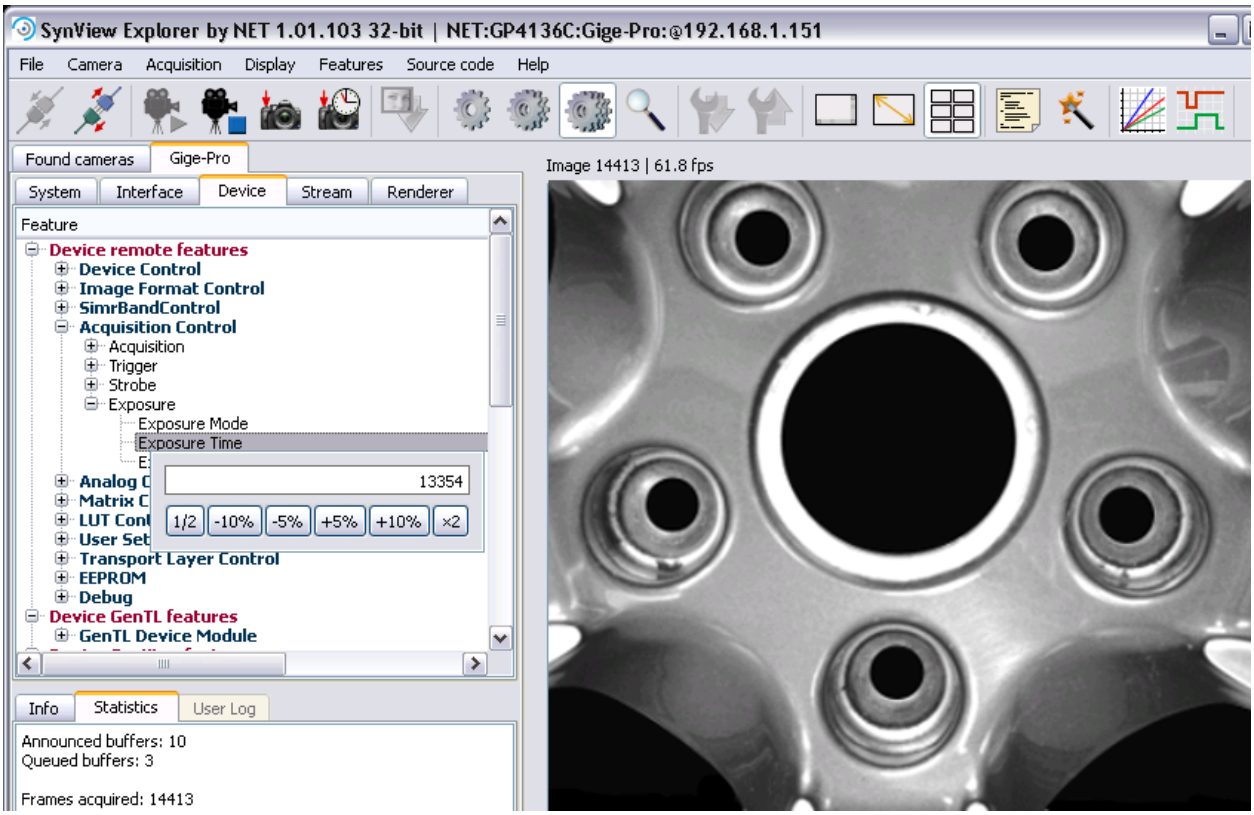


Figure 10: SynView Explorer: Play and Multiwindow Pre-view

6. When finished, click the Stop acquisition button and exit.

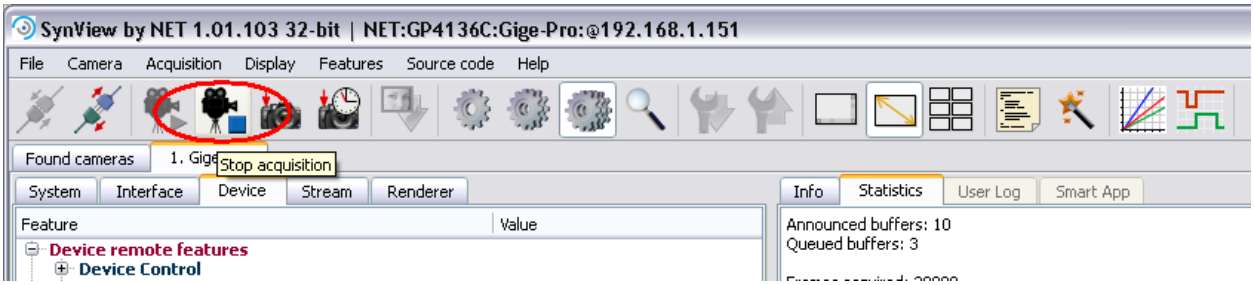


Figure 11: SynView Explorer: Stop Acquisition

Where to go next?

Once finished with the basic tests and verification, you might want to:

- Get more detailed information about installation process for all supported operating systems **Installation details** p.46 or information about customizing the linux installer ”
- **Custom** installation” p.64.
- Learn about SynView configuration options (”**SynView Settings utility and the configuration file**” p.67) and overview of troubleshooting and technical support approach (”**Troubleshooting & Support** p.73).
- Get overview about SynView package, its components, functionality and connectivity (”**SynView overview**SynView overview” p.18), including the SynView Explorer tool (”**SynView Explorer in detail**”p.24). Learn to understand the GenICam based camera feature tree and other important SynView API principles (”**Important principles behind SynView interface**” p.39).
- Know basics about the important machine vision standards, GenICam and GigE Vision (”**Vision Standards**” p.70) and how you can take advantage of advanced support for these standards in SynView when interfacing 3rd party software and hardware products.
- Study documentation for your hardware:
    - ***GigEPRO user manual***
    - ***CorSight user manual***
- Study documentation for the SynView API:
    - ***SynView programmer's guide***

SynView overview

This chapter gives a high level description of the SynView package, its components and the supported hardware.

Connecting your application with our cameras

The SynView package is available in versions for Windows and Linux operating systems, running on 32-bit and 64-bit PC architectures.
All supported platforms are since beginning treated with the same priority, share the equivalent feature range and provide the same toolset. Releases for all platforms are synchronized and all tests are executed on every platform. Migrating to a different operating system or upgrading to a 64-bit architecture is thus very straightforward and painless.
There are multiple ways how to include our cameras in your application.
The most natural one is to use the SynView software package and its SynView API library, which provides all the means for comfort work with the hardware and the acquired images. The sample code generator accompanying the library helps to integrate SynView in your application with very limited effort.
Because the low level hardware access library is a full featured GenICam GenTL Producer, the entire functionality of all our camera models is readily available in any image processing software package supporting GenTL standard, such as Halcon, ActivVisionTools or Common Vision Blox.
Furthermore, the GigE Vision based cameras (GigEPRO and GimagoEasy) are fully compatible also with those packages that did not yet implement GenTL, including LabView, VisionPro or MIL (Matrox Imaging Library).

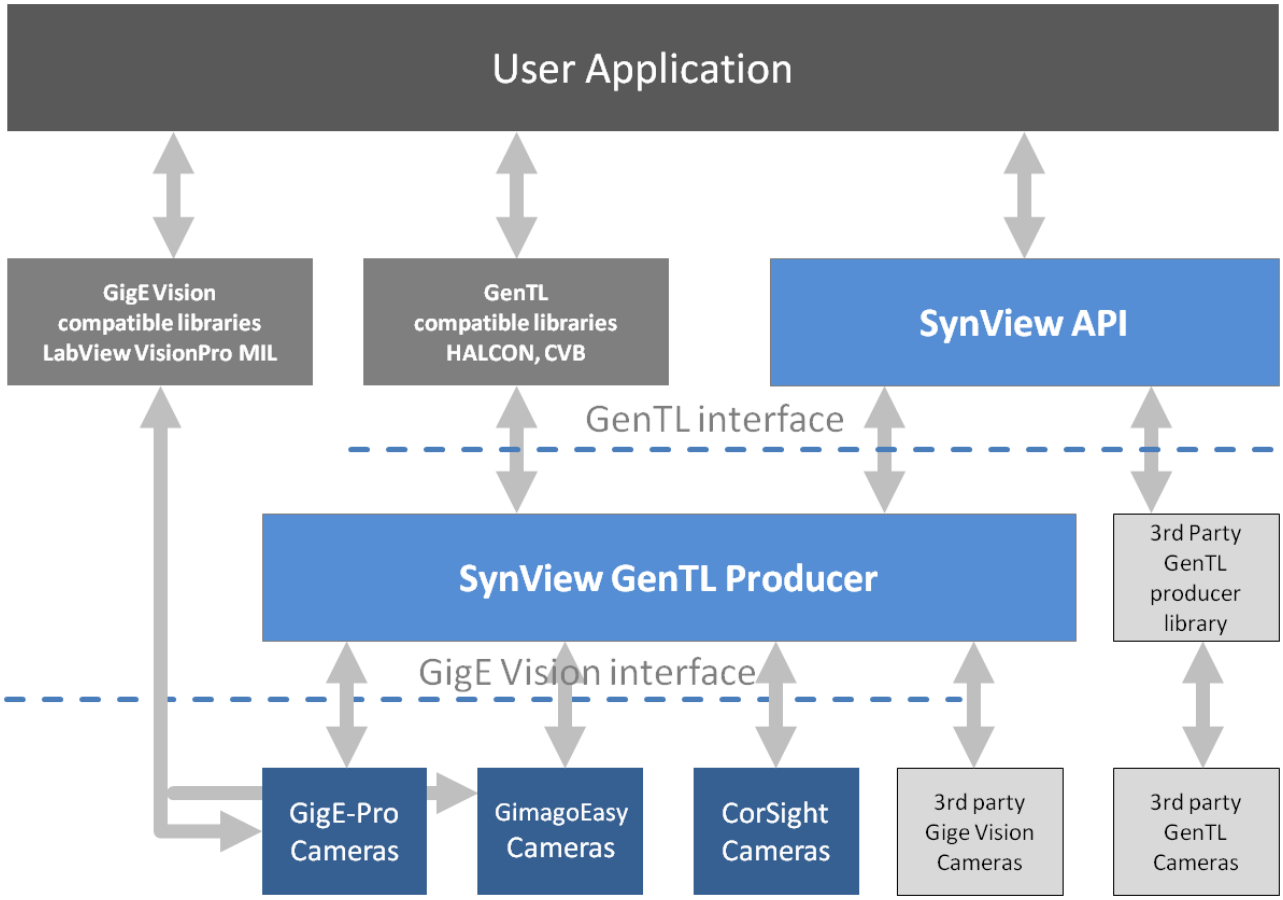


Figure 12: SynView API: connectivity capabilities

Thanks to SynView's wide compatibility with GenTL and GigE Vision standards, NET can also accommodate customers who need to combine NET cameras with a 3rd party model in a single project with a single API.

SynView GenTL Producer

The SynView's hardware access layer unifies access to the entire camera product range, covering all interface and sensor types and exposes that through a single interface. The interface is compatible with the GenICam GenTL standard (being a full featured GenTL Producer) and can thus be connected to SynView upper layers (SynView API) or to any GenTL Consumer library. SynView is also a fully GigE Vision compatible software.

SynView API

The diagram shows the basic architecture of the SynView API library, showing the most commonly used components.

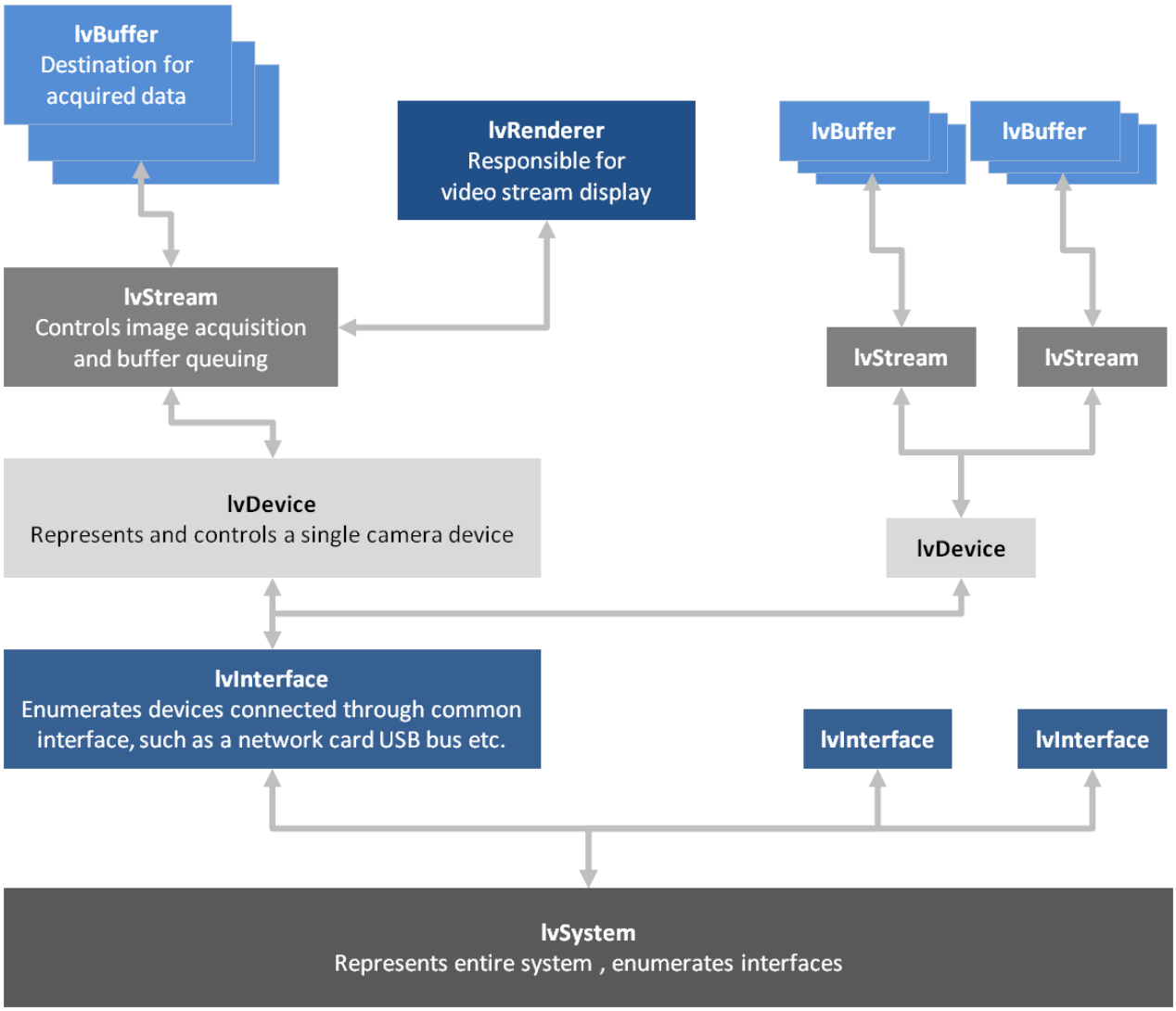


Figure 13: SynView API: basic architecture

The SynView API functionality is available either through a pure C interface, through an object-oriented C++ interface or as a .Net class library.
It supports various use cases, from trivial applications, where most functionality, including configuration, threading or display is handled internally for you by the library to complex scenarios, where user has fine grain control over everything.
The main features of the SynView API library were already outlined before.
Although the library provides the same functionality as any generic GenICam based package, SynView API goes way further, providing direct access to all common camera features, shortcuts for easy handling of complex feature sets. Of course it handles all obvious acquisition related tasks such as buffer queuing, saving images etc.

SynView API comes also with a preprocessing library that unifies access to preprocessing capabilities of some cameras and similar functionality in the software. When replacing a simple camera with a model capable of real-time preprocessing, the same user code will keep working seamless, only the task will get automatically offloaded from the host processor to the camera.
Last but not least, it also offers GUI components and dialogs for inclusion in user applications –such as the GenICam feature tree control.
While providing add-on features utilizing work with NET cameras, the SynView core is kept strictly generic, so that it works well with any GenICam compatible product.

SynView Explorer and SynView Source Code Generator

The SynView Explorer tool is a primary testing and demonstration program built on top of the SynView API — it provides easy access to the SynView API and camera functionality, available right after the installation without any programming. Besides that, the SynView Explorer offers additional built-in features designed to assist the developer to create a SynView API application quickly and easily.
The user interface of the SynView Explorer is discussed in "SynView Explorer in detail" p.24. Writing applications utilizing all advantages of modern cameras, in particular if the application should stay generic, error resistant and open for cameras of different models (possibly even different technologies or different vendors) requires good understanding of principles behind the GenICam standard. And that usually means quite a steep learning curve, even if the details are wrapped in a well designed API.
To simplify and streamline the development process, NET provides the SynView Source Code Generator tool.

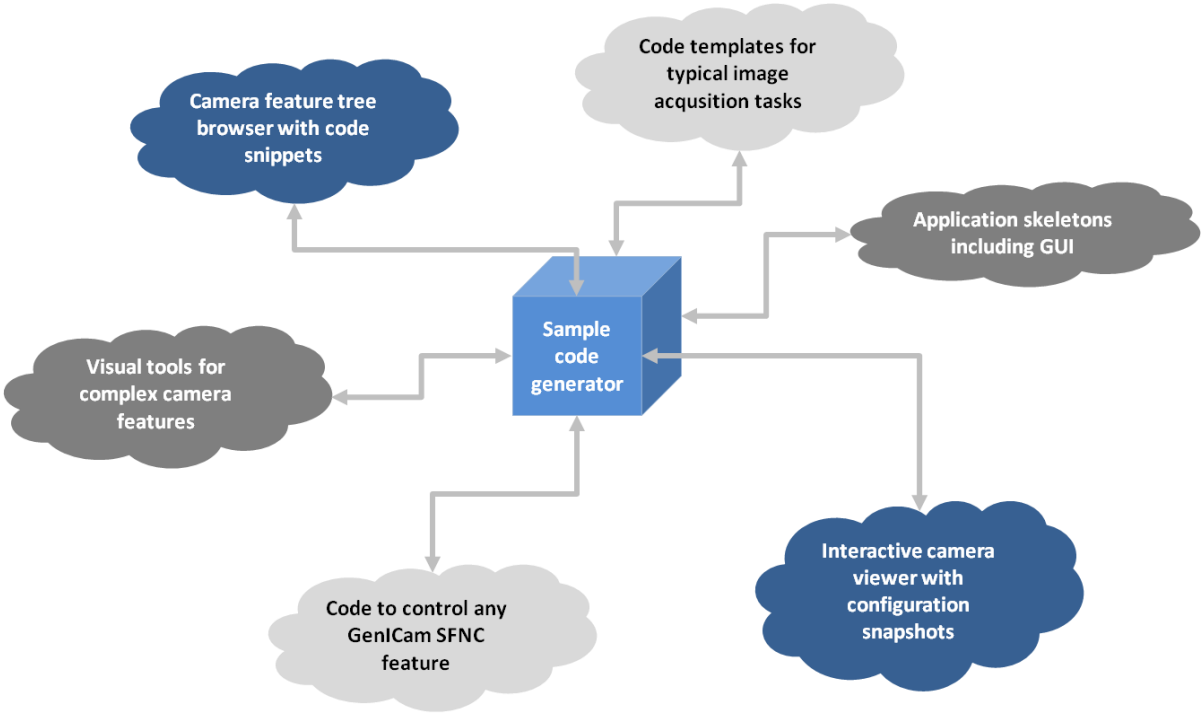


Figure 14: SynView Explorer: Source Code Generator features

- Single-line snippets to full-fledged examples including GUI.
- Templates for typical image acquisition and processing tasks, eg. camera enumeration, buffering techniques, HW/SW preprocessing etc.
- Browsing the camera's feature tree, generating code to control given feature. Browsing the feature tree including all standard SFNC features.
- Visual tools for complex camera features (trigger modes, complex digital I/O, LUT's and color transformations, AOI, etc.).
- Interactive camera configuration snapshot & reuse full camera settings.
- C, C++, C++.Net, C# or VB.Net code. Operating system independent code. Optional GUI elements based on Qt or Win32 API (eventually Xlib).
- Interactive code generation; cut & paste the code to your application.
- Fully generic, works with any GenICam compatible camera.
- Generating customized code examples through several mouse clicks. The tool runs under Windows and Linux.

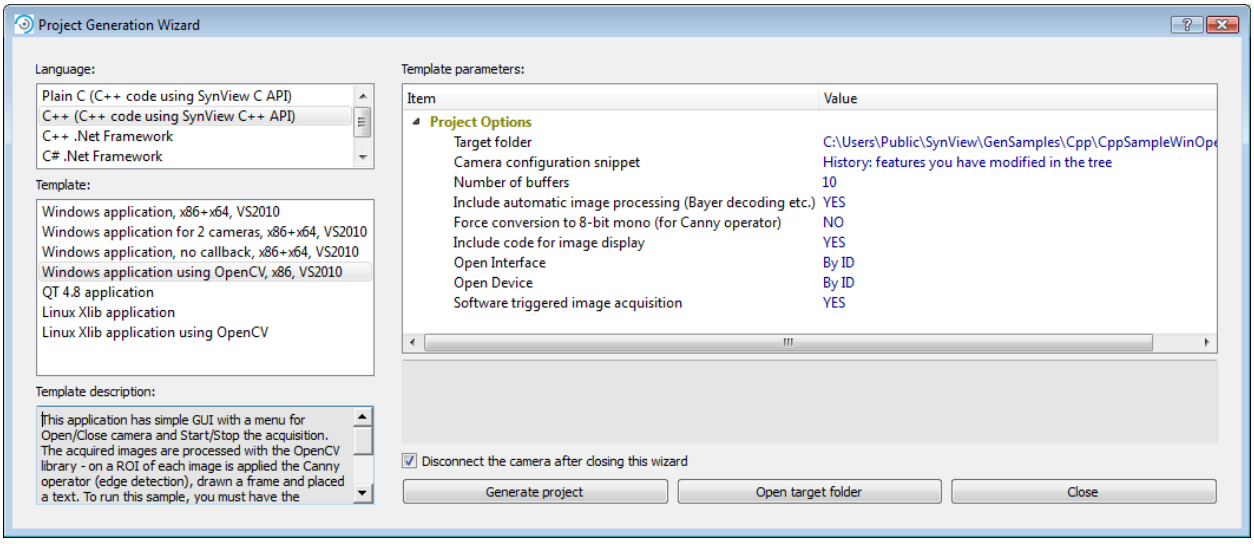


Figure 15: SynView Explorer: Source Code Generator configuration

Details about the SynView Source Code Generator with guidelines how to use it to quickly develop an application with SynView API are provided in ***SynView programmer's guide***.

Industry standards

SynView by NET uses the GenICam GenTL interface as a middle layer of the camera SDK. This allows maximum flexibility and both-side openness of the library. The industrial standards being used in SynView are listed below.

Table 1: Industrial Standards used in SynView SDK

GenICam GenApi	GenICam API, defining means to describe all camera features, their corresponding control registers and providing access to those features
GenICam GenTL	GenICam GenTL (Transport Layer), an interface between applications and camera-control libraries, covering device enumeration, image acquisition and related tasks.
GenICam SFNC	SFNC (Standard Features Naming Convention), standardizing names (ID's), types and other aspects for most common features of a machine vision camera
GigE Vision	Set of UDP/IP based protocols standardizing access to Gigabit Ethernet based cameras and devices.

More information about these industrial standards is provided in ”**Vision Standards**” p.70.

Licensing information

The SynView package and/or the software included in the cameras uses the following 3rd party software components:

- GenICam GenApi reference implementation — distributed under the GenICam license by the GenICam committee. The copyright is held by the GenICam committee. The package as well as the license is available from www.genicam.org. The GenICam GenApi reference implementation in turn uses Apache Xalan-C++ and Xerces-C++ libraries (Apache license, www.apache.org) and the modified MathParser library (LGPL license, kirya.narod.ru/mathparser.html).
- zlib compression library: available from www.zlib.net under zlib license.
- The Independent JPEG Group's JPEG software (libjpeg): available from www.ijg.org (Independent JPEG Group).
- OpenCV: available from opencv.willowgarage.com under BSD license.

SynView Explorer in detail

The SynView Explorer is a demonstration and developer tool delivered with SynView. It is written fully using the SynView API and therefore can serve as demo for both the camera and SynView API capabilities.
In Windows, the SynView Explorer can be started from the system menu:
Start →Programs → SynView →SynView Explorer.

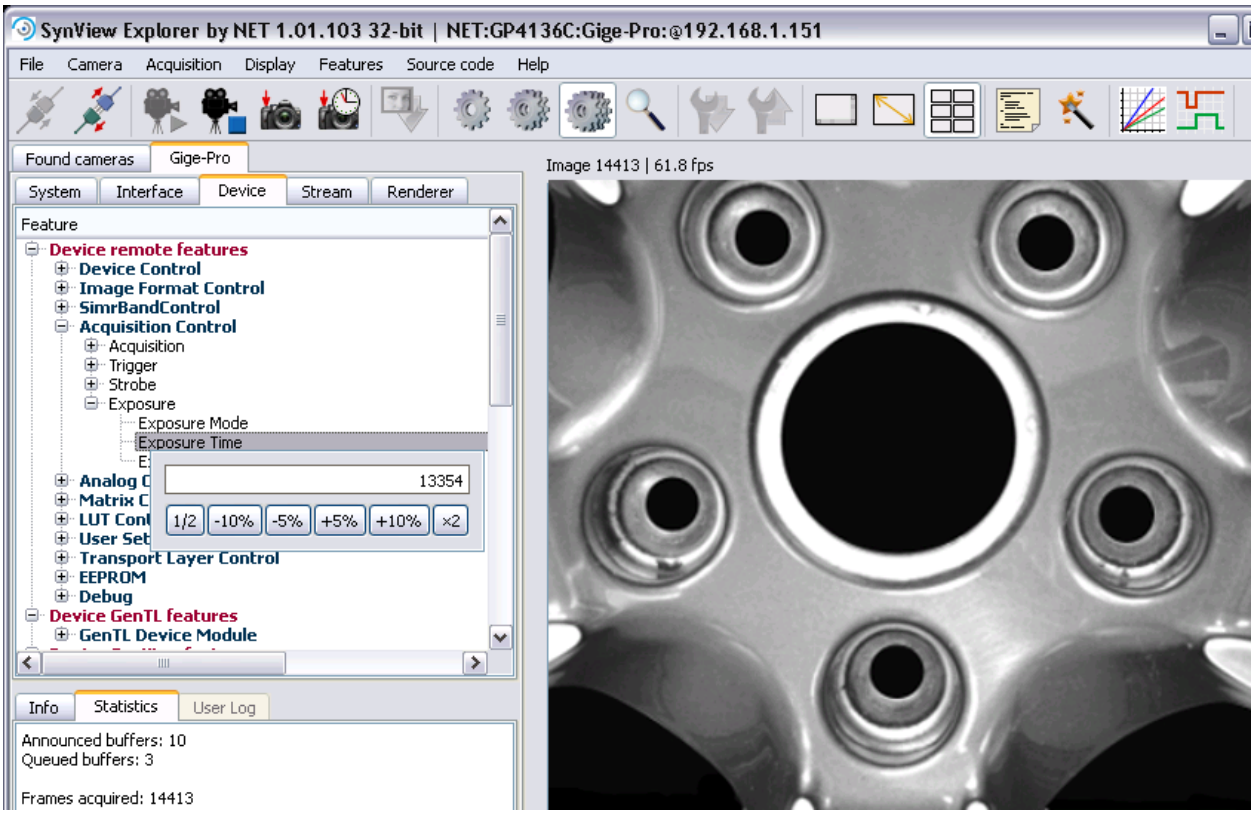


Figure 16: SynView Explorer running under Windows

In Linux, it can be started for example by executing the sv.explorer command from the shell, the installer attempts to add SynView executables to the system path. If not, it can still be invoked using full path: /opt/synview/bin/sv.explorer.
Only screen shots from Windows are shown. In Linux they would be identical (with the difference only in the display options).
SynView Explorer offers its functionality in a set of menus. Many menu items are duplicated as toolbar buttons, for convenience. In the screen shots we will show mostly the usage of the toolbar buttons.
SynView Explorer can be run in:

- Single camera (compact) mode, which uses only single application window. In this mode only one camera can be opened.
- Multiple cameras mode, which uses one window for controls and one window for each connected camera. In this mode you can connect multiple cameras and use them concurrently.

The mode can be set in the Settings dialog (see "Settings" p.37) and is applied when the Explorer is started next time.

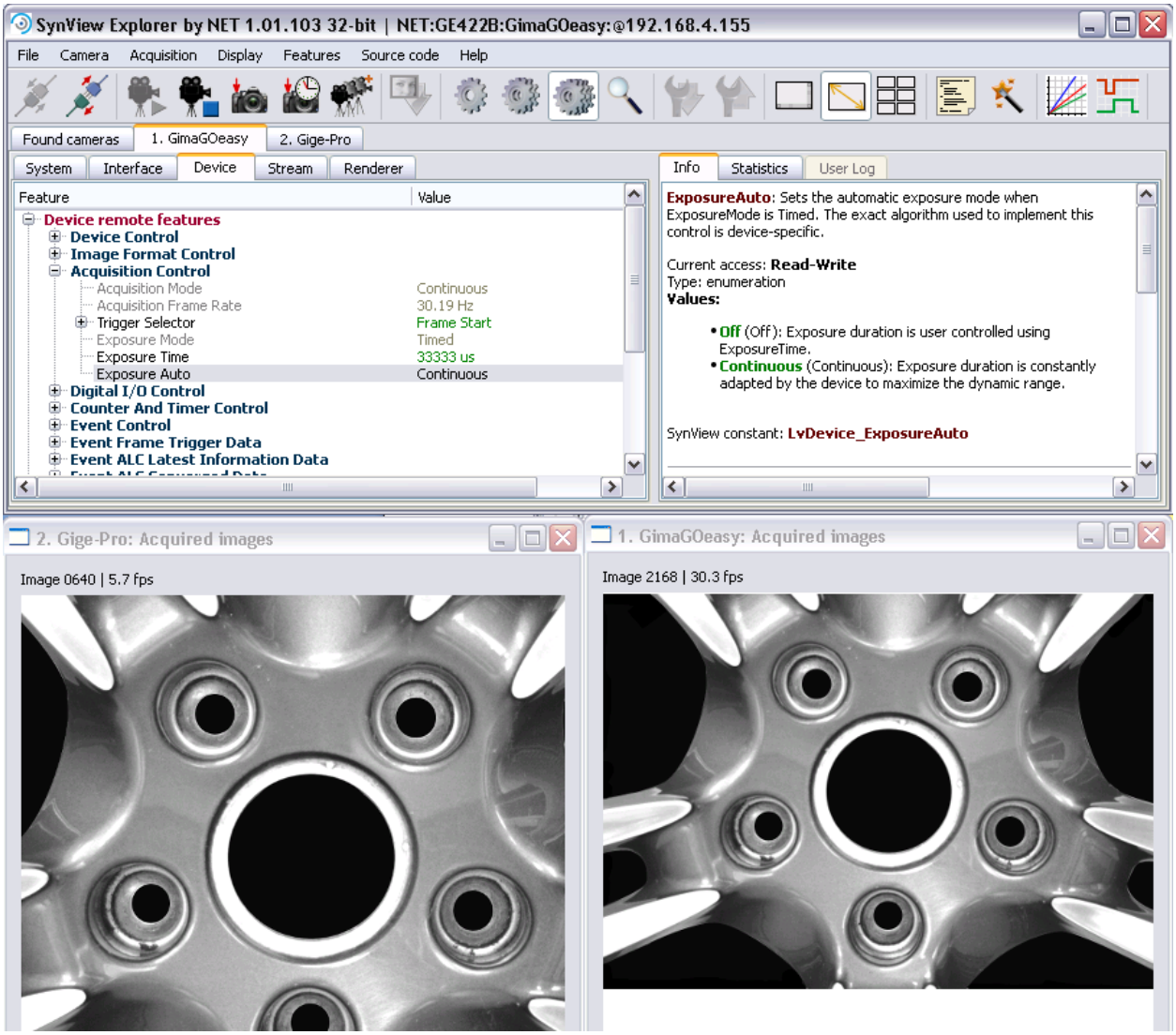


Figure 17: SynView Explorer running in multiple cameras mode

Connecting and disconnecting the camera

First step after starting the tool is to select and connect a camera. SynView Explorer uses regular SynView approach to enumerate cameras — depending on the configuration it can offer cameras from a single or from all GenTL Producer libraries available on the system. By default, it connects to the SynView GenTL Producer, which handles all NET cameras (GigEPRO ,GimagoEasy and CorSight families) as well as 3rd party GigE Vision compatible cameras.

All the found cameras are displayed in a tree on the Found Cameras tab. The tree lists all found Systems (GenTL producers), under each System it lists all Interfaces, containing at least one active device

(camera), and under each Interface it lists all found Devices. When you select a device, you can connect it by clicking on the Connect camera button, or simply by double-clicking on the Device.

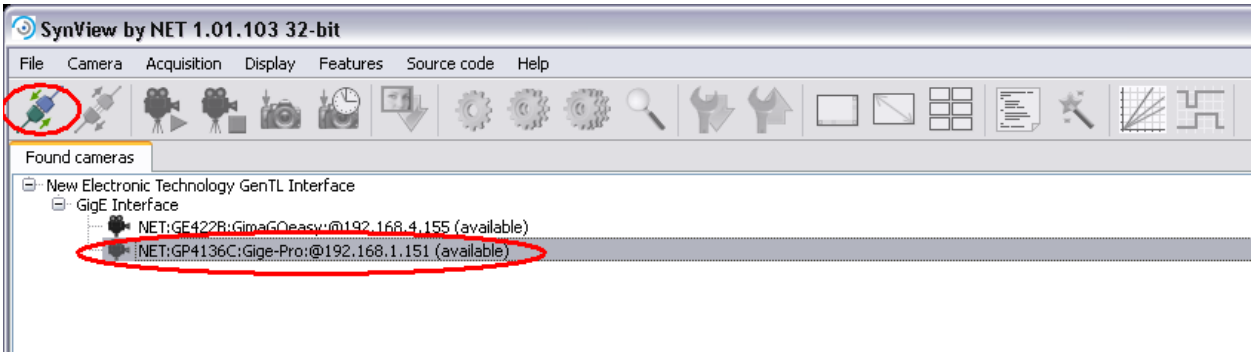


Figure 18: SynView Explorer: Connecting a camera

If the desired camera is not available, check again, if it is properly connected and powered. For GigE Vision cameras, check also the network configuration and firewall settings. SynView Explorer checks for the cameras existence at startup. You can also update the camera list during the Explorer run — use the Update camera list(s) menu item.

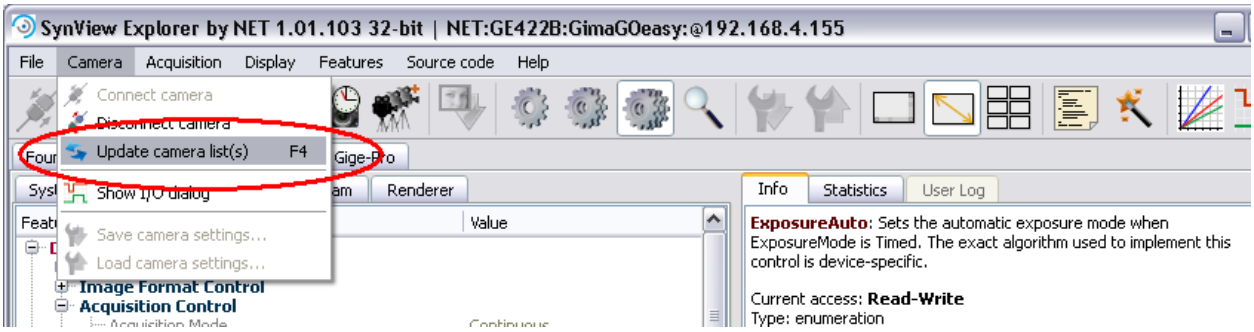


Figure 19: SynView Explorer: Updating the camera list

SynView Explorer connect the camera in exclusive access mode, that means other applications cannot connect to this camera, while is connected by the SynView Explorer. You can disconnect the camera and release it for other applications using the Disconnect camera button. Remember to do so namely when you come to the Source code wizard, where the generated application will need to connect to the same camera.

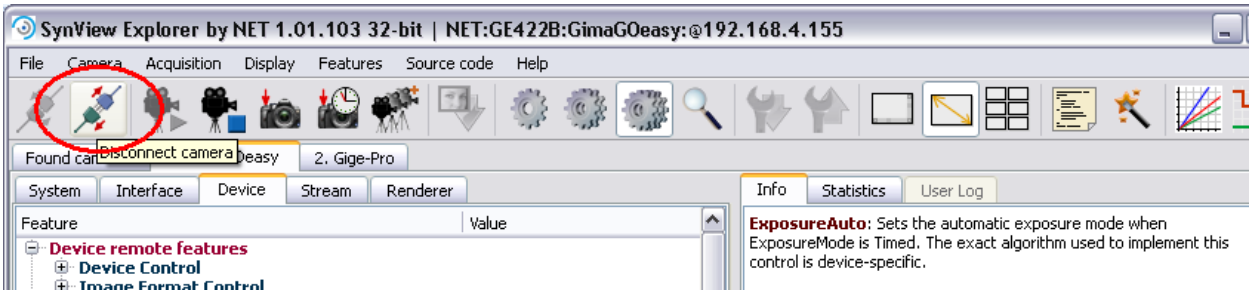


Figure 20: SynView Explorer: Disconnecting the camera

Features

Having the camera open, its feature tree is displayed, allowing configuring the camera. Actually, you will see 5 feature trees: from System, Interface, Device, Stream and Renderer. The role of these trees is explained in detail in the SynView programmer's guide, for the first steps use only the Device tab with the tree, containing the most important features.

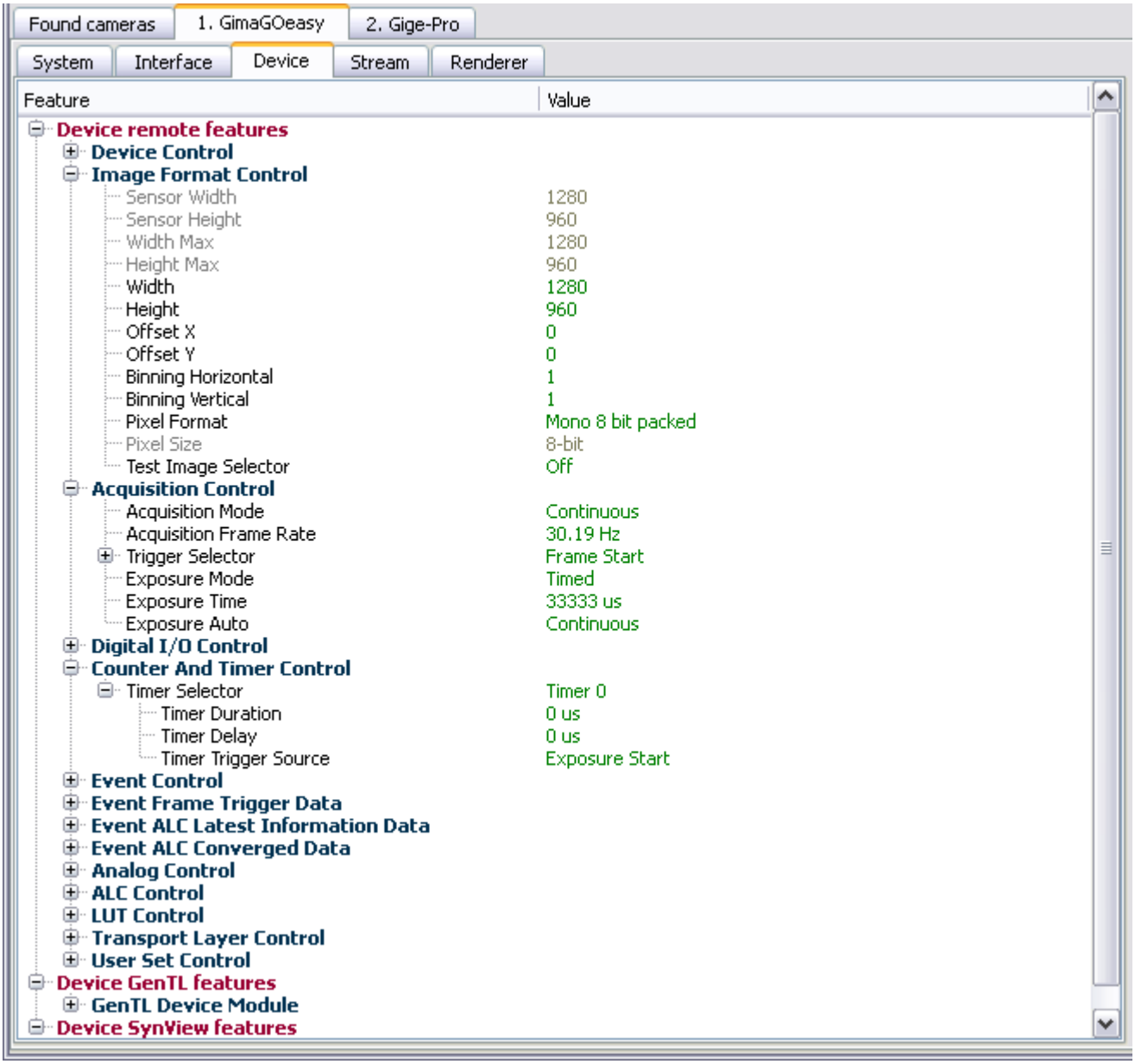


Figure 21: SynView Explorer: Device feature tree

When you browse the features in the feature tree, you can see that some of them are grayed out — these are the read-only features. Other features can be modified — these are read-write features.

This status need not be permanent, for example some features become read-only during the acquisition running (like the Width and Height device features).
You can also notice that features are of several types:

- a Boolean value — represented by a check box (for example Chunk Mode Active)
- an integer number — represented by a compound control with edit box, slider, inc/dec buttons (for example image Width and Height)
- a float number — also represented by a compound control with edit box, slider, inc/dec buttons (for example Exposure Time)
- an enumeration — represented by a combo box with available choices (for example Pixel Format)
- a string — represented by an edit box (for example Device User ID)
- a command — represented by a button for command execution (for example User Set Load)

Beginner-Expert-Guru level

Features are classified to 3 levels: Beginner, Expert and Guru. While the Beginner level (displayed as default) contains only a small subset of the most commonly used (and well understandable) features, the Guru level on the other hand includes all available features. In the menu or on the toolbar, you can switch to the Expert or Guru level, where a larger subset or all features are displayed.

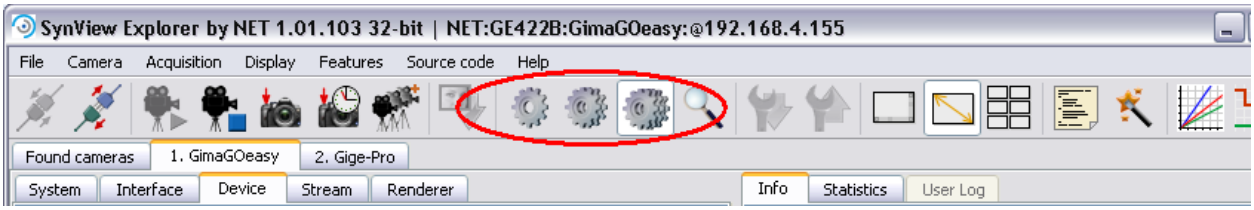


Figure 22: SynView Explorer: Feature levels

As you can see, in the Guru level the device provides quite a high number of features. We recommend using the Beginner level features only for initial experiments and switch to the Expert level, when doing more advanced tests. The Guru level should be avoided; since features in this level should not be usually touched during regular operation and altering them can lead to unexpected results.

Help for a feature

In a set of hundreds of features it may not be easy to guess the purpose of each feature. For this reason, the device provides also a description of each feature. In the Info panel Explorer shows the description for each selected feature:

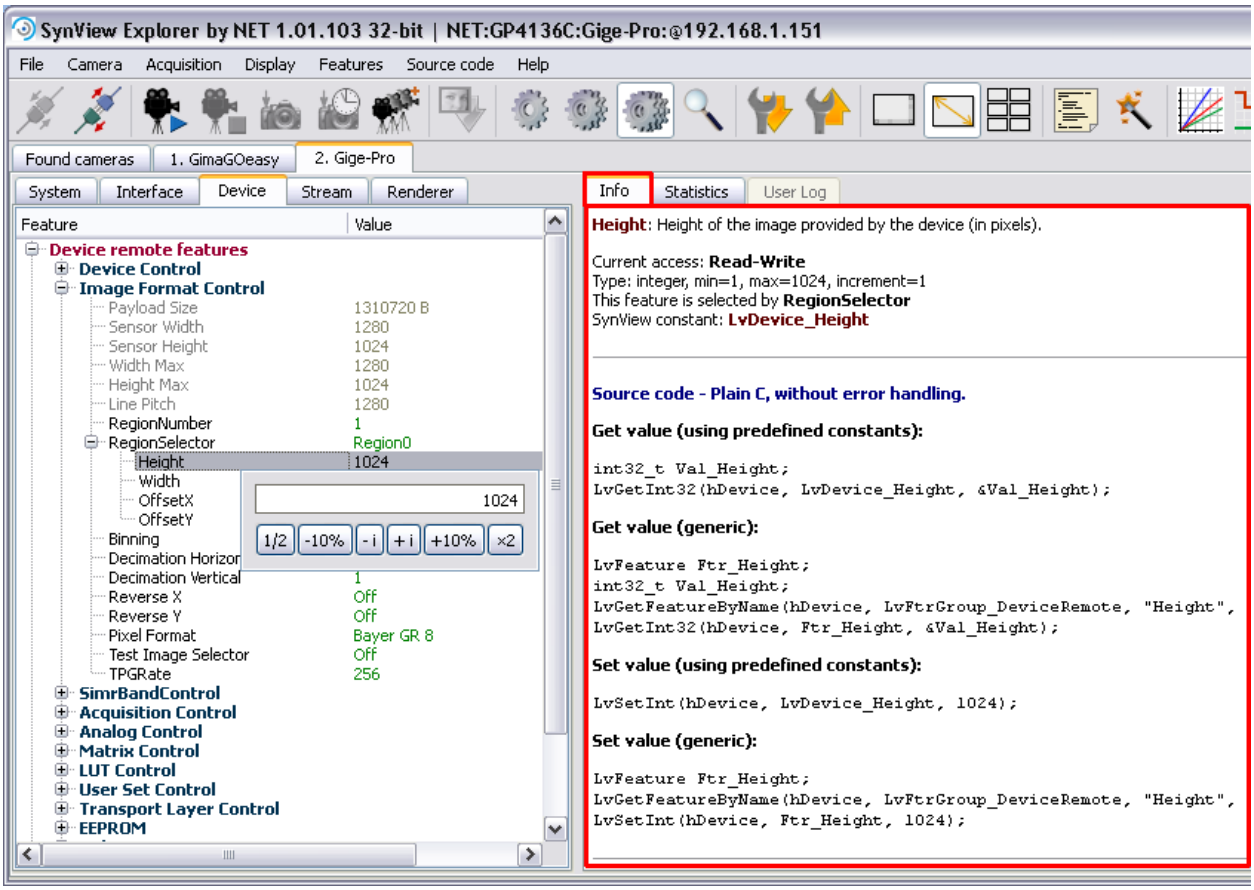


Figure 23: SynView Explorer: Info panel

Configuring the device

Typically, you first configure the device parameters and then start the acquisition. In the tree, you can configure any read-write feature, simply by clicking on it — a control window pops up, enabling you to change the feature. Note that the availability of one feature may depend on another —for example when the Image Width is set to Max Width, then the X Offset is not writable, because the Width + OffsetX must be less or equal to Max Width, so there is no space to increase the X Offset. However, when you make the Width smaller, you will see that the OffsetX becomes writable and that its maximum value follows the Max Width – Width formula. Furthermore, the availability of the offsets can be dependent on current ROI Mode. In some cases the dependencies may not be understandable at the first sight, but you can usually discover them fast by playing around with the features.

The tree of camera features follows the principles coming from the GenICam standard. Its subtleties are discussed in a dedicated "**Understanding the feature tree**" p.39.Getting familiar with that chapter will help you to effectively work with the camera.

Furthermore, the device configuration is also described in detail in the ***GigEPRO Manual*** and ***CorSight Manual***.

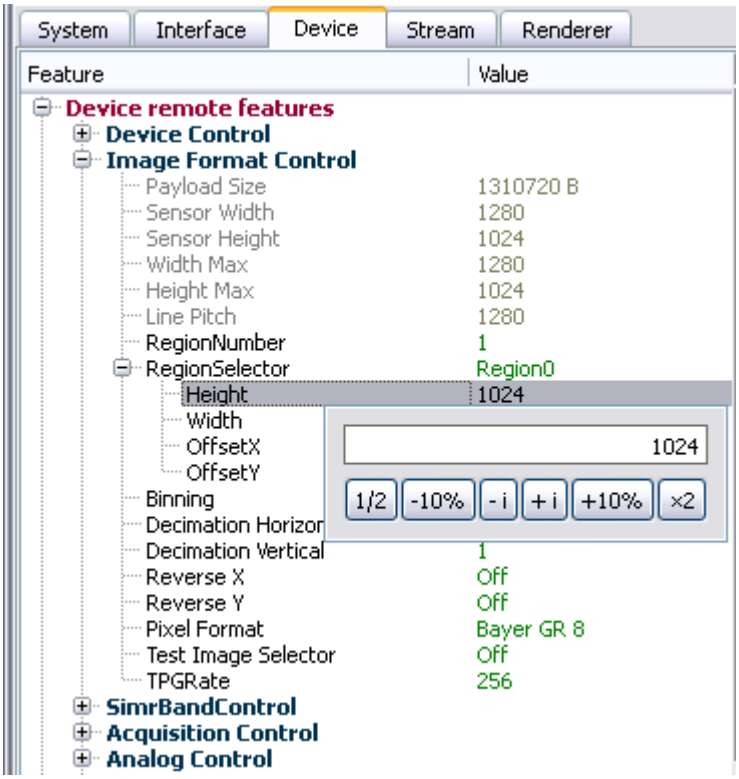


Figure 24: SynView Explorer: Setting a feature (Image Height)

Also note, that some features have so called selectors. For example the device can have several I/O ports and to configure a specific port, you should select it by the Line Selector and then the features in the tree under this selector are used for the selected port.

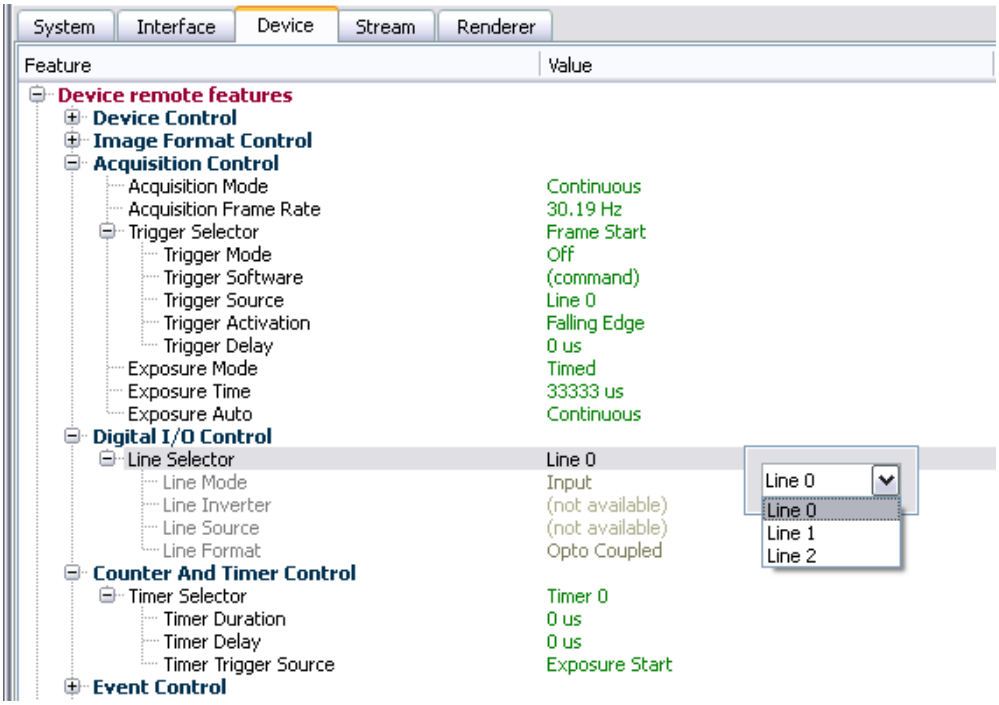


Figure 25: SynView Explorer: Using a selector (Line)

Finding a feature

In a complex feature tree it might be difficult to find desired feature. SynView Explorer offers a tool for finding a feature if you enter a part of its name or display name.

Figure 26: SynView Explorer: Find feature

During typing the feature name SynView Explorer updates a list of all features, in which the written substring occurs. By clicking on a feature in the list, the feature is focused in the tree.

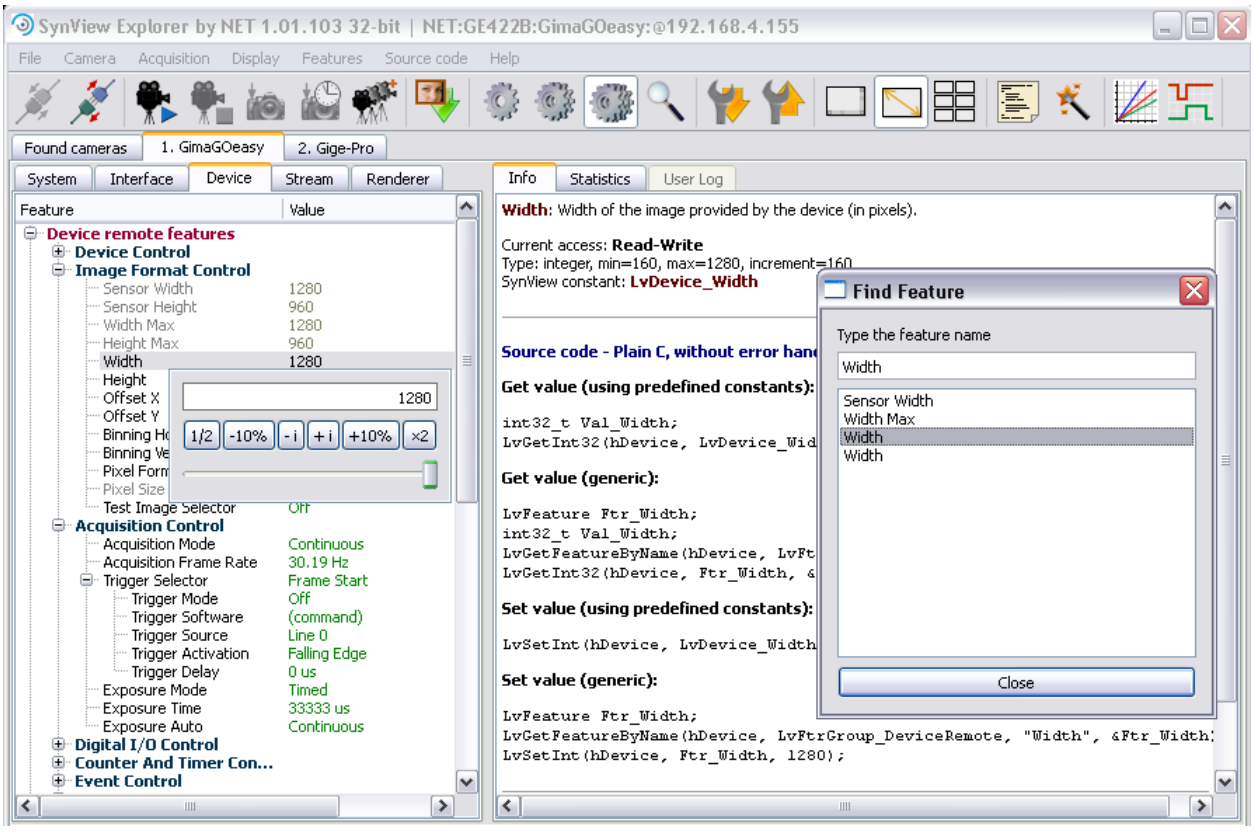


Figure 27: SynView Explorer: Find Feature dialog

Saving and restoring the configuration

It is possible to store full camera configuration (= all needed features) at any time to a file. That configuration can be then later reloaded either again in the SynView Explorer or in the user application (the SynView API provides a function to load settings). Using this functionality, the programmer does not need to care about the possible feature dependencies, SynView API handles that correctly. Furthermore, the stored configuration can be used for multiple connected cameras, provided that they are of the same type and firmware version.

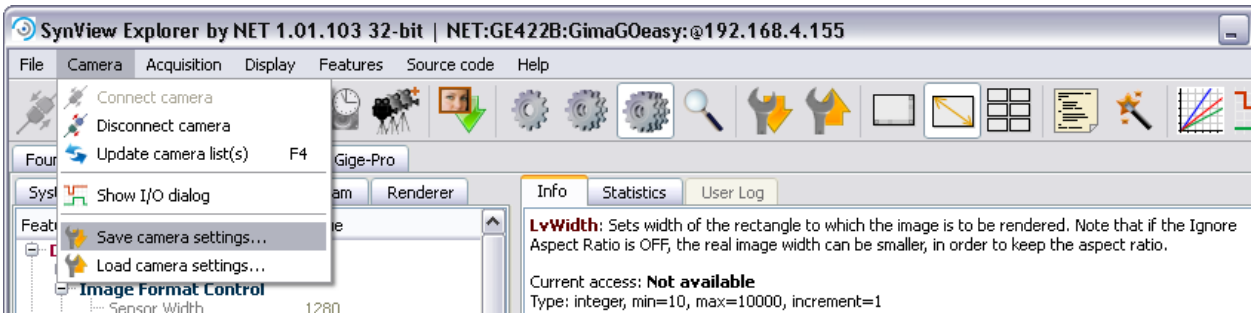


Figure 28: SynView Explorer: Save camera settings

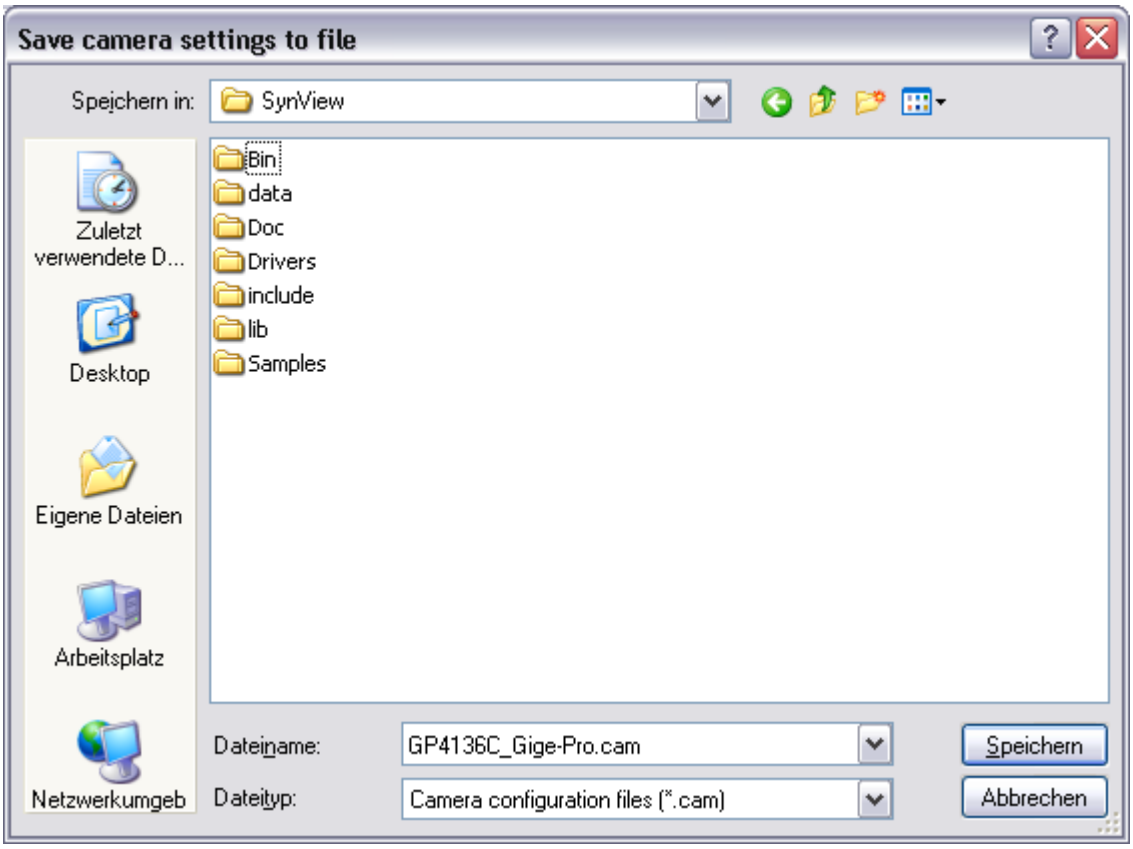


Figure 29: SynView Explorer: Save camera settings dialog

Start and stop the acquisition

Having the camera configured, you can start the acquisition using the Start acquisition button. If the acquisition does not start as expected, review again the camera feature settings. For GigE Vision cameras, once again, the network configuration and firewall settings are to be rechecked.

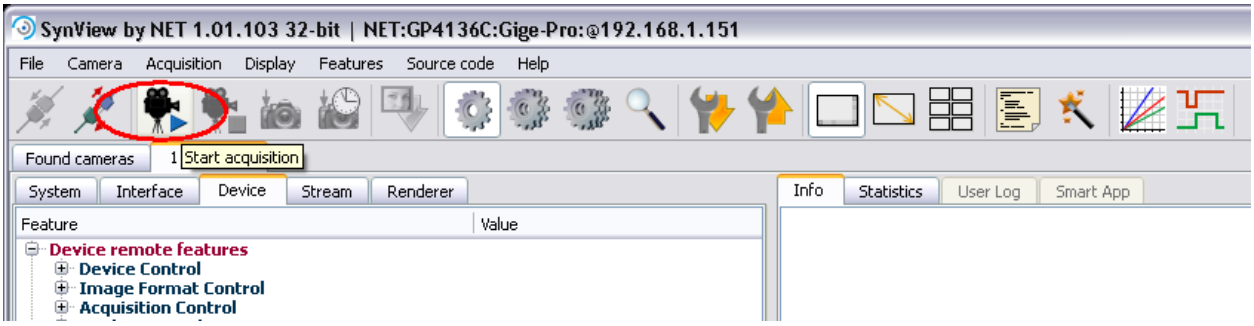


Figure 30: SynView Explorer: Start acquisition

In case you have configured the camera for triggered acquisition and the Software Trigger feature becomes available, you can trigger the camera by the button in the toolbar:

Figure 31: SynView Explorer: Trigger the camera

Note that some features become read-only when you start the acquisition, for example you cannot change the image size during the acquisition. And vice-versa: other features may become usable only after you start the acquisition — for example the Trigger Software command, if you use the triggered mode.

The acquisition can be stopped through the Stop acquisition button.

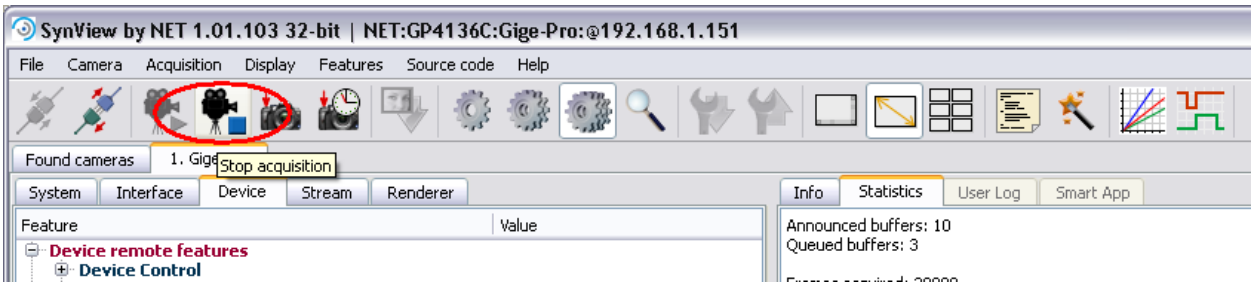


Figure 32: SynView Explorer: Stop acquisition

Display of images

The way how SynView Explorer displays the acquired images can be configured — the available options are full size (if the image does not fit in the display area, scroll bars are added), scale to fit (image is scaled to fit to the display area) and tiled display (SynView Explorer displays series of consecutive images in tiles). Note that the scale to fit and tile modes might not be available in Linux.
To be configured through the Display menu or corresponding buttons.

Figure 33: SynView Explorer: Display modes

Automatic image preprocessing

SynView Explorer makes automatic image preprocessing. This includes namely the Bayer decoding for color cameras, applying the LUT (Lookup Table), to which can be added parameters like white balance, gamma, brightness and contrast, and the color correction, by which can be adjusted the saturation. SynView automatically determines, if the required functionality is available in camera hardware, and if not, does the necessary operations by software (this increases the CPU load).
The automatic processing is by default switched on. You can switch it off in menu:

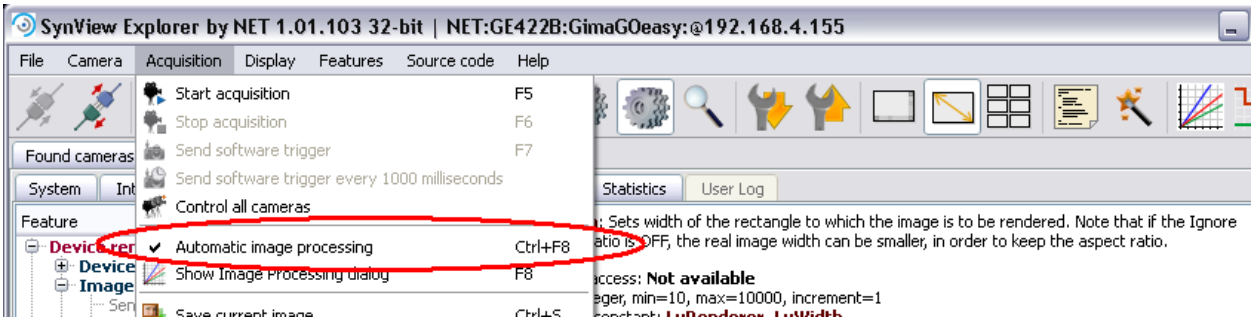


Figure 34: SynView Explorer: Automatic image processing

You can also display a dialog, where you can adjust the processing parameters:

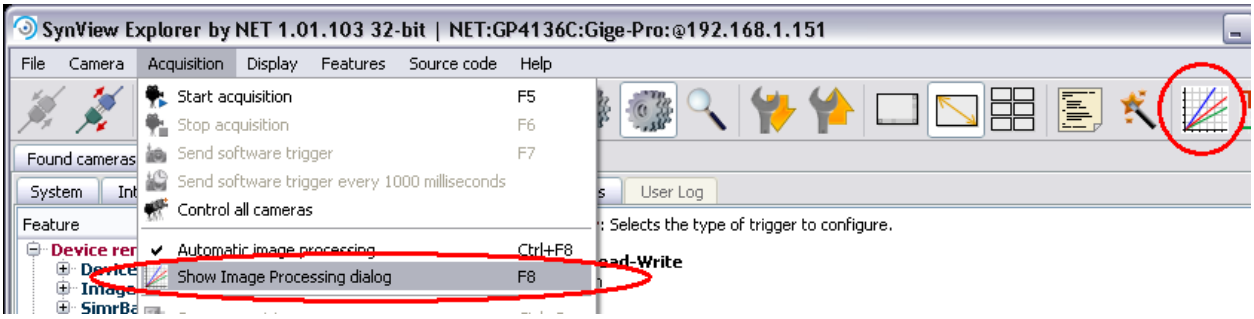


Figure 35: SynView Explorer: Show image processing dialog

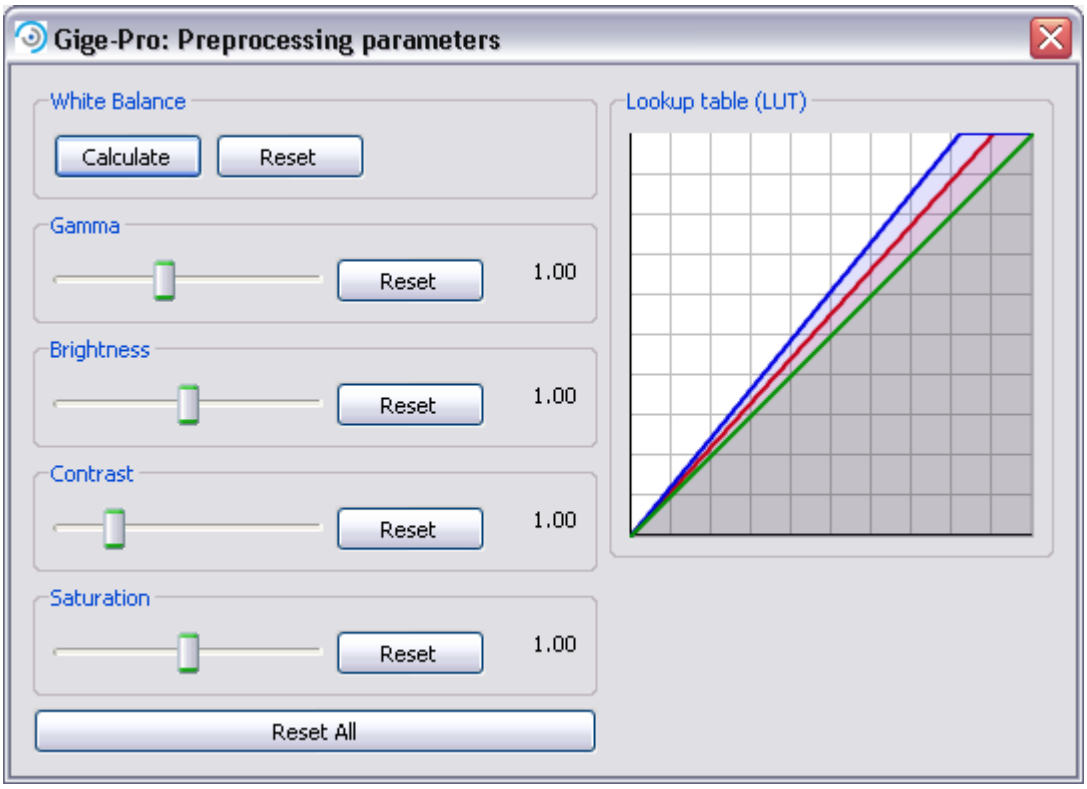


Figure 36: SynView Explorer: Image processing dialog

Note that the white balance factors are always calculated from the next acquired image, so after pressing Calculate button you will not see a change until a new image is acquired.

Saving acquired image

When you stop the acquisition, you can save the last acquired image, to a BMP, TIFF or JPEG file. If you have the automatic processing switched on, the processed image is saved, and otherwise the original image is saved. Before saving, the image is converted to suitable pixel format, for example when you save an image with a 12–bit mono pixel format to a BMP or JPEG file, it is automatically converted to 8–bit mono pixel format; to a TIFF file it is converted to 16–bit mono pixel format.

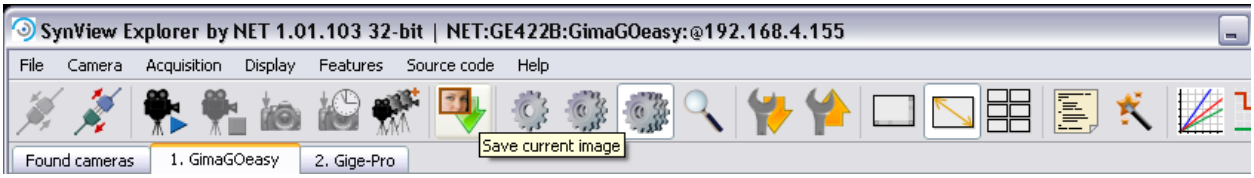


Figure 37: SynView Explorer: Save image

Select the image format in the Save image dialog:

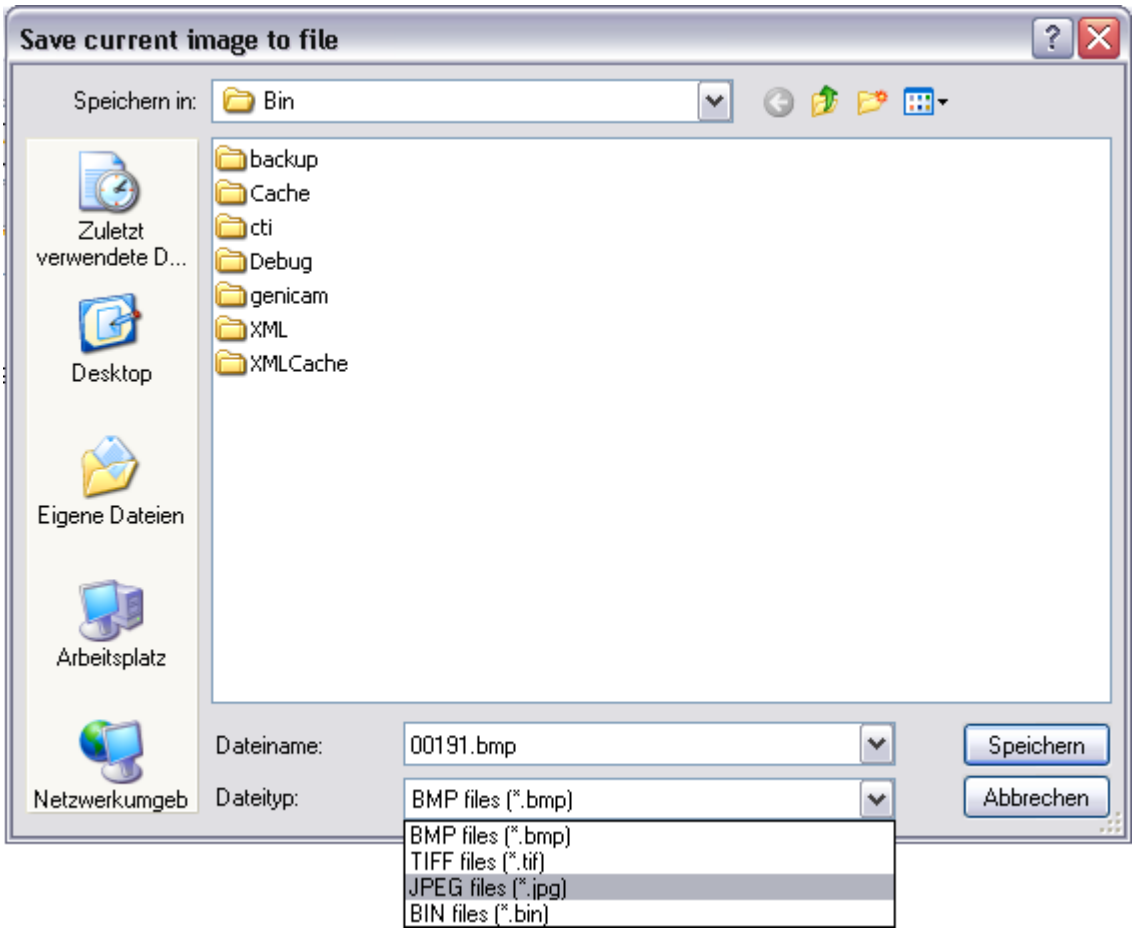


Figure 38: SynView Explorer: Save image dialog

Settings

SynView Explorer settings are available in the menu:

Figure 39: SynView Explorer: Settings

The following settings are available:

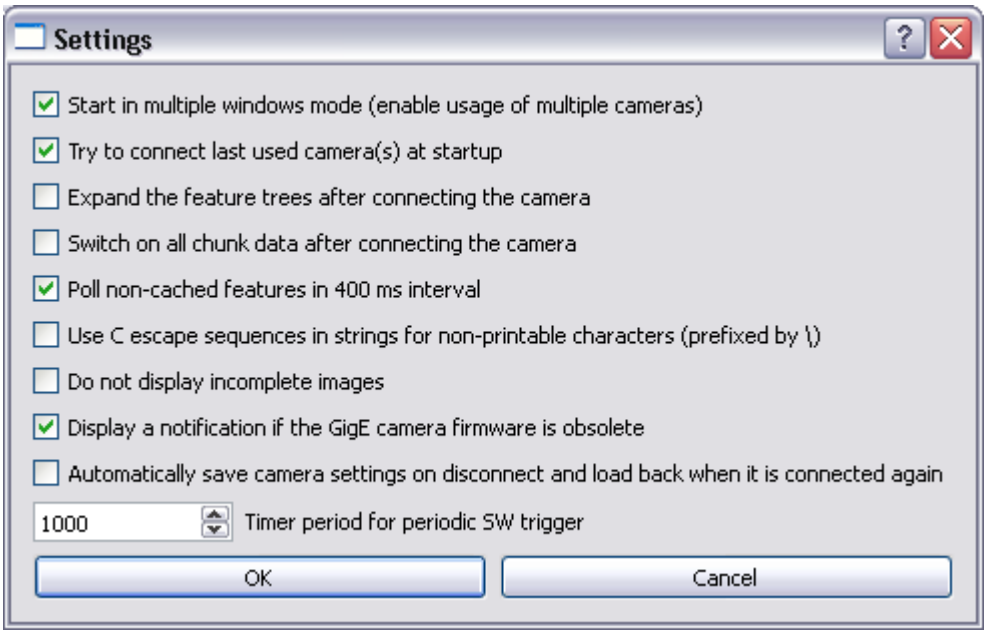


Figure 40: SynView Explorer: Settings dialog

- Start in multiple windows mode (enable usage of multiple cameras). If ON, SynView Explorer starts in multiple window mode enabling concurrent usage of multiple cameras. If OFF, it starts in a compact, single window mode, enabling to connect one camera only. This settings applies to the next SynView Explorer start (does not have an immediate effect).
- Try to connect last used camera at startup. If you close SynView Explorer without explicitly disconnecting the camera and this option is ON, the Explorer will attempt to automatically connect the same camera next time it is run (if the camera is still available).
- Expand the feature trees after connecting the camera. By default the trees are displayed as collapsed, that means only the top level items are displayed. When you set this option to ON, all the trees are fully expanded when the camera is connected.
- Switch on all chunk data after connecting the camera. If the camera delivers chunk data with each image and this option is ON, the Explorer automatically switches on all available chunk data features. If you do not use this option, you can always switch on selected chunk data features manually. Some of the chunk data are displayed automatically in a line above the image.

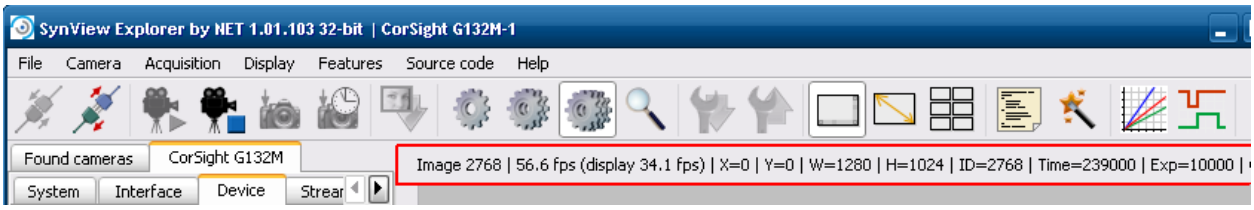


Figure 41: SynView Explorer: Chunk data

- Poll non-cached features in 400 ms interval. If set to ON, the Explorer polls the non-cached features (such as Device Temperature or Up Time) with the minimum period of 400 ms and updates their values in the feature tree.
- Use C escape sequences in strings for non-printable characters. In case some of the string features need to use a non-printable character (for example strings sent via RS-232 port), you can switch this option ON to be able to write and see such characters. Then a C language escape sequence syntax is used for non-printable characters. The sequence begins with a backslash followed by a single letter (r = Carriage Return, n = Line Feed, etc.), or a number in form of x + 2 hexadecimal digits or 3 decimal digits. The number expresses the character ordinal number. A backslash itself must be written as double backslashes.

```
\ + letter (b, t, n, v, f, r)
\ + xNN hexadecimal number
\ + NNN decimal number

\\ = \x5C = \092 (backslash)
\b = \x08 = \008 (backspace)
\t = \x09 = \009 (tab)
\n = \x0A = \010 (line feed)
\v = \x0B = \011 (vertical tab)
\f = \x0C = \012 (form feed)
\r = \x0D = \013 (carriage return)
```

Source code generator

SynView Explorer also provides a powerful set of tools for generating the source code. The detailed information about the developer assistive tools with guidelines how to use them during development is provided in the SynView source code generator chapter in the ***SynView programmer's guide***.

Important principles behind SynView interface

Understanding the feature tree

The “feature tree” is one of the most important patterns used within SynView. It is a tree of interdependent features used to configure individual hardware and SynView components, with a GenICam GenApi mechanism running under the hood. It is important to understand different aspects of the feature tree functionality to be able to configure the system effectively and reliably.

Feature tree instances

The feature tree paradigm is reused to configure the devices and various software components. It is important to understand that the set of supported features can vary significantly among different hardware (camera) models or even among different revisions of the same model. SynView Explorer displays all available feature tree instances, allowing configuring the entire system intuitively. The guide how to control individual features programmatically is provided in SynView programmer's guide.

Remote device features

Set of features used to configure every single device connected to the system. This feature tree is designed by the device manufacturer (by means of the GenICam standard) and SynView just exposes it to the user. The “device” is typically a camera, but it can be any configurable device connected to the system.

GenTL Producer features

Features configuring the GenTL Producer, which can be either directly SynView GenTL Producer or a 3rd party GenTL Producer controlling cameras of another vendor. There is one feature tree available for each distinct component describing the acquisition system:

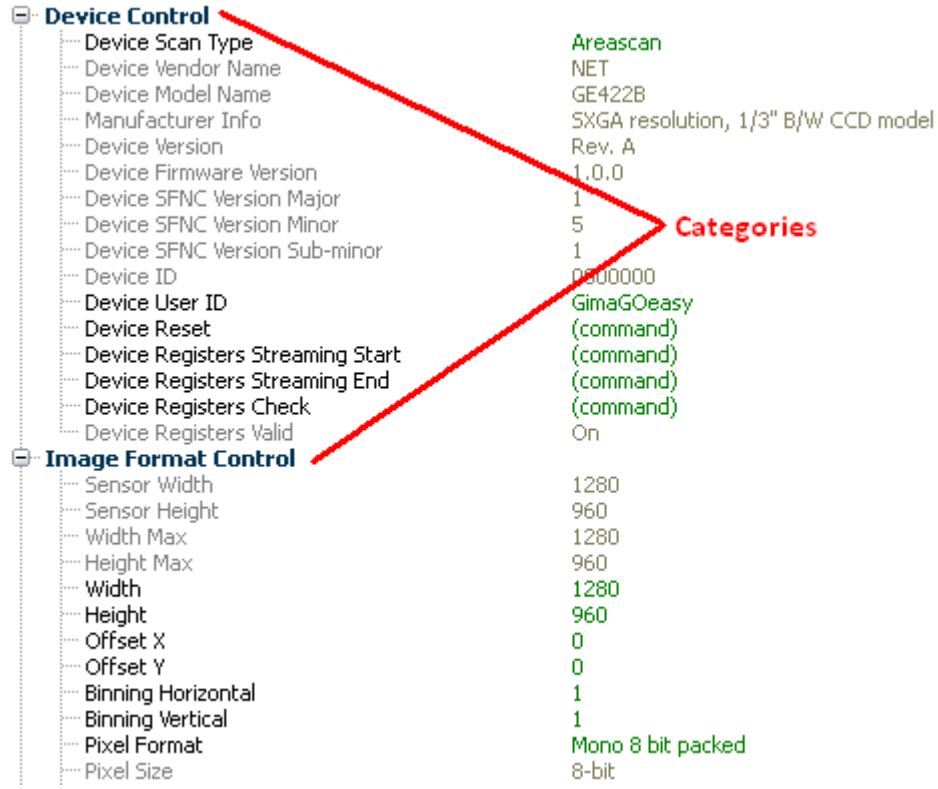
- System. Represents a GenTL Producer library. In most cases, there will be just a single system module available — representing the SynView GenTL Producer.
- Interface. Describes a physical interface used to connect cameras, for example a network segment interfacing GigE Vision cameras or a bus interfacing the CorSight camera modules.
- Device. Software representation of a device connected to the system.
- Data stream. Controls the stream of the image data coming from the device. In most cases there will be just a single data stream per device, but there might be devices featuring multiple data streams.
- Buffer. Allows to query information about individual image buffers.

SynView API features

On top of the configuration options provided by the camera itself (and eventually the supporting GenTL Producer library), SynView API itself also adds a rich set of features, which are, for the sake of consistency, also controlled through a feature tree. Again, SynView would provide a separate feature tree per component.

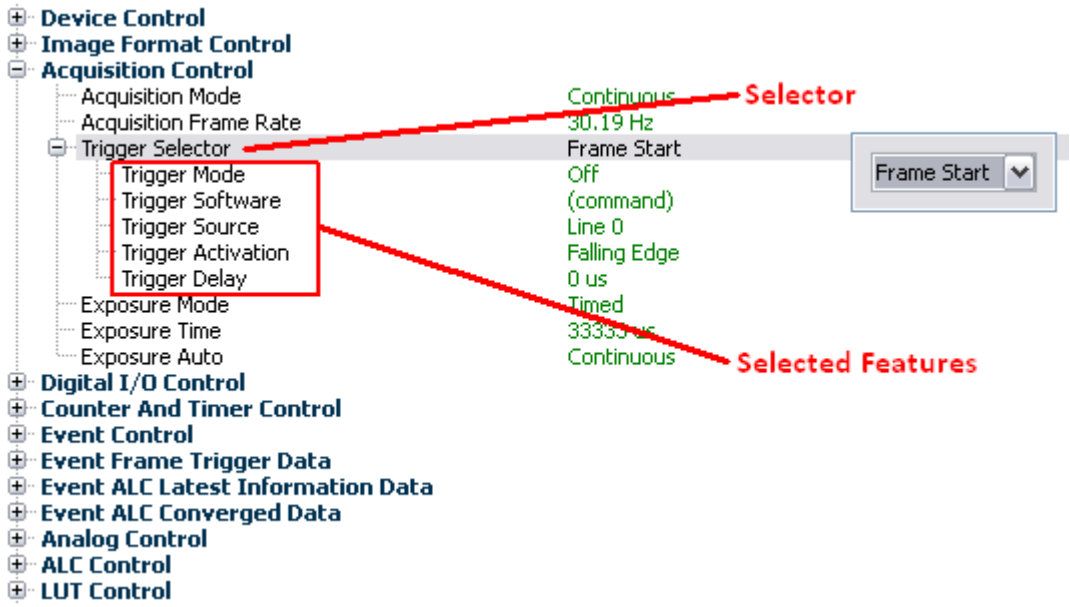
Categories

The features are sorted into categories, grouping together related parameters. The only purpose of categories is to improve the visual representation of a complex feature tree. From programmer's perspective (access of individual features), the categories are not important.



Selectors

Selectors are features controlling sets or arrays of identical features.
A selector can be viewed as an analogy of an array index. Similarly as changing an array index does not by itself modify any value stored in the array, changing a selector does never modify the selected features — it does not alter the actual camera status.
For example, switching the LineSelector value does not at all change configuration of the I/O lines. It just selects the line to be configured through the “selected” features, such as LineSource or LineInverter.

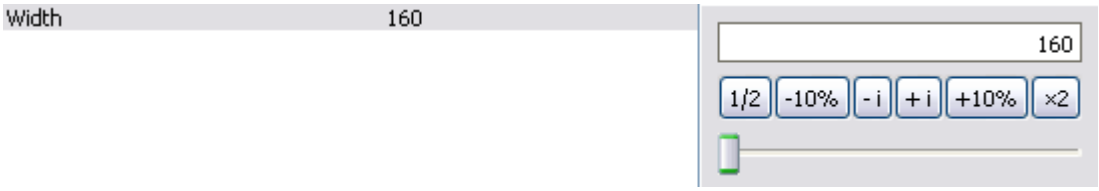


Feature interface types

Depending on the purpose of respective features, they can be represented through different inter faces.

Integer

Up to 64-bit integer value, which can have assigned min/max values and an increment (or a set of allowed values). The feature values must respect these constraints — if the application tries to set an invalid value, SynView API readjusts it to the closest valid one.



Float

Floating point number with optional min/max (or set of allowed values). For visualization purposes (slider in a GUI representation), the floating point features can also have an increment, but unlike with integers, setting the values aligned with the increment is not forced. The camera can, however, realign the value internally to a closest valid discrete value — the application can re-read the feature after setting to know the actually used exact value.

Enumeration

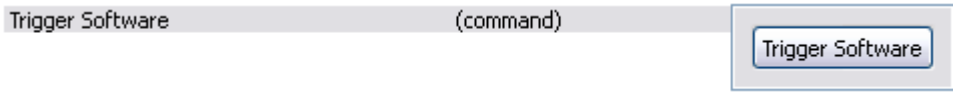
Feature allowing selecting from given set of entries. Some of the entries might have self-clearing behavior — they reset automatically to another entry. This behavior is similar to a command feature.

Boolean

Simple flags that can have only two values, true or false.

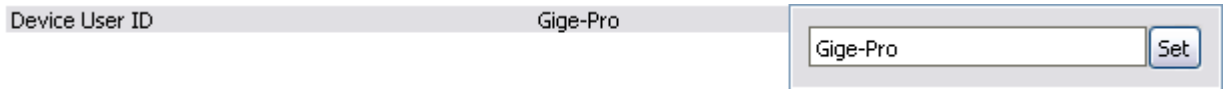
Command

Purpose of the command features is to init various actions (start acquisition, calibration, automatic white balance etc.). The command feature provides feedback when the action is finished.



String

String (character) based features.



Register

Memory blob with unspecified representation. Can be used for example to load/save the entire LUT in a single feature access.

Feature properties

The features have also additional properties, some of them dynamic, which can change during operation, others static (remain fixed during runtime). SynView API provides access to all these properties.

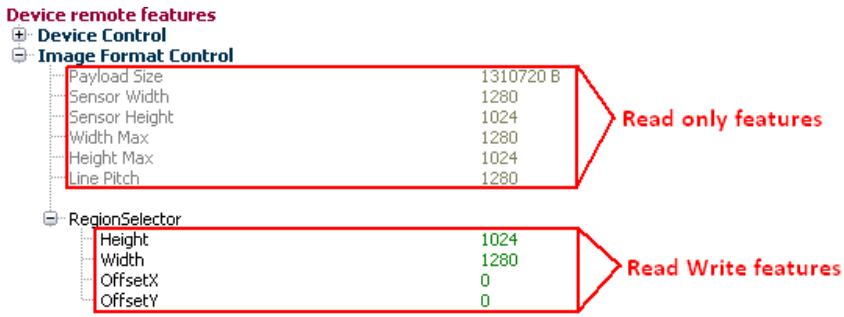
4.1.5.1. Dynamic properties

Access mode

The access mode specifies, whether the feature is available for reading and/or writing. It is important to know that the access mode can change during runtime, eg. based on other features. For example if TriggerMode is switched off, the depending features, such as TriggerSource might become unavailable. Note that in a GUI representation (eg. in the SynView Explorer), the temporarily unavailable or locked (read-only) features might be rendered for example as grayed or even disappear. Possible access modes are:

- Not implemented. The feature is not implemented. It will stay in this status during the entire operation. Features with this access mode will not be displayed at all in SynView Explorer GUI.

- Not available. The feature is temporarily unavailable (cannot be read nor written). It can become available later, typically based on other features in the same feature tree.
- Read only. The feature value is available for reading but not for writing.
- Write only. The feature value is available for writing but not for reading.
- Read-write. The feature value can be both read and written.



Caching

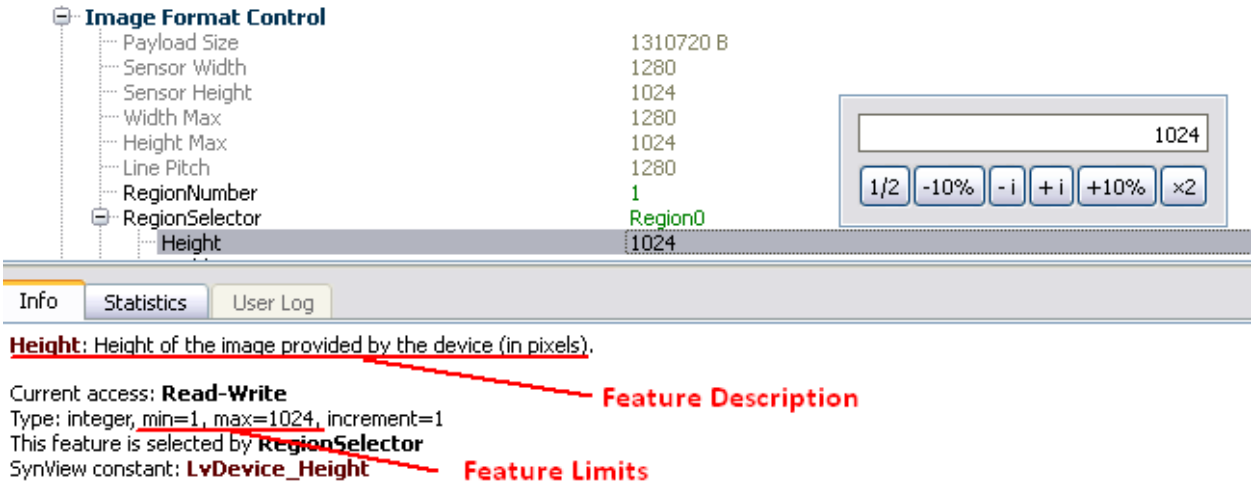
The feature values are cached internally, which is important especially when working with remote devices, such as GigE Vision cameras, so that the same value does not need to be re-read always over the network. The feature tree knows when it can use the case for every feature and when the cache should be invalidated. The application can rely fully on this mechanism without need to maintain own feature cache — SynView will communicate with the remote device only when necessary, thus optimizing the performance.

Furthermore, it is useful to distinguish between three types of features from the caching point of view:

- Write-through caching. Cached during write operation. This is the case for most features; the exact feature value is used.
- Write-around caching. Cached during read operation. Used typically for certain float features of the remote device, when the camera needs to readjust the value to a closest valid one. If the application needs to know the exact used value, it has to re-read the feature value immediately after writing.
- No caching. The value is not cached at all. Used for volatile features such as sensor temperature or similar. The application has to start SynView API polling thread to get the non-cached features updated regularly. Some features might switch to non-cached mode temporarily, for example the gain value will become non-cached when automatic gain mode is active.

Limits

Integer and float features have minimum, maximum and an increment (the increment is optional for floats). The feature value must fall within these limits. Remember that the limits can change during runtime, depending on other features.



4.1.5.2. Static properties

Unit

String representation of the physical unit represented by the value.

Display format

Provides hint for graphical representation of the feature value, such as whether the feature has linear or logarithmic behavior, if it should be displayed in decimal or hex form, etc.

Info texts

Each feature gets a display name (human readable name of the feature), tooltip (short info) and a description (longer info). These are useful again especially for display and user interaction.

Feature dependencies

The feature tree knows about possibly complex dependencies between individual features. Changing value of one feature can affect other feature(s) in following ways:

- Change other feature's value — for a feature which is a direct function of the first one.
- Change other feature's limits — eg. increasing a horizontal offset for the image' area of interest decreases maximum of the image width.
- Change other feature's access mode — eg. when disabling a trigger, individual trigger parameters might become unavailable, when starting acquisition, basic acquisition parameters might become locked.
- Change other feature's effective caching mode — eg. gain value is cached for the “manual” mode, but non-cached for the “automatic” gain mode.
- Invalidate the other feature — disabling its cache, for example when resetting the camera to a default status, all feature values must be re-read from the camera rather than from the now-invalid cache.

To understand the feature tree behavior and treat it properly, it is essential to remember about the possible feature relationships among certain features. The feature tree handles everything for you seamlessly, but you still need to keep in mind that feature status can change based on other features and

that for some feature groups the order of writing the features might be significant. To simplify work with such features, SynView API provides helper functions allowing setting the entire feature group in a single call or even store/reload entire camera configuration.

Chunk data based features

So called “chunk data” are additional data that a camera can deliver to the buffer together with an image. These data can contain information such as image timestamp, frame ID or acquisition parameters used to capture given image. These data can are also part of the feature tree — but they are not read by querying the camera's current status, but rather directly from the buffer itself. The feature tree internal logic knows how to access these additional data in the buffer. SynView API provides functions allowing specifying the “active” buffer that should be connected to the feature tree.

Event based features

Besides the image and chunk data, the remote device (camera) can also fire asynchronous events. These are useful for variety of purposes, including reporting of asynchronous error and status info, sending log messages and more. The event itself usually caries also additional data describing the event. These data are, yet again, integrated into the feature tree. SynView API provides notifications about each received event, so that the application can track them and possibly query the data associated with every event.

Feature update notifications in general

The application can instruct the SynView API to provide notification whenever the status of a given feature changes. The notification would be delivered when feature value has changed, as well as for other status updates (new access mode, feature cache invalidated, etc.).

Installation details

This chapter provides detailed overview of SynView installation process for all supported operating systems. For an express Windows based installation (when details are not important for you), please jump to "**Quick start**" p.7.

Before installation

It is important to know, that the system might not be immediately ready to use after the installation. The hardware (cameras) might need some basic configuration, as well as the system itself — eg. when using the GigEPRO cameras, the network and firewall must be properly configured. The details are always listed in the hardware manual of each respective hardware type. The basics are summarized in Section "**Connecting and configuring the camera(s)**" p.7.

Installation in Windows

System requirements

The SynView for Windows does not have any particular system requirements; the installer will install all the necessary dependencies, including the .NET Framework runtime. SynView is compatible with all "recent" Windows versions, including Windows 7 or Windows XP. Note that Windows versions older than Windows XP are not supported.

Note: The SynView package installation should be performed by a user with administrator rights.

Installation media

The SynView package can be installed from two types of installation media:

- If you obtained the installation CD-ROM, insert it to the CD-ROM drive. The installation will start automatically.
- If you don't have the installation CD-ROM or if you wish to install a newer SynView version, you can download the installer from the download area of NET camera on the NET website after registration. To register click on "login: free download" next to the SynView Installer icon. The link will guide through a procedure which asks to select a username and password for registration and which will give acess to the SynView SDK download area.
- The general download for NET products link is

http://www.net-gmbh.com/en/download.html

The installer file name follows pattern SynViewxyyzzz.exe, where xyyzzz stands for SynView version. Execute the installer. Depending on security adjustments of your system and whether you are running

the installer from a local disk or a network share, you might get following security warning. Simply proceed and click Yes.

Figure 42: SynView installation: Windows security warning

5.2.3. Installation procedure

The paragraphs below discuss all the dialogs of SynView's Windows installer and corresponding options. Note that some details might slightly differ between individual SynView versions, but the principles will always be the same. The installer first displays a welcome screen with basic information about the software you are going to install. Click Next to proceed.

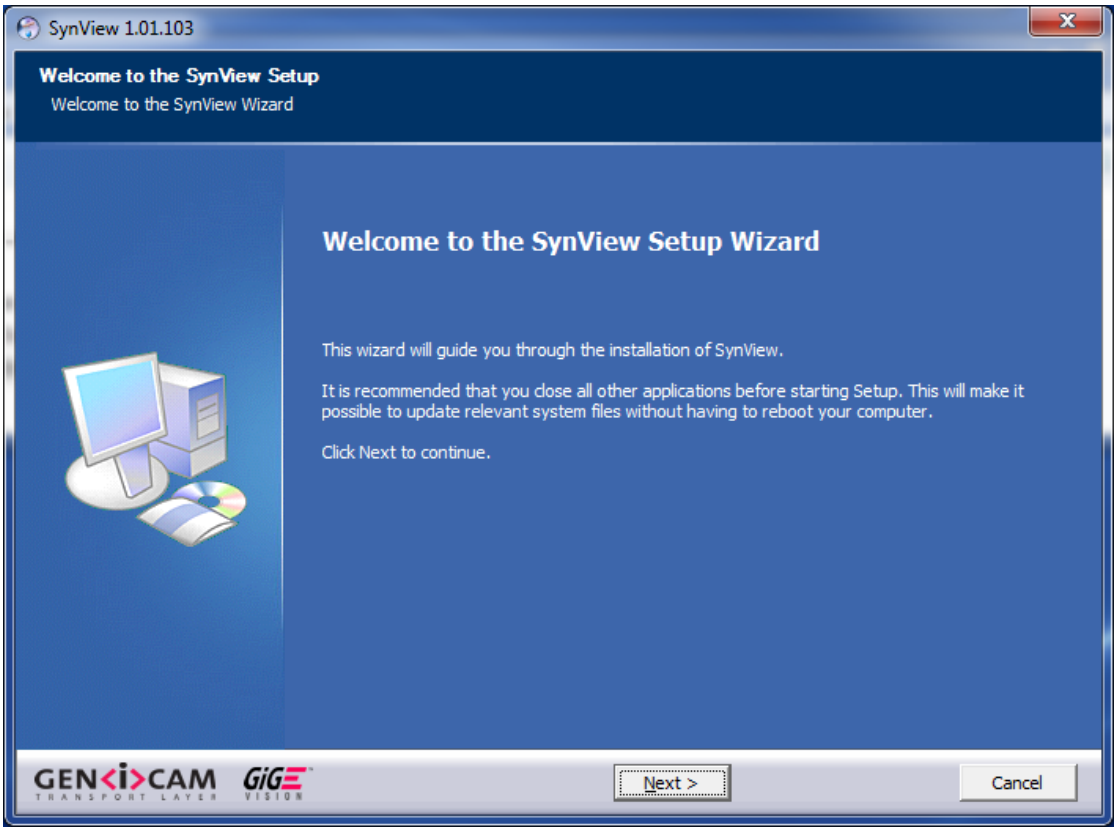


Figure 43: SynView installation: welcome screen

Next screen is the license agreement. Please read carefully the license text and if you agree, click the I Agree button to proceed with installation. Once confirmed, your full agreement with the license text is assumed.

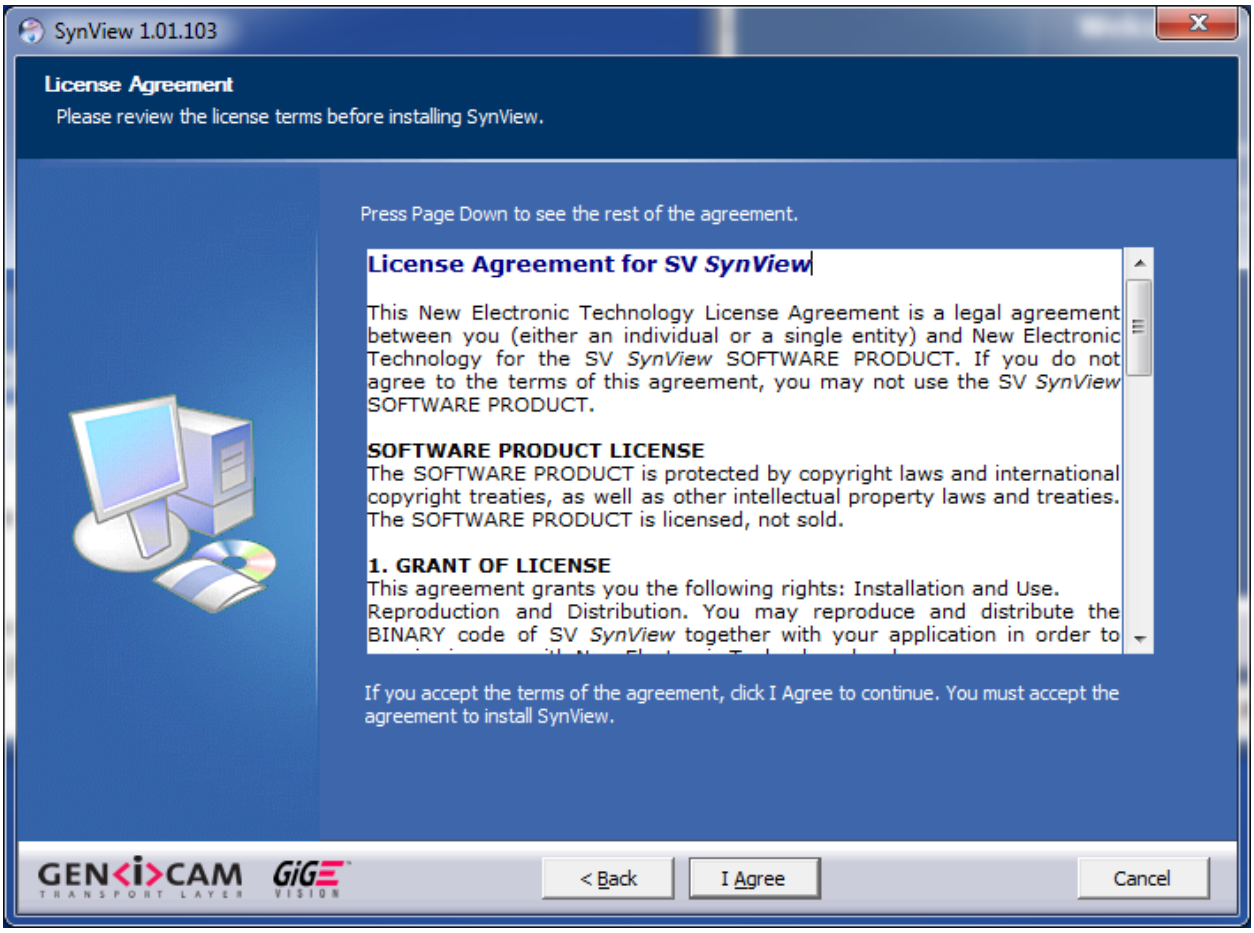


Figure 44: SynView installation: license agreement

Next the installer asks you about the location where to install SynView. The default location C:\Program Files\SynView should do well in most cases. If you prefer to install SynView in another directory, select it now. When finished, click the Next button.

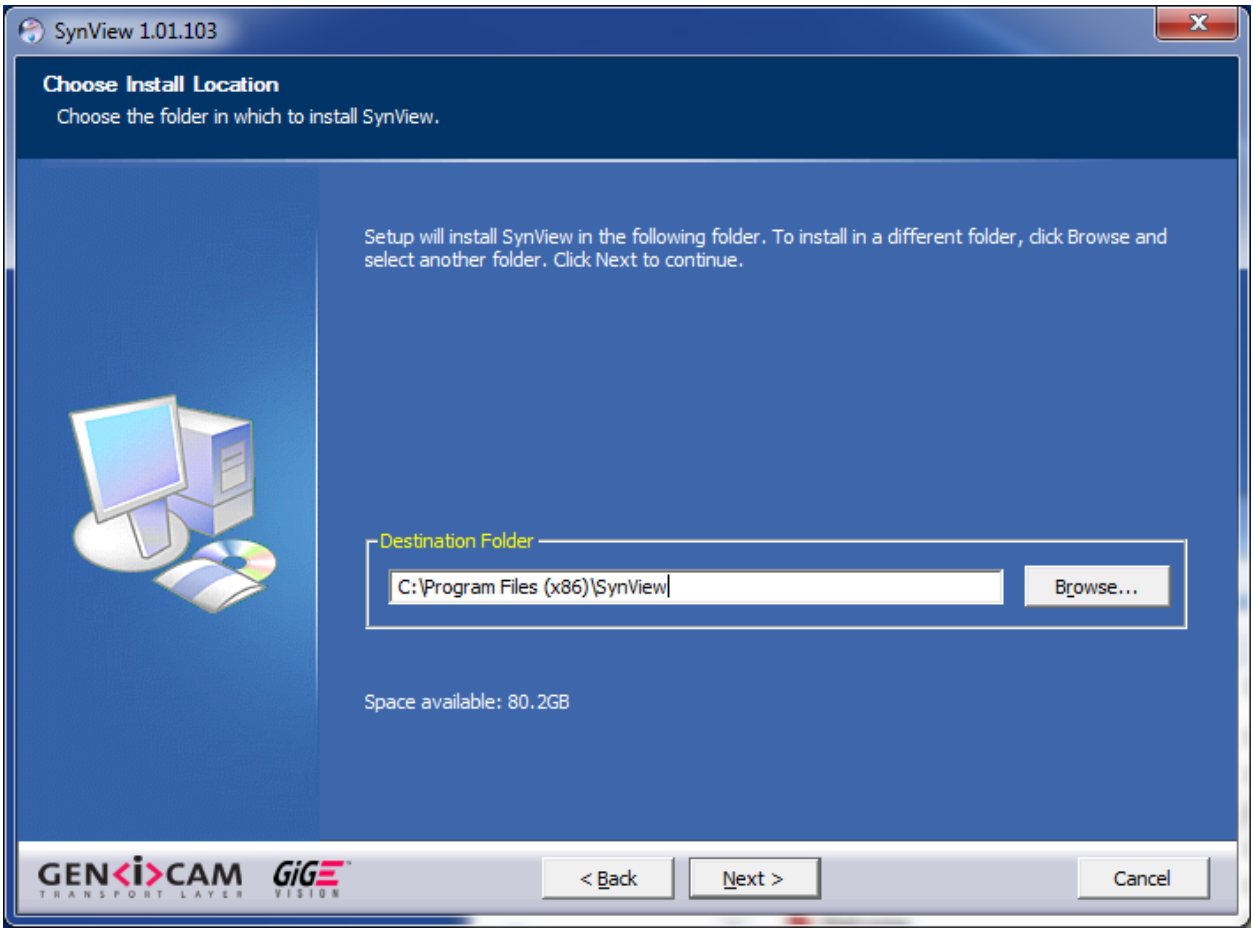


Figure 45: SynView installation: destination directory

If running Windows 7 or newer Windows version, the default software installation directory (C:\Program Files) is read only for regular applications. SynView, however, requires write access for certain files during operation (“**Installation layout**“ p.53). Therefore the installer will ask you where to put the application data requiring write access.

Note that in Windows versions older than Windows Vista (in particular in Windows XP) the C:\Program Files area is not protected and therefore all data can go to a single directory. Also, when installing to a non-standard location (outside C:\Program Files), the installer assumes the user chooses a writable area and installs everything to that directory. The dialog will be omitted in these cases.

You can choose from three options:

- C:\Users\Public\SynView: recommended when SynView will be used by multiple users.
- C:\Users\YourUserName\AppData\Local\SynView: recommended when SynView will be used by a single user (you).

User configured directory — if none of the options above fit your requirements, you can specify another directory. Be sure the directory has write access for all applications and is not protected by UAC.

Select the preferred option and click Next.

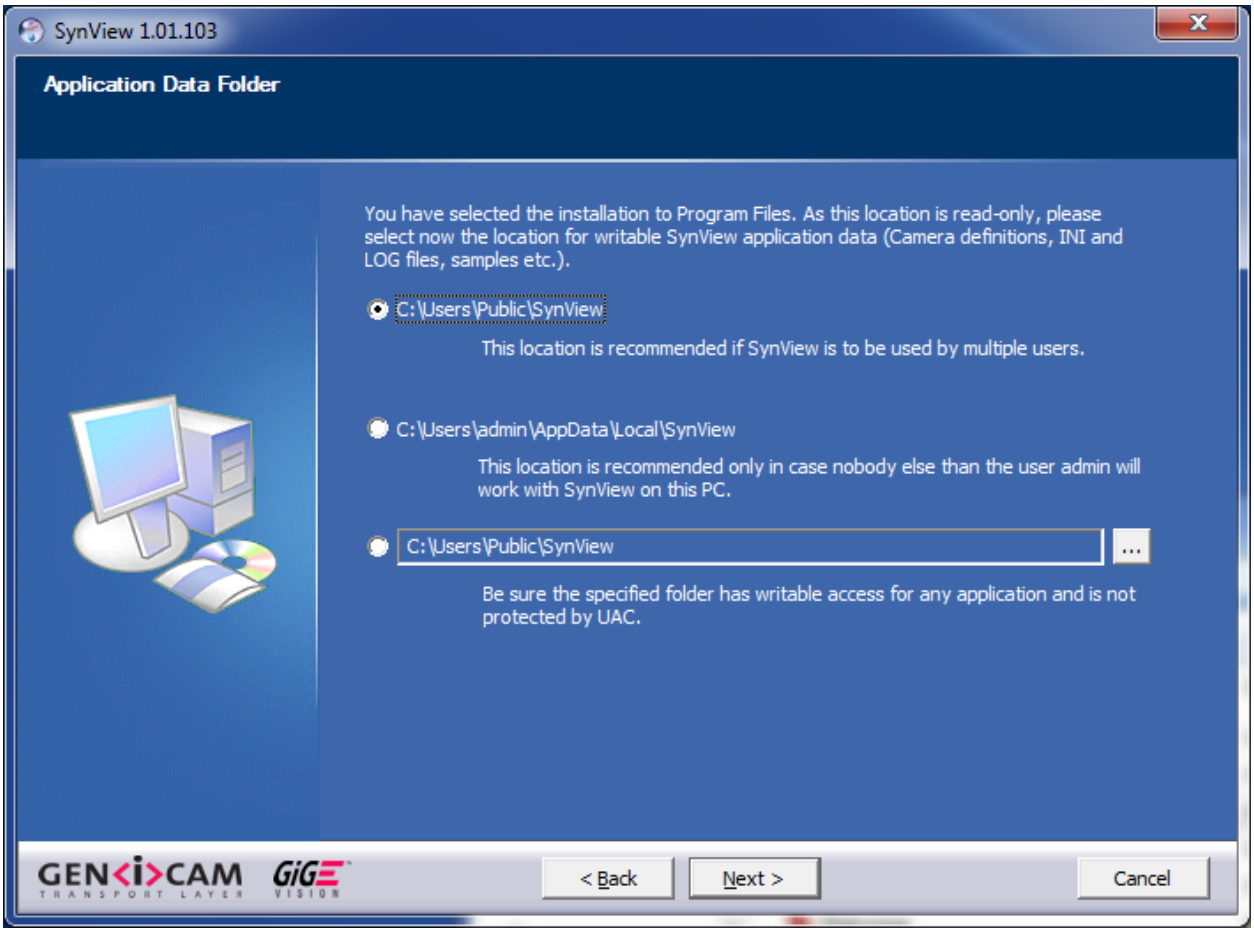


Figure 46: SynView installation: application data directory

In the next step you can select the SynView components to install. The set of actual options can slightly differ among individual SynView versions. The basic options are:

- Runtime files — files necessary to run every SynView application, these are always installed.
- Development files — essential files for compiling against SynView API as well as additional helper files. Files in this category are typically installed only on a development system, but not deployed with final runtime systems.
    - Include and library files — required for building SynView API applications.
    - Sample code
- Documentation — full documentation set for SynView package and all NET camera families.

Note that for testing the cameras, the "runtime" components suffice. However, for programming with SynView you need to make sure the Developer's tools are installed as well. When finished with the selection, click Install.

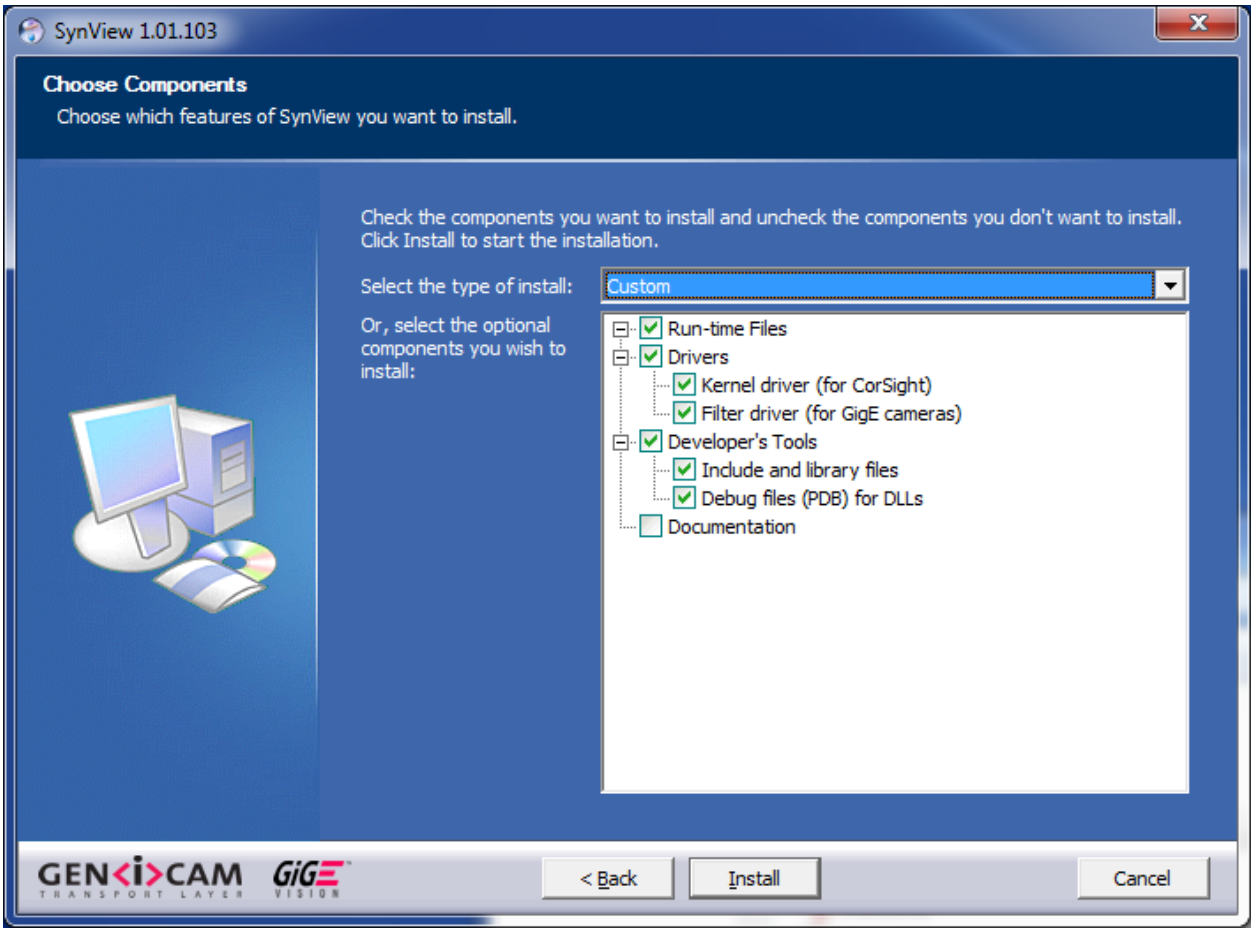


Figure 47: SynView installation: selecting components

After starting the installation, the selected components will be installed to your computer. The installer will inform you about the progress. Note that the installation includes Microsoft runtime libraries (Microsoft Visual C++ 2005 Redistributable) which sometimes take longer to be installed.

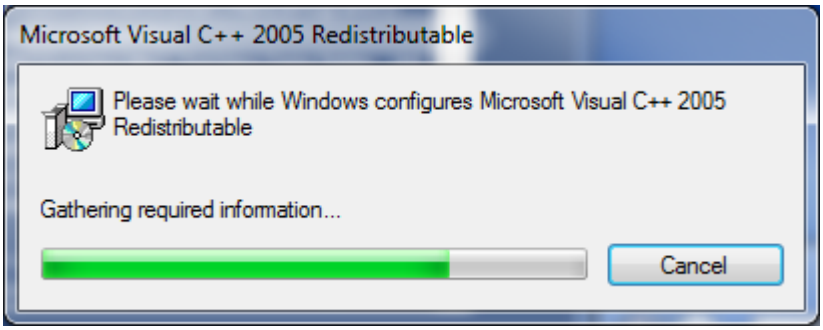


Figure 48: SynView Installation: Progress

After that the device driver for CorSight cameras as well as a network filter driver for GigE Vision based cameras (GigEPRO /GimagoEasy) will be installed. The filter driver installation might display a dialog that the driver has not passed the Windows Logo tests. This will occur several times.

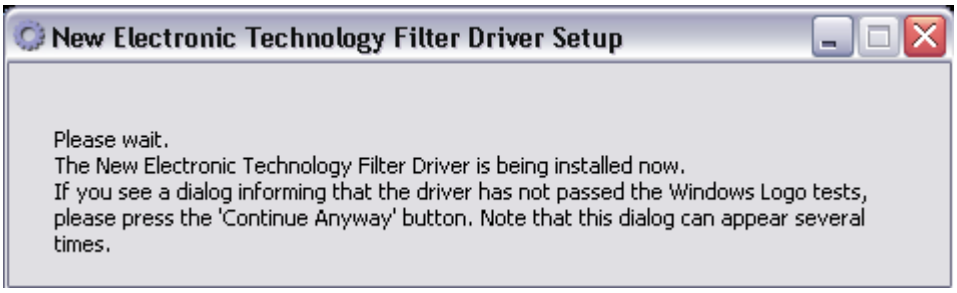


Figure 49: SynView installation: Filter Driver Message

When completed, the installer displays the final dialog. We recommend to keep the default option (Reboot now) and click Finish. After rebooting, SynView is ready to use.

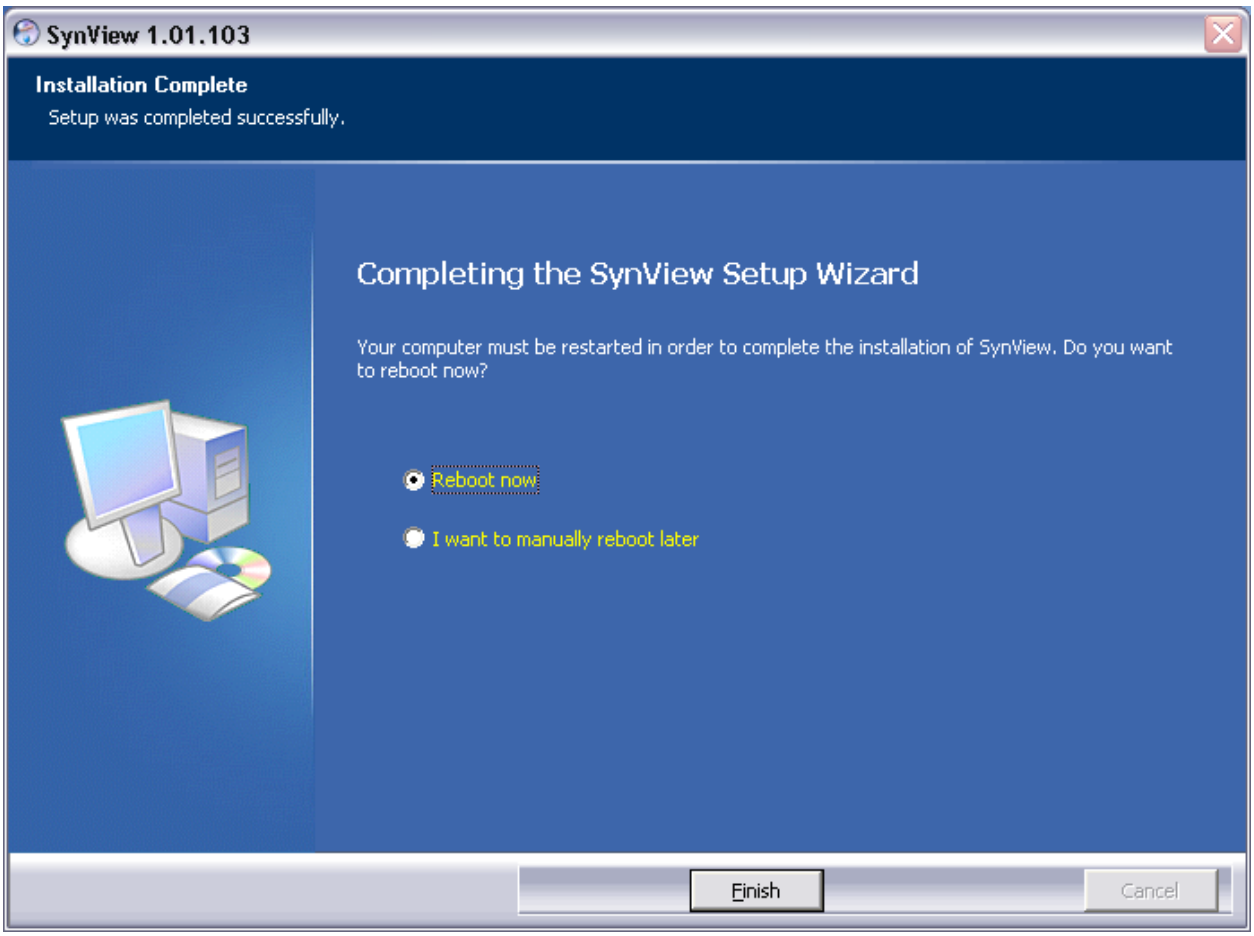


Figure 50: SynView installation: finishing installation

Installation layout

As mentioned above, the SynView files are installed to two destinations, the main directory and the application data directory. For the older Windows versions (Windows XP), these two are merged and the application data directory is actually equivalent to the bin subdirectory of the main installation destination.

Main installation directory

The main installation directory contains the following subdirectories:

- backup — backup of user modified files (such as the configuration files) that might be overwritten when upgrading to newer SynView version
- bin — all executable (*.exe) and library (*.dll files belonging to SynView
- data — tool-specific data, e.g. templates used by the SynView Source Code Generator
- doc — documentation files
- drivers — drivers (CorSight device driver, network filter driver)
- include — header files for SynView API development
- lib — library (*.lib) files for SynView API development
- samples — additional sample code (most of the samples can be, however, generated interactively using the SynView Source Code Generator)

Application data directory

The application data directory contains following main components:

- sv.synview.ini — SynView configurations file.
- sv.synview.log — SynView log file.
- XML — contains static XML files copied during installation. None of the files in this directory should be removed unless being explicitly advised by our support team.
- XMLCache — contains GenICam XML files downloaded from cameras.
- Cache — preprocessed versions of the GenICam XML files for performance improvement. These can be safely removed SynView will re-generate them upon next use.
- Debug — the .pdb files with debugging information for the SynView libraries. These might be useful for troubleshooting when contacting our support team.

Installing network filter driver under Windows

The SynView setup program can install the network filter driver automatically, if desired. The decision can be made on the setup program's dialog window, where installation components are selected. If not installed automatically, it is still possible to install the filter driver manually later, as described below. Before installing it, you might want to learn more information about the filter driver. In older SynView versions the setup did not install the filter driver automatically and the user was required to install it manually.

Manual filter driver installation

The instructions below describe how to install the network filter driver manually, if you did not enable it during installation of the SynView itself. Note that the instructions below assume that SynView is already successfully installed in the system.

Open the Network Connections applet in the Control Panel.

Windows 7

— menu Start →Control Panel →Network and sharing center →Manage Network Connection

Windows Vista

— menu Start →Control Panel →Network and internet →Network and sharing center →Manage Network Connection

Windows XP

— menu Start →Settings →Control Panel →Network Connections

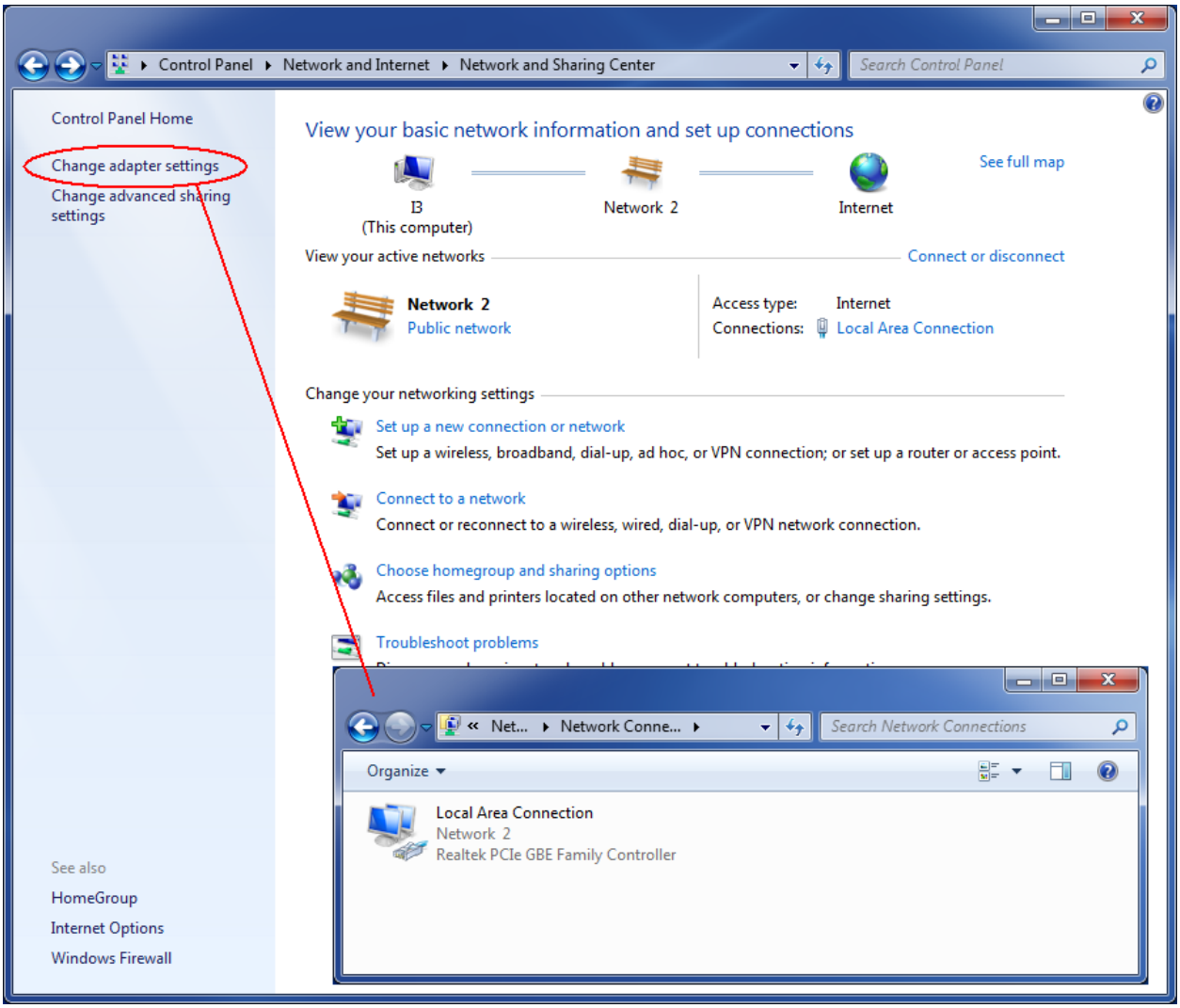


Figure 51: Windows 7: Installing the network filter driver, step 1

Click with the right mouse button on the network connection, which is used to connect the camera(s), select Properties from the context menu.

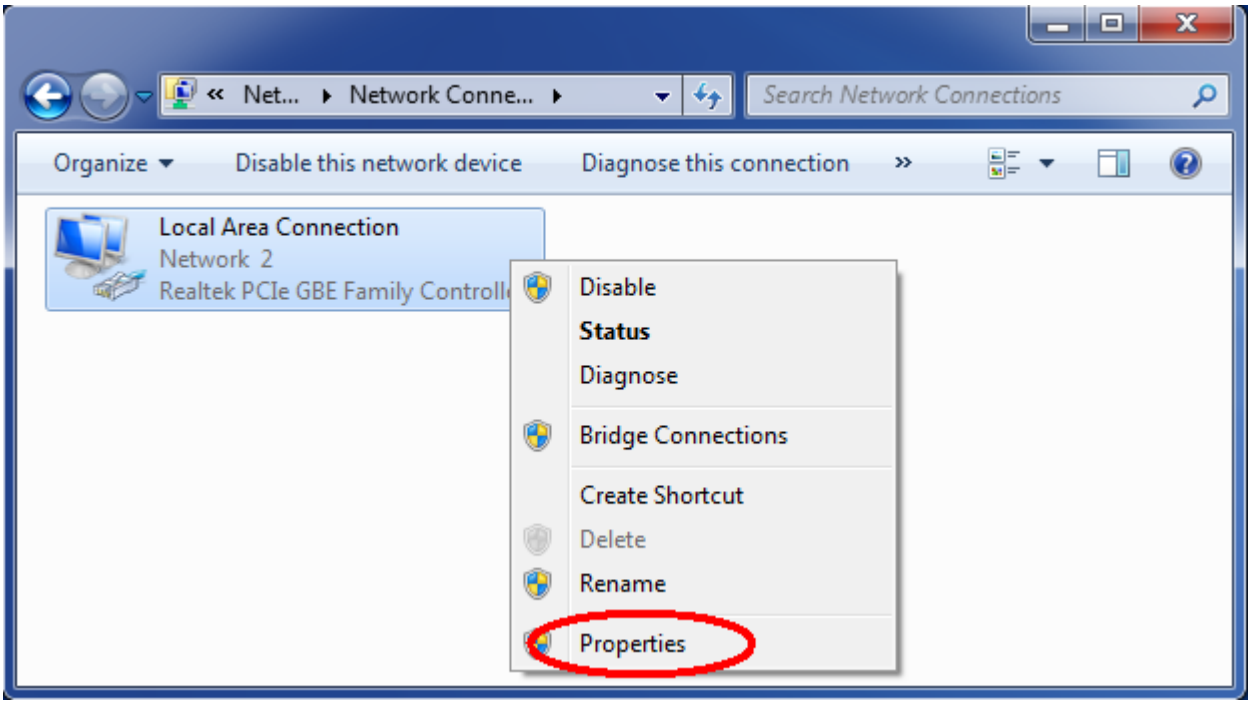


Figure 52: Windows 7: Installing the network filter driver, step 2

In the Properties dialog box, General tab select Install button to start the installation.

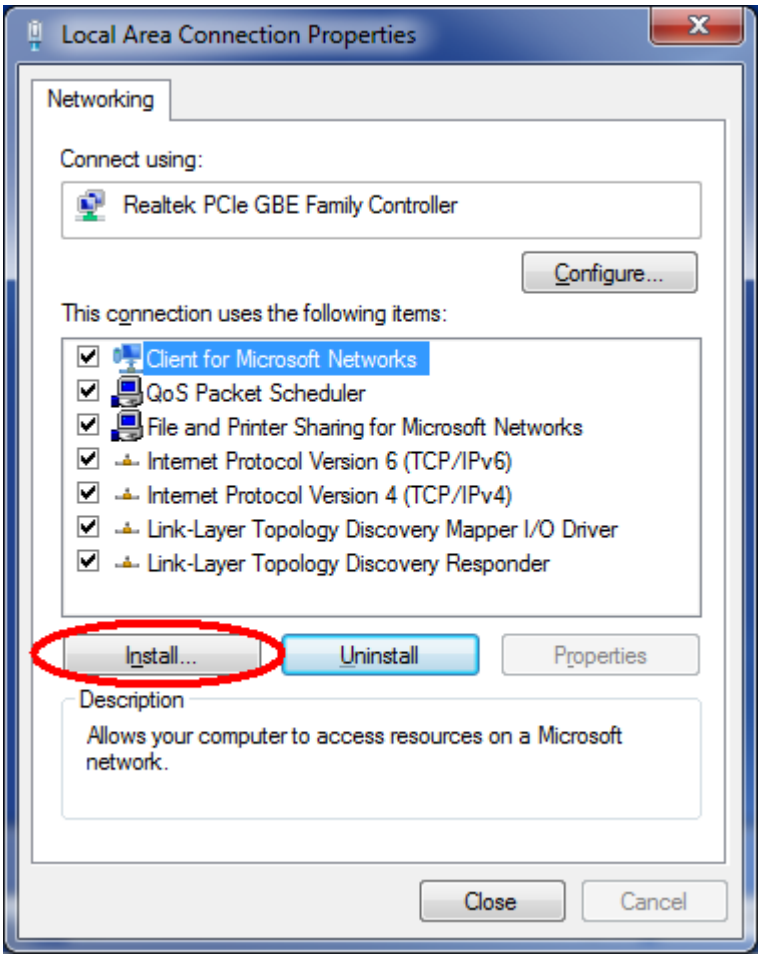


Figure 53: Windows 7: Installing the network filter driver, step 3

The Select Network Component Type dialog box appears. Select Service as the new network component type and click the Add... button.

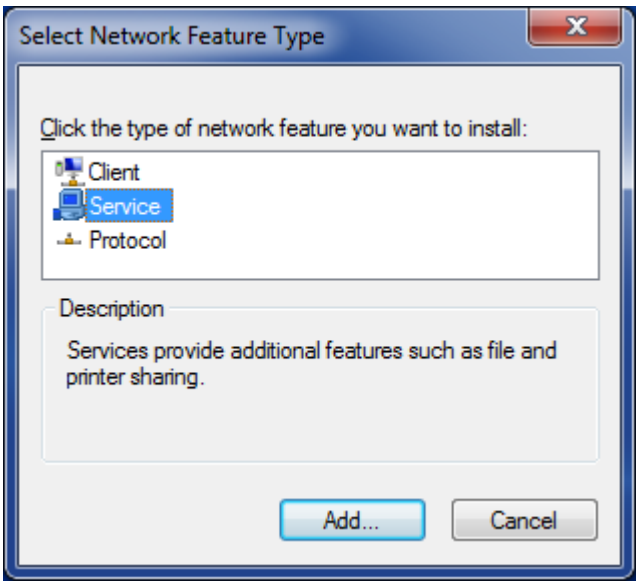


Figure 54: Windows 7: Installing the network filter driver, step 4

In the Select Network Service dialog box ignore the options possibly offered by the system and click the "Have Disk..." button to specify the driver location.

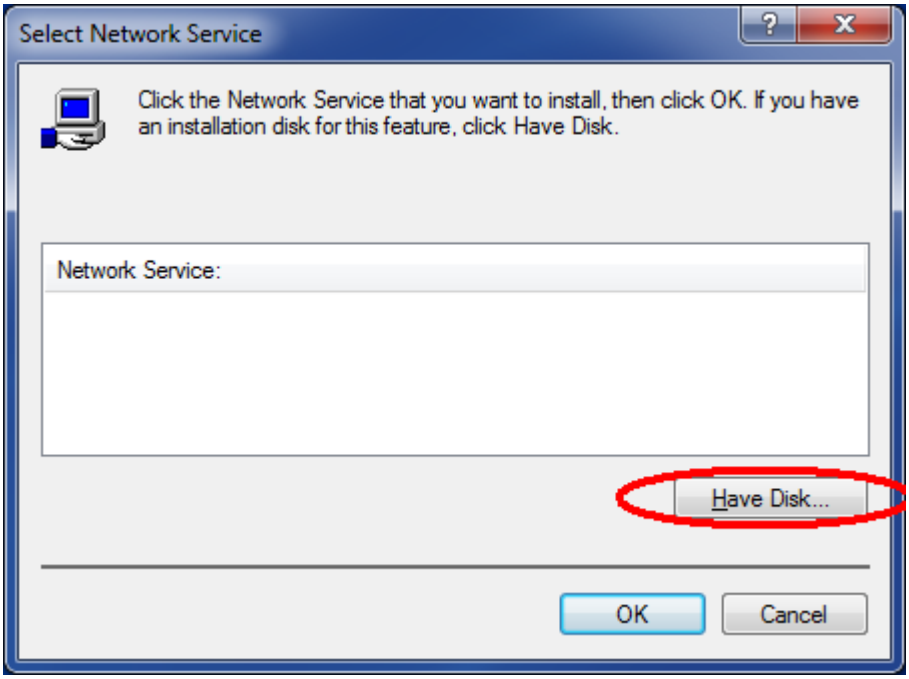


Figure 55: Windows 7: Installing the network filter driver, step 5

In the next dialog browse to the location where the filter driver files are stored. The directory is $(NET_SYNVIEW)\drivers\Windows32 , where $(NET_SYNVIEW) is the SynView installation directory C:\Program Files\SynView by default. When located, click Next.

For 64-bit Windows the driver files are located in $(NET_SYNVIEW)\drivers\Windows64.
The driver consists of 4 files that should in any case be available to the system in the same directory (that you need to locate in the dialog as described above).
The files are: SvGevDrv.sys (the driver executable itself), the SvGevDrv.inf (the file used to install/uninstall the driver), the SvGevDrv_m.inf (auxiliary) and the SvDrv.cat with a digital signature.
Windows should successfully find the proper driver and offer it for installation as "New Electronic Technology SynView Filter Driver". Select that option and click the OK button. Depending on SynView and Windows versions used one or more dialogs might appear to inform you that the driver is not tested for Windows compatibility. In such case, just click through them, allowing to finish the installation successfully.
When finished the filter driver will be successfully installed and ready to use. You should see it listed in the Local Area Connection properties dialog box.

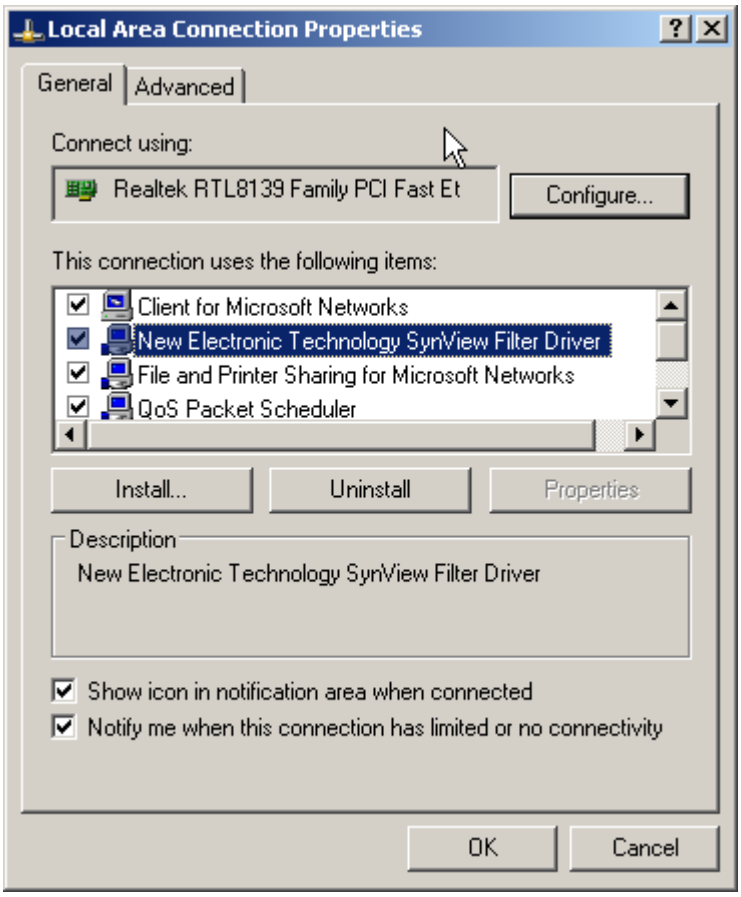


Figure 56: Windows XP: Filter Driver listed as network service

Installation in Linux

This section discusses SynView package installation under Linux with notes specific for the most popular distributions and overview of system requirements.

Installation package types

Linux world does not have a single standard way for software installation. The SynView Linux package is therefore distributed in multiple forms to suit all systems.

RPM format

- Used on most commercial distributions and their community variants, including Fedora/Red Hat, OpenSUSE/SUSE.
- The package file name is synview-x.yy.zzz.rpm, where x.yy.zzz stands for SynView version.
- To install, execute rpm -i synview-x.yy.zzz.rpm with superuser privileges.
- To uninstall, execute rpm -e SynView with superuser privileges.

DEB format

- Used on Debian based distributions, including Debian, Ubuntu (and variants) or Mint.
- The package file name is synview-x.yy.zzz.deb, where x.yy.zzz stands for SynView version.

- To install, execute dpkg -i synview-x.yy.zzz.deb with superuser privileges.
- To uninstall, execute dpkg -r SynView with superuser privileges.

Compressed archive (TARGZ)

- The .tar.gz archive is intended for systems, where the rpm/deb formats cannot be used or when a customized installation is required for serious reasons. Details are provided in a separate section "Custom installation in Linux" p.64.

System requirements

Following list enumerates the most important requirements for installing and using SynView. Note that the list might not be complete and is subject to change between SynView versions.

- Kernel version: kernels of the 2.6 line. The oldest tested version to date is 2.6.18, SynView is kept aligned with all the newest kernel version.
- Compiler: g++/gcc 4.1 and newer, libstdc++ v. 6 (GLIBCXX 3.4.5), pthreads implementation NPTL.
- When installing on CorSight, the kernel driver needs to get compiled and configured, therefore the basic infrastructure for kernel development needs to be available, in particular the basic compiling tools (gcc and make) and kernel headers. The name of the package containing the kernel headers might differ among individual distributions (eg. linux-headers, kernel-devel or similar). Additional system requirements might apply for CorSight systems, if in doubt, contact us to get full list of distributions supported by the CorSight cameras.
- LSB compatibility and xdg-utils are advantage, but they come by default on most of the usual distributions.
- To compile samples generated by the SynView Source Code Generator, one needs basic development tools (make, g++) plus packages for Xlib/Xt development.

Standard installation layout

The SynView installation follows the File system Hierarchy Standard. Its files are distributed in several directories.

Main installation directory

The main installation directory is /opt/synview.

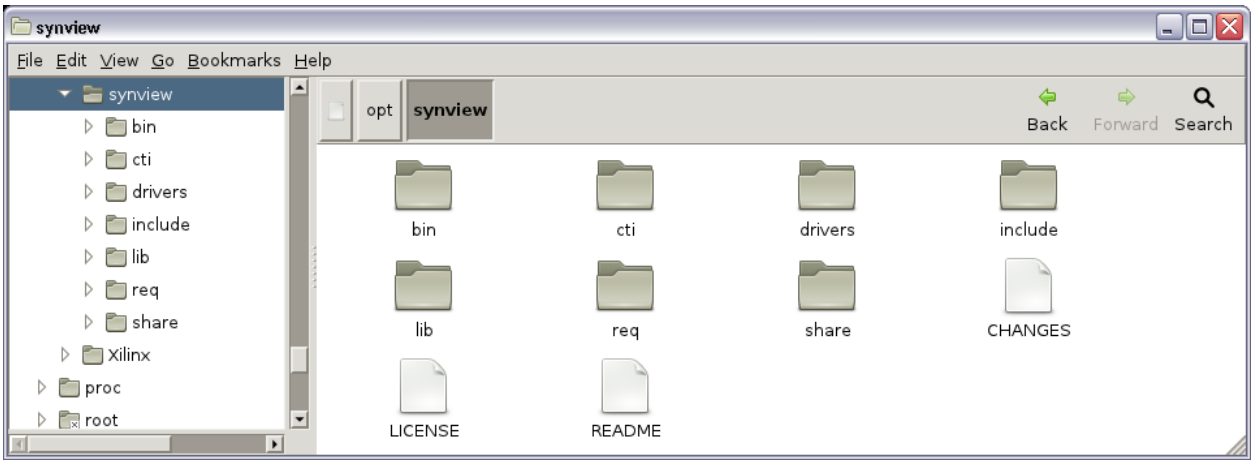


Figure 57: Linux Installation: main directory

The main installation directory contains all the runtime binaries and development files. It is organized in following subdirectories:

- bin — tools and test/demo programs, in particular SynView Explorer (sv.explorer) and SynView Settings (sv.settings)
- lib — all SynView API libraries, i.e. a directory where the linker should be pointed at; on 64-bit systems the lib directory contains the 32-bit versions, while the 64-bit ones reside in lib64
- cti — contains the SynView GenTL Producer library; on 64-bit systems the cti directory contains the 32-bit versions, while the 64-bit ones reside in cti64
- include — SynView API header files
- share — static data files used by SynView
- share/xml — static XML files copied during installation
- req — foreign requisite components used by SynView, in particular the GenICam runtime
- drivers — scripts and sources necessary to build and install SynView device drivers (used only on CorSight systems)

Configuration files

The configuration files are stored in /etc/opt/synview.

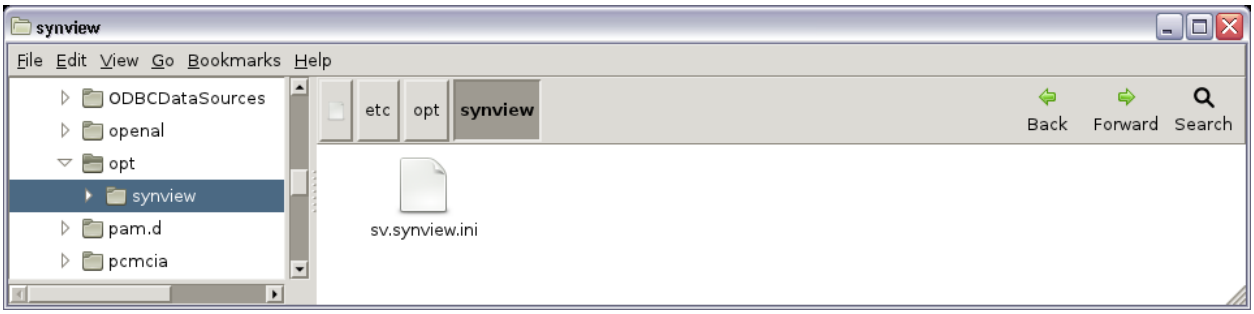


Figure 58: Linux Installation: configuration file

The directory contains in particular the main SynView configuration file, sv.synview.ini. The file should be edited with superuser privileges. See Section "SynView Settings utility and the configuration file" p.67 for details.

Application data

The various application data, such as logs and cache go to /var/opt/synview.

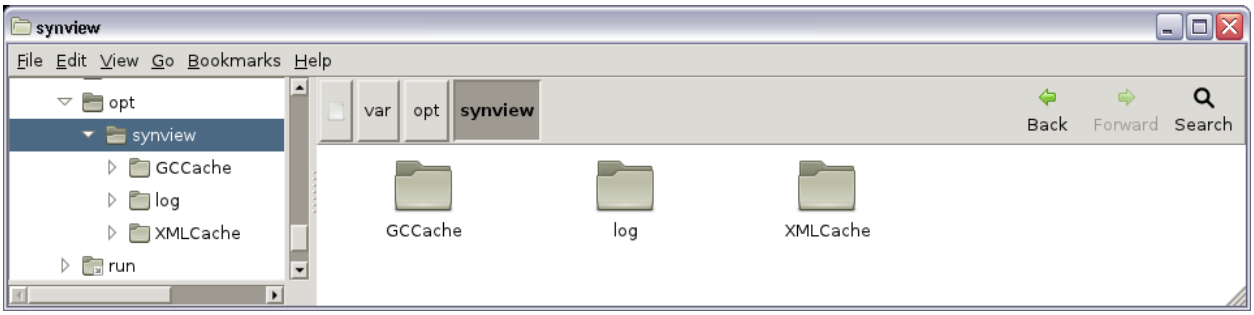


Figure 59: Linux Installation: application data

The data are stored in several subdirectories, in particular:

- log — default destination for SynView log files, in particular the sv.synview.log that are useful for any troubleshooting ("Troubleshooting & Support p.73).
- XMLCache — contains GenICam XML files downloaded from cameras.
- GCCache — preprocessed versions of the GenICam XML files for performance improvement. These can be safely removed. SynView will re-generate them upon next use.

User specific files

Possible user specific files (such as demos'/tools' configuration) are stored in ~/.synview.

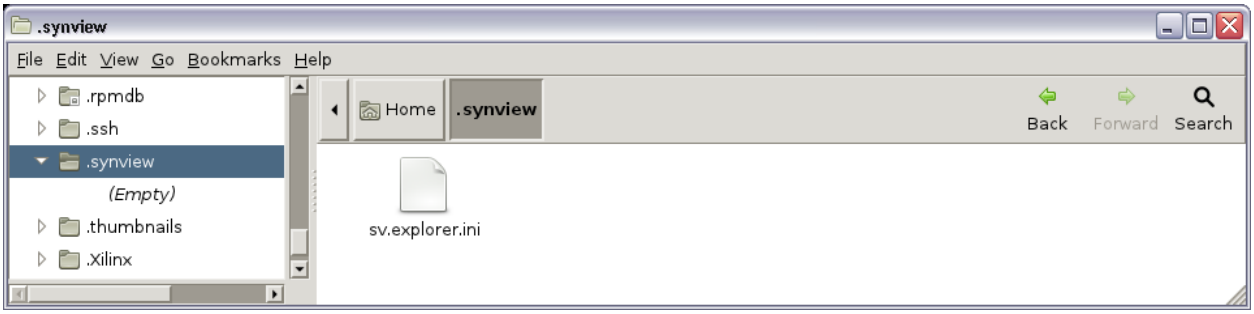


Figure 60: Linux Installation: user specific data

This directory is used solely to store user specific settings or preferences for SynView tools, such as SynView Explorer or SynView Settings. It will not be needed for usual operation on a runtime system.

Other actions performed by the installer

Besides copying the files and adjusting user rights, the installer performs some additional actions, including:

- When installing on a CorSight system (detected automatically), the necessary device driver is built and installed.

- For convenience, the directory /opt/synview/bin is added to PATH environment variable, so that the tools are directly accessible from the shell. This is performed by adding an entry to /etc/profile.d — on systems using other mechanism, the system administrator might need to adjust the path manually.
- Directory /opt/synview/cti is added to the GENICAM_GENTL32_PATH environment variable (and analogically the 64-bit variant), so that the SynView GenTL Producer can be located by 3rd party GenTL consumers. This is performed by adding an entry to /etc/profile.d — on systems using other mechanism, the system administrator might need to adjust the path manually.
- For convenience, the directory /opt/synview/lib (and its 64-bit variant) is added to the search path of the dynamic linker. This is performed by adding an entry to ld.so.conf.d — on systems using other mechanism, the system administrator might need to adjust the path manually.

Notes for the most common distributions

The SynView package is tested with the most popular distributions. Brief notes for those distributions are listed below. Note that this list by no means suggests that other distributions are not supported, SynView should work on any system fulfilling the basic requirements (“System requirements” p.60). Also the versions listed for each distribution are just a hint, showing which was the original version used for tests. Each newer version of given distribution should work well and older versions would work if fulfilling the mentioned system requirements.
For each distribution, the tests were done with the most basic “default” setup including X, but without any special additions.
The notes below are related only to SynView itself, assuming the Linux operating system was properly installed and configured. Special requirements related to operating system installation might apply on CorSight systems —. To install SynView on CorSight, kernel development infrastructure (gcc, make, kernel headers) must be installed to successfully build and configure the device driver.

Ubuntu

Originally tested version: 10.04 LTS (Lucid Lynx)
Packaging type: deb format
Works out of the box

Debian

Originally tested version: 6.0 (Squeeze)
Packaging type: deb format
Works out of the box

OpenSUSE (community version of SUSE Linux Enterprise)

Originally tested version: 11.2
Packaging type: rpm format
Default installation switches on a paranoid firewall, which has to be configured (or switched off) if working with GigE Vision cameras (GigEPRO /GimagoEasy)

Fedora (community version of Red Hat Enterprise Linux)

Originally tested version: 14 (Laughlin)
Packaging type: rpm format

Default installation switches on a paranoid firewall, which has to be configured (or switched off) if working with GigE Vision cameras (GigEPRO /GimagoEasy)

Custom installation

Custom installation in Linux

In some situations, the official SynView installers (RPM or DEB) cannot be used, either because given system does not properly support either of them or when installing to other than default locations is required for serious reasons. To cover those situations, the SynView package can be on request delivered in the .tar.gz format. This allows full flexibility during the software installation; however, certain rules need to be followed to make the installation fully functional. Note that the installation details are subject to change — in case of doubts or experiencing problems with the installation, contact our support team. It is highly recommended to use the standard installers wherever possible.
Before planning the installation, you might want to read the section describing the regular installation, including the system requirements and the default installation layout: Section "**Installation in Linux**" p.59.

Extracting the archive

The package file name is synview-x.yy.zzz.tar.gz, where x.yy.zzz stands for SynView version. As a first step, the files from the archive should be extracted to the system:

- The opt directory should be extracted to /opt. Alternatively it can be extracted to a custom location, provided that the environment is properly adjusted, see below.
- The etc directory should be extracted to /etc. Alternatively, it can be extracted to a custom location, provided that the environment is properly adjusted, see below.
- The var directory (if present in the archive) should be extracted to /var. Alternatively, it can be extracted to a custom location, provided that the environment is properly adjusted, see below. If the directory is not present in the archive, we recommend creating it in the target system, including (empty) subdirectories log, XMLCache and GCCache.

Adjusting the environment

The path to the SynView GenTL Producer needs to be adjusted according to the GenTL standard requirements:

- On 32-bit installations, the variable GENICAM_GENTL32_PATH must point to the cti subdirectory of the main SynView installation.
- On 64-bit installations, the variable GENICAM_GENTL64_PATH must point to the cti64 subdirectory of the main SynView installation.

On systems supporting the /etc/profile.d mechanism, this will be handled automatically by the file etc/profile.d/synview.sh in the .tar.gz archive. The file, however, assumes default installation locations — in case of custom installation, its contents need to be adjusted accordingly.

The PATH variable could be adjusted to include the bin subdirectory of the main SynView installation directory, so that the individual SynView applications can be loaded from the console without specifying the full path.

Dynamic linking

If desired, the directory with the SynView shared libraries should be added to the search path of the dynamic linker. The shared libraries are located in lib or lib64 (depending on architecture) subdirectory of the main SynView installation.
On systems supporting the /etc/ld.so.conf.d mechanism, this will be handled automatically by the file etc/ld.so.conf.d/synview.conf in the .tar.gz archive. The file, however, assumes default installation locations — in case of custom installation, its contents need to be adjusted accordingly. To apply the configuration, the ldconfig command will have to be executed.
The dependencies of the individual SynView libraries and executables should be resolved automatically — the corresponding links are stored in given files using the DT_RPATH/DT_RUNPATH ELF tags. The tags contain both the absolute path (pointing to the default installation locations) and the relative path (using the "$ORIGIN" convention). This means that dependencies should be automatically resolved even when installing to a custom location.
Note that in certain situations (such as when using suid binaries), the $ORIGIN based path might be ignored by the dynamic linker. In such situation, you might need to adjust the runpaths to match your installation, or use another equivalent mechanism (such as LD_LIBRARY_PATH) of your choice.

Access rights

The execution rights for all the executables should be adjusted according to your system policy. The default installers enable the execution rights for all users. Additionally, you might want to add the setuid bit for the sv.fwupgrade so that the tool is allowed to automatically shutdown the CorSight system after firmware update.
The /var/opt/synview directory (or its equivalent specified through LVS_VAR environment variable) and its subdirectories must have write access for all users.
All other installed files should have at least read access for all users enabled.

Kernel drivers

Depending on the hardware you are going to use, you might need to install, build and configure the kernel driver(s) delivered with SynView. This applies in particular to the CorSight device driver (not needed when using only GigEPRO cameras), but in future SynView versions other drivers might be available.
The driver(s) are available in the driver subdirectory of the main SynView installation. There's one subdirectory per driver. Each driver comes with a control script (typically named control_driver). This script is used by the default installers to build, install and configure the driver. The script is easy to follow and commented, so it should be easy for the system administrator to understand its purpose and eventually adapt the script to match the desired target system. Note that the script is subject to change any time between SynView versions without prior notice.

Optional files

Some of the files distributed with the SynView package might not be necessary for the basic runtime operation.

- The tools installed in the bin subdirectory are very useful in the development phase, but might not be necessary in the target runtime installation. In such case they can be freely omitted without any harm.
- The 3rd party prerequisite libraries, which are not typically installed on every system, or might be present in different version, are distributed together with SynView in the req subdirectory. This covers especially GenApi, Qt, Xerces/Xalan. On embedded, space-sensitive installations, where the system already contains one or more of these components, the “original” copies can be used instead of those delivered in the SynView package. It is, however, fully responsibility of the system administrator to determine and use the proper versions of these prerequisites for every given SynView version and configure the dynamic linking properly. Do this only at your own risk.
- The support files for the SynView Source Code Generator (in share/srcgen) can be omitted if SynView Source Code Generator functionality is not needed.
- The contents of the driver’s directory might or might not be needed, depending on the hardware you are going to use. It is fully responsibility of the system administrator to install and configure all the required drivers.
- The configuration files from the etc tree can be omitted if you compensate their purpose by other means (most of them were discussed above). The only exception is the sv.synview.ini file, which must always be present.

De-installation/upgrade

When uninstalling or upgrading the SynView package, all relevant steps corresponding with your custom installation scenario should be performed to keep the system consistent.

Configuration

SynView Settings utility and the configuration file

The SynView configuration options are concentrated in a single file, sv.synview.ini. On Windows systems the file is located in the “application data” part of the installation. On Linux systems it's stored by default in /etc/opt/synview.
The file can be edited directly — every single option is well documented directly in the file, including its possible values. Note that on Linux superuser privileges are required to edit the file.
Instead of editing the file directly, however, the recommended alternative is to edit the file through a helper utility, SynView Settings (Windows: menu Start →Programs→SynView→Tools→SynView Settings; Linux: /opt/synview/bin/sv.settings as superuser).
The tool provides a comfortable way to edit the sv.synview.ini file.

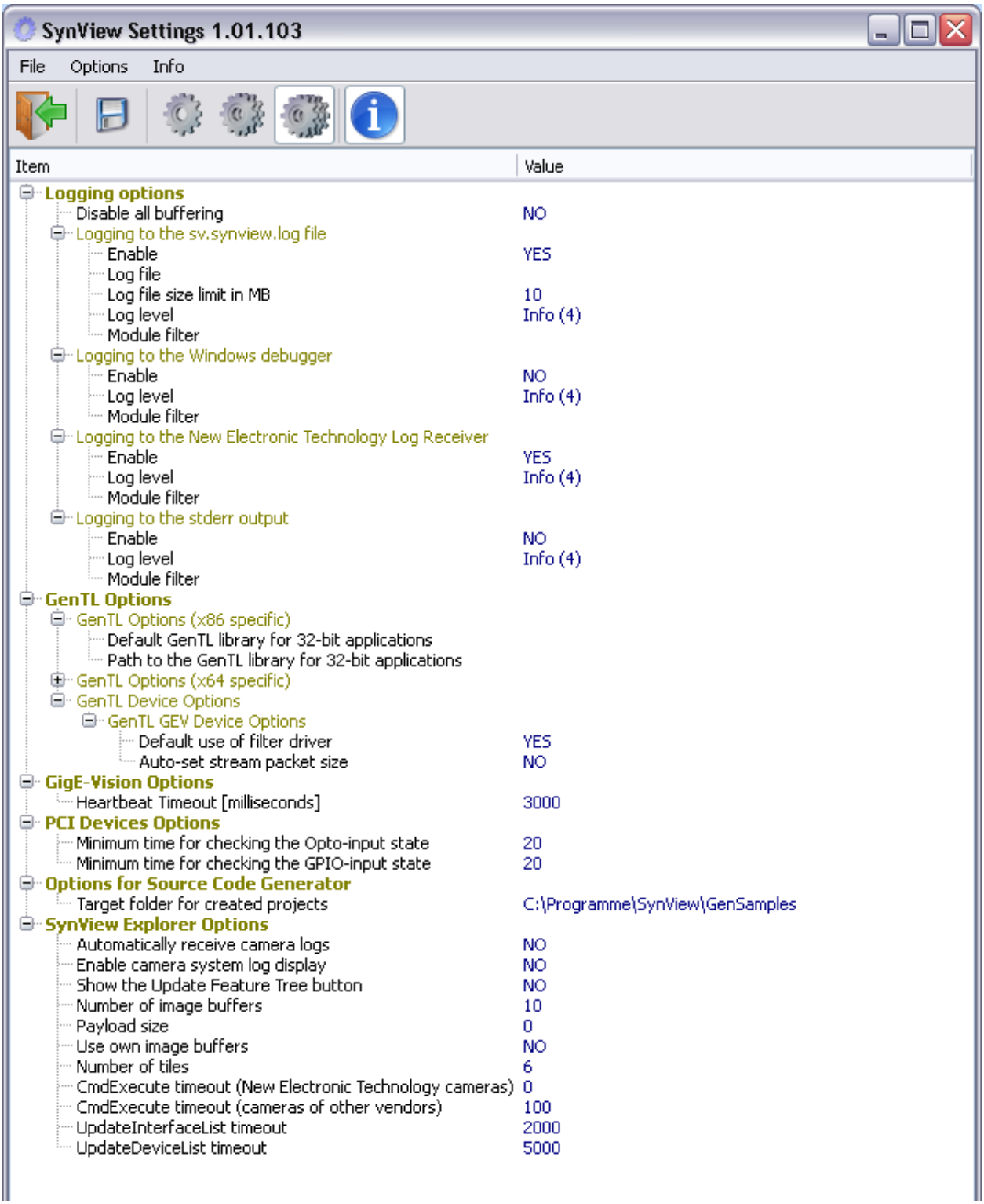


Figure 61: SynView: Settings utility

The tool's interface can be controlled threw a handful of menu options (or corresponding buttons). You can show/hide the info pane with info about the selected configuration option. You can switch between

three configuration levels (beginner, expert, guru) — the higher level, the more detailed options are shown. Finally, you can store the changes.
The set of options itself is ever evolving, so the SynView Settings tool itself (or the sv.synview.ini file) is the most complete documentation. The options include detailed logging configuration (see also **Troubleshooting & Support** p.73), system installation related info and configuration of individual SynView subsystems.

Selected configuration options

The documentation for all the SynView configuration options is provided directly in the SynView Settings utility and the configuration file. Here we'll mention just few selected important options:
Logging options
The SynView log is an important tool for any support and troubleshooting. For regular operation, the log might be kept at low level (default is level Info (4)) or even switched off to optimize the performance. However, during application development or when debugging a problem, we recommend to switch the Log level configuration option to Trace (6). All log files sent to our support team should be created with level Trace (6).
The logging infrastructure is optimized for high performance. For example, the logging to file is buffered to minimize the disk access overhead. This can have a drawback, however, that in case of an application crash, the last part of the log might be missing in the file. When troubleshooting application crashes, we recommend switching on the option Disable all buffering.
Heartbeat timeout
Heartbeat timeout is an important parameter, related to GigE Vision cameras only (GigEPRO/ GimagoEasy). This timeout, in milliseconds, defines how frequently the application SW has to send the “heartbeat” messages to the camera, indicating that it's still alive. When the camera does not receive the message within the specified timeout period, it assumes the application has failed and disconnects itself. The default value used in SynView is 3 s. At most times, the heartbeat handling is fully transparent to the user. However, during debugging, when a developer stops the application at a breakpoint for single stepping, the SynView heartbeat handling thread would also stop, fail to deliver the heartbeat messages and the camera would disconnect. To prevent that, we recommend increasing the heartbeat timeout parameter to a higher value. Please keep in mind that with a high heartbeat value the camera will be available later for reconnection in case of an unexpected disconnect from the application side. The parameter is configurable through camera features at runtime. Its default value can be adjusted through the SynView Settings utility.

Vision Standards

The following paragraphs provide brief information about the most important machine vision standards built into SynView and the GigEPRO /GimagoEasy and CorSight camera families.

GenICam

The main idea of the GenICam standard, published first in 2006 under EMVA (European Machine Vision Association), is to provide a unified application programming interface (API) to the users of machine vision cameras. It enables an easy integration of individual components, such as cameras, image processing libraries, drivers or frame grabbers.

GenICam is independent on the transport layer technology. The GigE Vision standard uses GenICam to access camera configuration. GenICam is also implemented by manufacturers of cameras based on "older" interface technologies (Camera Link, USB, IIDC DCAM) and efforts are made to integrate GenICam back into these standards. More importantly, newly developed machine vision standards are expected to build upon GenICam and smart camera platforms, such as NET CorSight also heavily rely on GenICam based interoperability. GenICam is thus supported by virtually all important manufacturers of machine vision components. GenICam consists of three main modules, GenApi, GenTL, and SFNC.

GenICam GenApi

GenApi (GenICam Application Programming Interface) is a basic building block of GenICam. It allows describing complete camera functionality using an XML file with precisely defined syntax. Individual camera features are described by their name, type (integer, float, boolean, command, string, etc.), address, length and other parameters of the register controlling the feature, as well as other details including complex description of logical and mathematical relationships between individual features. The XML file is usually stored directly on the camera, which further simplifies the automatic configuration process. The standard clearly defines the way, how the XML configuration file should be interpreted, so that the host-side implementation (application itself) and camera can cooperate. GenICam also provides quality reference implementation in form of a set of C++ libraries with license similar to BSD. All GenICam software products known to the author are using the reference implementation. GenApi is fully independent on the transport layer, which only has to supply two functions for reading and writing the camera registers by means of its corresponding protocol (GVCP in case of GigE Vision). The GenApi principle is revolutionary among other aspects in the degree of flexibility available for the camera description. GenICam GenApi allows exact camera description, while leaving full freedom for layout and implementation of individual registers, as well as for defining custom features, not specified by the standard.

GenICam GenTL

GenTL (Transport Layer) is the newest addition to GenICam. It defines interface for acquisition of image sequences (or additional non-image data) independently on the transport layer technology and platform (operating system, programming language, etc.).

GENICAM TRANSPORT LAYER

GenTL allows to enumerate and identify the devices (cameras) connected to the system, control access to them from individual applications ("GenTL Consumers"), configure them (by means of GenApi), configure the pool of buffers used for the acquired data and control the acquisition itself, including synchronization, buffer locking, control of the input and output buffer queues, reading additional data bound to individual images, etc.
The basic camera functionality is available through programming interface compatible with the ISO norm for the C programming language. More advanced GenTL features can be configured by means of a GenApi interface, similarly as the camera itself. In such case the GenTL Producer implements a virtual register space, described by a similar XML file, which is used also for description of the cameras.
Thanks to GenTL, the applications can use cameras across the transport layer technologies, without knowing any low-level details about each respective technology.

GenICam SFNC

SFNC (Standard Features Naming Convention) is kind of superstructure on top of GenApi. It defines a convention for naming features typical for machine vision cameras. SFNC defines a universal camera model, which might be implemented by most of the cameras on the market. Generic applications and libraries can, thanks to this standard model, automatically comprehend virtually any newly connected camera and configure it according it needs, without "manual" intervention from the programmer or user. SFNCC thus practically takes the full freedom of GenApi and ties it again to a clearly defined model, so the features listed in the XML configuration file can be automatically (ie. without study of the documentation) bound to a particular meanings. Nevertheless, the camera implements this model only to the level possible and practical for its design. The designer has still full freedom for adding his own extensions and features not defined by the standard.

GigE Vision

GigE Vision is relatively young but already very mature and widely adopted standard published in 2006 under AIA (Advanced Imaging Association). It defines interface (set of communication protocols) for industrial cameras using Ethernet as its transport media. The goal of the standard was to reach high degree of interoperability between individual manufacturers. Despite of the name, suggesting that it was originally designed for gigabit Ethernet, the standard itself does not define any particular transport media and enables thus transition to faster media (10G Ethernet) in future.
Main advantages of GigE Vision against competing technologies are:

- Gigabit Ethernet infrastructure is readily available in most current systems and is well known.
- Sufficient bandwidth for most of current camera designs.
- Cable length up to 100 meters without regeneration, with switches the length is practically unlimited.
- Number of cameras available in the system is practically unlimited.
- Wide spectrum of network topologies, support for multicasting etc. enable to build specific systems with eg. multiple cameras simultaneously available (and simultaneously acquiring) to multiple host computers.
- Plug&play, support of GenICam standard.
- Cost savings thanks to utilization of cheap, off-the-shelf equipment such as network cards (instead of specialized frame grabbers), standard network cables, etc. Total costs can compete even with analog systems.

GigE Vision is actually set of protocols built on top of UDP/IP. UDP was selected to achieve maximum possible performance. Error checking and repairing mechanisms are available directly by means of the GigE Vision protocols, if it is required by the application.
Main components of the standard are Device Discovery, GVCP, GVSP, Bootstrap registers:

Device discovery

Device discovery is the mechanism allowing discovery and identification of GigE Vision compatible cameras available in the system. The application issues (broadcast) a request for identification of the cameras. Individual cameras must react to this request by sending their card, containing for example the camera's IP address, name of the camera manufacturer and model name, as well as other basic information.
The application can change the IP configuration of the camera (e.g. to bring it to its own subnet) and request exclusive rights to access the camera.

GVCP

Once the connection is established, the application communicates with the camera using GVCP (GigE Vision Control Protocol). Its task is to control all the communication with the camera, especially read and write all the camera's control registers and allow thus the actual image acquisition.

GVSP

As soon as the camera is configured through GVCP, it can start sending the image and other data. GigE Vision defines GVSP (GigE Vision Streaming Protocol) for this purpose. GVSP mainly defines how the image data are packetized and it also offers means for recovery from possible errors occurring during the data transfer.

Bootstrap registers

The most basic camera parameters, allowing its identification and establishing the connection are summarized in a mandatory block of registers, so called bootstrap registers. The layout of additional registers defining specific camera functionality is left completely up to the camera implementation. The camera describes that layout using XML file compatible with the GenICam standard. This configuration file is usually stored directly on the camera and can be downloaded by means of GVCP.

Troubleshooting & Support

Getting support of NET products

Before you contact the support team

Make sure the problem is reproducible, preferably with the SynView Explorer and that you capture the sv.synview.log file (See next chapter: “Gathering information about the problem”).
Attach the captured log file (one per each session if multiple test scenarios are described in the report) as well as other important information, describe all details about the problem, see “**Reporting the problem**“ p.74. Following these guidelines will guarantee that your problem can be solved effectively, in the shortest possible time.
Send the problem report to proper e-mail address, depending on your location and the problem nature, as explained in Section “**Technical Support**“ p.77.

Gathering information about the problem

- When collecting information about your problem, please follow the steps listed hereafter:
- Be sure you use the latest version of SynView. It may be that you have an old version of SynView installed and you are experiencing a bug, which was already removed in the meantime. The version of currently installed SynView is visible in the SynView Explorer tool (menu Help →About). The latest SynView version can be downloaded from the SynView download area.
- Enable logging. Before attempting to reproduce the problem, be sure the logging is enabled through the SynView Settings tool (or directly the sv.synview.ini file (refer to Section “**SynView Settings utility and the configuration file**“ p.67)) enable logging to file, preferably with highest log level. If the application crashes and the log is not complete, switch on the Disable all buffering log configuration option.
- Test the hardware functionality with a supplied demo program “SynView Explorer”. As the first step use the “standard” demo program, SynView Explorer, which covers most of the possibilities of NET hardware usage. If you are able to reproduce the problem in SynView Explorer, then the problem is probably not in your code. However, it still need not be a bug, a problem can be also caused by an improper configuration.
- Camera configuration. Check especially whether the camera is set in a proper working mode. See camera documentation to learn how the camera should be adjusted.
- Check another hardware. Sometimes the problem can be caused by the HW/SW environment or by a defective piece of hardware. Try if the same problem happens on another PC. If you have another camera of the same type available, try if the problem happens also after the camera replacement. For GigE Vision based cameras, review the network configuration and try to connect the camera directly, without additional network components in the path.

- Check your application. If the problem could not be reproduced in the previous steps, reproduce the problem in your application. If possible, extract and send us the smallest part of your code that can demonstrate where the problem occurs. Even better is if you can make the extracted code fragment compilable, so that the problem can be reproduced with a small application that you can send to us. Make sure that your application does proper error handling when accessing SynView functions. Ignoring the error status of SynView functions can lead to unexpected behavior of your application.
- Screenshots. In case of disturbances in the acquired images, make and send us the screenshots of them, either as bitmaps with lossless compression (.bmp, .png) or as .jpg with a high quality (small compression).
- Send us the report. Send all the collected information (and especially the sv.synview.log file) to us, following guidelines from the next chapter: “Reporting the problem”.

Reporting the problem

Each problem report should contain following information:

- The e-mail subject should give a brief but specific description of the problem and your company name (separated by a “@” character). The subject will be used to identify evolution of the support case, so if you need to submit a different problem, create a new subject string, do not reuse the old one.
    - Example of a good subject: Changing exposure time has no effect @ My Company Inc.
    - Example of a bad subject: Problem with camera
- Describe your problem in detail, including all its symptoms, conditions under which it occurs and all other information that can help resolving the problem. List all the information you gathered while testing and reproducing the problem, as described in Chapter “**Gathering information about the problem**“ p.73.
- Attach the required sv.synview.log file (one per each reported session). The file absolutely essential for our support team and failing to attach it will usually only lead to a response asking you to collect and send it, which will in effect make the time needed for the resolution longer. Be sure that you send the log file generated during the problematic session you are mentioning in your report reporting.
- If troubleshooting a CorSight camera running Linux, attach also the device driver log (
- **Linux driver** logging“ p.76) and contents of /proc/lvsm. The driver logs are appended to the standard kernel log and can be retrieved for example using the dmesg command.
- If the problem occurs only under some conditions or only with some particular piece of hardware or software, describe the differences between the two cases and attach log files generated during both the successful and problematic sessions. If only a specific sequence of steps leads to the problem, describe this sequence.

Troubleshooting NET products

Troubleshooting common issues

The lists below mention some most common issues that can be faced by customers, but have easy solution. It is a good idea to check this list first when encountering a problem. It is also a good idea to check the next Chapter “SynView logging” whether some hints can be found there.

- If you are not getting any image, if you get different image data than expected or if you have problems with acquisition timing, check again carefully that the camera is set in proper working mode and that it is not in some error status (see camera manual for possible description of such situations).

If you have problems with acquisition from a GigEPRO /GimagoEasy camera, check whether your firewall settings do not interfere with the acquisition.

SynView logging

All the SynView components are using a common logging subsystem, which can be configured through the sv.synview.ini and SynView Settings (“**SynView Settings utility and the configuration file** **p.67**).
The log can be directed to various outputs, while the most important of them (especially for support) is the file output. The log file path can be adjusted however the default name (sv.synview.log) as well as the default location (application data directory under Windows, /var/opt/synview/log under Linux) will be just right in most situations.
The logging level can be adjusted. The default level Info (4) provides most useful information about the system and SynView operation. For specific troubleshooting and support requests, it is often useful to switch to the most verbose level Trace (6).
In case of troubleshooting problems involving an application crash, when the log might not be written completely (log buffers not flushed completely at the time of application crash), the logging subsystem should be configured to avoid any buffering. In such case, please switch on the Disable all buffering option.
The logging configuration offers advanced options, the most important of them being the max allowed size of the log file. The log can be also directed to other outputs, but those are mainly intended for use during advanced debugging by qualified personnel.
Several logging mechanisms are used by various SynView components. They are separated to utilize log analysis, to separate domain-specific info from the general purpose logs and to improve performance, logging only what needs to get logged in a given situation. Creation of these log files can usually be controlled by the user.

Note: All information in the logs is informative and its main purpose is to assist the NET support team when trying to resolve possible problems. It is not designed as a tool that should be necessarily well understandable by users, it might be misleading and all the information is context-specific. If a word “error” or “warning” appears in the log, it does not necessarily mean a real error occurred, it might be valid situation in a given context. You don't need to get alarmed because of that and contact support, especially if the application is otherwise working well.

Linux driver logging

On the CorSight systems, a device driver controlling the acquisition device is running. The driver can also provide useful logs, but those are not routed through SynView user level logging subsystem, but rather through standard Linux kernel logging mechanisms. Depending on the system configuration, there might be multiple ways to access that log, but the most convenient is usually to simply execute the dmesg command and store its output to a file. Apart of the kernel log messages, other (mostly static) information can be available through reading the /proc/svsm.

Debugging with GigE Vision cameras

When debugging an application using GigE Vision cameras (GigEPRO /GimagoEasy), it's necessary to pay attention to a low level GigE Vision communication parameter, the heartbeat timeout.
This timeout, in milliseconds, defines how frequently has the camera send the “heartbeat” messages to the camera, indicating that it's still alive. When the camera does not receive the message within the specified timeout period, it assumes the application has gone away and disconnects itself. The default value used in SynView is 3 s (other software packages might use another value).
At most times, the heartbeat handling is fully transparent to the user. However, during debugging, when a developer stops the application at a breakpoint for single stepping, the SynView heartbeat handling thread would also stop, fail to deliver the heartbeat messages and the camera would disconnect. To prevent that, we recommend to increase the heartbeat timeout parameter to a higher value. It has to be understood that when the application stops unexpectedly during debugging the camera would not disconnect properly and it will take according to the debug heartbeat setting longer time before the camera becomes available again for connection. The heartbeat parameter is configurable through camera features at runtime (the feature's GenICam name is GevHeartbeatTimeout), while its default value can be adjusted through the SynView Settings utility, configuration option Heartbeat timeout.
Note that some software packages attempt to increase the timeout automatically, whenever they find out that a debugger is attached to the process. We don't do that since we believe that the correct heartbeat timeout value can only be determined by the application developer, for reasons described above.

Debugging symbols

When debugging certain kinds of problems, such as application crash, it might be very useful if you can provide the support team with extended diagnostic information, such as the stack trace showing the crash point. The installer of SynView for Windows allows to install “PDB files” — files containing the debugging information (symbols) for the SynView libraries. We recommend to install these debug files on the development and testing systems, so that it is possible to generate the stack trace info whenever encountering a problem. The files are located in the bin\Debug subdirectory of the SynView installation.
Note that in Linux version of SynView the symbols are directly included in the libraries and executables.

Technical Support

NET ensures the conformity of its product to be reliable and free from defects during manufacturing by testing all the cameras before release. However, unexpected problems and technical issues may come up due to the complexity of the product.

In case you require technical support, contact the agent near you or contact NET directly at the following locations:

Websites

Europe	www.net-gmbh.com
France	www.net-france-sas.fr
Italy	www.net-italia.it
USA	www.net-usa-inc.com
Asia	www.net-japan.com

Email

Europe	info@net-gmbh.com
France	info@net-france-sas.fr
Italy	info@net-italia.it
USA	info@net-usa-inc.com
Asia	info@net-japan.com

Phone

Europe	+49 8806 92 34-0
France	+33 450 452 292
Italy	+39 305 237 163
USA	+1 219 934 9042
Asia	+81 454 781 020

Fax

Europe	+49 8806 92 34-77
Italy	+39 305 237 163
USA	+1 219 934 9047
Asia	+81 45 476 2423

In case of an RMA, you must first contact NET and obtain an RMA Number before sending the product to us. We are not responsible for any problems caused by not following the RMA procedure.

IMPRINT

NET New Electronic Technology GmbH

Address:
Lerchenberg 7
D-86923 Finning
Germany

Contact:
Phone: +49-88 06-92 34-0
Fax: +49-88 06-92 34-77
www.net-gmbh.com
E-mail: info@net-gmbh.com

Managing Directors:
Jean-Pierre Heinrichs
Thomas Huber
Uwe Post

VAT- ID: DE 811948278
Register Court: Augsburg HRB 18494