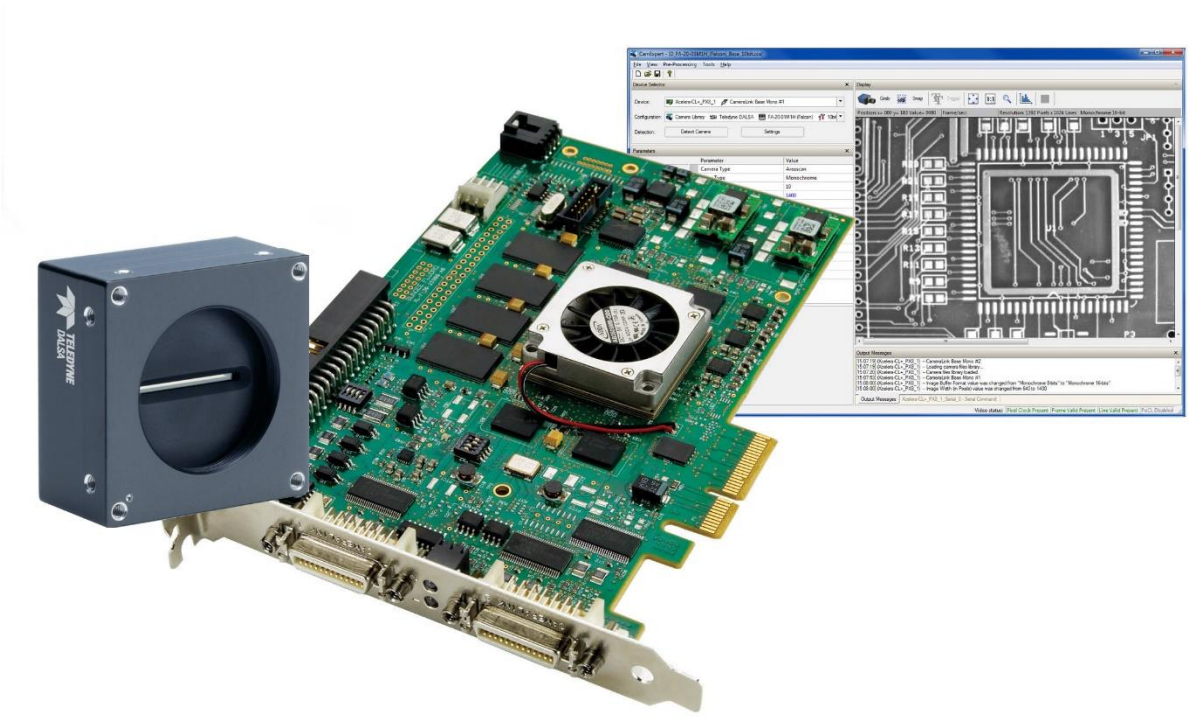
Sapera LT™ 8.7

Getting Started Manual for Frame Grabbers

sensors | cameras | frame grabbers | processors | **software** | vision solutions



P/N: OC-SAPM-INTR0

www.teledynedalsa.com

TELEDYNE DALSA
Everywhereyoulook™

Notice

Document Date: 2022-08-12

About Teledyne DALSA, a business unit of Teledyne Digital Imaging Inc.

Teledyne DALSA is an international leader in high-performance digital imaging and semiconductor technologies that designs, develops, manufactures, and markets digital imaging products and solutions, in addition to providing semiconductor products and services.

Teledyne DALSA is part of Teledyne Imaging, a group of leading-edge technology companies aligned under the Teledyne umbrella. With unrivalled expertise across the spectrum and decades of experience, the collective offers world-leading capabilities in sensing, signal generation and processing, enabling customers to succeed across a vast range of applications and industries.

Contents

Introduction to Sapera LT

Sapera™ LT is a software API for controlling image acquisition devices such as frame grabbers and camera. Sapera LT libraries support Teledyne DALSA cameras and frame grabbers as well as hundreds of third-party camera models across all common interface formats like GigE Vision®, Camera Link®, USB3 Vision®, as well as emerging new image acquisition standards such as CLHS.

The manual introduces the Sapera LT API and is designed to help programmers with installation, hardware setup and validation.

Sapera LT Licensing

Sapera LT is available free of charge, both SDK and runtime versions, when used with Teledyne DALSA frame grabbers or camera products.

The Sapera LT APIs

Sapera LT includes everything you need to acquire and display images, using one of its 3 application programming interfaces (API):

- Sapera LT++ classes (based on C++ language)
- Sapera LT .NET classes (based on .NET languages)
- Sapera LT Standard API (based on C language)

It is targeted at developers that have their own image processing libraries and want to interface those libraries to a Sapera LT compatible device. Sapera LT includes tools such as CamExpert to speed up application development.

Hardware independent classes allow one application to control different Teledyne DALSA devices through the same API. It also guarantees seamless migration to any future Teledyne DALSA hardware product supported by Sapera LT.

Sapera Tools and Utilities

The Sapera LT SDK and runtime installations include the following a set of tools and utilities:

- Sapera Explorer: quick access to all tools, demos, examples, and source code
- CamExpert: acquisition and device configuration, including image display
- Sapera Monitor: real-time event viewing of applications
- Sapera Log Viewer: error and other message log for applications and hardware
- Sapera Configuration: memory resource configuration and frame grabber configuration
- PCI Diagnostic: frame grabber hardware debugging
- Sapera Color Calibration: color correction utility.

Sapera LT Manual Descriptions

Sapera LT is supported by the following manuals.

API/Topic	Title	Description
General	**Sapera LT Getting Started Manual for Frame Grabbers**	Provides a general overview of the Sapera LT APIs with frame grabbers, possible hardware configurations and a quick overview of Sapera programming.
General	**Sapera LT Getting Started Manual for GigE Vision Cameras & 3D Sensors**	Provides a general overview of the Sapera LT APIs with GigE Vision cameras and 3D sensors, a quick overview of Sapera programming as well as troubleshooting guidelines.
General	**Sapera LT Getting Started Manual for USB3 Vision Cameras**	Provides a general overview of the Sapera LT APIs with USB3 Vision cameras, a quick overview of Sapera programming as well as troubleshooting guidelines.
General	**Sapera LT User's Manual**	Introduces Sapera LT ++ API and Sapera .NET API programming procedures, including sample code for typical operations in C++ and C#.
GigE Vision	**Network Imaging Package for Sapera LT Optimization Guide**	Network setup and optimization guide for GigE Vision cameras.
.NET	**Sapera LT .NET Programmer's Manual**	Provides a complete reference of the Sapera .NET Framework for Visual Studio. Sapera .NET reflects the underlying low-level Sapera LT architecture.
C++	**Sapera LT ++ Programmer's Manual**	Provides a complete reference of all the Sapera LT ++ classes. Sapera LT ++ is based on the C++ language.
C++	**Sapera LT Legacy Classes Reference Manual**	Describes the obsolete C++ classes that continue to be supported but have been replaced or retired.
C	**Sapera LT Acquisition Parameters Reference Manual**	Describes the Sapera LT low-level acquisition parameters and capabilities (based on the C language).

Supported Operating Systems

- Windows 10 (32-bit or 64-bit versions)
- Windows 11 (64-bit)

Supported Sapera LT Development Environments

- PCI-bus IBM PC or compatible with Pentium class or later processor
- C/C++ and .NET language compilers, for both 32-bit and 64-bit development:
  - Microsoft Visual Studio 2013
  - Microsoft Visual Studio 2015
  - Microsoft Visual Studio 2017
  - Microsoft Visual Studio 2019

Installation Types

When installing Sapera LT, you are provided with a choice of different setup types, each with its own dedicated installation program.

The full SDK (software development kit) installation provides access to all available Sapera LT functions, tools, demos, and utilities, such as CamExpert or Z-Expert (for 3D profile sensors), and the Sapera Network Imaging Package (required when using GigE cameras) or USB3 Vision Interface (for USB3 Vision cameras).

Installation Type	Notes
Sapera LT SDK (for developers)	Full installation of the software development kit, including all demos, tools, and utilities. Installation options allow you to install components for frame grabbers, GigE Vision cameras (includes the Sapera Network Imaging Package), 3D sensors or USB3 Vision cameras.
Sapera LT Runtime (32-bit + 64-bit)	Runtime installation for application deployment, including tools and utilities (includes the Sapera Network Imaging Package if required).
Sapera LT Runtime (WoW64*)	
Sapera LT Runtime + CamExpert (32-bit + 64-bit)	Runtime installation for application deployment, including tools and utilities, along with CamExpert (includes the Sapera Network Imaging Package if required).
Sapera LT Runtime + CamExpert (WoW64*)	
Sapera Camera SDK	Installation of Sapera Camera SDK for Gen CP CameraLink cameras. Feature control only.
Sapera Camera SDK Runtime	Runtime installation for application deployment of Sapera Camera SDK for GenCP CameraLink cameras. Feature control only.

* Use only for developing 32-bit applications to run on 64-bit OS. Do not use for native 64-bit development.

These executable files are available for download directly from the Teledyne DALSA website:

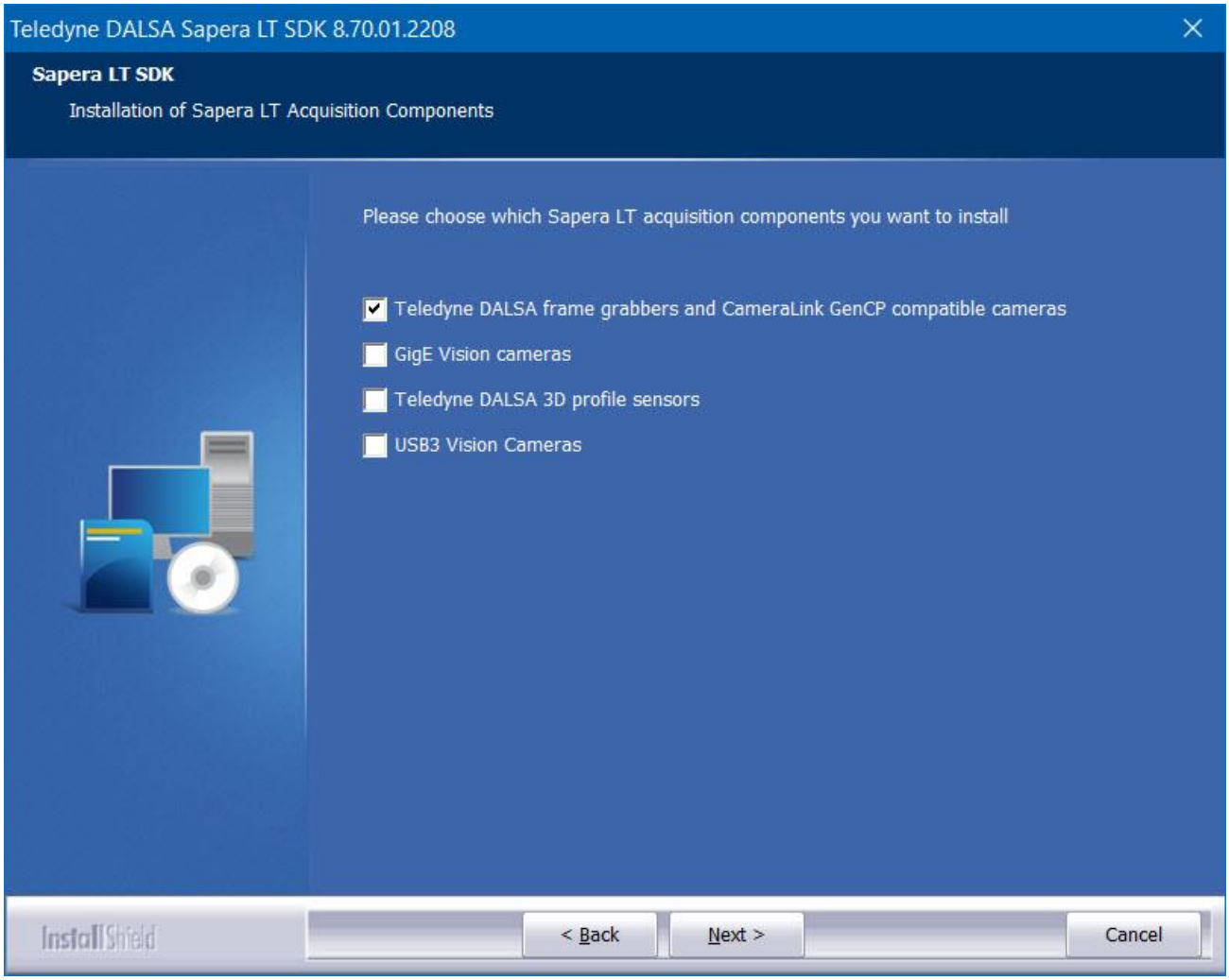
http://teledynedalsa.com/imaging/support/downloads/sdks/

The Sapera Camera SDK is also available for use when using Teledyne DALSA Camera Link cameras with third-party frame grabbers only. It is provided as part of the Teledyne DALSA camera installation.

Installing Sapera LT for use with frame grabbers and CameraLink GenCP compatible cameras

To install Sapera LT, it is strongly recommended that the installation be started locally instead of from a network location.

During the installation process, you are prompted to choose the Sapera LT acquisition components to install. Select **Teledyne DALSA frame grabbers and CameraLink GenCP compatible cameras**.



Upgrading Previous Versions of Sapera LT

Sapera LT 8.6 or Higher

When upgrading from Sapera LT version 8.6x, uninstalling is not required; the same acquisition component options (for frame grabbers, GigE Vision cameras, etc.) will be installed (persistent installation).

However, if you want to change the components to install, you must uninstall Sapera LT first.

To verify the acquisition components currently installed, check the Teledyne DALSA Sapera LT SDK entry on the **Start** menu under **Settings** > **Apps** > **Apps & features**. For example, if only the **Frame grabbers and CameraLink cameras** option was selected, the entry appears as follows:

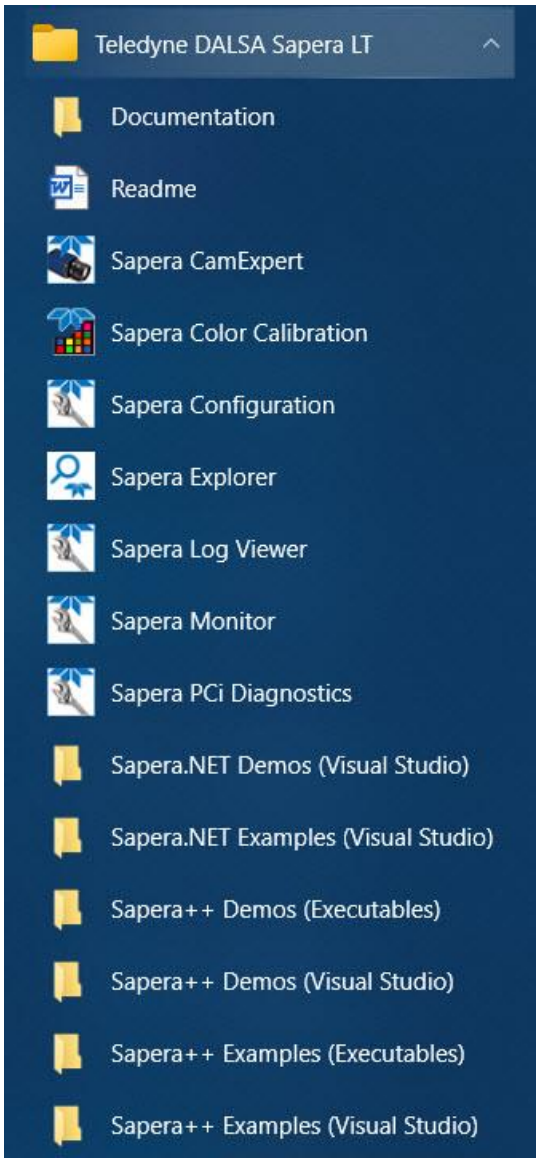
If more than one option was selected (for example, **GigE Vision cameras** and **Teledyne DALSA 3D profile sensors**), the entry appears as follows:

Sapera LT 8.5 and Lower

Uninstall any version Sapera LT 8.5 or older before installing Sapera LT 8.7. Use Windows **Settings** > **Apps** > **Apps & features** page.

Start Menu Shortcuts

You can access Sapera LT utilities, demos and examples from the **Start** menu, under **Teledyne DALSA Sapera LT**. The Network Configuration Tool will be found under **Teledyne DALSA**.



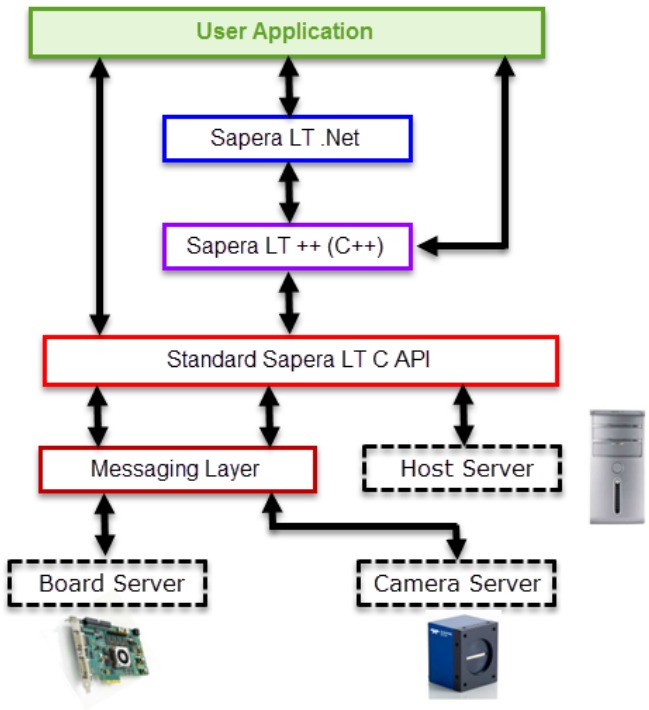
See Appendix: File Location for a complete description of directory contents.

Sapera LT Architecture Overview

The following section describes application architecture, related terms, and illustrates Sapera LT's library architecture.

Application Architecture

Whichever API is used (Sapera LT ++, Sapera LT .NET, or Standard C), the Sapera LT modular architecture allows applications to be distributed on different Sapera LT servers. Each server can run either on the host computer or on a Teledyne DALSA device. Sapera LT calls are routed to different servers via the Sapera LT messaging layer in a fashion completely independent of the underlying hardware.



What is a server?

A Sapera Server is an abstract representation of a physical device like a frame grabber, a camera, or a desktop PC. In general, a Teledyne DALSA board is a server. Some processing boards, however, may contain several servers; this is true when using multi-processor boards.

A server allows Sapera applications to interact with the server's resources. The server name consists of the product model name and an index. For example, the Xtium CLHS PX8 frame grabber has a server name "Xtium-CLHS_1". If more than one device of the same type is available, the index differentiates them.

A server may also have different resources available, identified by a resource index. For example, the Xtium_CLHS_1 server has two types of acquisition resource available to select, depending on the configuration required: Camera Link HS Mono or Camera Link RGB.

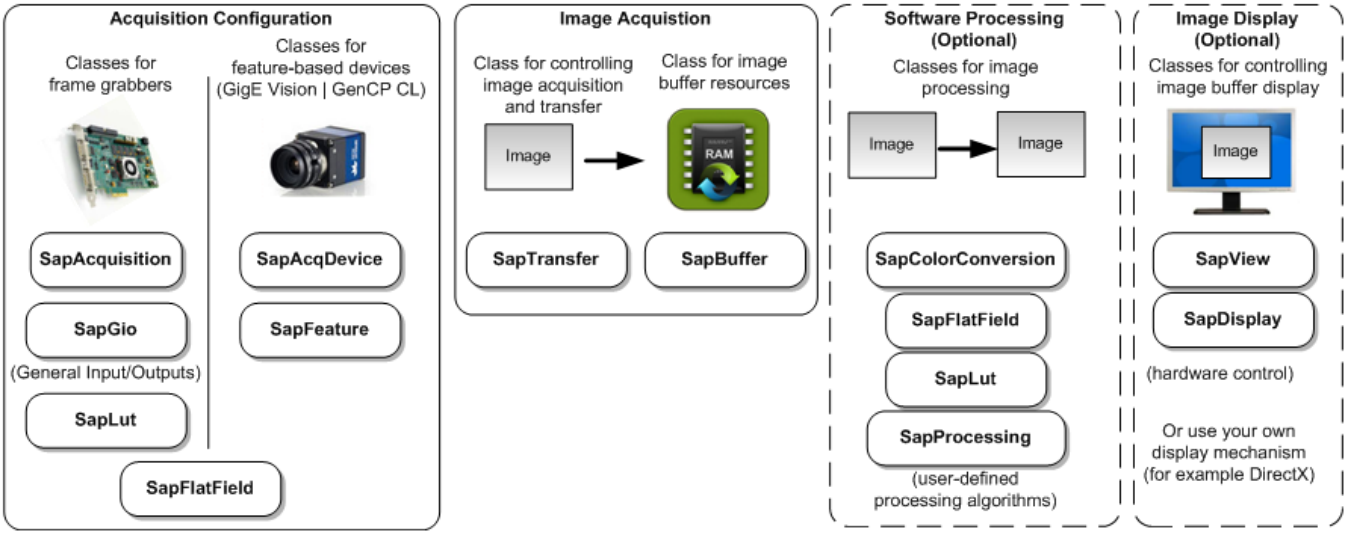
Library Architecture

The typical machine vision application requires configuration of acquisition resources, image capture and transfer to memory buffers. These image buffers can then be processed or displayed, analyzed, with results determining subsequent processes. Events can also be monitored to trigger appropriate responses. The Sapera LT library architecture is organized around these basic machine vision functional blocks.

Vision applications developed with Sapera LT are typically programmed in either the Sapera LT++ API (Application Programming Interface) or the Sapera LT .NET API. In general, both APIs use similar classes and naming conventions.

Sapera LT Standard C API is available for programmers who prefer working with the underlying Sapera LT C layer or who are maintaining legacy code. It provides access most of the same functionality as the higher level, object-oriented programming C++ and .NET APIs (for example, it does not support flat field calibration and software correction). For more information refer to the Sapera LT Acquisition Parameters Reference Manual.

The following block diagram, while not exhaustive of all the classes available in Sapera LT, illustrates the major functional blocks with the corresponding classes.



For the complete reference to the Sapera LT APIs refer to the Sapera LT ++ Programmer's Manual or Sapera LT .NET Programmer's Manual.

In addition, the Sapera LT User's Manual provides explanations and multiple code snippets, in both C++ and .NET languages, for typical application operations.

It is always recommended to use the source code provided with the demos and examples as both a learning tool and a starting point for your applications. For a complete list and description of the demos and examples included with Sapera LT see Demos and Examples.

Configuration Files

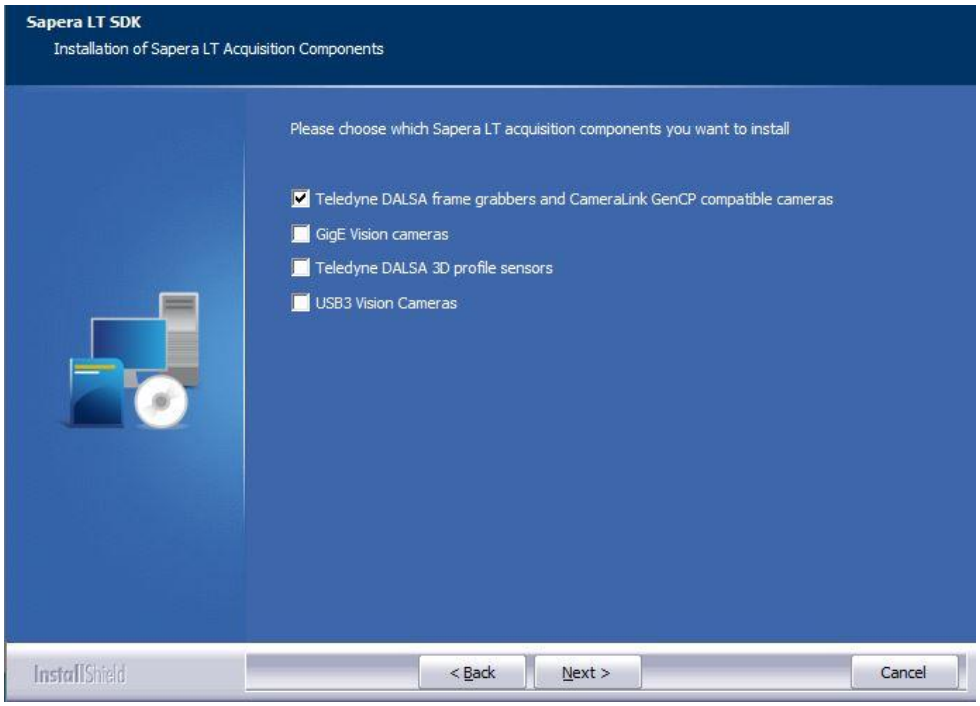
Sapera LT applications need to load the appropriate acquisition configuration file before acquiring images from a camera. The camera configuration file has the extension .ccf. For applications that use both a frame grabber and camera, both devices' parameter settings are saved in this file.

CamExpert, the camera interfacing tool provided with Sapera LT, supplies several predefined camera files (available in the Sapera\CamFiles directory with the .cca extension) for many cameras, both digital and analog. Use CamExpert to generate a .ccf file for cameras for which there are currently no available .cca files, or to modify a distributed .ccf file with parameters as required by your imaging project.

The SapAcquisition and SapAcqDevice constructors, for frame grabbers and GeniCam-compliant cameras respectively, have prototypes that use .ccf files. For cameras, if no .ccf file is available, the camera default parameters are used.

Quick Start Guide

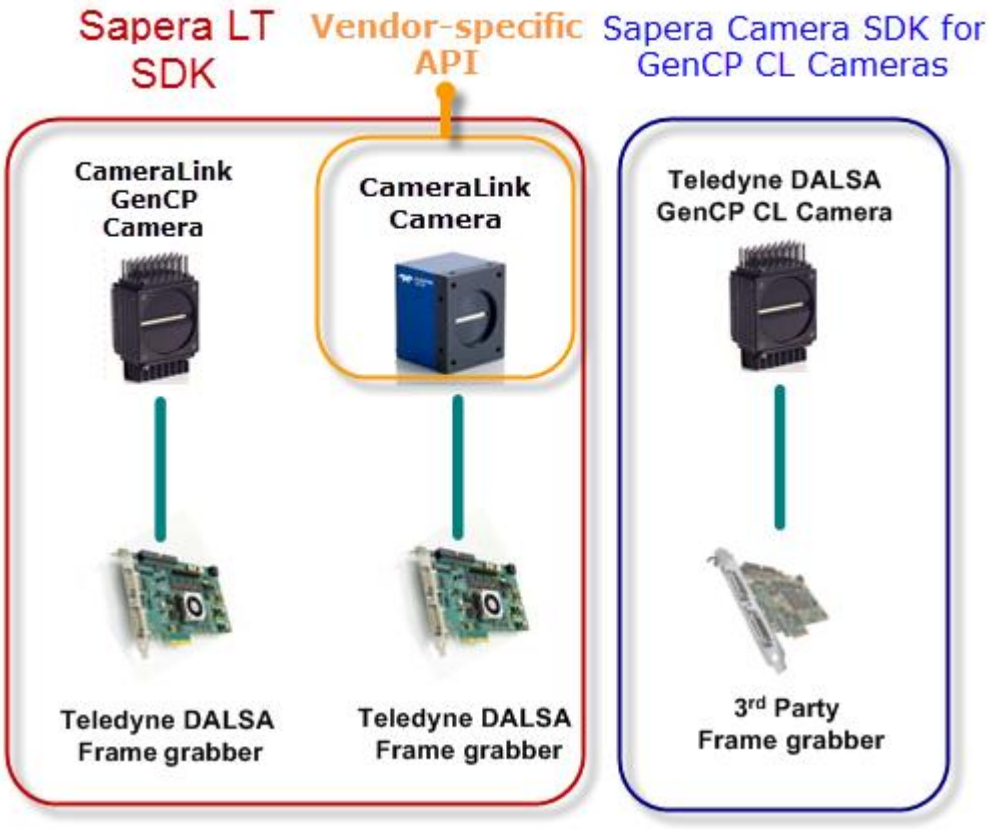
For Teledyne DALSA frame grabbers and Camera Link cameras, during installation, choose the option to install Sapera LT for **Teledyne DALSA frame grabbers and Camera Link GenCP compatible cameras** (select other components as required).



Note that if you try to install Sapera LT from a network location you will not be able to install the Sapera Network Imaging Package; copy the installation executable locally.

In general, for frame grabber applications there are 3 possible application configurations:

- **GenCP Protocol CameraLink camera** with **Teledyne DALSA frame grabber**
- **ASCII text-based (or other) CameraLink camera** with **Teledyne DALSA frame grabber**
- **Teledyne DALSA GenCP CL Camera** with **3rd-party frame grabber** (use the Sapera Camera SDK for GenCP CL Cameras in this scenario instead of the full Sapera LT installation: this provides access to the camera features, but does not provide acquisition since this must be handled by the frame grabber API)



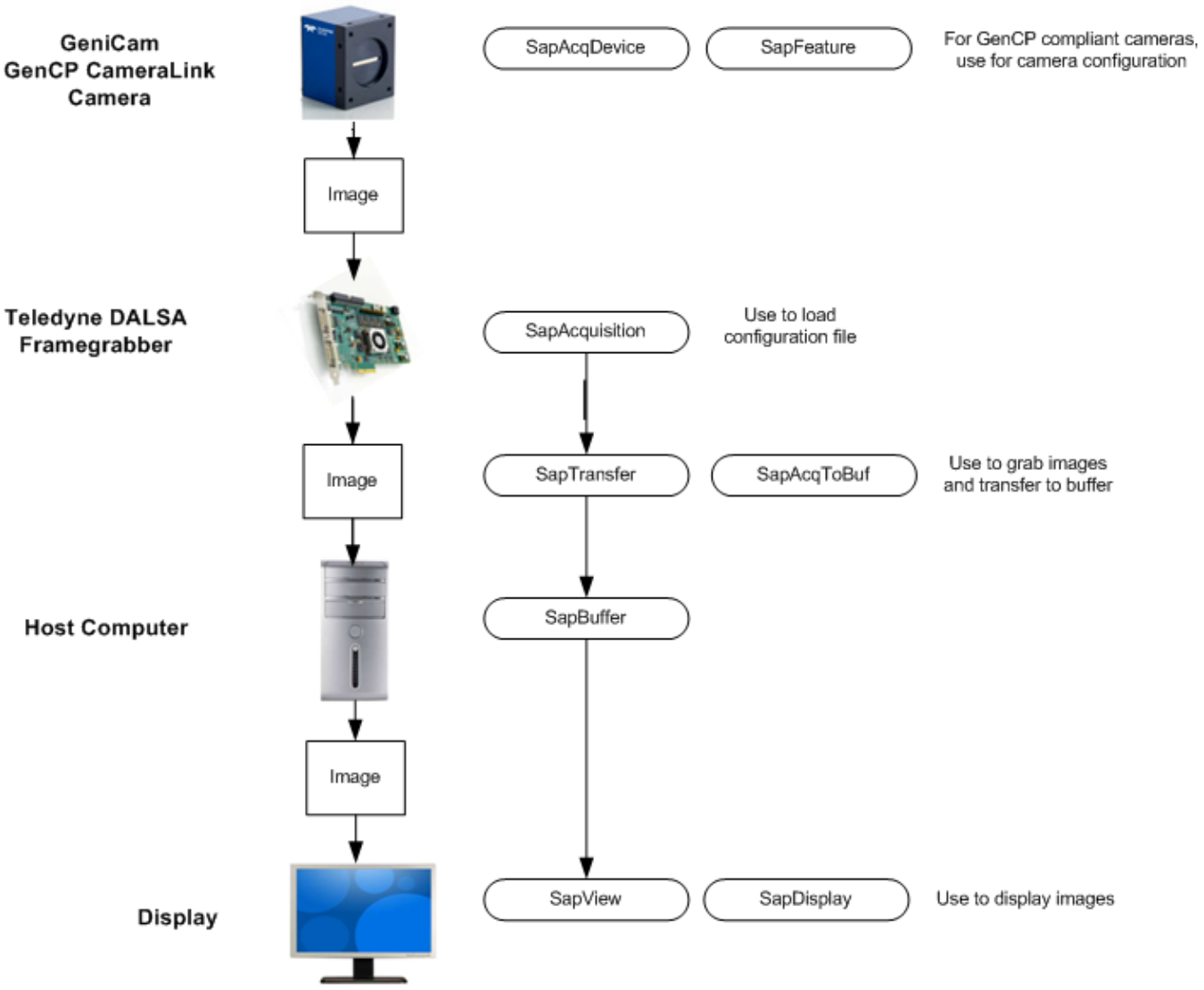
Currently, Sapera LT supports GenICam GenCP CL and GigE Vision standards (including all mandatory feature requirements).

	When using CameraLink cameras it is important to verify if the camera complies with the GenCP protocol or requires the vendor-specific API. For more information on the Sapera LT implementation of GenICam, refer to the either of the Sapera LT ++ or .NET Programming Manuals.

Using Sapera LT with a Teledyne DALSA GenCP CameraLink Camera and Frame Grabber

Sapera LT Classes for GenCP CL Cameras and Frame Grabber Applications

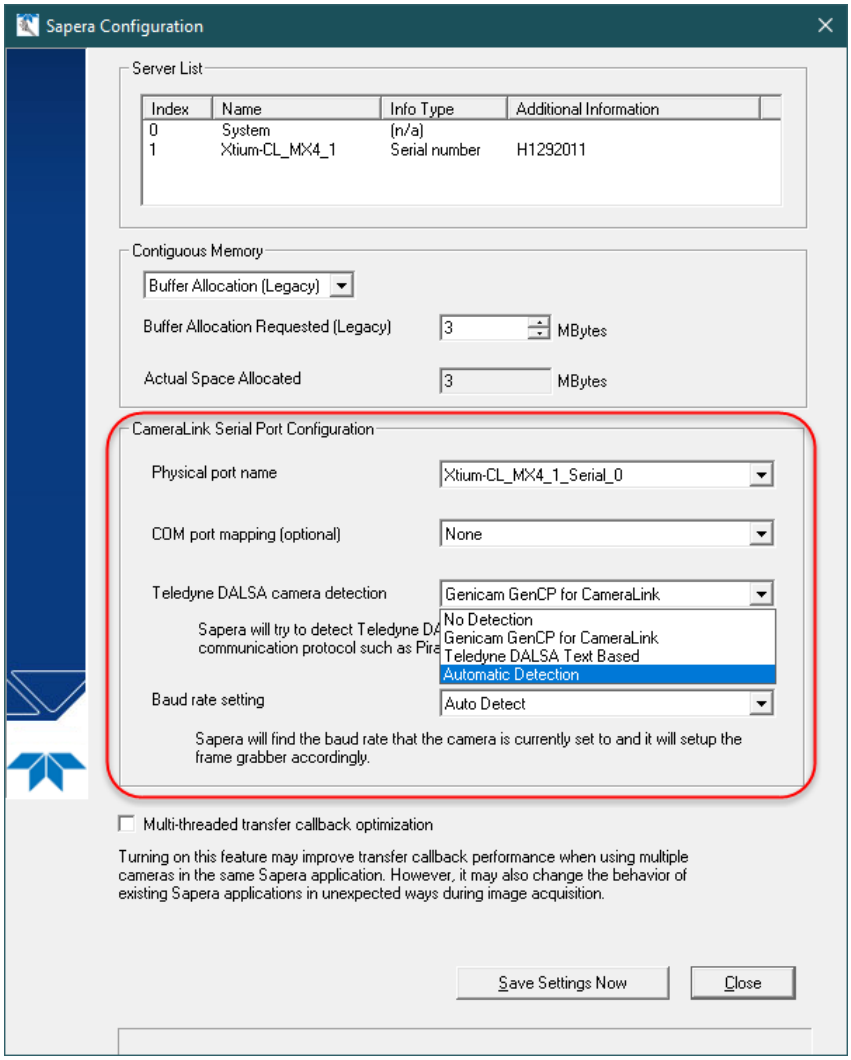
The following diagram represents a typical application flow showing the Sapera LT object types associated with each component or stage:



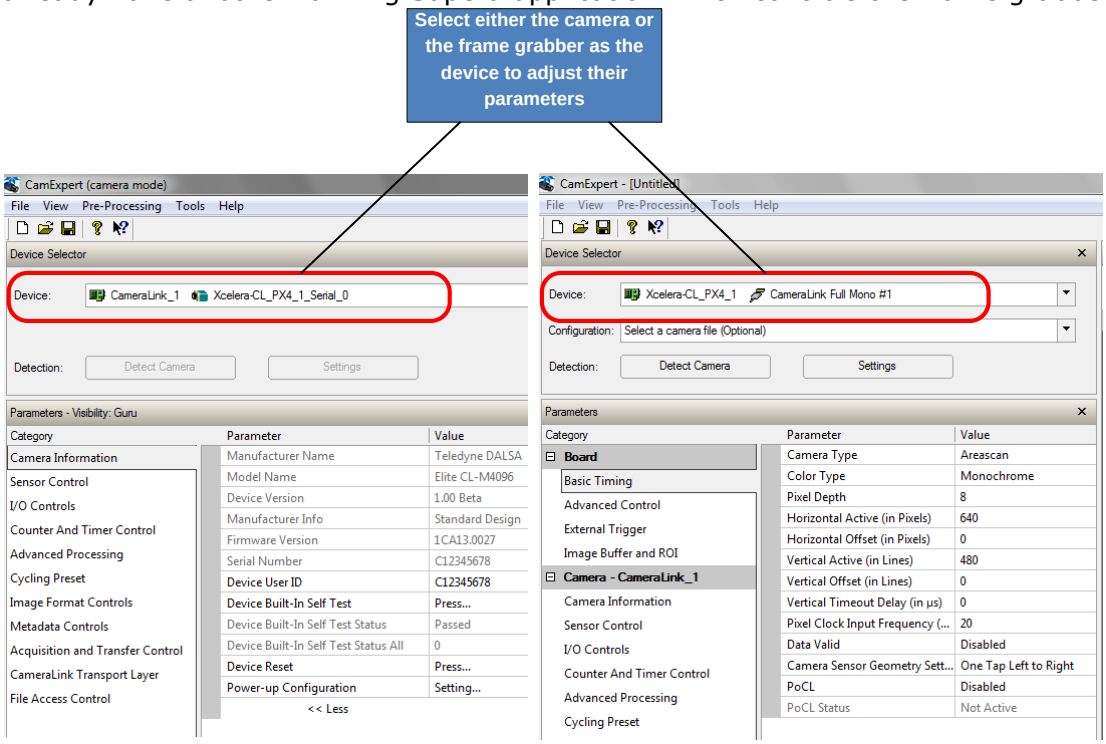
Teledyne DALSA CameraLink cameras either support the GenCP standard or can be programmed using ASCII commands on CameraLink serial ports; refer to the camera documentation to determine which method the camera supports.

To configure Teledyne DALSA GenCP CameraLink Cameras

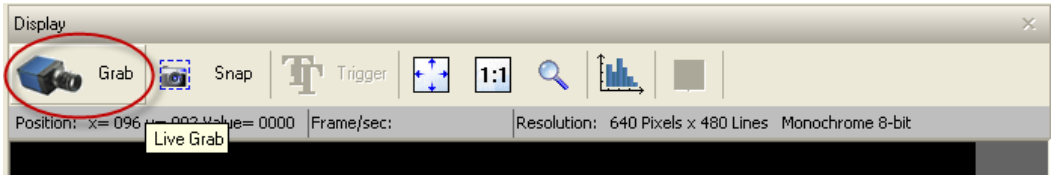
1. Run the Sapera Configuration utility, select the frame grabber serial port connected to the camera, and set the **Teledyne DALSA camera detection** parameter to one of the three following options:
    - **GenICam GenCP for CameraLink**: CamExpert will try to detect Teledyne DALSA cameras that support the GenCP communication protocol.
    - **Teledyne DALSA Text Based**: CamExpert will try to detect Teledyne DALSA cameras that support the Teledyne DALSA ASCII-based communication protocol, also known as Three Letter Commands (TLC).
    - **Automatic Detection**: CamExpert will try to detect Teledyne DALSA cameras on this COM port using both GenCP and text-based protocols.



2. Start CamExpert. In the **Device Selector** pane, select either the appropriate frame grabber port or the camera. If you select the former, then you can easily switch between frame grabber and camera parameters, as the two are shown together in the CamExpert GUI. If you select the latter, then only camera parameters are available. This can be useful if you already have another running Sapera application which controls the frame grabber only.



3. Modify the camera and frame grabber parameter settings as required and test the image acquisition by clicking **Grab**.



4. Save the frame grabber configuration to a new .ccf file.

Using Sapera LT with an ASCII Text-based (or other) CameraLink Camera and Teledyne DALSA Frame grabber

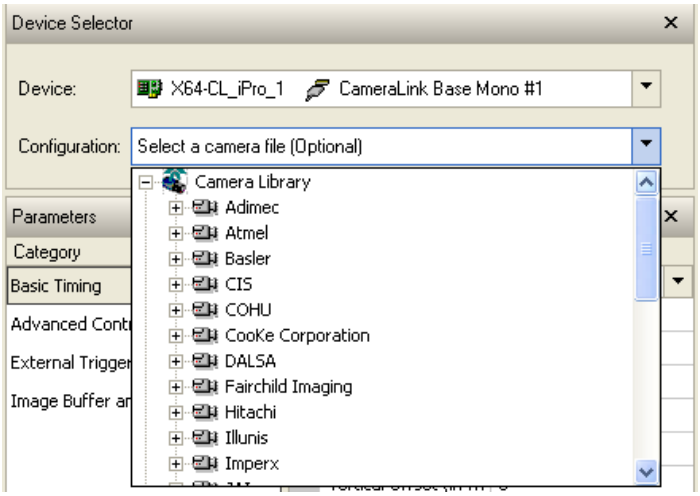
The Teledyne DALSA frame grabber requires a camera configuration file (*.ccf) to program the frame grabber to match the characteristics of the ASCII text-based (non-GenCP) CameraLink Teledyne DALSA or 3rd-party camera (for example, cameras that use three letter serial port commands). Teledyne DALSA provides a number of .cca files for popular cameras.

To configure the camera parameters, refer to the camera user documentation.

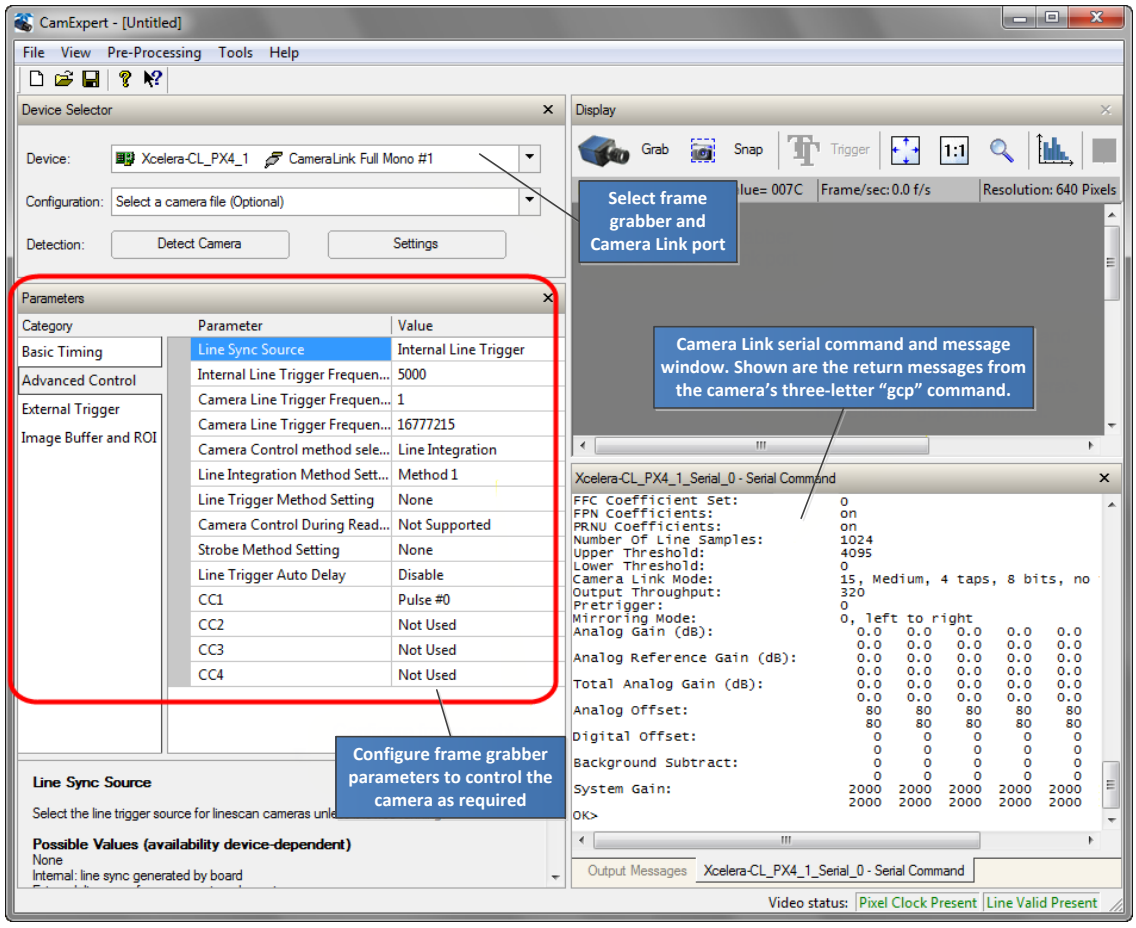
For certain 3rd party applications, it may be necessary to map frame grabber serial ports to Windows COM ports. The Sapera Configuration utility can be used to do so; refer to the Serial Port Mapping section.

To configure cameras that support ASCII commands only

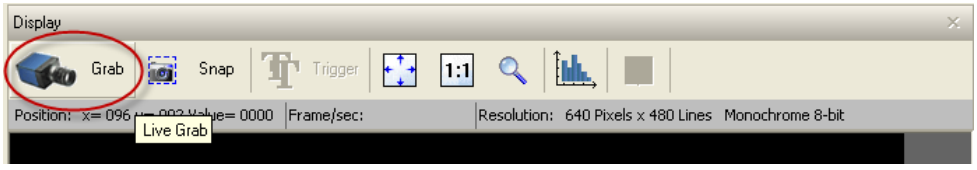
1. Start CamExpert. In the **Device Selector** pane, select the required frame grabber, so that you can adjust its parameters.
2. If available, select the camera configuration file for the camera.



3. Use the **CameraLink Serial Command** tab (next to the **Output Messages** tab) to send ASCII commands to the camera, so that you can adjust its parameters.



4. After modifying the camera and frame grabber parameter settings as required, test the image acquisition by clicking the **Grab** button.

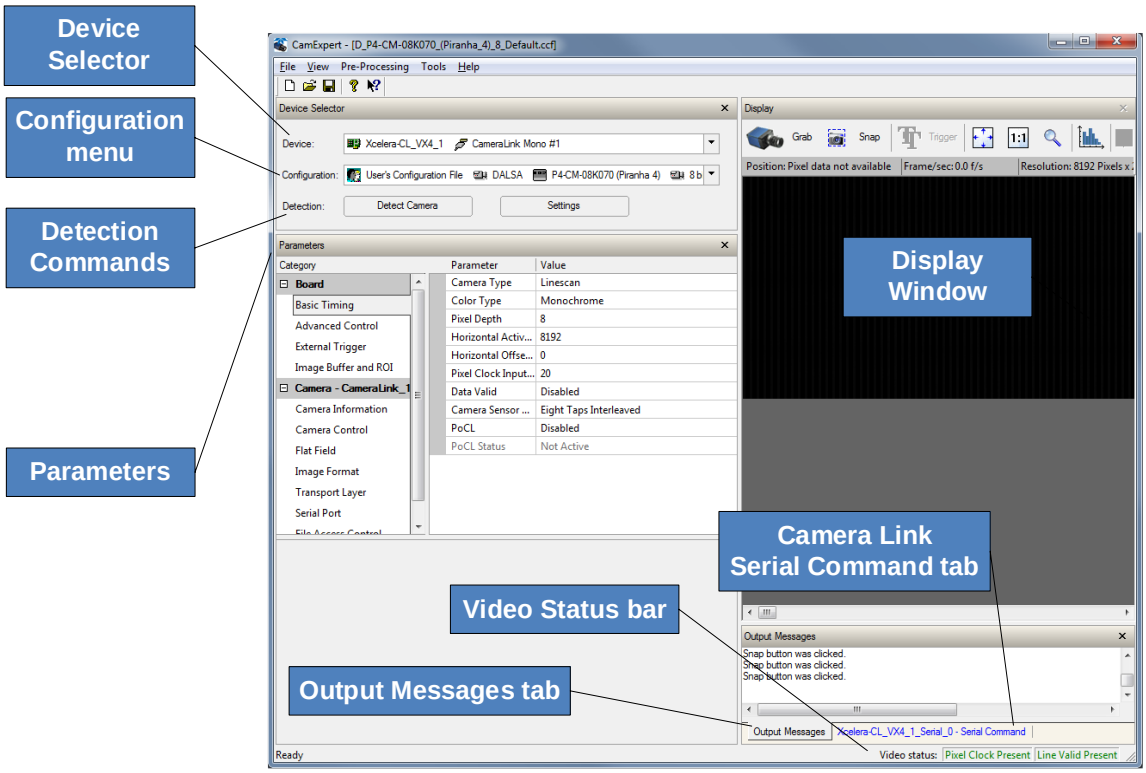


5. Save the frame grabber configuration to a new .ccf file.

Using the CamExpert Tool

The CamExpert application uses various panels to simplify choosing and configuring camera files or acquisition parameters for the installed devices. CamExpert:

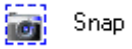
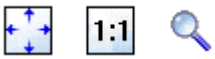
- Supports all Teledyne DALSA hardware currently supported by Sapera LT.
- Creates and modifies camera configuration (.ccf files).
- Supports Teledyne DALSA GigE Vision cameras such as the Genie series by presenting the camera features as defined by the camera's XML file.
- Groups acquisition parameters into related categories for easier access to any specific parameter.
- Enables intelligent editing of video timings through a locking mechanism that allows explicit modification of some values and automatic recalculation of the remaining ones.
- Provides live acquisition display window which allows immediate verification of timing or control parameters without the need to run a separate acquisition program.



The CamExpert window is divided into panels.

- **Device Selector**: The Device menu allows you to view and select from any installed Sapera acquisition device. After a device is selected, CamExpert only presents acquisition parameters applicable to that device.
  The Configuration menu allows selecting any camera file that is included with the Sapera installation. Only camera files supported by the selected acquisition device are displayed. When there is more than one acquisition server, such as monochrome and RGB, selecting an inappropriate camera file will produce a message prompting you to select the correct acquisition server.
  The Detection command buttons allow you to detect cameras that comply with the GenICam GenCP protocol or Teledyne DALSA cameras that use the three-letter protocol. Clicking the Settings button allows you to set the camera detection communication settings.

- **Parameters**: Allows viewing or changing all acquisition parameters supported by the acquisition device. CamExpert displays parameters only if those parameters are supported by the installed device. This avoids confusion by eliminating parameter choices when they do not apply to the hardware in use.
- **Display**: Provides a live or single frame acquisition display. Frame buffer parameters are shown in an information bar above the image window. The Display pane includes CamExpert control buttons.

	**Acquisition control button:** Click once to start live grab, click again to stop.
	**Single frame grab:** Click to acquire one frame from device.
	**Software trigger button:** With the I/O control parameters set to Trigger Enabled / Software Trigger type, click to send a single software trigger command.
	**CamExpert display controls:** (these do not modify the frame buffer data) Stretch image to fit, set image display to original size, or zoom the image to any size and ratio.
	**Histogram / Profile tool:** Select to view a histogram or line/column profile during live acquisition.

- **Output Messages**: Displays messages from CamExpert or the device driver.
- **Camera Serial Link Command tab**: Use this to send ASCII commands from CamExpert to Teledyne DALSA CameraLink cameras.
- **Video Signal Status bar**: Located on the lower right of the CamExpert window, displays color coded camera signal status information is displayed. These are in green for valid signals detected, and in red for missing or incorrect signals. Video status items may differ with different devices.

Additional Information

For additional information about Sapera acquisition parameters, refer to the Sapera Acquisition Parameters Reference Manual.

See either the corresponding device user's manual or search within this manual for limitations applicable to specific Teledyne DALSA hardware.

Sapera LT Utilities

Sapera LT includes the utilities that can be used to monitor Sapera LT hardware and software events, error messages, as well as frame grabber configuration and diagnostics.

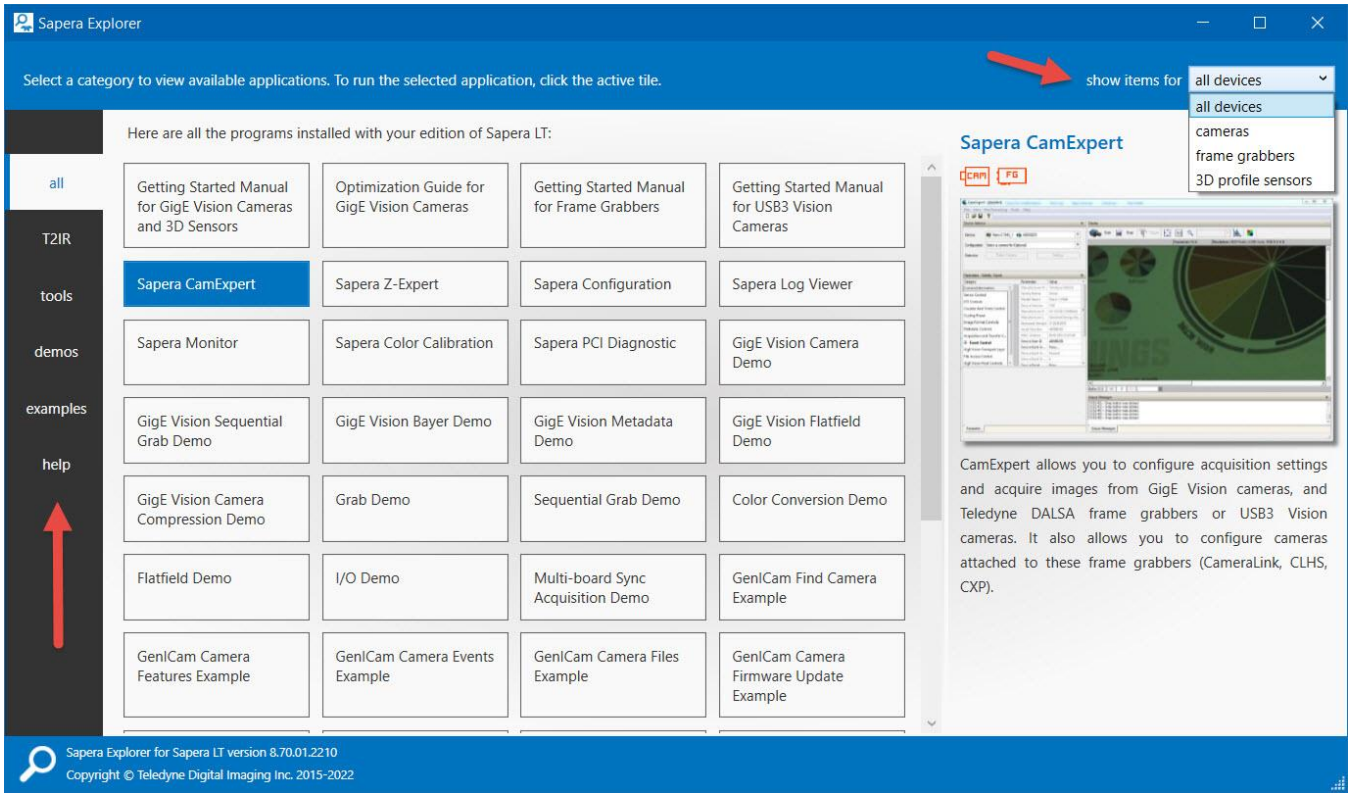
Sapera Explorer	Provides quick access to the Sapera LT tools, documentation, demos, examples, and source code.
Sapera Monitor	Monitors user selected events generated by a Sapera LT application.
Sapera Log Viewer	Displays error and other messages generated by Sapera LT applications and Teledyne DALSA hardware.
Sapera Configuration	Configuration utility for Teledyne DALSA frame grabbers.
Sapera PCI Diagnostics	Low-level diagnostic utility for Teledyne DALSA frame grabbers.
Sapera Color Calibration	Color correction utility for cameras (if supported).

Sapera Explorer

The Sapera Explorer application provides quick access to the Sapera LT tools, documentation, demos, examples, and source code.

To open Sapera Explorer

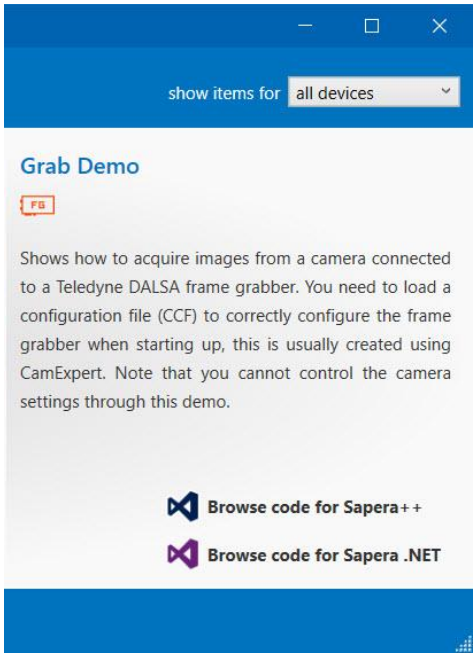
- On Windows **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera Explorer**, or
- Use the desktop shortcut that is typically created upon installation of Sapera LT.



Usage tips

- Select a category on the left to filter applications or documents.
- Use the **show items for** list to filter items according to type of device.
- Click a tile to display, on the right pane, a description of the selected item.
  The CAM, FG and 3D icons indicate the type of device to which the selected item applies (cameras, frame grabbers, 3D sensors).

- Click an active tile to open the application or document.
- For demos and examples, click **Browse code for Sapera ++** or **Browse code for Sapera .NET** to open the source code directory.


show items for
all devices
Grab Demo
FG
Shows how to acquire images from a camera connected to a Teledyne DALSA frame grabber. You need to load a configuration file (CCF) to correctly configure the frame grabber when starting up, this is usually created using CamExpert. Note that you cannot control the camera settings through this demo.
Browse code for Sapera++
Browse code for Sapera .NET

Sapera Monitor

As part of the Trigger-to-Image-Reliability (T2IR) framework, the Sapera Monitor tool allows users to view the acquisition and transfer events generated by an acquisition device in real time. Sapera Monitor can run concurrently with CamExpert or with your own application, and therefore can be useful for debugging applications and identifying problems without having to code event handlers.

Older driver versions of Teledyne DALSA devices may not support Sapera Monitor. Check the Teledyne DALSA website for updated drivers for your device that support Sapera Monitor.

Before starting Sapera Monitor, you must start a Sapera application that uses the Teledyne DALSA GigE devices you want to monitor (such as CamExpert).

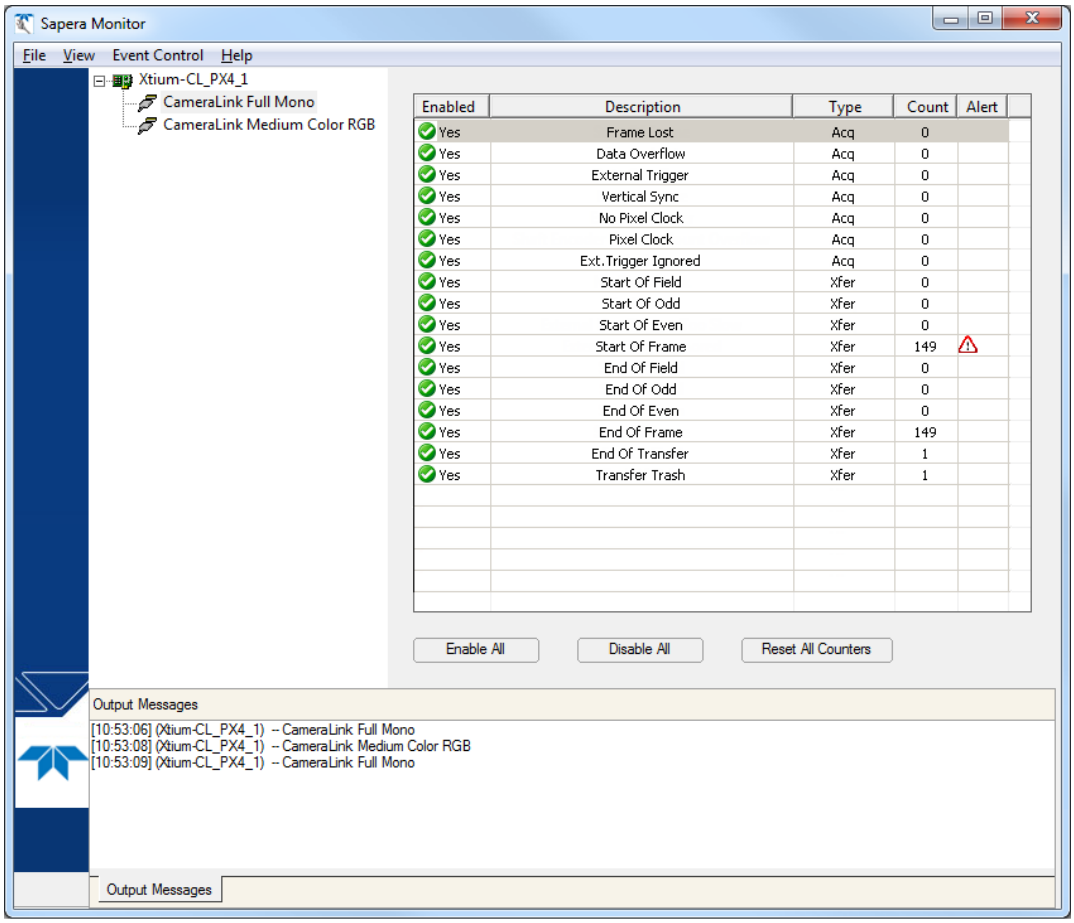
To open Sapera Monitor

- From the **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera Monitor**.

The Sapera Monitor Window

The Sapera Monitor Window is divided into three panes:

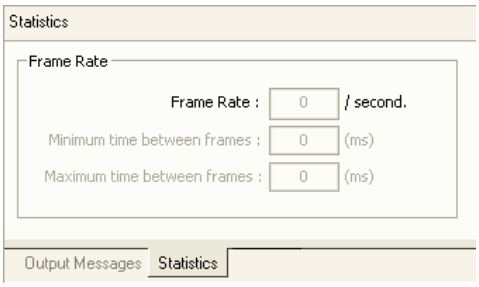
- **Device directory tree**: displays the available acquisition devices to monitor.
- **Event table**: displays the available events to monitor for the selected device.
- **Output Messages**: displays the messages generated by the selected monitored events.



Statistics Tab

Clicking on the **Statistics** tab displays various real-time acquisition statistics, such as the Frame Rate.

Note that different devices can support different statistics and not all devices support all statistics. In addition, these real-time acquisition statistics are not included in generated reports. Thus, **depending on the selected device, the Statistics tab may not be available**.



The Sapera Monitor Menu Commands

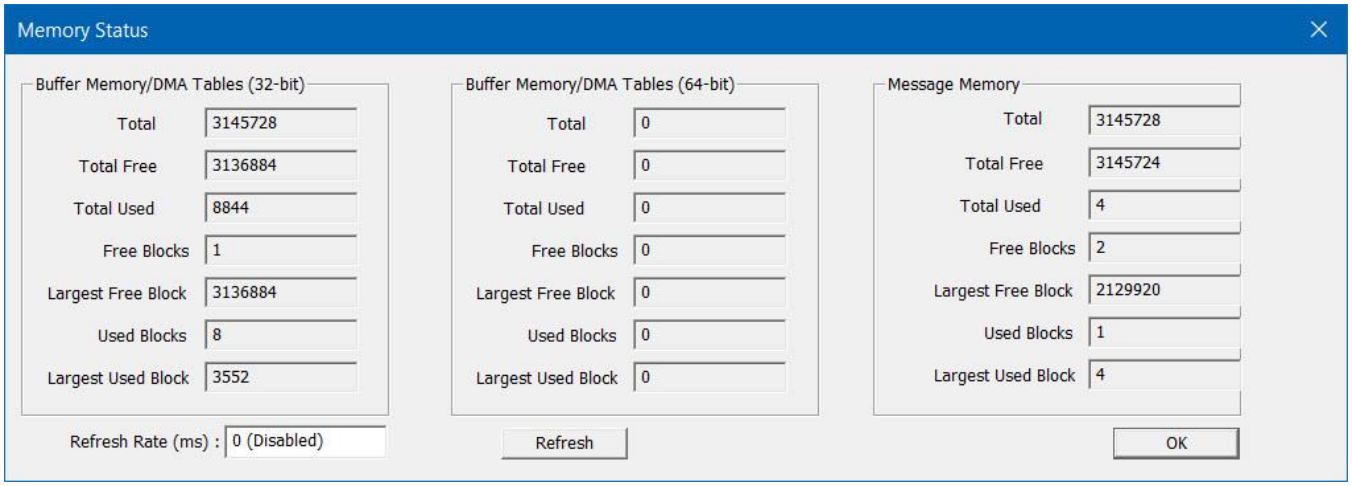
The Sapera Monitor menu provides access to **File**, **View**, and **Event Control** commands.

File Menu Commands

- **Generate Report.** Generates a text file report that includes all event settings and messages included in the current **Output Messages** pane.
- **Clear Log Information.** Clears the current **Output Messages** pane.

View Menu Commands

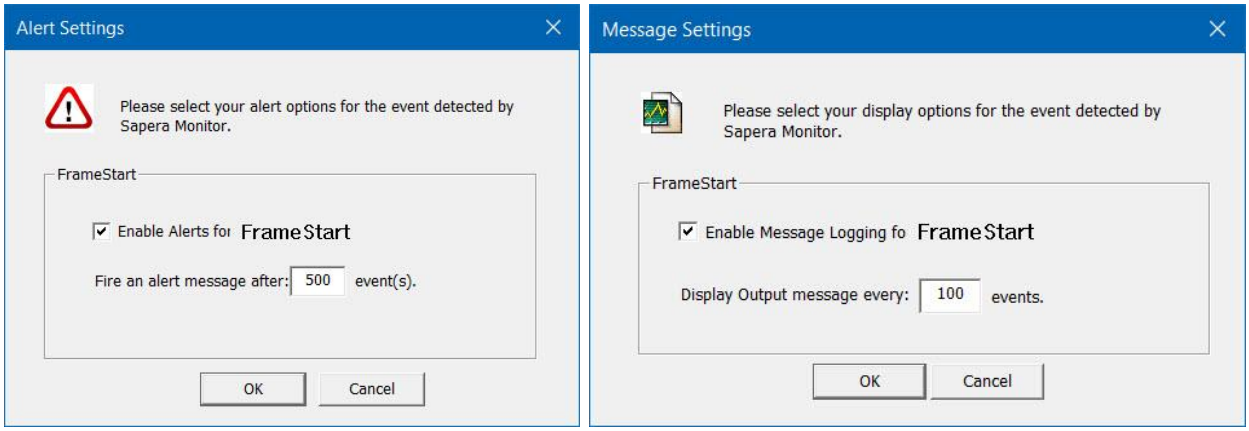
- **Always on Top.** Displays the Sapera Monitor on top of any other windows that may be visible on the desktop.
- **Select Events.** Opens the Sapera Monitor Events Display Settings dialog which allows you to specify the events to display in the Event table.
- **View Memory Status**. Opens a Memory Status dialog with memory usage information.



Event Control Menu Commands

Note that the commands are also available by right-clicking on an event, which opens a shortcut menu.

- **Enable** and **Disable.** Starts or stops the currently selected event.
- **Reset event counter.** Returns the event counter to zero for the currently selected event.
- **Alert Settings.** Opens the Alert Settings dialog, where you can enable or disable alerts for the selected event and set the number of events required to generate an alert.
- **Message Settings.** Opens the Message Settings dialog, where you can enable message logging and set the number of events required to generate a log message. The log messages appear in the **Output Messages** pane.



When an alert is generated, the Alert icon is displayed in the Alert column of the event.

In the **Output Messages** pane, messages appear in blue, while alert messages appear in red.

Using Sapera Monitor

To use the Sapera Monitor tool to monitor a device

- Open a Sapera application, such as CamExpert, that uses the device.
- Open the Sapera Monitor application.
- In the Sapera Monitor Device directory tree, select a device to monitor.
- In Event table, select the events to monitor.
  - Double-click on an event to toggle between the *Enabled* and *Disabled* state. Alternatively, you can use the **Enable All** and **Disable All** buttons to quickly enable or disable all events at once. Note that these commands do not change the alert and message settings of the events.
  - Use the **Event Control** menu or shortcut menu to modify the settings for an event.
- Start acquisition with the device.

Sapera Log Viewer

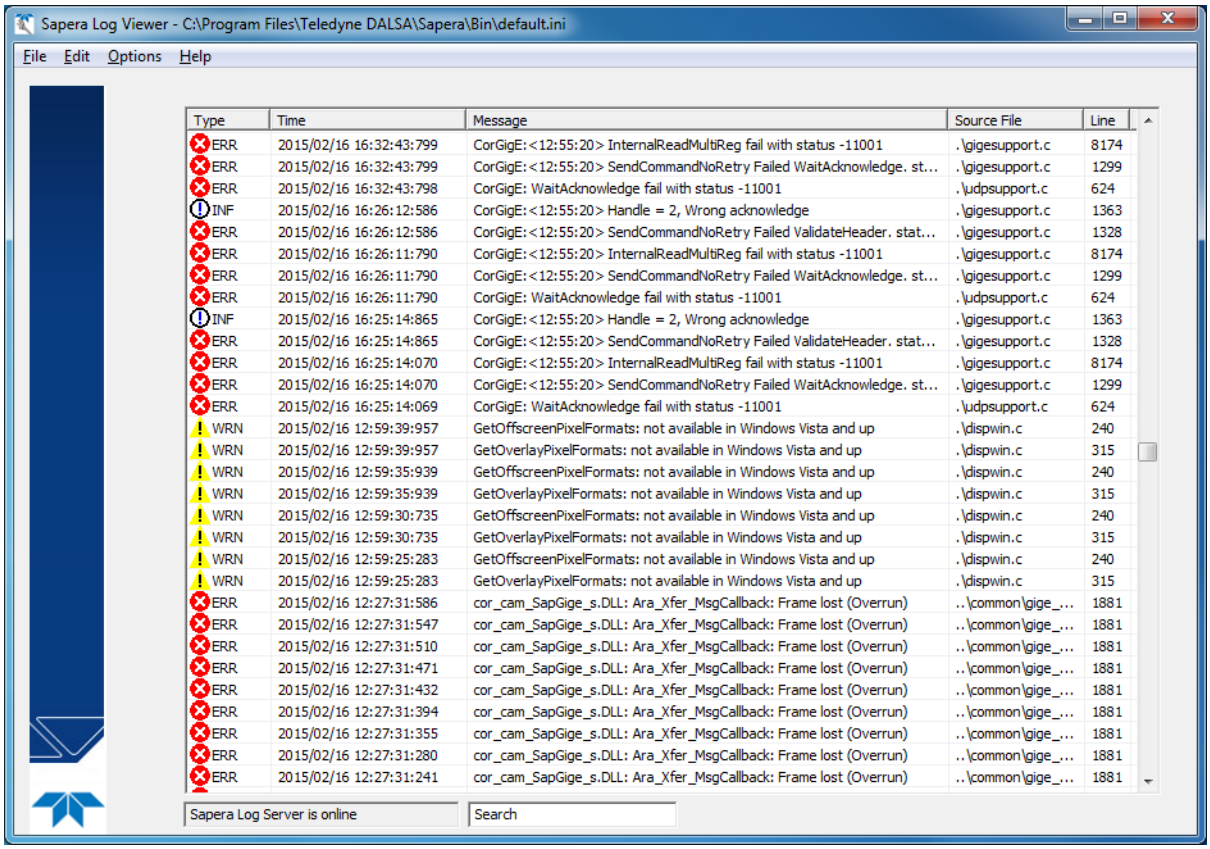
The Sapera Log Viewer utility program included with Sapera LT provides an easy way to view error and other types of messages generated by Sapera LT applications and Teledyne DALSA hardware, such as cameras and frame grabbers. Typically, the Sapera Log Viewer application is used by technical support to troubleshoot software and hardware problems.

During development it is recommended to start the Sapera Log Viewer before your application and then let it run so it can be referred to any time a detailed error description is required. However, errors are also stored by a low-level service (running in the background), even if the utility is not running. Therefore, it is possible to run it only when a problem occurs with your application.

Refer to the utility's online help for more information on using the Sapera Log Viewer.

To open Sapera Log Viewer

- From the **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera Log Viewer**.

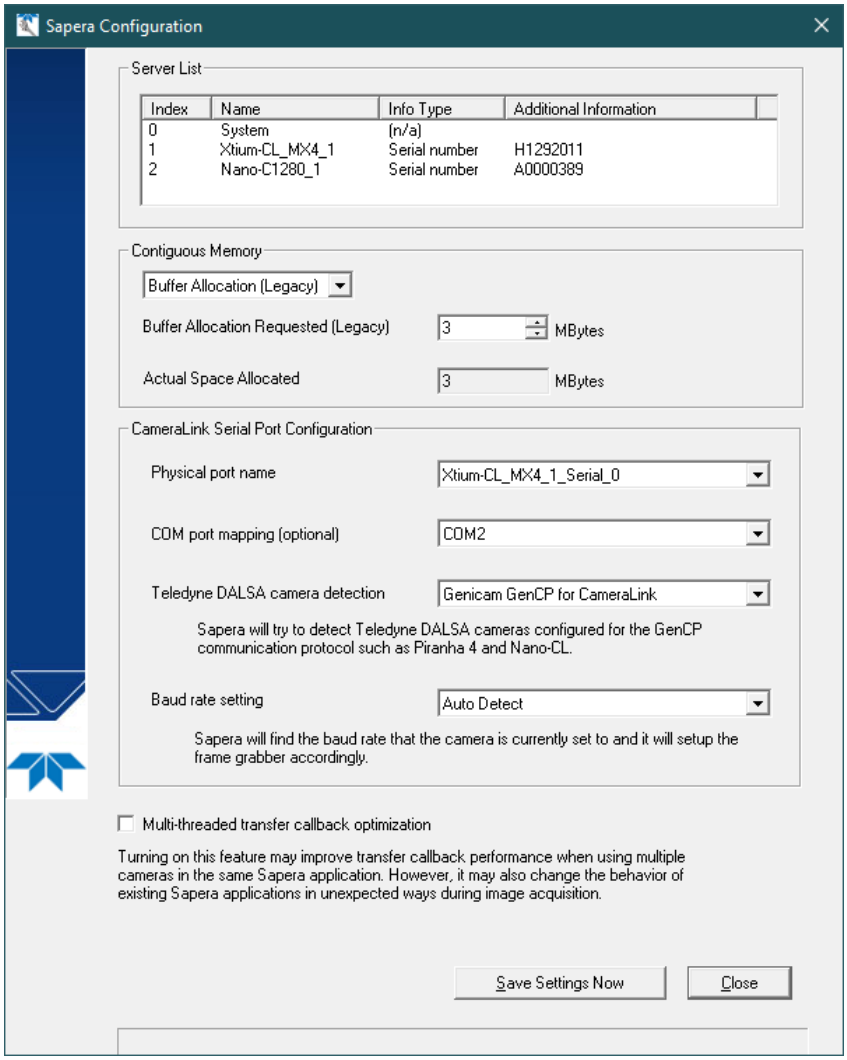


Sapera Configuration

The Sapera Configuration program displays all the Sapera LT-compatible devices present within your system, together with their respective serial numbers. It can also adjust the amount of contiguous memory to be allocated at boot time or map serial ports.

To open Sapera Configuration

- From the **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera Configuration**.



The *System* entry in the **Server List** represents the system server. It corresponds to the host machine (your computer) and is the only server that should always be present. The other servers correspond to the devices present within the system.

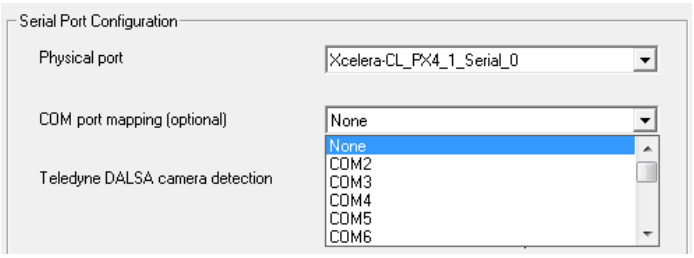
The **CameraLink Serial Port Configuration** section allows you to select the frame grabber's serial port, map COM ports, specify the type of Teledyne DALSA camera detection, as well as configure the baud rate.

Configuring Frame Grabber Board Serial Ports

Certain frame grabber boards provide an onboard serial port for direct camera control by the frame grabber. Refer to the specific board user manual for information on how to configure and use it.

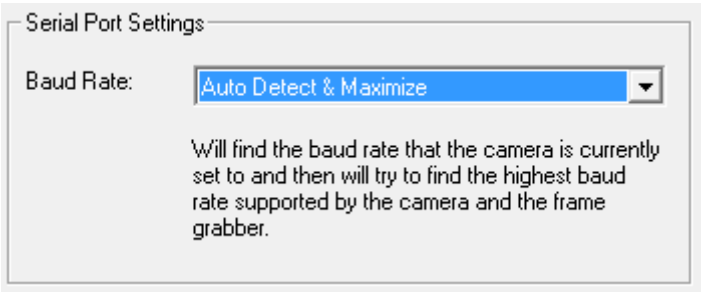
Serial Port Mapping

For certain third-party applications, it may be necessary to map frame grabber serial ports to Windows COM ports. To do so, run the Sapera Configuration utility, select the frame grabber serial port connected to the camera, and set the **CameraLink Serial Port Configuration** > **COM port mapping (optional)** parameter. For new mappings to take effect, a system reboot is generally required.

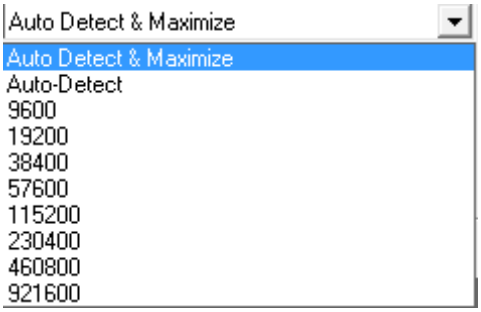


Baud Rate

By default, the baud rate is set to **Auto Detect & Maximize**, which will find the baud rate that the camera is currently set to and then find the highest baud rate supported by both the camera and the frame grabber.



Otherwise, the baud rate can be specifically chosen from among the options available in the drop-down list.

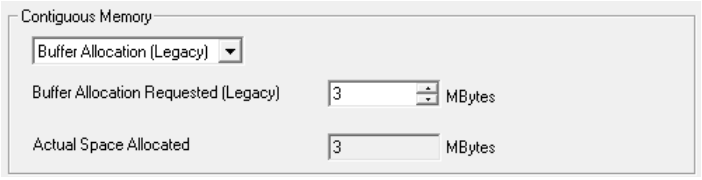


Configuring Contiguous Memory

The **Contiguous Memory** section lets you specify the total amount of contiguous memory to be reserved for allocating **buffers** and **messages**. This RAM memory is used by frame grabbers to allocate DMA tables. In general, contiguous memory is used by legacy applications with older frame grabbers; most applications should use scatter-gather type memory. However, a certain amount of contiguous memory is required for Sapera LT buffer descriptors and 1 MB for every 3000 buffers should be allocated.

Contiguous memory is reserved at boot time for the allocation of dynamic resources used for frame buffer management such as scatter-gather list, DMA descriptor tables plus other kernel needs.

Adjust this value higher if your application generates any out-of-memory error while allocating host frame buffers or when connecting the buffers via a transfer object.



You can approximate the worst-case scenario amount of contiguous memory required as follows:

- Calculate the total amount of host memory used for one frame buffer
  `[number of pixels per line * number of lines * (2 if buffer is 10/12/14 or 16 bits)]`
- Provide 200 bytes per frame buffer for Sapera buffer resources.
- Provide 64 bytes per frame buffer for metadata. Memory for this data is reserved in chunks of 64 kB blocks.
- Provide 48 bytes per frame buffer for buffer management. Memory for this data is reserved in chunks of 64 kB blocks.
- For each frame buffer DMA table, allocate 24 bytes + 8 bytes for each 4 kilobytes of buffer. For example, for a 120x50x8 image:
  `120 * 50 = 6000 = 1.46 4 kB blocks -> roundup to 2 4 kB blocks.`
  Therefore 24 bytes + (2 * 8 bytes) = 40 bytes for DMA tables per frame buffer.
- Memory for this data is reserved in chunks of 64 kB blocks.
- If vertical flipping is enabled, one must add 16 bytes per line per buffer.
- For example, for an image 4080x3072 image:
  `16 bytes * 3072 = 49152 bytes`

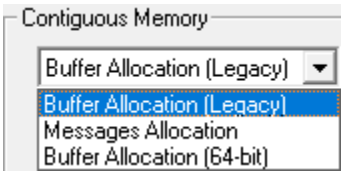
Test for any memory error when allocating host buffers. Simply use the Buffer menu of the Sapera Grab demo program (see Acquiring with Grab Demo) to allocate the number of host buffers required for your acquisition source. Feel free to test the maximum limit of host buffers possible on your host system – the Sapera Grab demo will not crash when the requested number of host frame buffers is not allocated.

The following calculation is an example of the amount of contiguous memory to reserve beyond 5 MB with 80,000 buffers of 2048x1024x8:

a) (80000 * 64 bytes)
b) (80000 * 48 bytes)
c) (80000 * (24 + (((2048*1024)/4 kB) * 8))) = 323 MB
d) Total = a (rounded up to nearest 64 kB) + b (rounded up to nearest 64 kB) + c (rounded up to nearest 64 kB).

Note: All Sapera LT demos and examples **do not use contiguous memory** other than the 1 MB per 3000 buffers required for buffer descriptors. Therefore, you should not modify these settings unless your application requires contiguous memory.

The list specifies the memory type to allocate.



- **Buffer Allocation (Legacy)**: Contiguous memory is allocated in the first 4 GB of host RAM memory. In practice, not all 4 GB is available since it is also used by other hardware resources; the actual amount available can range from 2 GB to 3.5 GB. RAM amounts greater than 4 GB are unused since there is no address space to map it to. Sapera LT buffer descriptors are allocated in this space; 1 MB for every 3000 buffers should be allocated.

  For applications requiring memory resources greater than 4 GB, use 64-bit buffer allocation which is not affected by the legacy 32-bit 4 GB RAM limitation.

  The Xcelera series frame grabbers and earlier only support 32-bit buffer allocation.

- **Buffer Allocation (64-bit)**: Contiguous memory is allocated in all of host RAM memory. By default, Sapera LT does not allocate any contiguous memory. This is supported by newer frame grabbers, such as the Xtium and Xtium2 series.

  The current value for the selected buffer type determines the total amount of contiguous memory reserved at boot-time for the allocation of dynamic resources (for example, buffers, lookup tables, kernels). Adjust this value according to the need of your application for contiguous memory.

- **Messages Allocation**: Determines the total amount of contiguous memory reserved at boot-time for the allocation of messages. This memory space is used to store arguments when a Sapera LT function is called. Increase this value if you are using functions with large arguments, such as arrays. The use of this messaging memory is for support of legacy applications using older frame grabbers.

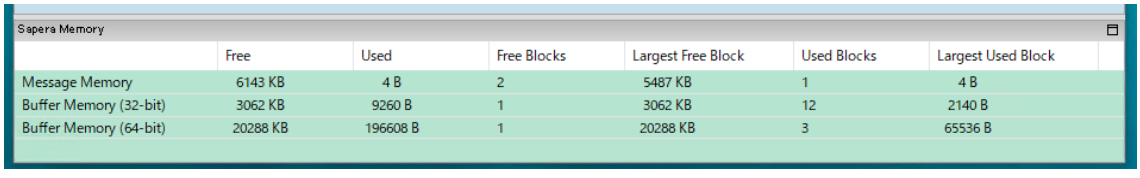
For each item of the list:

- The **Allocation Requested** value displays what was requested.
- The **Actual Space Allocated** value displays the amount of contiguous memory that was allocated successfully.

Sapera LT cannot always allocate the amount of memory requested by the user in one contiguous bloc of memory. In this case, it will try to allocate smaller chunks of memory. The number of blocks is the number of chunks of contiguous memory it needed to allocate the requested memory size.

For example, in the 64-bit buffer memory, if 16 GB is requested, at boot time, Sapera LT will try to allocate one block of 16 GB of memory. If this fails, it will try 2 x 8 GB, then 4 x 4 GB, etc., until it is able to allocate the amount requested. In this case, if 4 blocks of 4 GB each are allocated, the largest block would show as 4 GB and you would see that there are 4 blocks. But once memory starts being used by driver, some fragmentation of the memory will occur, and the number of blocks will increase. But once Sapera LT application is closed, the memory manager performs 'garbage collection' and the initial condition is restored.

The frame grabber Diagnostic Tool (included in the frame grabber installation) displays the memory resources; refer to the frame grabber documentation for more information on using the Diagnostic Tool.



	Free	Used	Free Blocks	Largest Free Block	Used Blocks	Largest Used Block
Message Memory	6143 KB	4 B	2	5487 KB	1	4 B
Buffer Memory (32-bit)	3062 KB	9260 B	1	3062 KB	12	2140 B
Buffer Memory (64-bit)	20288 KB	196608 B	1	20288 KB	3	65536 B

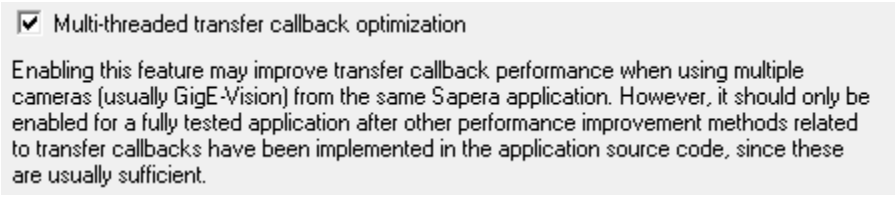
Host Computer Frame Buffer Memory Limitations

When planning a Sapera application and its host frame buffers used, plus other Sapera memory resources, do not forget the Windows operating system memory needs.

A Sapera application using the preferred scatter gather buffers could consume most of the remaining system memory, with a large allocation of frame buffers. If using frame buffers allocated

as a single contiguous memory block, Windows will limit the allocation dependent on the installed system memory. Use the Buffer menu of the Sapera Grab demo program to allocate host buffer memory until an error message signals the limit allowed by the operating system used.

Multi-Threaded Transfer Callback Optimization



Usage Notes When Writing Sapera Applications

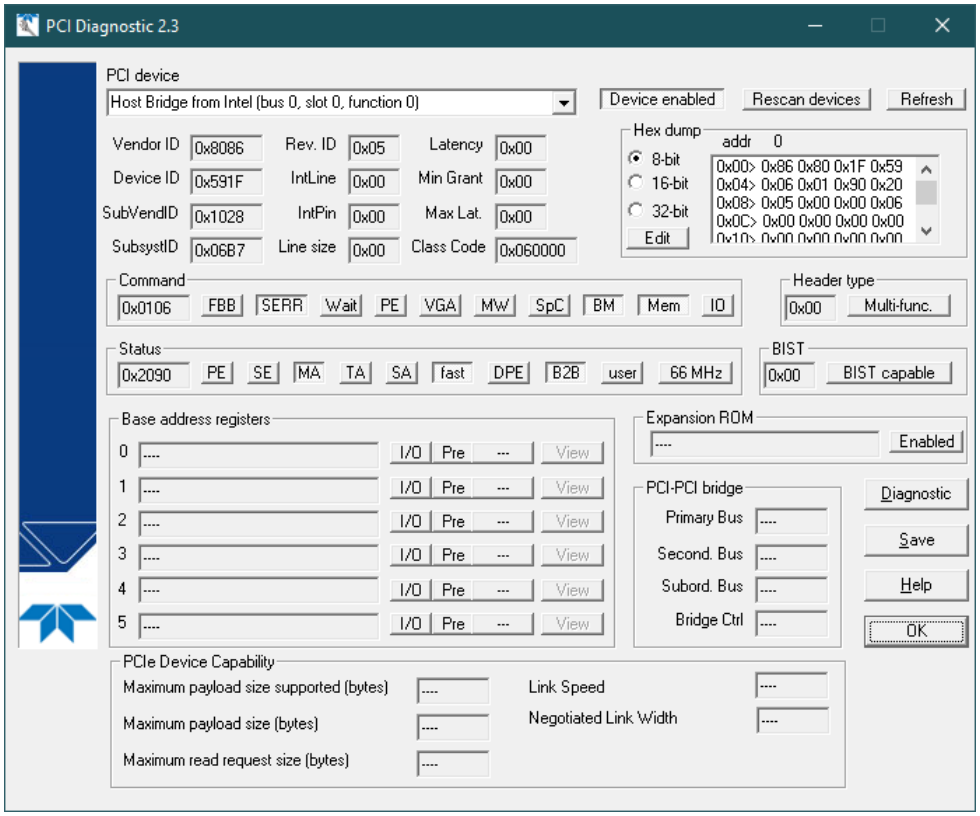
- Always disable the **Multi-threaded transfer callback optimization** option (the default) while developing and thoroughly testing the application, especially making sure that appropriate robustness standards are met.
- If the application does not meet performance requirements, all the known performance improvements that can be implemented in application code must be tried (for example, limiting operations as much as possible in the transfer callback function).
- If performance requirements are still not met, and there is only one camera per running instance of the application, then still leave disabled since it provides no performance benefit.
- Only consider enabling if performance requirements are not met with multiple cameras in the same running instance of the application.
- If enabling does not improve performance, then disable it.
- If enabling improves performance, the application must be once again thoroughly tested to prove that it still meets the same robustness requirements as before.

Sapera PCI Diagnostic

The Sapera PCI Diagnostic tool is used for debugging frame grabber hardware issues.
PCI Diagnostic reads the content of the PCI configuration space and detects memory and I/O conflicts between PCI devices. Use it to verify the integrity of your system before and after installing a new PCI device. Refer to the utility's online help for more information on using the PCI Diagnostic utility.

To open Sapera PCI Diagnostic

- From the **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera PCI Diagnostic**.



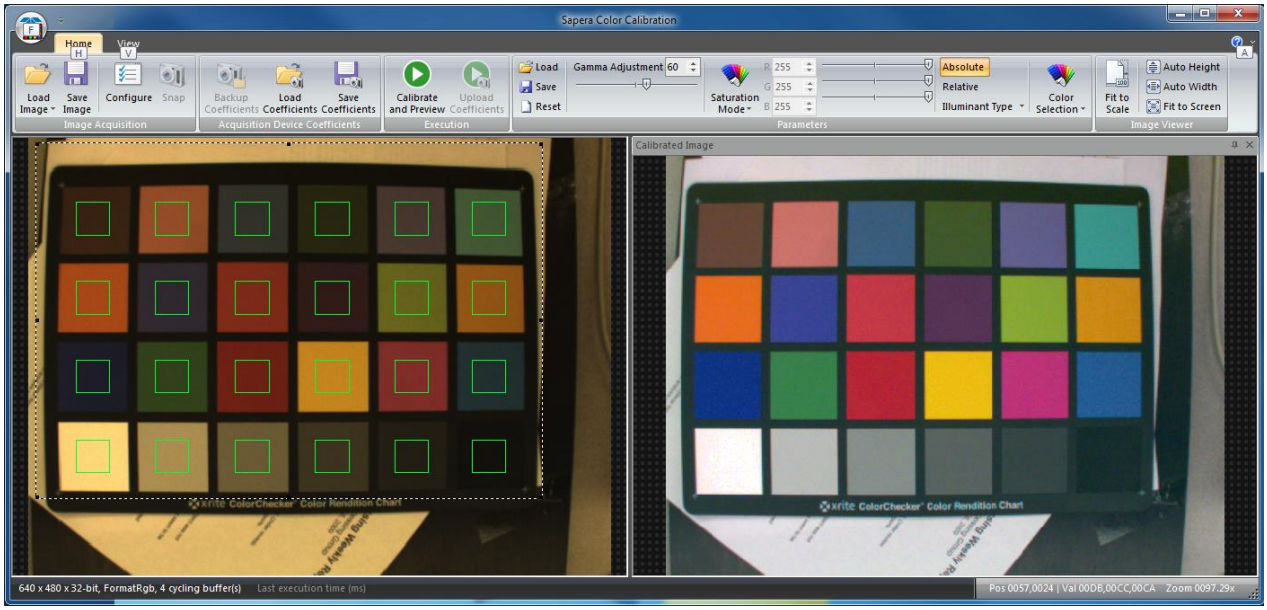
Sapera Color Calibration

Sapera Color Calibration allows you to generate color correction coefficients that are used to adjust the camera sensor's color response for known colors at a specific illumination. This ensures that the camera outputs the correct color for a given scene.

The color correction coefficients are saved as a *.ccor file that can be uploaded to the camera to perform real-time correction using the camera's hardware (instead of performing correction on the host computer after the image transfer).

To open Sapera Color Calibration

- From the **Start** menu, select **Teledyne DALSA Sapera LT** > **Sapera Color Calibration**.



Demos and Examples

Several generic demos and examples are available for both Sapera ++ and Sapera .NET. Complete source code is provided for projects in Microsoft Visual Studio 2013/2015/2017/2019.

Note that the project files support Unicode character set only. The executables were compiled with Visual Studio 2019.

If your application requires a user interface, Sapera LT includes the GUI classes used by many of the demos to create commonly used dialog boxes. The GUI classes include a set of Microsoft® Foundation Classes (MFC) based dialog boxes designed to implement some of the most commonly used tasks for Sapera LT applications, such as loading an acquisition configuration file. They, however, do not constitute an official API. Rather, they are provided 'as is' with source code so that you may modify them at your discretion. For more information on these classes refer to the Sapera LT GUI Class Reference manual.

Certain device driver installations provide other demos and examples that demonstrate the specific usages and capabilities of the device. Refer to a specific device user's manual for further details.

Demo Source Code

Several demo programs are available with Sapera. They are GUI applications, more complete than the supplied examples. There are demos that cover Sapera LT ++ and Sapera .NET.

The demos' main purpose is to provide the user with a starting application that can be modified to become the user's end application.

The Sapera LT ++ and Sapera LT .NET demo source code for the supported compilers are found in the Sapera\Demos directory.

Projects are also provided to allow you to recompile all the demos in a batch, together with the Sapera LT ++ GUI Classes.

Example Source Code

Several example programs are available within Sapera. They are essentially basic console applications demonstrating simple tasks like grabbing an image and loading an image file from the disk.

The main purpose of the examples is to provide the user with code samples that can be easily extracted and integrated into an application. Examples cover both Sapera LT ++ and Sapera .NET.

The Sapera LT ++ and .NET example source code for the supported compilers are found in the Sapera\Examples directory.

Projects are also provided to allow you to recompile all the examples in a batch.

Demos and Examples for Frame Grabbers

The Sapera Explorer utility provides easy access to all demos and examples, along with their source code. Note that all executables were compiled with Visual Studio 2019.

Demo Programs

Demo Name	Description
GrabDemo	This program demonstrates the basic acquisition functions included in Sapera LT ++. It allows you to acquire images either in continuous or in one-shot mode, while adjusting the acquisition parameters. For a tutorial on using this demo, see Acquiring with Grab Demo. The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital camera.
SeqGrabDemo	Sequential Grab Demo. This program demonstrates how to grab a sequence of images into memory and then display them. The program allows you to record several images and then load and save AVI files. Each image is stored in its own buffer and can be reviewed. A small number of images are allocated by default, but they can be increased using the buffer options inside the demo. The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital camera.
MultiBoardSyncGrabDemo	Multi-board Sync Acquisition Demo. This program demonstrates how to grab from two different frame grabbers simultaneously (SapAcquisition servers), place both images in a single buffer, and display it. The minimum requirements for this demo are two Sapera-compatible frame grabbers, each with an analog or digital camera.
ColorConvDemo	Color Conversion Demo. This program demonstrates the color conversion functionality, for both Bayer and other color formats, included in Sapera LT ++. It allows you to acquire images either in continuous or in one-shot mode, while adjusting the acquisition parameters. It includes interactive control of color conversion parameters. You may optionally apply filtering to acquired images. The minimum requirements to run this demo are a Sapera-compatible frame grabber with a digital Bayer filter or other supported color format camera.
FlatFieldDemo	This program demonstrates the flat field correction functionality included in Sapera LT ++. Flat Field Correction (FFC) includes Fixed Pattern Noise (FPN), Pixel Replacement, Photo Response Non-Uniformity (PRNU), and Shading Correction. The demo allows you to acquire images either in continuous or in one-shot mode, while adjusting the acquisition parameters. It includes interactive calibration of flat field gain and offset settings. You may optionally apply flat field correction to acquired images. For a tutorial on using this demo, see Using the Flat Field Demo. The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital monochrome camera.

Demo Name	Description
IODemo	I/O Demo. This program demonstrates the usage of General I/O functionality included in Sapera LT ++. It allows you to monitor the current state of the input and output pins and change their state between low and high polarity. It also allows you to monitor interrupts on the input pins and to count the number of incoming interrupts. The minimum requirements to run this demo are a Sapera-compatible board with General I/O capabilities. This example is *not available in .NET*.

Console Examples

Example Name	Description
GrabLut	This example shows how to grab images from a selected camera into a buffer in the host computer's memory. This program also shows how to enable, save, and load a lookup-table (LUT) into the hardware processing engine and display the image with the selected LUT applied.
GrabCPP	Grab Console. This example shows how to grab an image from a selected camera into a Sapera buffer and then display it. The buffer is created according to the camera settings. Any Sapera compatible frame grabber can be used. This example is named Grab in .NET.
GrabConsoleMultiformat	This example is similar to the Grab Console example but shows how to use multiformat buffers (images that contain both RGB and monochrome (IR) data). This example is *not available in .NET*.

Demos and Examples for GenCP CameraLink Cameras

All frame grabber demos apply to GenCP CameraLink Cameras.

Example Name	Description
CameraEvents	This example shows how to list all the available events with SapAcqDevice. Using the registering and unregistering callback mechanism, it shows also how to track when a specific event occurs.
CameraFeatures	This example shows how to enumerate available features on a camera (for example, Genie or GigE). It also shows how to retrieve feature specific information (for example, access mode), and how to change feature values.
CameraFiles	This example shows how to upload/download files for GenCP cameras that support file access such as firmware upload and LUTs.
CameraFirmwareUpdate	This example shows how to update firmware for GenCP cameras that support file access, allowing automatic firmware updates at the application level. (GigE and CLHS cameras are also supported by this example.)
FindCamera	This example shows how to list all detected cameras when more than one camera is present, listing them by user name, serial number, model name or server name. By uncommenting a part of code, you will be able to change the user defined name of the camera.
GrabCameraLink	This example shows how to acquire images using a Teledyne DALSA frame grabber (using the SapAcquisition API class) together with a Teledyne DALSA camera which is accessible through the SapAcqDevice API class (such as Piranha4).

Generic Sapera LT Examples

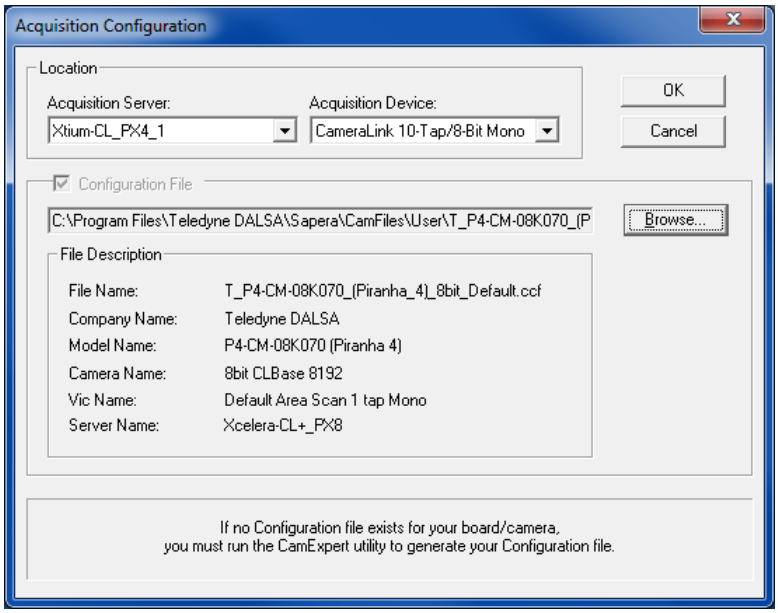
The generic Sapera LT examples do not require an acquisition device.

Example Name	Description
ColorSplit	Shows how to split and merge color images into single monochrome components. An RGB image is loaded, split into three monochrome components, then a simple processing is applied to the three components before they are merged back to RGB as output.
FileLoadCPP	This example shows how to load an image file from the disk into a Sapera buffer and then display it. The buffer is created according to the image file properties. One of several images (monochrome, RGB, or YUV) can be selected for loading. This example is named FileLoad in .NET.

Acquiring with Grab Demo

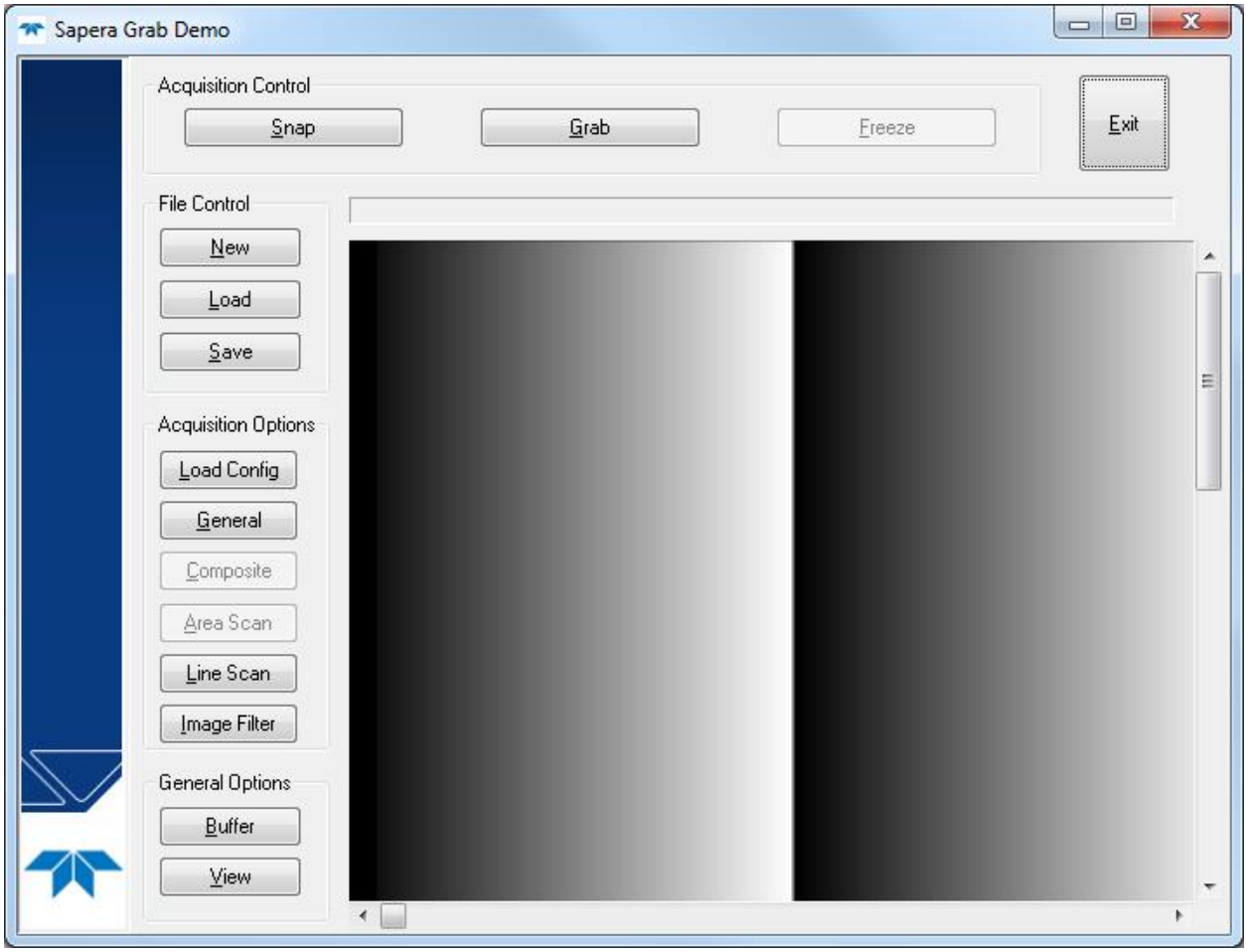
The Sapera LT Grab Demo program allows you to grab and display a live image in a host buffer. It can accommodate any Sapera-compatible board with any camera. This demo is a good starting point to verify that your camera and frame grabber are properly installed.

The following dialog box appears when starting the program.



- Select the **Acquisition Server** and the **Acquisition Device**. The first one corresponds to the board you want to grab from; the second represents the acquisition device on this board (some boards may have more than one).
- (Optional) Select an acquisition configuration file (CCF File) compatible with your camera. CamExpert must be used to generate CCF files (for example, external trigger, cropping window, and so forth).
- Click **OK** to start the demo.

If this dialog box does not appear, then no frame grabber was found within your system. In such a case, verify that the device driver corresponding to your frame grabber was correctly installed. The **Sapera LT Configuration Program** gives you a list of boards present within your system. This demo can work without a frame grabber. However, it only allows you to load the file from the disk.



By using Grab Demo you can now:

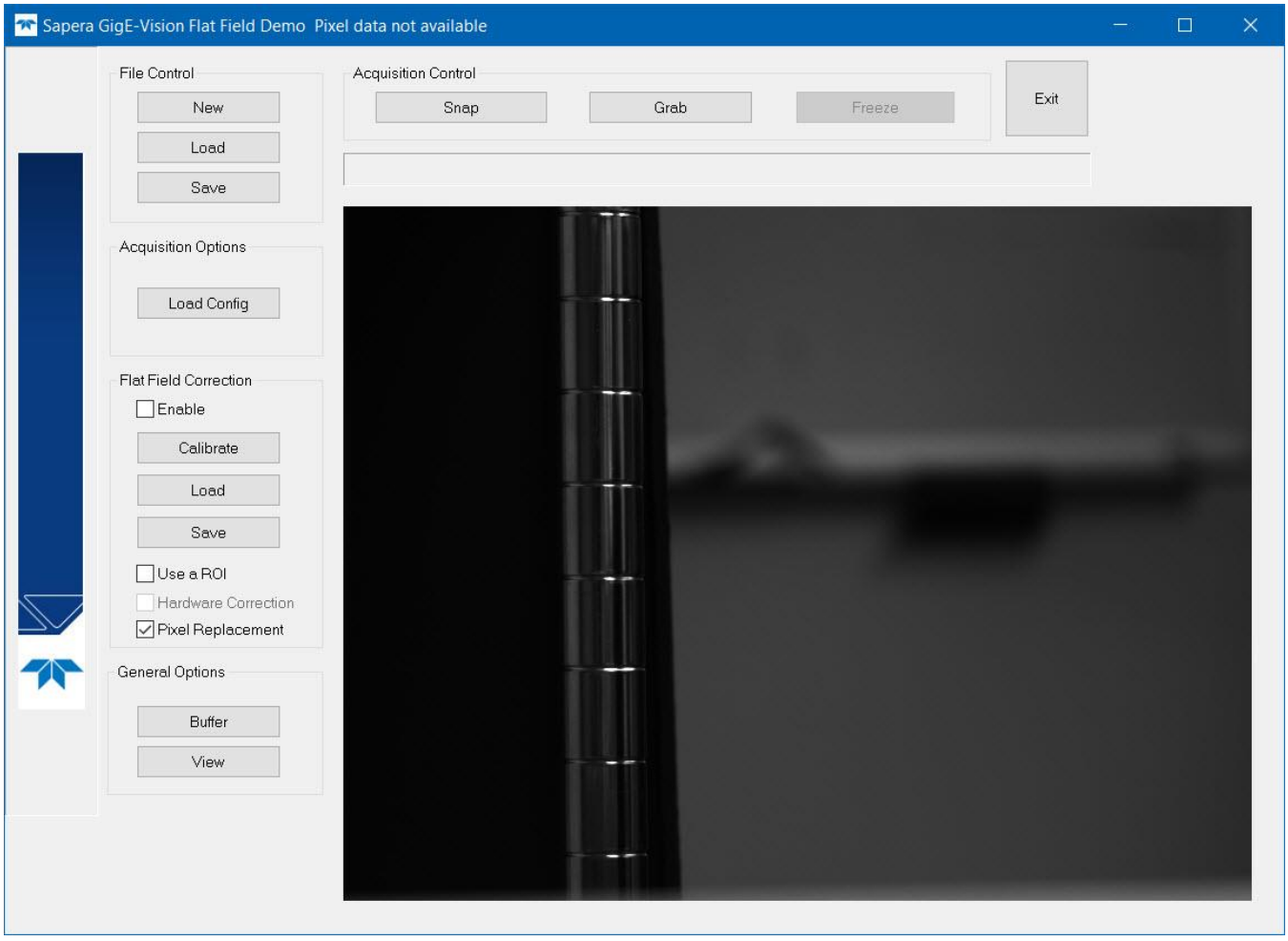
- Control the acquisition using the **Snap**, **Grab**, **Freeze,** and **Abort** buttons.
- Load/save images from/to disks using the **Load** and **Save** buttons.
- Dynamically adjust the acquisition parameters through the **General**, **Area scan**, **Linescan**, and **Composite** buttons.
- Change the CCF file using the **Load Config** button.
- Adjust the hardware-based acquisition **Image Filter** (if supported).
- Use the **Buffer** button to change the number of buffers used for internal cycling and the type of buffer used (contiguous, scatter-gather, off-screen, or overlay).
- Use the **View** button to adjust the scaling and the select the bit range to display when grabbing images with formats greater than 8-bits.

Using the Flat Field Demo

Flat Field Correction is the process of eliminating small gain differences between pixels in a camera. When calibrated flat field correction is applied to the image, the sensor exposed to a uniformly lighted field will have no gray level differences between pixels. The Flat Field demo automatically functions both with hardware supporting flat field processing or with software processing via the Sapera library on the host system processor. If the selected board does not support onboard flat field processing, a message is displayed stating that processing will be done on the host system.

Flat Field Demo Main Window

The demo main window provides acquisition control buttons and a central area for displaying the grabbed image. Developers can use the demo source code as a foundation to quickly create and test the desired imaging application.



There are two steps to using flat field correction:

- Performing flat field calibration.
- Enabling flat field correction on your acquisitions.

Flat Field Calibration

Calibration is the process of taking two reference images, one of a dark field and one of a light gray field (not saturated), to generate correction data for images captured by the camera. Each pixel data is modified by the correction factor generated by the calibration process, so that each pixel now has an identical response to the same illumination.

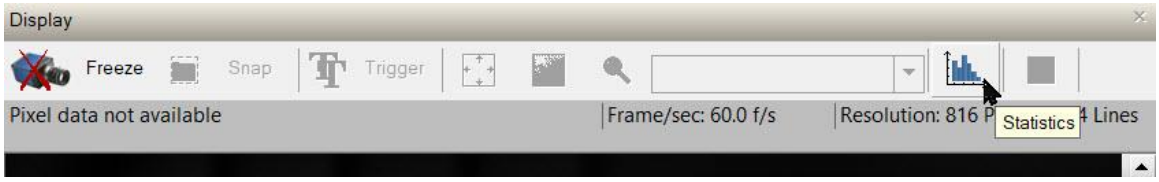
Preparing for Flat Field Calibration

To get ready for flat field calibration, you will need to perform acquisitions of dark images and of light images using the Histogram Tool. You will need to grab a flat light gray level image (not saturated), such as a clean evenly lighted white wall or non-glossy paper with the lens slightly out of focus. Ideally, a controlled diffused light source aimed directly at the lens should be used. Note the lens iris position for a bright but not saturated image. Additionally, check that the lens iris closes well or have a lens cover to grab the dark calibration image.

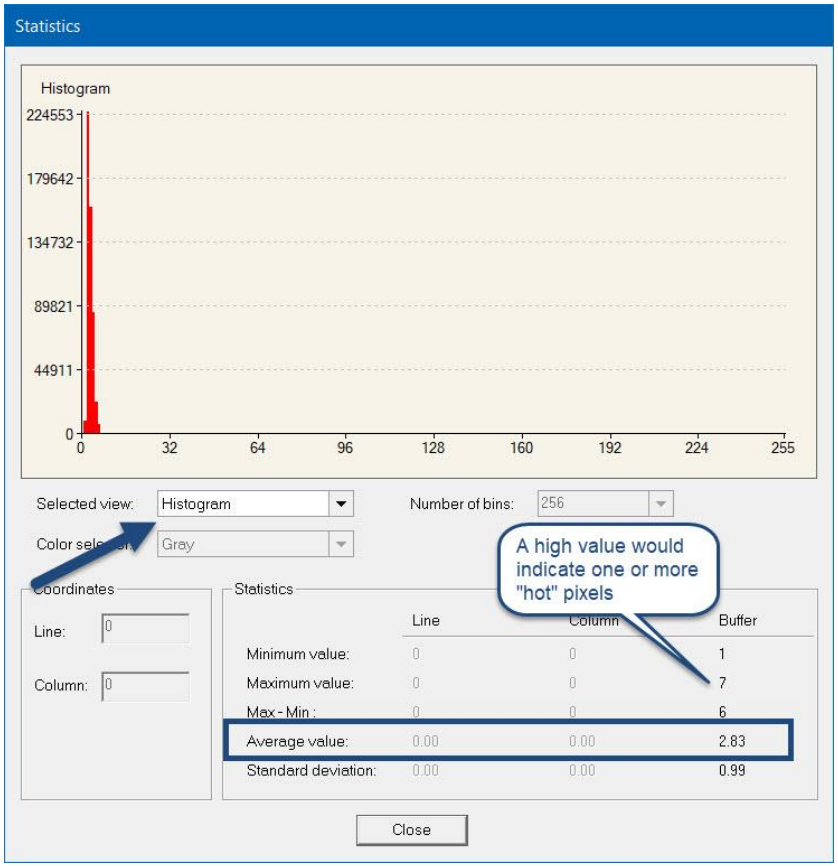
Use CamExpert to make the acquisition.

Dark Image Acquisition

- Close the camera lens iris and cover the lens with a lens cap.
- In CamExpert, click **Grab**.
- While grabbing, click the **Statistics** button on the Display toolbar to open the Statistics window, and select *Histogram* on the **Selected View** list.



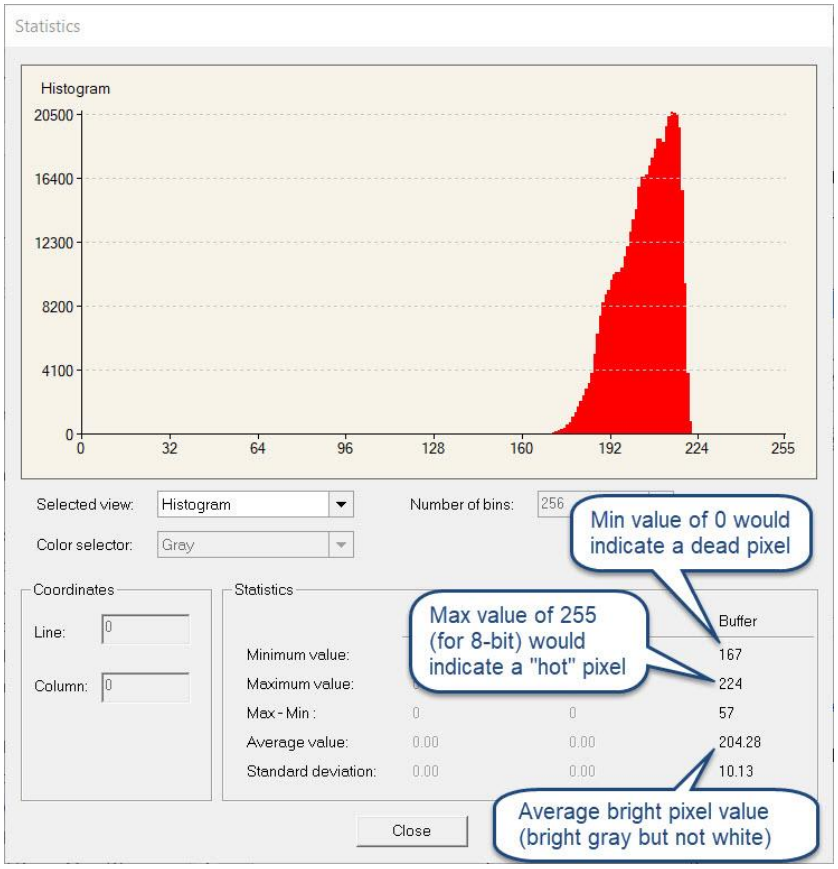
The following figure shows a typical histogram for a camera grabbing a very dark image.



Important: In this example, the **average** pixel value for the frame is close to black. Also note that most sensors will show a much higher maximum pixel value due to one or more "hot pixels". The sensor specification accounts for a small number of hot or stuck pixels (pixels that do not react to light over the full dynamic range specified for that sensor).

Bright Image Acquisition

- Aim the camera at a diffused light source or evenly lit white wall with no shadows falling on it.
- In CamExpert, click **Grab**.
- While grabbing, click the **Statistics** button on the Display toolbar to open the Statistics window, and select *Histogram* on the **Selected View** list.
- Use the lens iris to adjust for a bright gray approximately around a pixel value of 200 (for 8-bit pixels). The following figure shows a typical histogram for a bright gray image.
- Note the camera position and lens iris position to be able to repeat it during the calibration procedure.



Important: In this example, the **average** pixel value for the frame is bright gray. Also note that sensors may show a much higher maximum or a much lower minimum pixel value due to one or more "hot or dead pixels". The sensor specification accounts for a small number of hot, stuck, or dead pixels (pixels that do not react to light over the full dynamic range specified for that sensor).

Performing Flat Field Calibration

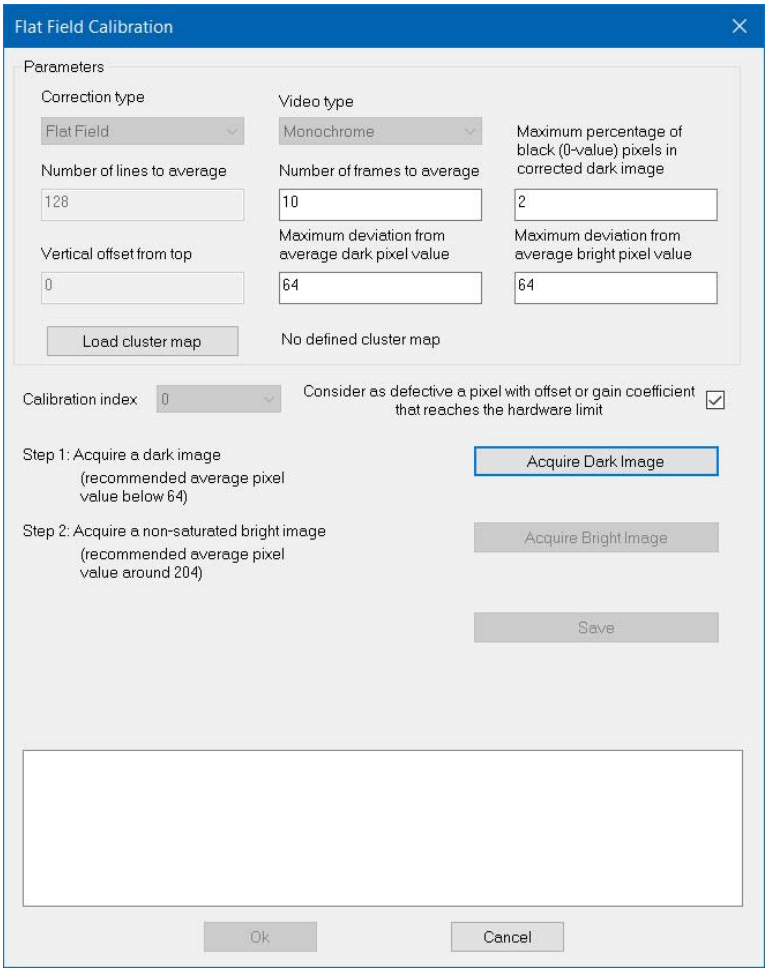
Calibration is the process of taking two reference images, one of a dark field and one of a light gray field (not saturated), to generate correction data for images captured by the camera. **Before calibrating**, look at section Preparing for flat field calibration on how to set up those dark/white acquisitions.

After calibration, each pixel data is modified by the correction factor generated by the calibration process, so that each pixel now has an identical response to the same illumination.

The calibration process for an area scan camera can be over the whole frame or a selected number of lines within the frame. For line-scan cameras the user simply selects the number of input lines to average. The calibration control overview follows.

To perform calibration

- Click **Calibrate**. The calibration window opens as shown.



- First select the **Correction Type** as flat field or single flat line. Note that when using a line-scan camera, only flat line calibration is available.
- **Video type** will default to the acquisition type defined in the loaded camera file.
- Set the **Number of frames to average** during each calibration step. This should be set to more than one to avoid false data from random pixel noise.
- The field for **Maximum deviation from average** defaults to 25% of the gray level range captured, (64 for 8-bit capture, 256 for 10-bit capture, and so forth). This value sets the threshold for detecting static dead pixels – both dark or light. Users will need to adjust this field to best isolate dead pixels from their imaging source.
- The field for **Maximum percentage of black pixels in corrected dark image** defaults to 2. A high percentage of 0 value pixels should be avoided so that gain adjustments can be properly calibrated.
- When doing a single line calibration to apply to the captured frame, use the two selection fields **Number of lines to average** and **Vertical offset from top**, to select which video line will be used.
- Setup the camera to capture a uniform dark image. Black paper with no illumination and the camera lens' iris closed to minimum can provide such a black image. See Dark Image Acquisition.
- Click on **Acquire Dark Image**. The flat field demo will grab a video frame, analyze the pixel gray level spread, and present the statistics. If acceptable, use the image as the black reference.
- Setup the camera to acquire a uniform white image (but not saturated white). Even illumination on white paper can be used. It is preferable to prepare for the white level calibration before the calibration procedure. See Bright Image Acquisition.

- Click on **Acquire Bright Image**. The flat field demo will grab a video frame, analyze the pixel gray level spread, and present the statistics. If acceptable, use the image as the white reference.
- Test the calibration by enabling flat field correction during a live grab. If necessary, adjust the dead pixel detection threshold and repeat the calibration.
- Save multiple versions of calibration data to compare for best imaging or for different imaging setups.

Flat Field Correction

Once flat field calibration is done, you can enable flat field correction.

Verify camera acquisition

First ensure that the camera is functioning and that the acquisition board is capturing live images. The Flat Field Demo main window provides acquisition controls to confirm image capture.

Enable

To use real time flat field correction, first click in the **Enable** box. Then do image snaps or continuous live grab.

Save

Click on the **Save** button to store files with the flat field gain and offset data gathered with the calibration procedure. Files are saved as .bmp images and can be named as required to reference the camera used.

Load

Click on the **Load** button to retrieve files with the flat field gain and offset data gathered with the calibration procedure.

Appendix: File Locations

The table below describes the contents of the Teledyne DALSA installation directory, usually **C:\Program Files\Teledyne DALSA**.

Directory	Contents
GenICam 3.20	GenICam standard support files
Network Interface	Sapera Network Imaging Package
Sapera	Sapera LT Readme and version history documents
Sapera\Bin	Utility programs
Sapera\CamExpert	CamExpert frame grabber and camera configuration utility
Sapera\CamFiles	Camera configuration files for frame grabbers
Sapera\Classes	Sapera LT ++ header files (Basic and GUI Classes) Sapera LT ++ source code (GUI Classes only)
Sapera\Components\NET	.NET classes
Sapera\Demos	Source code for GUI-based demo applications
Sapera\Demos\Binaries	Executable files for GUI-based demo applications
Sapera\Examples	Source code for console-based demo applications
Sapera\Examples\Binaries	Executable files for console-based demo applications
Sapera\Explorer	Utility program for accessing Sapera examples, demos and documentation.
Sapera\Help	Online documentation
Sapera\Images	Image files used by demos and examples
Sapera\Include	Header files for C libraries
Sapera\Lib	Import libraries
Sapera\Licenses	License agreements for Teledyne and other products
C:\Windows\system32	Dynamic Link Libraries (DLLs)
C:\Windows\system32\drivers	Device drivers

Contact Information

Sales Information

Visit our web site:	http://www.teledynedalsa.com/
Email:	mailto:info@teledynedalsa.com

Technical Support

Submit any support question or request via our web site:

Technical support form via our web page:	
Support requests for imaging product installations	http://www.teledynedalsa.com/en/support/options/
Support requests for imaging applications	
Camera support information	
Product literature and driver updates	

When encountering hardware or software problems, please have the following documents included in your support request:

- The Sapera Log Viewer .txt file
- The PCi Diagnostics PciDiag.txt file (for frame grabbers)
- The Device Manager BoardInfo.txt file (for frame grabbers)

Note, the Sapera Log Viewer and PCi Diagnostics tools are available from the Windows **Start** menu under **Teledyne DALSA Sapera LT**.
The Device Manager utility is available as part of the driver installation for your Teledyne DALSA device and is available from the Windows **Start** menu under **Teledyne DALSA *Device Name*** > **Device Manager**.