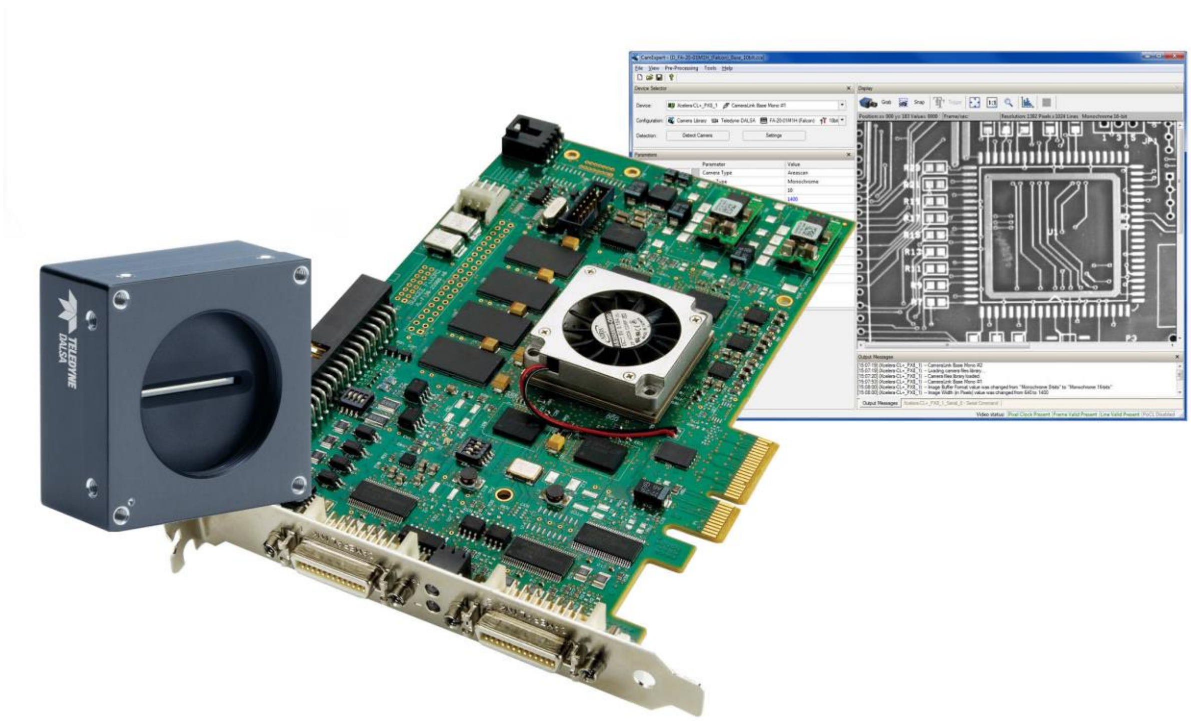
Sapera LT™ 8.60

Getting Started Manual
for Frame Grabbers

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



P/N: OC-SAPM-INTR0
www.teledynedalsa.com

TELEDYNE DALSA
Everywhere**you**look™

NOTICE

Printed on October 15, 2020

Document Number: OC-SAPM-INTR0
Printed in Canada

About This Manual

This manual exists in Windows Help, and Adobe Acrobat® (PDF) formats (printed manuals are available as special orders). The Help and PDF formats make full use of hypertext cross-references. The Teledyne DALSA home page on the Internet, located at http://www.teledynedalsa.com/imaging, contains documents, software updates, demos, errata, utilities, and more.

About Teledyne DALSA

Teledyne DALSA, a business unit of Teledyne DALSA Inc., is an international high-performance semiconductor and electronics company that designs, develops, manufactures, and markets digital imaging products and solutions, in addition to providing wafer foundry services.

Teledyne DALSA Digital Imaging offers the widest range of machine vision components in the world. From industry-leading image sensors through powerful and sophisticated cameras, frame grabbers, vision processors and software to easy-to-use vision appliances and custom vision modules.

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 3rd party camera models across all common interface formats like GigE Vision®, Camera Link®, as well as emerging new image acquisition standards such as CLHS.

> If your application requires image processing or GPU optimization, Sapera Essential, a full-featured image processing library, is available as a separate software package. For more information see www.teledynedalsa.com/imaging/products/software/.

The manual introduces the Sapera LT API and is designed to help programmer's with installation, and quickly perform 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 grabber 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: provides 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: frame grabber configuration
- PCI Diagnostic: frame grabber hardware debugging

Sapera LT Manual Descriptions

Sapera LT is supported by the following manuals in PDF, and compiled HTML help formats.

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 using Sapera LT APIs with GigE Vision cameras and 3D sensors, possible hardware configurations, and a quick overview of Sapera programming.
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++, C# and VB .NET.
GigE Vision	**Network Optimization Guide**	Network setup and optimization guide for GigE Vision cameras as well as troubleshooting guidelines.
.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 GUI Classes Reference Manual**	Describes the C++ GUI (graphical user interface) helper classes used to create common application dialogs. These classes are used in the Sapera demo programs and are provided to help users with applications that require a GUI.
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).
C	**Sapera LT Basic Modules Reference Manual**	Lists in detail the Sapera LT low-level module functions as well as data definitions, file formats, and macros (based on the C language).

Supported Operating Systems

- Windows 10 (32-bit or 64-bit versions)
- Windows 7 SP1 with security update KB3033929 (SHA-2 Code Signing Support)

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 2010
  - Microsoft Visual Studio 2012
  - Microsoft Visual Studio 2013
  - Microsoft Visual Studio 2015
  - Microsoft Visual Studio 2017

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, and utilities, such as CamExpert, for 32 or 64-bit application development, and optionally the GigE Vision module (if required).

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

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

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

Alternatively, a Sapera Vision Software disk is also available. Use the Sapera Vision Software installation disk autorun program and select the Software Installation option and follow the instructions. The software disk also allows you to select any Teledyne DALSA frame grabber driver software to install.

Note: The Sapera LT 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.

Upgrading Previous Versions of Sapera LT

Sapera LT 7.30 and Higher

When upgrading from Sapera LT version 7.30 or higher, uninstalling is not required; proceed directly to installation of Sapera LT 8.40; the same directory structure and component configuration are used.

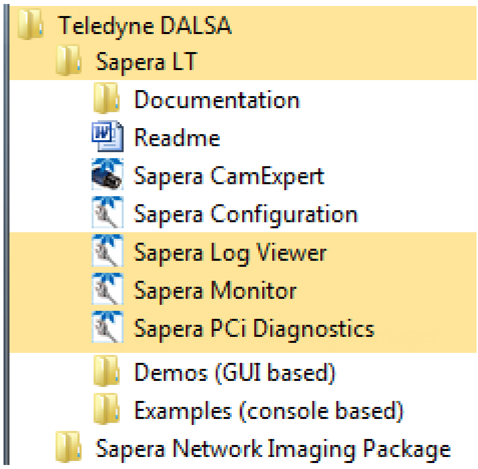
Sapera LT 7.20 and Lower

Before installing Sapera LT 8.40, uninstall Sapera LT from the Windows Control Panel > Programs and Features.

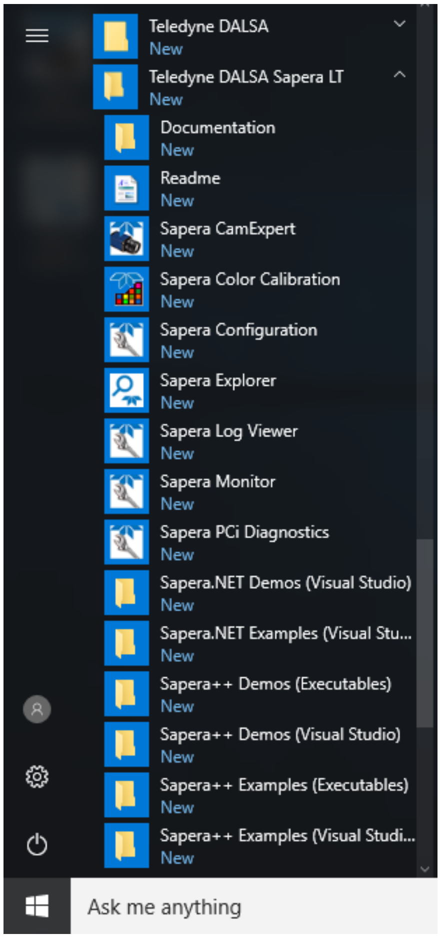
Start Menu Shortcuts

For Windows 7, Start menu shortcuts for Sapera LT are available under **All Programs > Teledyne DALSA > Sapera LT** and **> Sapera Network Imaging Package.** For Windows 10, similar shortcuts are available from the Start menu.

The following screenshots display the menus for the full Sapera LT SDK.



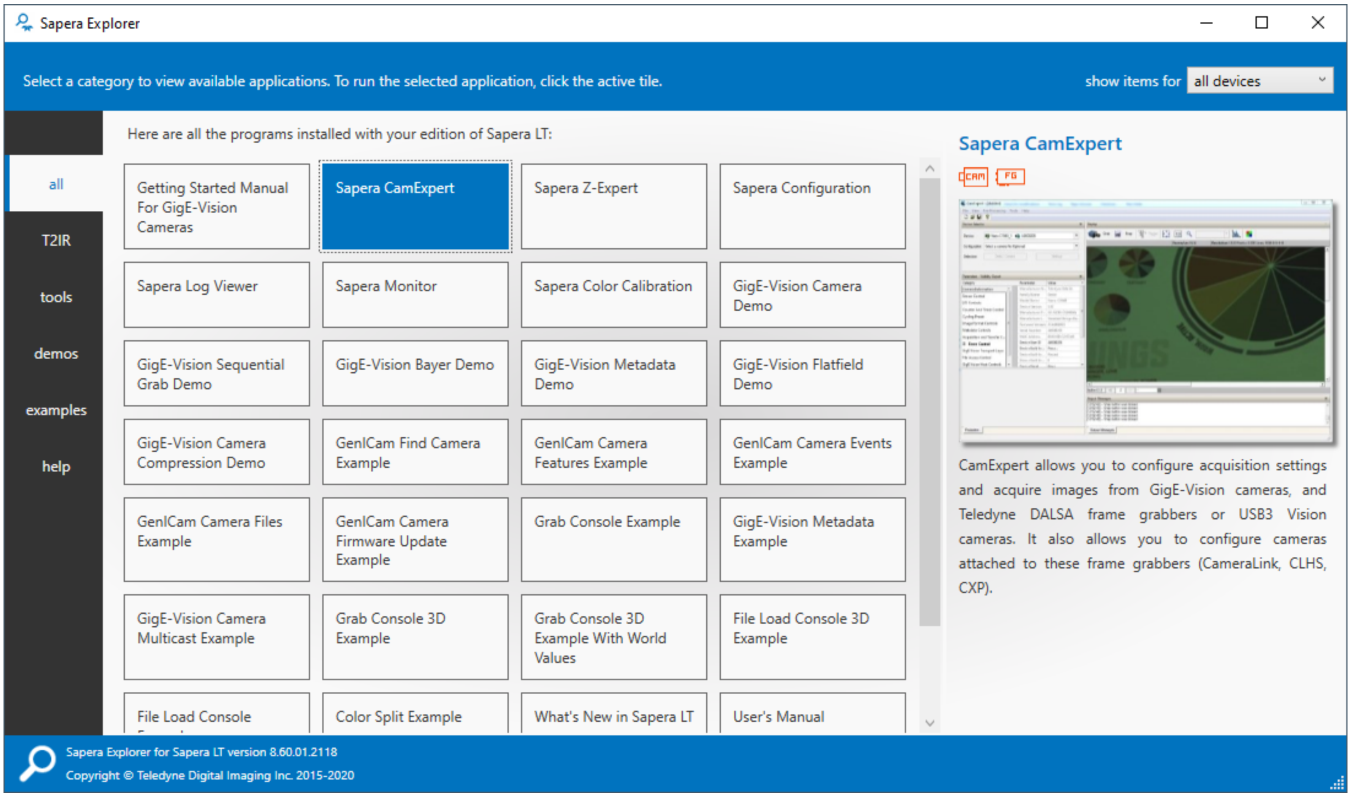
Windows 7



Windows 10

Sapera Explorer

The Sapera Explorer application provides quick access to the Sapera LT tools documentation, demos, examples and source code, grouped by category. Select a category to view available applications. To run the selected application, click the active tile.



Sapera Explorer

Select a category to view available applications. To run the selected application, click the active tile.

show items for: all devices

Here are all the programs installed with your edition of Sapera LT:

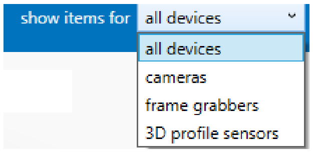
Getting Started Manual For GigE-Vision Cameras	Sapera CamExpert	Sapera Z-Expert	Sapera Configuration
Sapera Log Viewer	Sapera Monitor	Sapera Color Calibration	GigE-Vision Camera Demo
GigE-Vision Sequential Grab Demo	GigE-Vision Bayer Demo	GigE-Vision Metadata Demo	GigE-Vision Flatfield Demo
GigE-Vision Camera Compression Demo	GenICam Find Camera Example	GenICam Camera Features Example	GenICam Camera Events Example
GenICam Camera Files Example	GenICam Camera Firmware Update Example	Grab Console Example	GigE-Vision Metadata Example
GigE-Vision Camera Multicast Example	Grab Console 3D Example	Grab Console 3D Example With World Values	File Load Console 3D Example
File Load Console	Color Split Example	What's New in Sapera LT	User's Manual

Sapera CamExpert

CamExpert allows you to configure acquisition settings and acquire images from GigE-Vision cameras, and Teledyne DALSA frame grabbers or USB3 Vision cameras. It also allows you to configure cameras attached to these frame grabbers (CameraLink, CLHS, CXP).

Sapera Explorer for Sapera LT version 8.60.01.2118
Copyright © Teledyne Digital Imaging Inc. 2015-2020

Items can be filtered using the **show items for** drop-down list:



show items for: all devices

- all devices
- cameras
- frame grabbers
- 3D profile sensors

For Demos and Examples, click **Browse code for Sapera ++** or **Browse code for Sapera .NET** to open the source code directory.

Browse code for Sapera++

Browse code for Sapera .NET

The CAM, FG and 3D icons indicate whether the demo or example is supported by cameras, frame grabbers or 3D sensors.

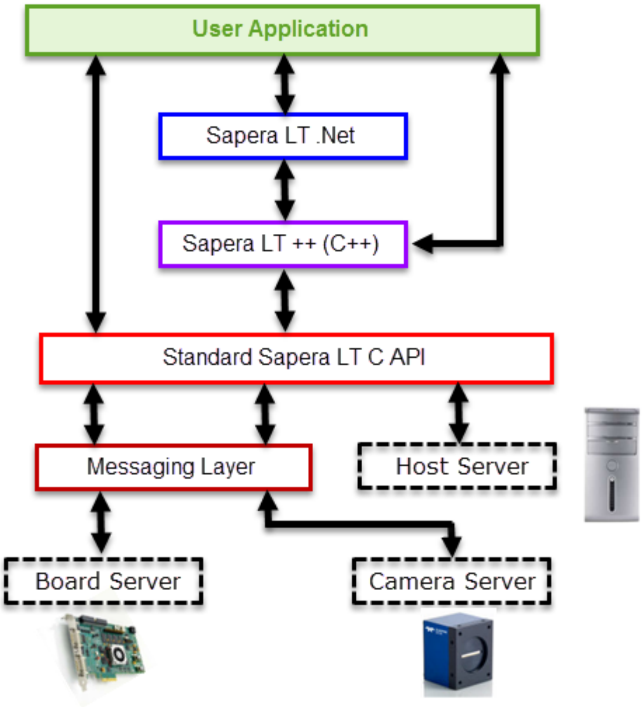
CAM FG 3D

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.

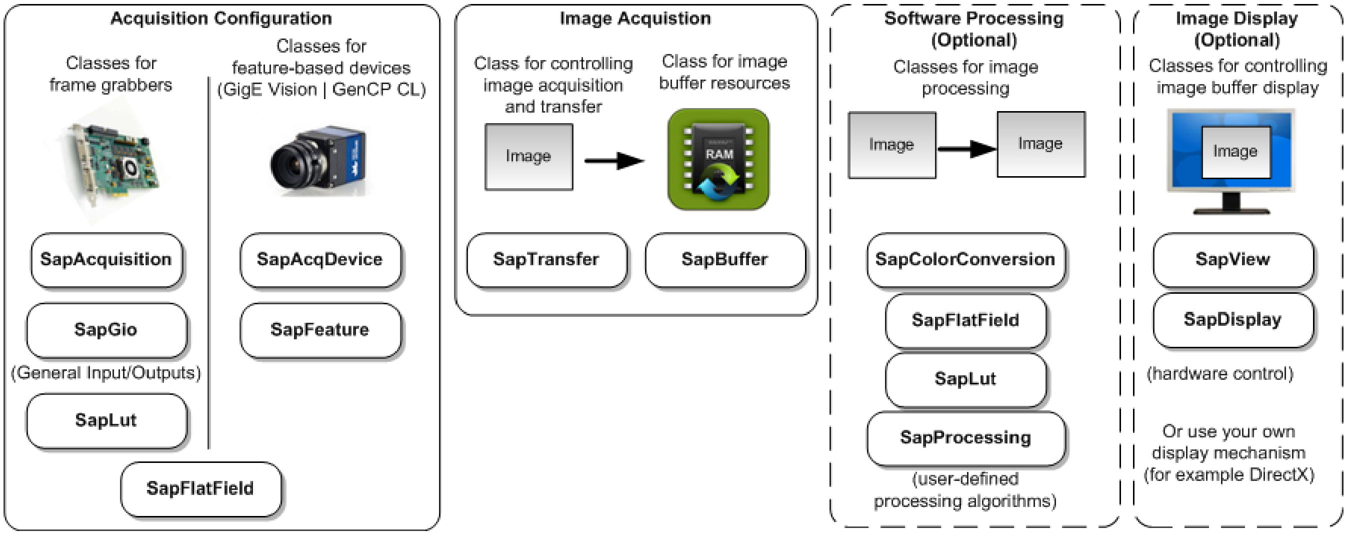
Library Architecture

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 Basic Modules Reference Manual and 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.


Acquisition Configuration
Classes for frame grabbers
Classes for feature-based devices (GigE Vision | GenCP CL)
SapAcquisition
SapGio
(General Input/Outputs)
SapLut
SapAcqDevice
SapFeature
SapFlatField
Image Acquistion
Class for controlling image acquisition and transfer
Class for image buffer resources
Image
RAM
SapTransfer
SapBuffer
Software Processing (Optional)
Classes for image processing
Image
Image
SapColorConversion
SapFlatField
SapLut
SapProcessing
(user-defined processing algorithms)
Image Display (Optional)
Classes for controlling image buffer display
Image
SapView
SapDisplay
(hardware control)
Or use your own display mechanism (for example DirectX)

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

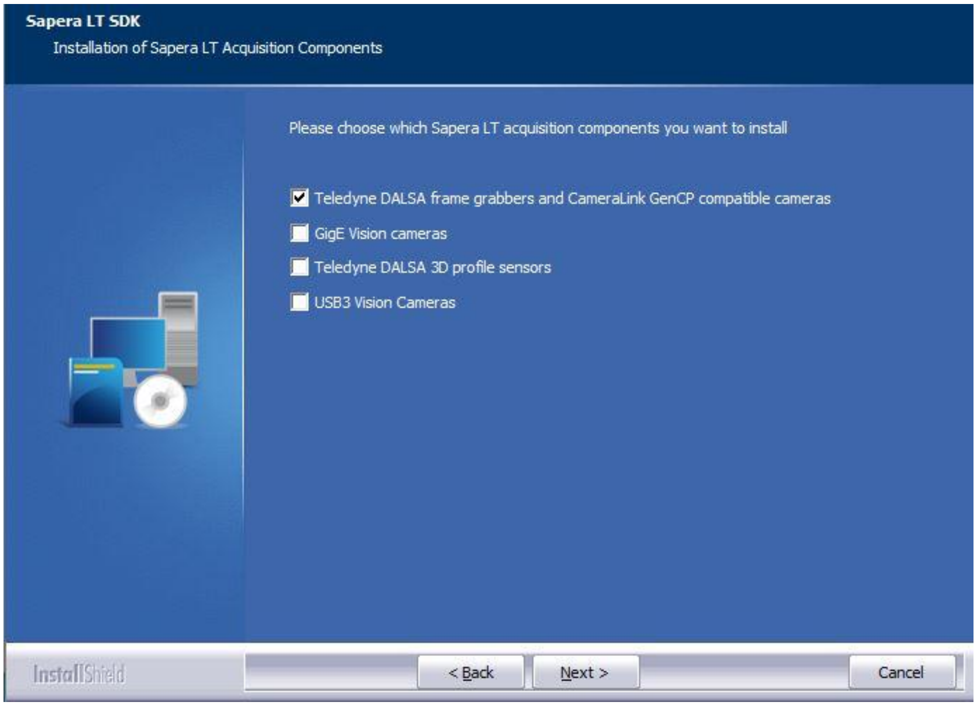
Sapera LT applications need to load the appropriate acquisition configuration file before acquiring images from a camera. The camera configuration file has the extension **.ccf**. For applications that use both a frame grabber and camera, both device’s parameter settings are saved in this file.

CamExpert, the camera interfacing tool provided with Sapera LT, supplies sets of 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, both 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 cameras (select other components as required). Currently, Sapera LT supports GenICam GenCP CL and GigE Vision standards (including all mandatory feature requirements).



Sapera LT SDK

Installation of Sapera LT Acquisition Components

Please choose which Sapera LT acquisition components you want to install

☑ Teledyne DALSA frame grabbers and CameraLink GenCP compatible cameras

□ GigE Vision cameras

□ Teledyne DALSA 3D profile sensors

□ USB3 Vision Cameras

When upgrading from a previous version this dialog is not displayed and the same directory structure and component configuration are used.

⚠	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.

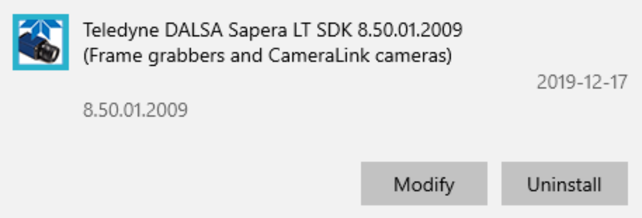
Upgrading Previous Versions of Sapera LT

Sapera LT 7.30 and Higher

When upgrading from Sapera LT version 7.30 or higher, uninstalling is not required; proceed directly to installation of Sapera LT 8.50.

⚠	Note: The installation option is persistent. When upgrading to a newer version of Sapera LT, the previously chosen option (frame grabbers only/GigE Vision cameras only/All components) is used. If you want to change option you must uninstall Sapera LT first.

To verify the current installation option, check the Sapera LT SDK entry in the **Windows Settings > Apps and features** section. For example, if the **Frame grabbers and CameraLink cameras** option was selected, it is indicated in the Teledyne DALSA Sapera LT SDK program name as shown below (click the entry to view the full text).



Teledyne DALSA Sapera LT SDK 8.50.01.2009
(Frame grabbers and CameraLink cameras)

2019-12-17

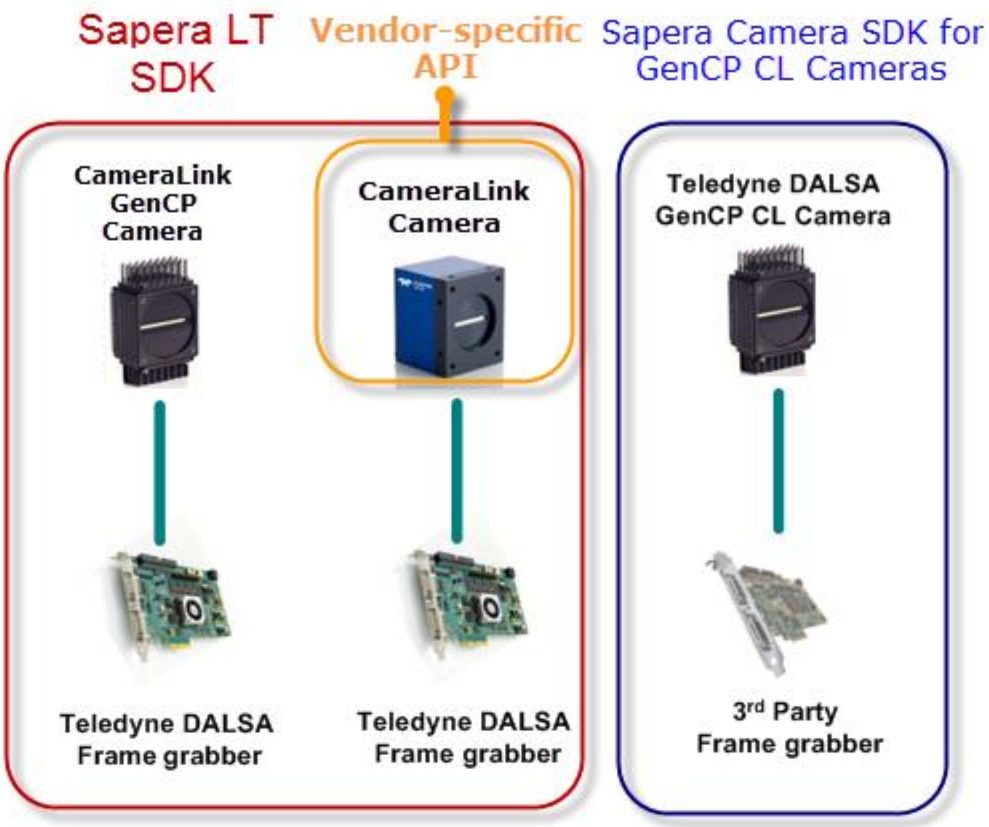
8.50.01.2009

Sapera LT 7.20 and Lower

Before installing Sapera LT 8.50, uninstall Sapera LT from the **Windows Settings > Apps and features** section.

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)


Sapera LT SDK
Vendor-specific API
Sapera Camera SDK for GenCP CL Cameras
CameraLink GenCP Camera
CameraLink Camera
Teledyne DALSA GenCP CL Camera
Teledyne DALSA Frame grabber
Teledyne DALSA Frame grabber
3rd Party Frame grabber

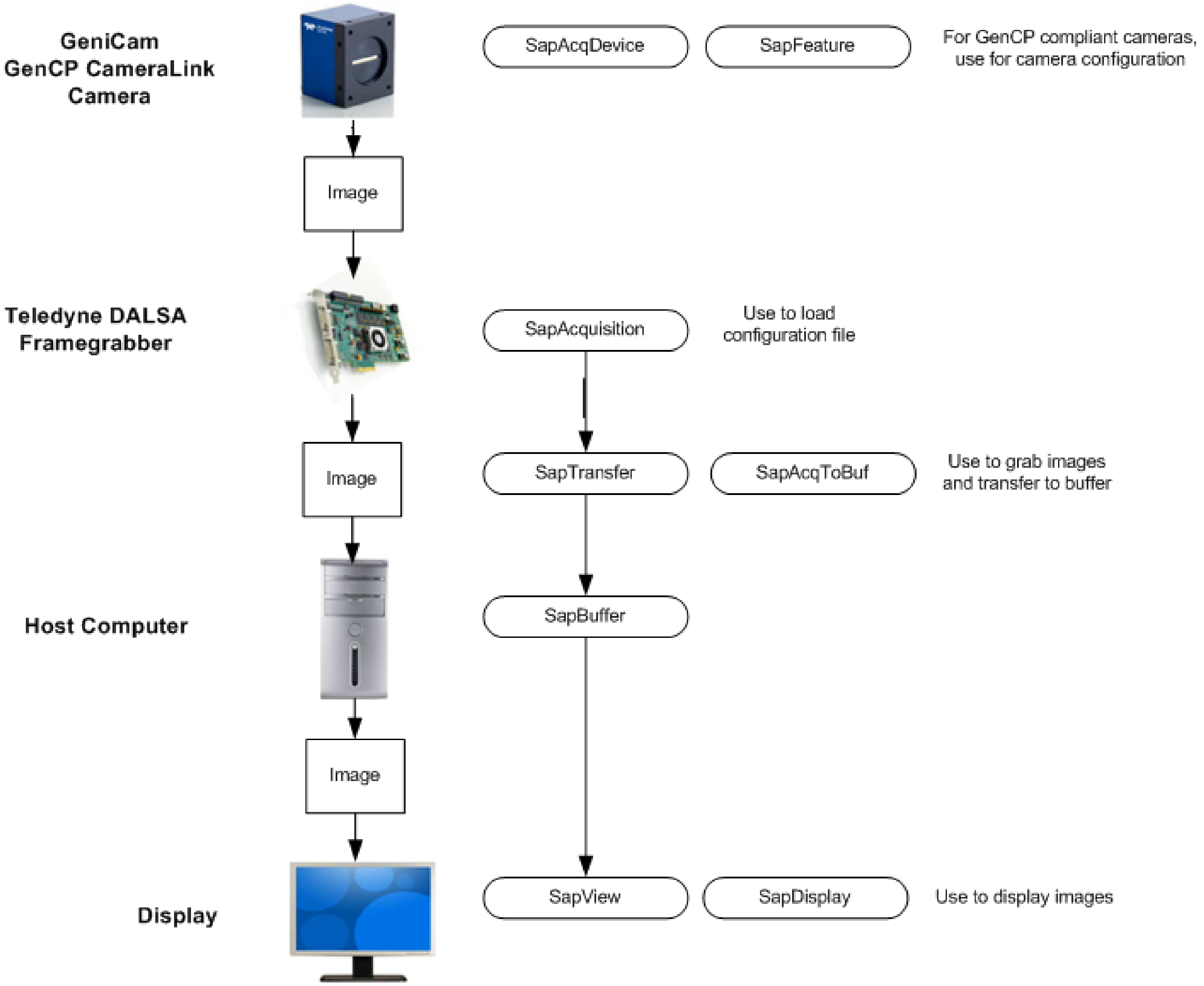
Currently, Sapera LT supports GenICam GenCP CL and GigE Vision standards (including all mandatory feature requirements).

⚠	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

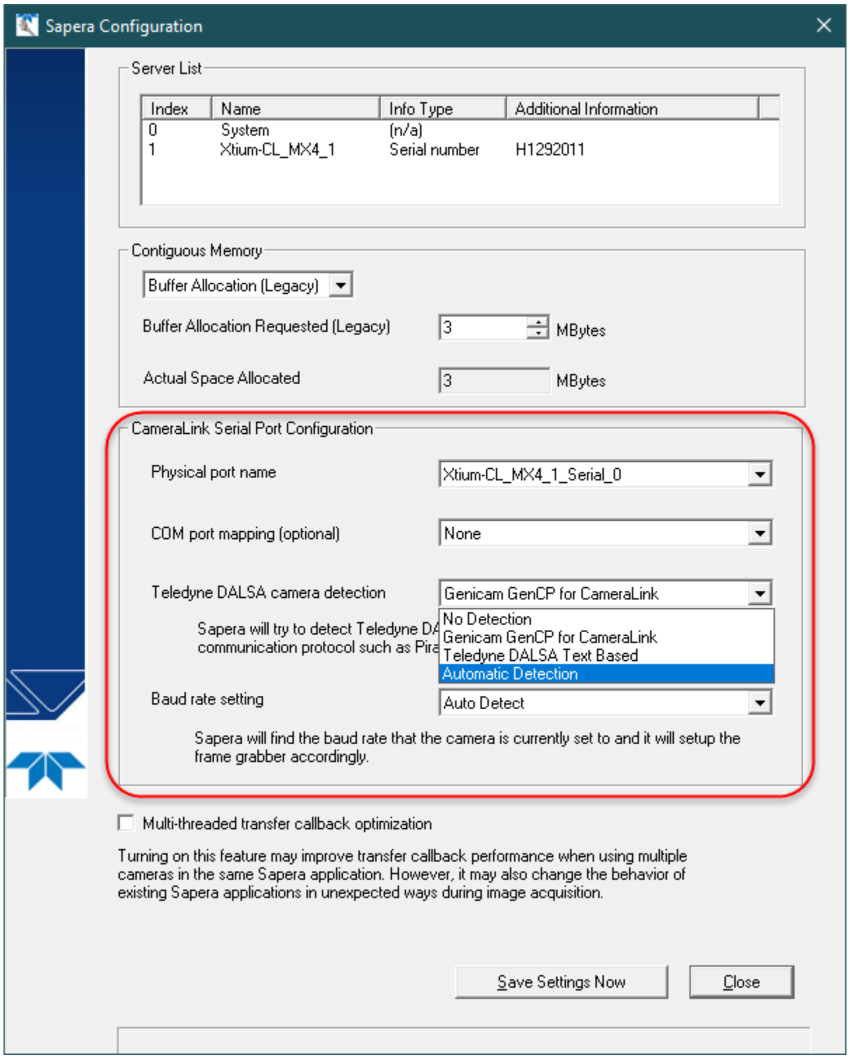
The following diagram represents a typical application flow showing the Sapera LT object types associated with each component or stage:



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.



Sapera Configuration

Server List

Index	Name	Info Type	Additional Information
0	System	(n/a)	
1	Xtium-CL_MX4_1	Serial number	H1292011

Contiguous Memory

Buffer Allocation (Legacy)

Buffer Allocation Requested (Legacy) 3 MBytes

Actual Space Allocated 3 MBytes

CameraLink Serial Port Configuration

Physical port name: Xtium-CL_MX4_1_Serial_0

COM port mapping (optional): None

Teledyne DALSA camera detection: Genicam GenCP for CameraLink

Sapera will try to detect Teledyne DA
communication protocol such as Pira

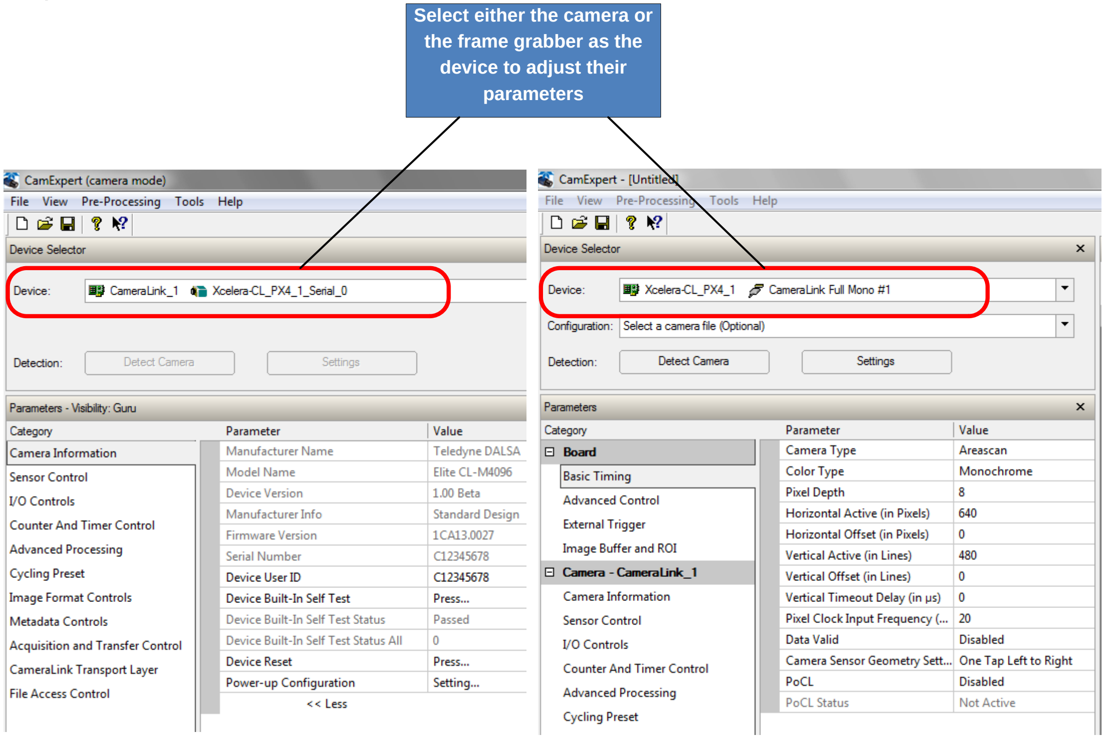
Baud rate setting: Auto Detect

Sapera will find the baud rate that the camera is currently set to and it will setup the frame grabber accordingly.

□ Multi-threaded transfer callback optimization

Turning on this feature may improve transfer callback performance when using multiple cameras in the same Sapera application. However, it may also change the behavior of existing Sapera applications in unexpected ways during image acquisition.

2. Start the CamExpert application. In the Device tab, 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.


Select either the camera or the frame grabber as the device to adjust their parameters

CamExpert (camera mode)

Device: CameraLink_1 Xcelera-CL_PX4_1_Serial_0

Detection:

Parameters - Visibility: Guru

Category	Parameter	Value
Camera Information	Manufacturer Name	Teledyne DALSA
Sensor Control	Model Name	Elite CL-M4096
I/O Controls	Device Version	1.00 Beta
Counter And Timer Control	Manufacturer Info	Standard Design
Advanced Processing	Firmware Version	1CA13.0027
Cycling Preset	Serial Number	C12345678
Image Format Controls	Device User ID	C12345678
Metadata Controls	Device Built-In Self Test	Press...
Acquisition and Transfer Control	Device Built-In Self Test Status	Passed
CameraLink Transport Layer	Device Built-In Self Test Status All	0
File Access Control	Device Reset	Press...
	Power-up Configuration	Setting...
	<< Less	

CamExpert - [Untitled]

Device: Xcelera-CL_PX4_1 CameraLink Full Mono #1

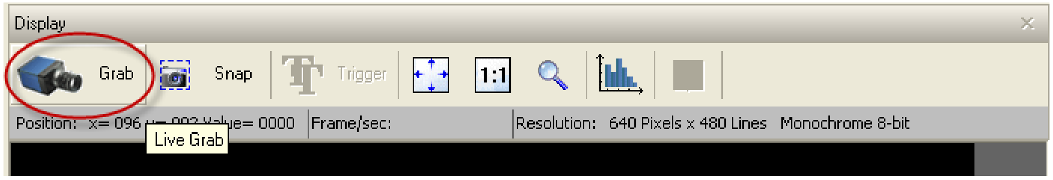
Configuration: Select a camera file (Optional)

Detection:

Parameters

Category	Parameter	Value
⊟ Board	Camera Type	Areascan
Basic Timing	Color Type	Monochrome
Advanced Control	Pixel Depth	8
External Trigger	Horizontal Active (in Pixels)	640
Image Buffer and ROI	Horizontal Offset (in Pixels)	0
⊟ Camera - CameraLink_1	Vertical Active (in Lines)	480
Camera Information	Vertical Offset (in Lines)	0
Sensor Control	Vertical Timeout Delay (in µs)	0
I/O Controls	Pixel Clock Input Frequency (...	20
Counter And Timer Control	Data Valid	Disabled
Advanced Processing	Camera Sensor Geometry Sett...	One Tap Left to Right
Cycling Preset	PoCL	Disabled
	PoCL Status	Not Active

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



Position: x= 096 y= 002 Value= 0000 Frame/sec: Resolution: 640 Pixels x 480 Lines Monochrome 8-bit

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

Any ASCII Text-based (or other) CameraLink Camera and Teledyne DALSA Frame grabber

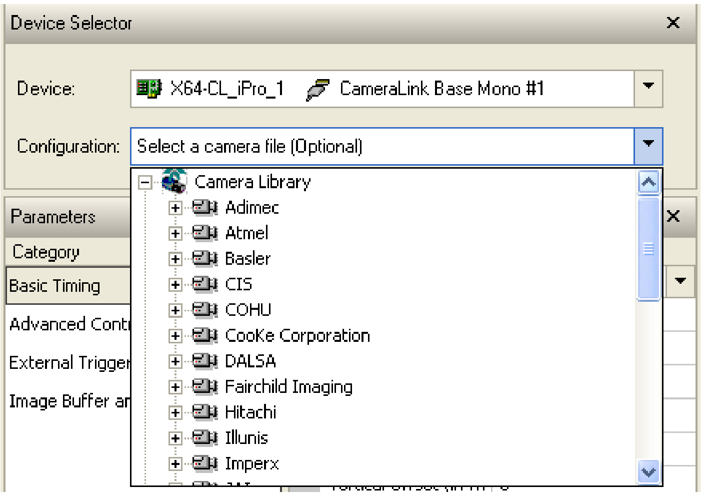
The Teledyne DALSA frame grabber requires a camera configuration file (**.ccf*) to program the frame grabber to match the characteristics of the ASCII text-based (non-GenCP) CameraLink Teledyne DALSA or 3rd party camera (for example, cameras that use three letter serial port commands). Teledyne DALSA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

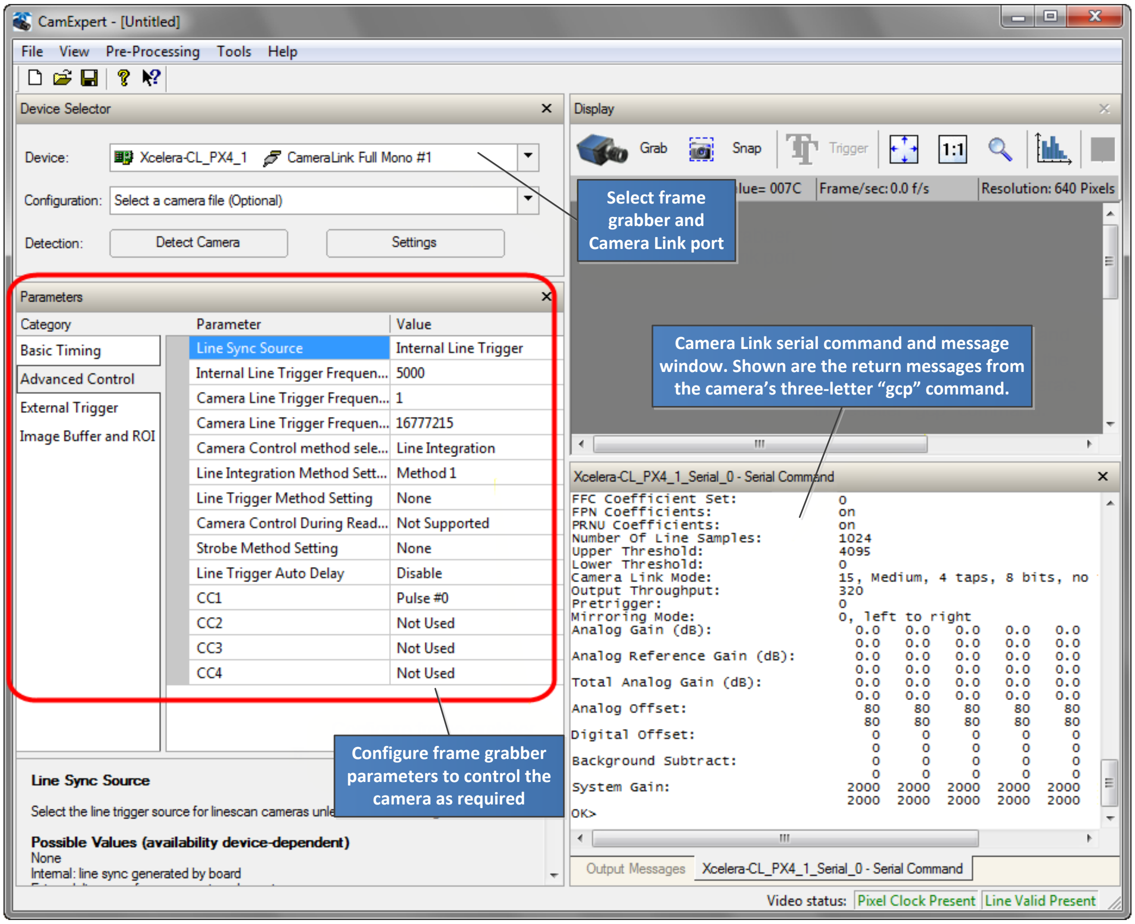
1. Start the CamExpert application. In the Device tab, select the required frame grabber, so that you can adjust its parameters.
2. If available, select the camera configuration file for the camera.



Device: X64-CL_iPro_1 CameraLink Base Mono #1

Configuration: Select a camera file (Optional)

3. Use the CameraLink Serial Command tab in CamExpert to send ASCII commands to the camera, so that you can adjust its parameters.



Device: Xcelera-CL_PX4_1 CameraLink Full Mono #1

Configuration: Select a camera file (Optional)

Detection: Detect Camera | Settings

Select frame grabber and Camera Link port

lue= 007C | Frame/sec: 0.0 f/s | Resolution: 640 Pixels

Camera Link serial command and message window. Shown are the return messages from the camera's three-letter "gcp" command.

Parameters

Category	Parameter	Value
Basic Timing	Line Sync Source	Internal Line Trigger
Advanced Control	Internal Line Trigger Frequen...	5000
External Trigger	Camera Line Trigger Frequen...	1
Image Buffer and ROI	Camera Line Trigger Frequen...	16777215
	Camera Control method sele...	Line Integration
	Line Integration Method Sett...	Method 1
	Line Trigger Method Setting	None
	Camera Control During Read...	Not Supported
	Strobe Method Setting	None
	Line Trigger Auto Delay	Disable
	CC1	Pulse #0
	CC2	Not Used
	CC3	Not Used
	CC4	Not Used

Line Sync Source

Select the line trigger source for linescan cameras unle

Possible Values (availability device-dependent)
None
Internal: line sync generated by board

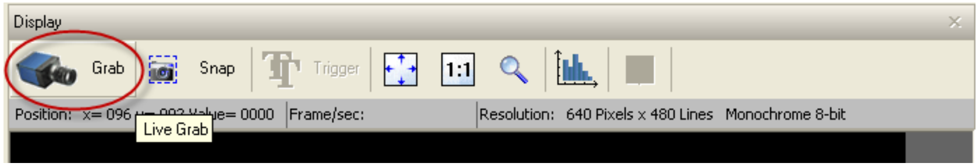
Configure frame grabber parameters to control the camera as required

Xcelera-CL_PX4_1_Serial_0 - Serial Command

```
FFC Coefficient Set:           0
FPN Coefficients:              on
PRNU Coefficients:             on
Number Of Line Samples:        1024
Upper Threshold:               4095
Lower Threshold:               0
Camera Link Mode:              15, Medium, 4 taps, 8 bits, no
Output Throughput:             320
Pretrigger:                    0
Mirroring Mode:                0, left to right
Analog Gain (dB):                0.0    0.0    0.0    0.0    0.0
                                 0.0    0.0    0.0    0.0    0.0
Analog Reference Gain (dB):      0.0    0.0    0.0    0.0    0.0
                                 0.0    0.0    0.0    0.0    0.0
Total Analog Gain (dB):          0.0    0.0    0.0    0.0    0.0
                                 0.0    0.0    0.0    0.0    0.0
Analog Offset:                    80     80     80     80     80
                                  80     80     80     80     80
Digital Offset:                    0      0      0      0      0
                                   0      0      0      0      0
Background Subtract:               0      0      0      0      0
                                   0      0      0      0      0
System Gain:                    2000   2000   2000   2000   2000
                                2000   2000   2000   2000   2000
OK>
```

Video status: Pixel Clock Present | Line Valid Present

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



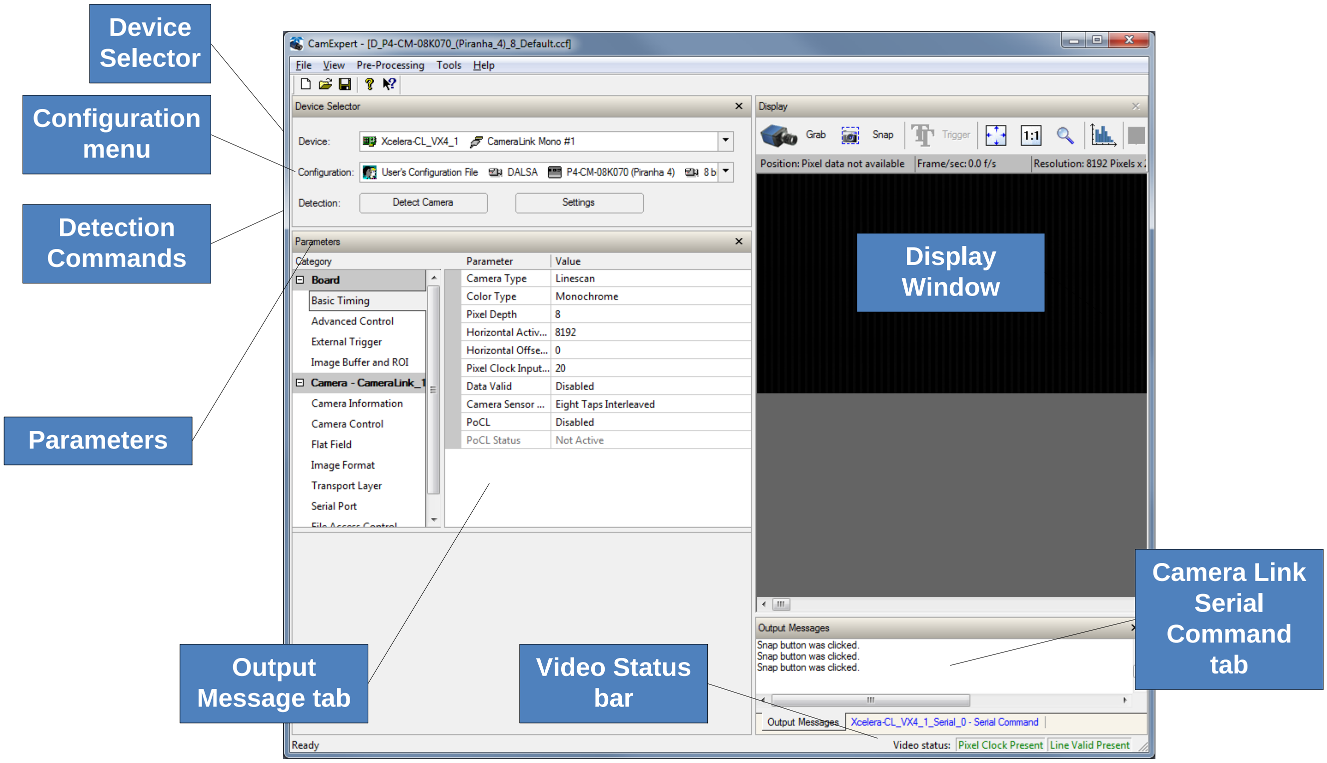
Position: x= 096 ... Value= 0000 | Frame/sec: | Resolution: 640 Pixels x 480 Lines Monochrome 8-bit

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. Features include:

- 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 controls as defined by the camera’s XML file.
- Groups acquisition parameters into related categories for easier access to any specific parameter.
- Intelligent editing of video timings through a locking mechanism that allows explicit modification of some values and automatic recalculation of the remaining ones.
- Live acquisition display window which allows immediate verification of timing or control parameters without the need to run a separate acquisition program



Device: Xcelera-CL_VX4_1 CameraLink Mono #1

Configuration: User's Configuration File DALSA P4-CM-08K070 (Piranha 4) 8 b

Detection:

Parameter	Value
Camera Type	Linescan
Color Type	Monochrome
Pixel Depth	8
Horizontal Activ...	8192
Horizontal Offse...	0
Pixel Clock Input...	20
Data Valid	Disabled
Camera Sensor ...	Eight Taps Interleaved
PoCL	Disabled
PoCL Status	Not Active

Position: Pixel data not available | Frame/sec: 0.0 f/s | Resolution: 8192 Pixels x

```
Snap button was clicked.
Snap button was clicked.
Snap button was clicked.
```

Video status: Pixel Clock Present | Line Valid Present

- **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 panel**: 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 panel**: 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. These are:

Grab Freeze	**Acquisition control button:** Click once to start live grab, click again to stop.
Snap	**Single frame grab:** Click to acquire one frame from device.
T Trigger	**Software trigger button:** With the I/O control parameters set to Trigger Enabled / Software Trigger type, click to send a single software trigger command.
✥ 1:1 🔍	**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 Panel**: 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, 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 following utilities that can be used to monitor Sapera LT hardware and software events, error messages, as well as frame grabber configuration and diagnostics:

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 Tool	Generates color correction coefficients (if supported) that are used to adjust the camera sensor’s color response.

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. This is very useful since one can use the Sapera Monitor tool to debug application s and identify problems without having to code event handlers.

The key advantage to Sapera Monitor is that it can run concurrently with CamExpert or your own application. This can be useful for debugging applications and identifying problems without having to code event handlers.

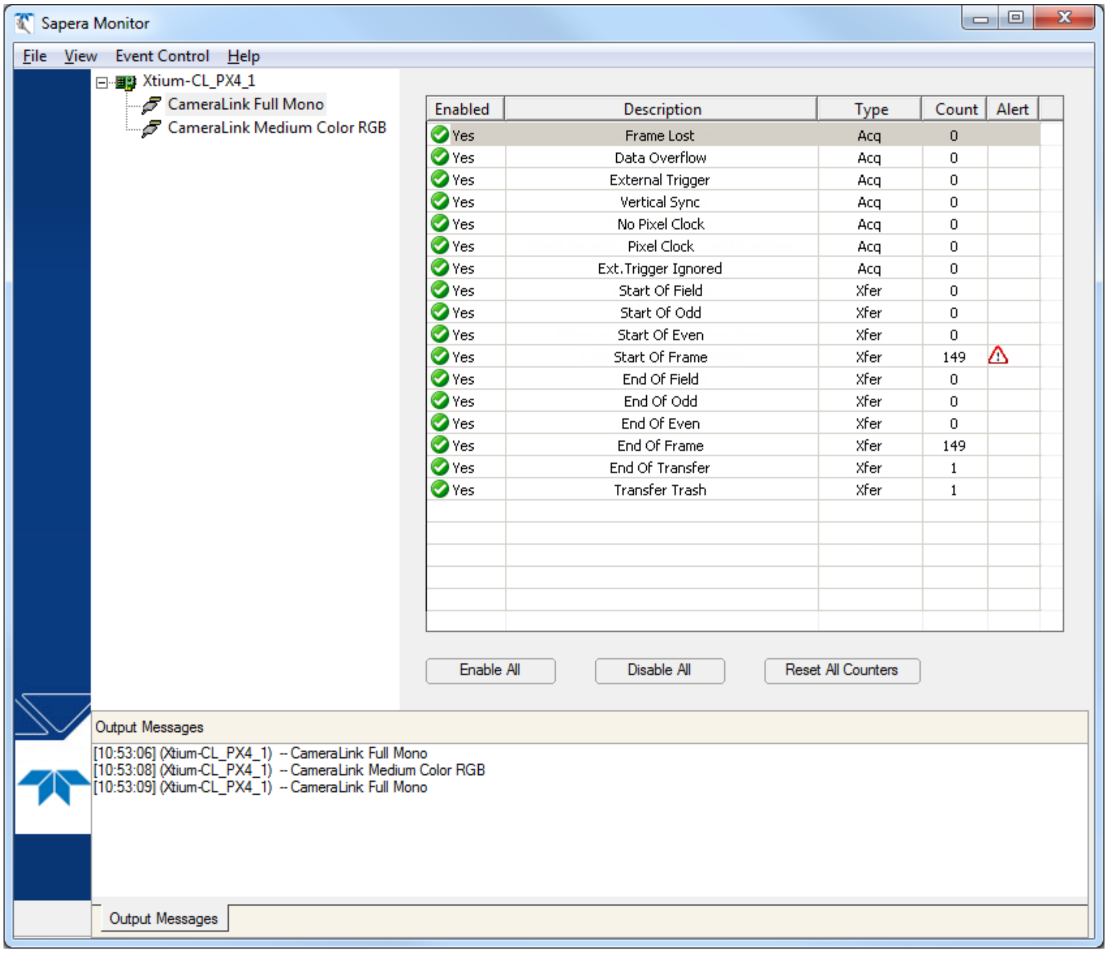
To launch the Sapera Monitor use the **Start•All Programs•Teledyne DALSA•Sapera LT•Sapera Monitor** menu shortcut.

⚠	Note: 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. In addition, when using Teledyne DALSA Genie devices, you must start a Sapera application, such as CamExpert, that uses the device you want to monitor, before launching 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 pane: displays the messages generated by the selected monitored events.

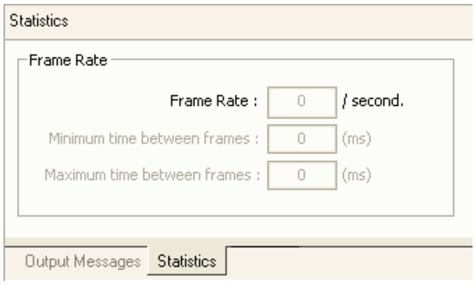


Enabled	Description	Type	Count	Alert
✓ Yes	Frame Lost	Acq	0	
✓ Yes	Data Overflow	Acq	0	
✓ Yes	External Trigger	Acq	0	
✓ Yes	Vertical Sync	Acq	0	
✓ Yes	No Pixel Clock	Acq	0	
✓ Yes	Pixel Clock	Acq	0	
✓ Yes	Ext.Trigger Ignored	Acq	0	
✓ Yes	Start Of Field	Xfer	0	
✓ Yes	Start Of Odd	Xfer	0	
✓ Yes	Start Of Even	Xfer	0	
✓ Yes	Start Of Frame	Xfer	149	⚠
✓ Yes	End Of Field	Xfer	0	
✓ Yes	End Of Odd	Xfer	0	
✓ Yes	End Of Even	Xfer	0	
✓ Yes	End Of Frame	Xfer	149	
✓ Yes	End Of Transfer	Xfer	1	
✓ Yes	Transfer Trash	Xfer	1	

```
[10:53:06] (Xtium-CL_PX4_1)  -- CameraLink Full Mono
[10:53:08] (Xtium-CL_PX4_1)  -- CameraLink Medium Color RGB
[10:53:09] (Xtium-CL_PX4_1)  -- CameraLink Full Mono
```

Statistic Tab

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



Frame Rate : 0 / second.

Minimum time between frames : 0 (ms)

Maximum time between frames : 0 (ms)

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.

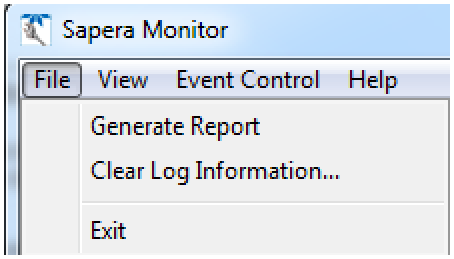
Sapera Monitor Menu Commands

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

File Menu Commands

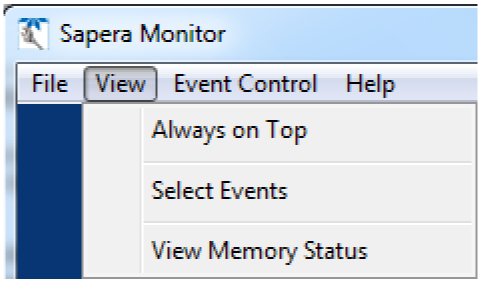
The **File•Generate Report** command generates a text file report that includes all event settings and messages included in the current Output Messages pane.

The **File•Clear Log Information** command clears the current Output Messages pane.

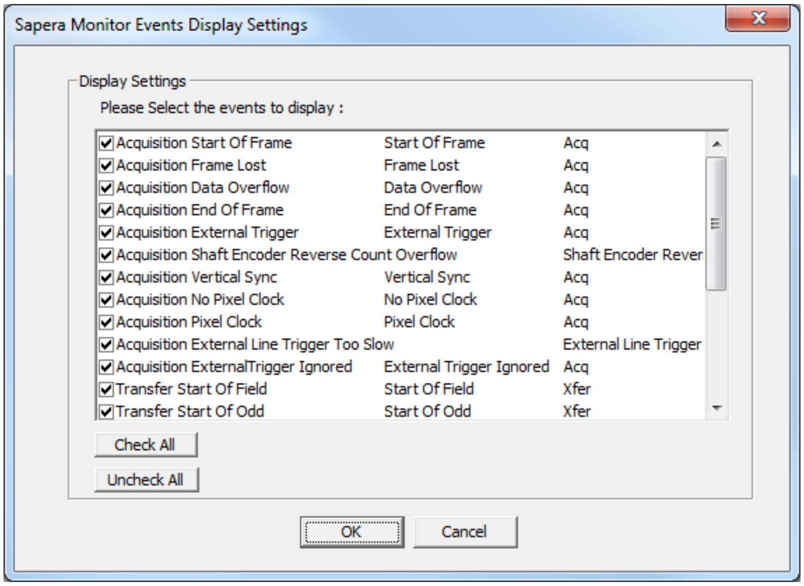


View Menu Commands

The **View•Always on Top** command displays the Sapera Monitor on top of any other windows that may be visible on the desktop.



The **View•Select Events** command opens the Sapera Monitor Events Display Settings dialog which allows you to specify the events to display in the Event table.


Sapera Monitor Events Display Settings

Display Settings

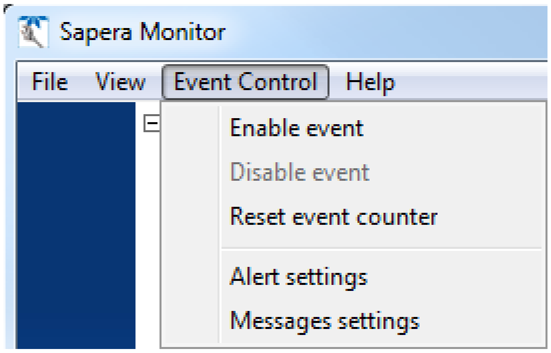
Please Select the events to display :

☑ Acquisition Start Of Frame	Start Of Frame	Acq
☑ Acquisition Frame Lost	Frame Lost	Acq
☑ Acquisition Data Overflow	Data Overflow	Acq
☑ Acquisition End Of Frame	End Of Frame	Acq
☑ Acquisition External Trigger	External Trigger	Acq
☑ Acquisition Shaft Encoder Reverse Count Overflow		Shaft Encoder Rever
☑ Acquisition Vertical Sync	Vertical Sync	Acq
☑ Acquisition No Pixel Clock	No Pixel Clock	Acq
☑ Acquisition Pixel Clock	Pixel Clock	Acq
☑ Acquisition External Line Trigger Too Slow		External Line Trigger
☑ Acquisition ExternalTrigger Ignored	External Trigger Ignored	Acq
☑ Transfer Start Of Field	Start Of Field	Xfer
☑ Transfer Start Of Odd	Start Of Odd	Xfer

Event Control Menu Commands

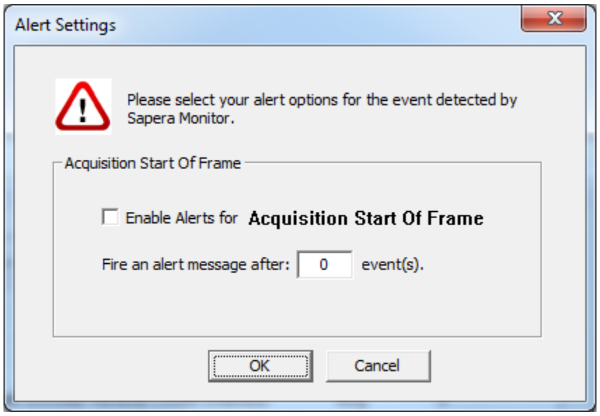
The **Event Control•Enable** and **Disable** commands enable or disable the currently selected event in Event table.

The **Event Control•Reset event count** command returns the event counter to zero for the currently selected event in Event table.



Alert Settings

You can specify the alert options for each available event using the Alert Settings command, available through the Event Control menu or by right-clicking on the selected event. The Alert Settings dialog allows you to enable or disable alerts for the event and to specify the number of events required to generate an alert.



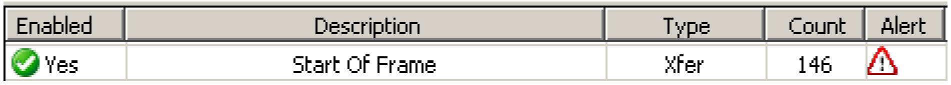
Please select your alert options for the event detected by Sapera Monitor.

Acquisition Start Of Frame

☐ Enable Alerts for **Acquisition Start Of Frame**

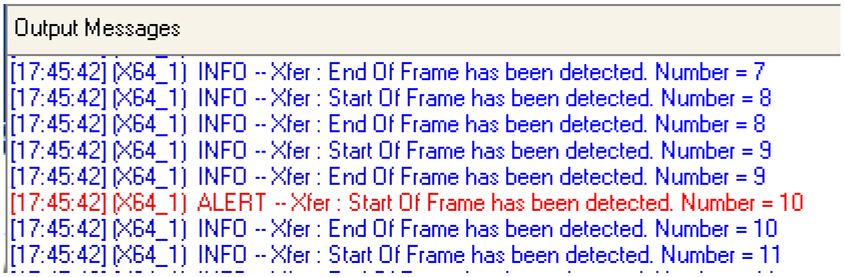
Fire an alert message after: 0 event(s).

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



Enabled	Description	Type	Count	Alert
✓ Yes	Start Of Frame	Xfer	146	⚠

An alert message, in red, also appears in the Output Messages Settings pane.

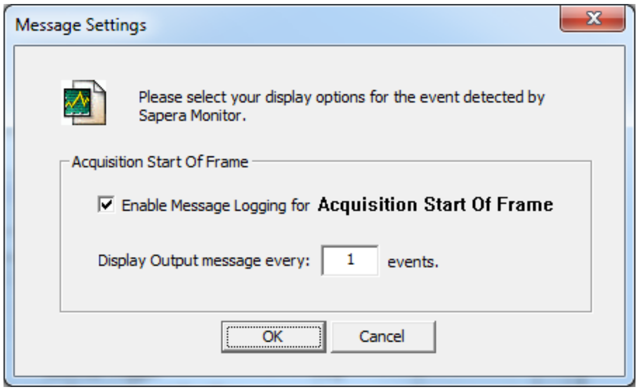


Output Messages

```
[17:45:42] [X64_1] INFO -- Xfer : End Of Frame has been detected. Number = 7
[17:45:42] [X64_1] INFO -- Xfer : Start Of Frame has been detected. Number = 8
[17:45:42] [X64_1] INFO -- Xfer : End Of Frame has been detected. Number = 8
[17:45:42] [X64_1] INFO -- Xfer : Start Of Frame has been detected. Number = 9
[17:45:42] [X64_1] INFO -- Xfer : End Of Frame has been detected. Number = 9
[17:45:42] [X64_1] ALERT -- Xfer : Start Of Frame has been detected. Number = 10
[17:45:42] [X64_1] INFO -- Xfer : End Of Frame has been detected. Number = 10
[17:45:42] [X64_1] INFO -- Xfer : Start Of Frame has been detected. Number = 11
```

Message Settings

You can specify the events to enable message logging and the number of events required to generate a log message using the Message Settings command, available through the Event Control menu or by right-clicking on the selected event. The log messages appear in the Output Message pane.



Message Settings

Please select your display options for the event detected by Sapera Monitor.

Acquisition Start Of Frame

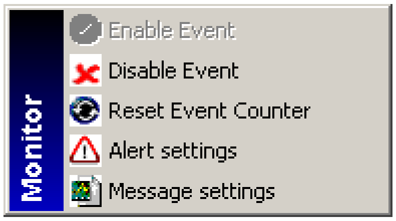
☑ Enable Message Logging for **Acquisition Start Of Frame**

Display Output message every: 1 events.

Using Sapera Monitor

To use the Sapera Monitor tool to monitor a device:

- Run a Sapera application, such as CamExpert, that uses the device.
- Launch the Sapera Monitor application
- In the Sapera Monitor Device directory tree, select the device.
- In the Sapera Monitor Event table, select the events to monitor. Double-clicking on the row containing the event toggles the Enabled status between ✓ Yes and ✗ No.
  Right-clicking on the event opens a context menu which provides easy access to commands to reset the event counter and modify the event Alert settings or Message settings.



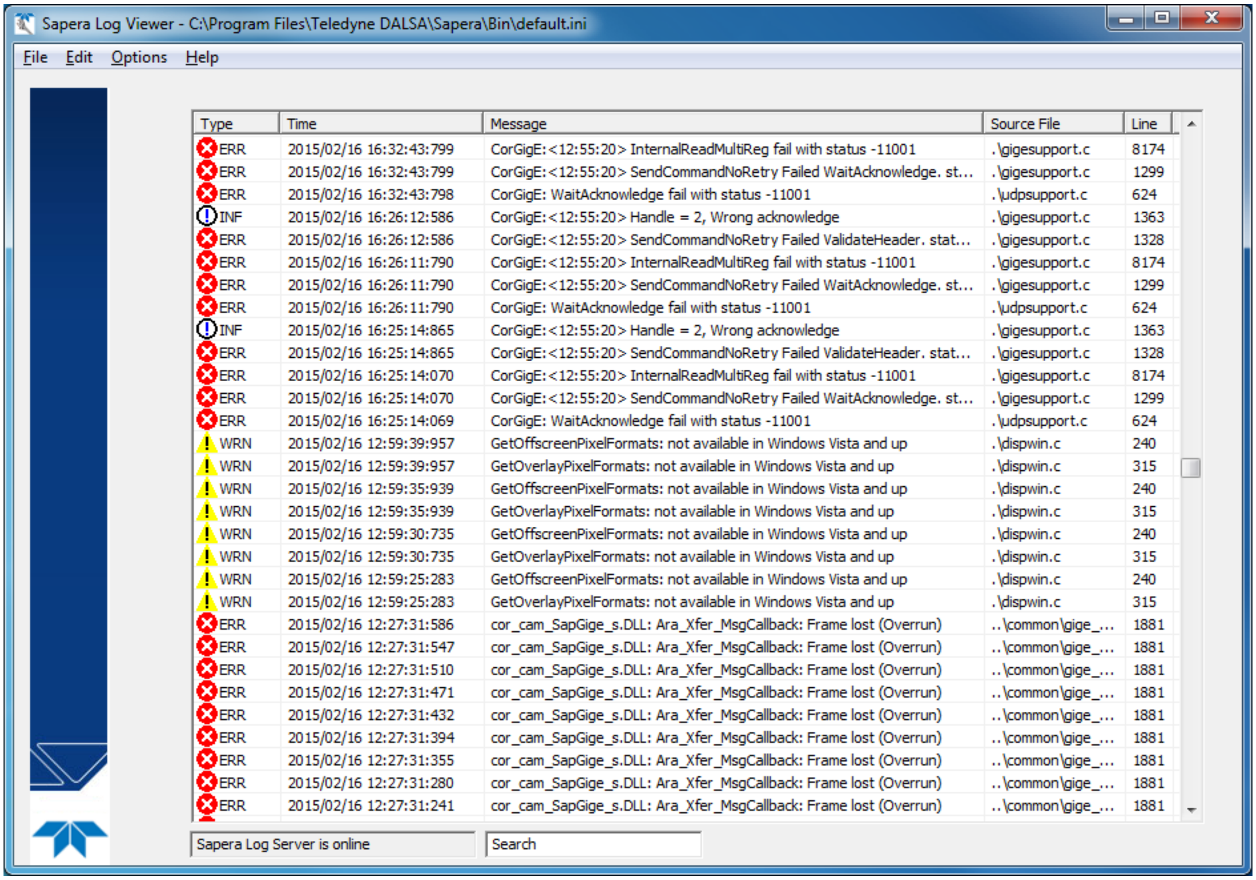
These commands are also available through the Event Control Menu.
Alternatively, you can use the Enable All and Disable All buttons to quickly clear or select events.

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.

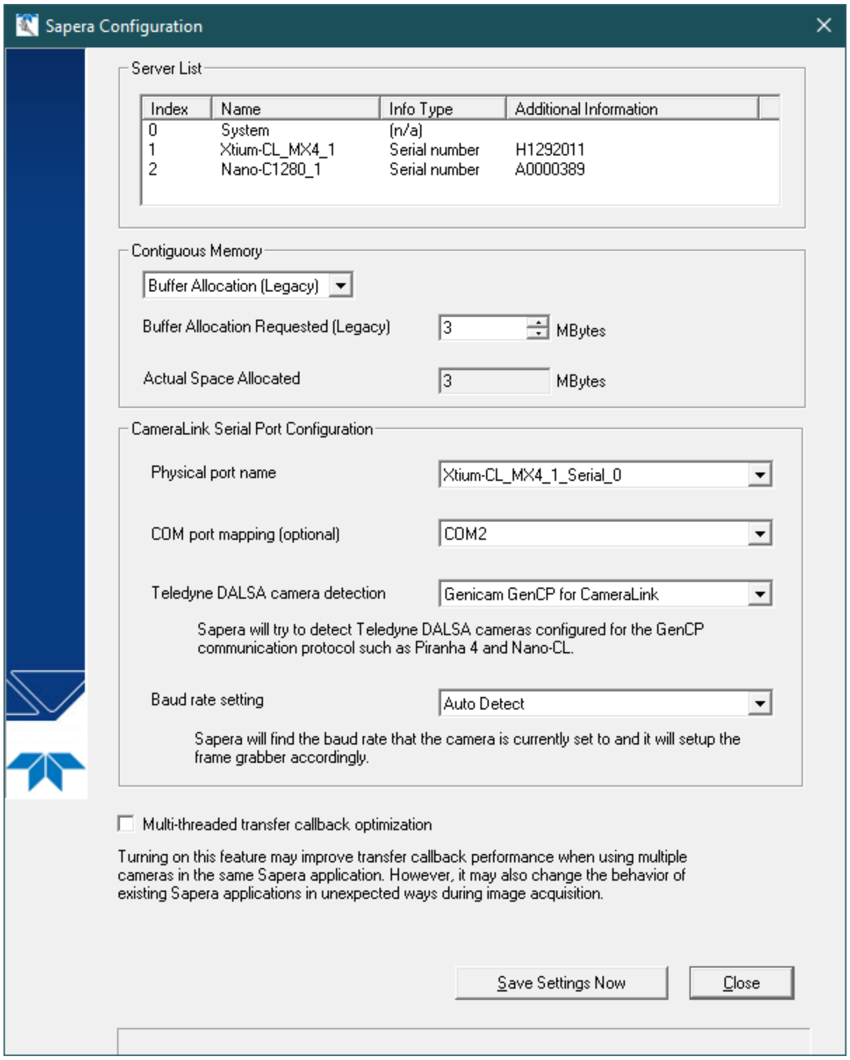
Refer to the utility's online help for more information on using the Sapera Log Viewer.


Sapera Log Viewer - C:\Program Files\Teledyne DALSA\Sapera\Bin\default.ini
File
Edit
Options
Help
Sapera Log Server is online
Search

Type	Time	Message	Source File	Line
ERR	2015/02/16 16:32:43:799	CorGigE:<12:55:20> InternalReadMultiReg fail with status -11001	.\gigesupport.c	8174
ERR	2015/02/16 16:32:43:799	CorGigE:<12:55:20> SendCommandNoRetry Failed WaitAcknowledge. st...	.\gigesupport.c	1299
ERR	2015/02/16 16:32:43:798	CorGigE: WaitAcknowledge fail with status -11001	.\udpsupport.c	624
INF	2015/02/16 16:26:12:586	CorGigE:<12:55:20> Handle = 2, Wrong acknowledge	.\gigesupport.c	1363
ERR	2015/02/16 16:26:12:586	CorGigE:<12:55:20> SendCommandNoRetry Failed ValidateHeader. stat...	.\gigesupport.c	1328
ERR	2015/02/16 16:26:11:790	CorGigE:<12:55:20> InternalReadMultiReg fail with status -11001	.\gigesupport.c	8174
ERR	2015/02/16 16:26:11:790	CorGigE:<12:55:20> SendCommandNoRetry Failed WaitAcknowledge. st...	.\gigesupport.c	1299
ERR	2015/02/16 16:26:11:790	CorGigE: WaitAcknowledge fail with status -11001	.\udpsupport.c	624
INF	2015/02/16 16:25:14:865	CorGigE:<12:55:20> Handle = 2, Wrong acknowledge	.\gigesupport.c	1363
ERR	2015/02/16 16:25:14:865	CorGigE:<12:55:20> SendCommandNoRetry Failed ValidateHeader. stat...	.\gigesupport.c	1328
ERR	2015/02/16 16:25:14:070	CorGigE:<12:55:20> InternalReadMultiReg fail with status -11001	.\gigesupport.c	8174
ERR	2015/02/16 16:25:14:070	CorGigE:<12:55:20> SendCommandNoRetry Failed WaitAcknowledge. st...	.\gigesupport.c	1299
ERR	2015/02/16 16:25:14:069	CorGigE: WaitAcknowledge fail with status -11001	.\udpsupport.c	624
WRN	2015/02/16 12:59:39:957	GetOffscreenPixelFormats: not available in Windows Vista and up	.\dispwin.c	240
WRN	2015/02/16 12:59:39:957	GetOverlayPixelFormats: not available in Windows Vista and up	.\dispwin.c	315
WRN	2015/02/16 12:59:35:939	GetOffscreenPixelFormats: not available in Windows Vista and up	.\dispwin.c	240
WRN	2015/02/16 12:59:35:939	GetOverlayPixelFormats: not available in Windows Vista and up	.\dispwin.c	315
WRN	2015/02/16 12:59:30:735	GetOffscreenPixelFormats: not available in Windows Vista and up	.\dispwin.c	240
WRN	2015/02/16 12:59:30:735	GetOverlayPixelFormats: not available in Windows Vista and up	.\dispwin.c	315
WRN	2015/02/16 12:59:25:283	GetOffscreenPixelFormats: not available in Windows Vista and up	.\dispwin.c	240
WRN	2015/02/16 12:59:25:283	GetOverlayPixelFormats: not available in Windows Vista and up	.\dispwin.c	315
ERR	2015/02/16 12:27:31:586	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:547	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:510	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:471	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:432	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:394	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:355	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:280	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881
ERR	2015/02/16 12:27:31:241	cor_cam_SapGige_s.DLL: Ara_Xfer_MsgCallback: Frame lost (Overrun)	..\common\gige_...	1881

Sapera Configuration Utility

The **Sapera Configuration** program (**SapConf.exe)** allows you to see 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. After activating this program, it displays all the servers related to the installed devices as shown in the figure below (64-bit version shown).



Index	Name	Info Type	Additional Information
0	System	(n/a)	
1	Xtium-CL_MX4_1	Serial number	H1292011
2	Nano-C1280_1	Serial number	A0000389

Buffer Allocation (Legacy)

Buffer Allocation Requested (Legacy) 3 MBytes

Actual Space Allocated 3 MBytes

Physical port name: Xtium-CL_MX4_1_Serial_0

COM port mapping (optional): COM2

Teledyne DALSA camera detection: Genicam GenCP for CameraLink

Sapera will try to detect Teledyne DALSA cameras configured for the GenCP communication protocol such as Piranha 4 and Nano-CL.

Baud rate setting: Auto Detect

Sapera will find the baud rate that the camera is currently set to and it will setup the frame grabber accordingly.

☐ Multi-threaded transfer callback optimization

Turning on this feature may improve transfer callback performance when using multiple cameras in the same Sapera application. However, it may also change the behavior of existing Sapera applications in unexpected ways during image acquisition.

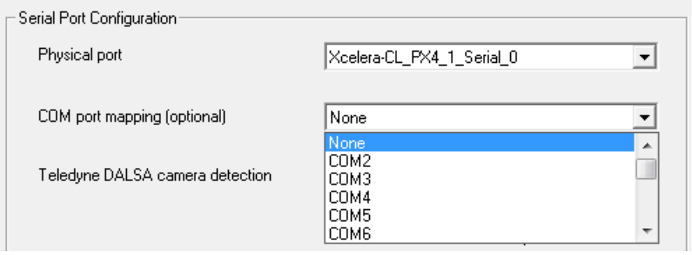
- The **System** entry represents the system server. It corresponds to the host machine (your computer) and is the only server that should always be present. The other servers correspond to the devices present within the system.
- The **CameraLink Serial Port Configuration** section allows you to select 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 3rd 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 **Serial Port Configuration > COM port mapping** parameter. For new mappings to take effect, a system reboot is generally required.



Serial Port Configuration

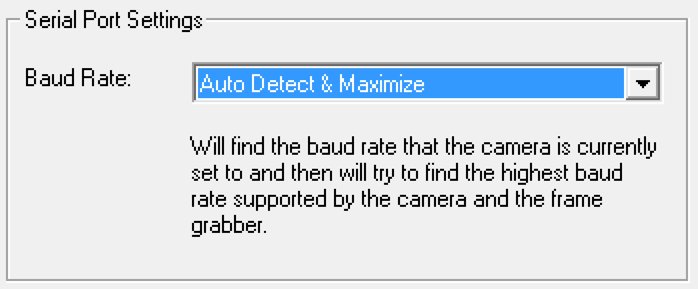
Physical port: Xcelera-CL_PX4_1_Serial_0

COM port mapping (optional): None

Teledyne DALSA camera detection

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.

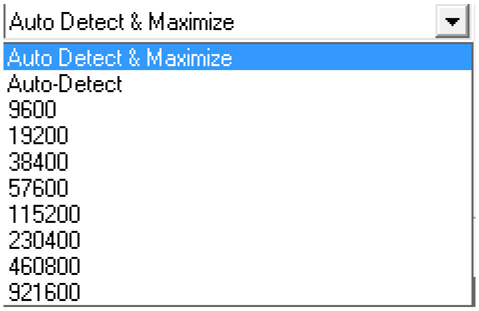


Serial Port Settings

Baud Rate: Auto Detect & Maximize

Will find the baud rate that the camera is currently set to and then will try to find the highest baud rate supported by the camera and the frame grabber.

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

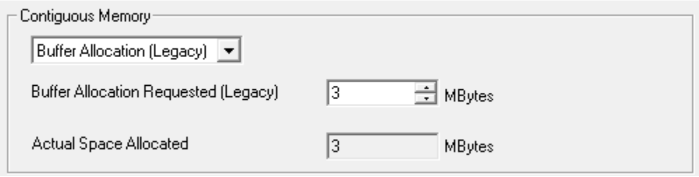


Auto Detect & Maximize

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 1MB 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.



Buffer Allocation Requested (Legacy): 3 MBytes

Actual Space Allocated: 3 MBytes

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 64kB blocks.
- Provide 48 bytes per frame buffer for buffer management. Memory for this data is reserved in chunks of 64kB blocks.
- For each frame buffer DMA table, allocate 24 bytes + 8 bytes for each 4kB of buffer. For example, for a 120x50x8 image:
  `120 * 50 = 6000 = 1.46 4kB blocks -> roundup to 2 4kB blocks.`
  Therefore 24 bytes + (2 * 8 bytes) = 40 bytes for DMA tables per frame buffer.
- Memory for this data is reserved in chunks of 64kB 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`

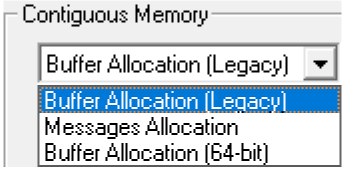
Test for any memory error when allocating host buffers. Simply use the Buffer menu of the Sapera Grab demo program (see Acquiring with Grab Demo) to allocate the number of host buffers required for your acquisition source. Feel free to test the maximum limit of host buffers possible on your host system – 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 5MB with 80,000 buffers of 2048x1024x8:

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

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

The drop-down list specifies the memory type to allocate.


Contiguous Memory
Buffer Allocation (Legacy)
Buffer Allocation (Legacy)
Messages Allocation
Buffer Allocation (64-bit)

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

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

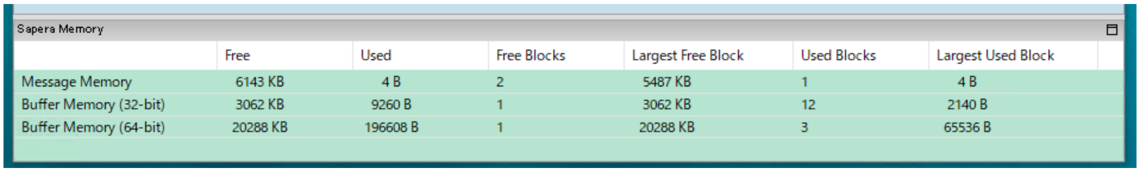
The **Requested** value displays what was requested.

The **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 16GB is requested, at boot time, Sapera LT will try to allocate one block of 16GB of memory. If this fails, it will try 2 x 8GB, then 4 x 4 GB... until it is able to allocate the amount requested. In this case, if 4 blocks of 4GB each are allocated, the largest block would show as 4GB 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.



Sapera Memory

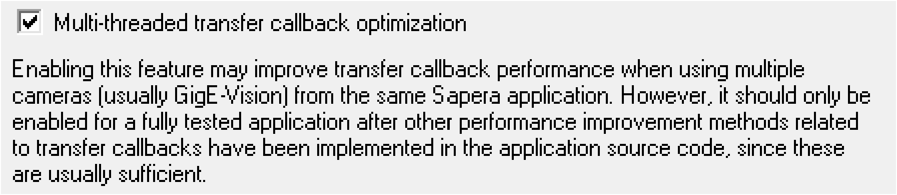
	Free	Used	Free Blocks	Largest Free Block	Used Blocks	Largest Used Block
Message Memory	6143 KB	4 B	2	5487 KB	1	4 B
Buffer Memory (32-bit)	3062 KB	9260 B	1	3062 KB	12	2140 B
Buffer Memory (64-bit)	20288 KB	196608 B	1	20288 KB	3	65536 B

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



☑ Multi-threaded transfer callback optimization

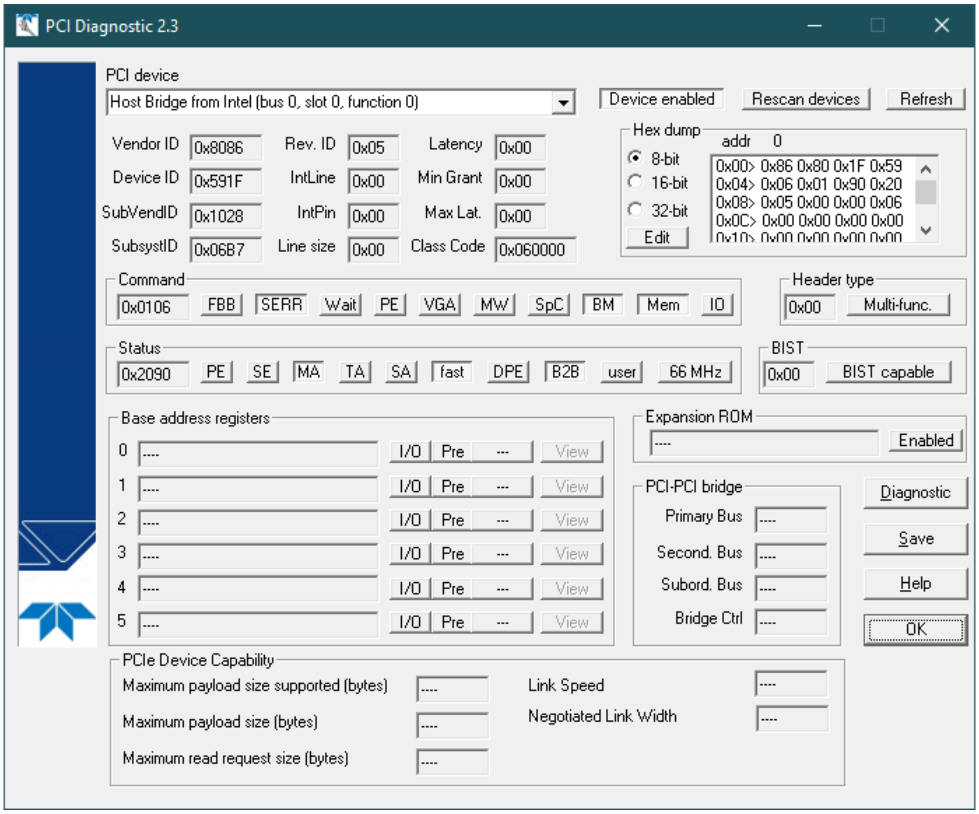
Enabling this feature may improve transfer callback performance when using multiple cameras (usually GigE-Vision) from the same Sapera application. However, it should only be enabled for a fully tested application after other performance improvement methods related to transfer callbacks have been implemented in the application source code, since these are usually sufficient.

Usage Notes When Writing Sapera Applications

- Always disable this 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 which 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.

PCI Diagnostic

The 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.



PCI Diagnostic 2.3

PCI device: Host Bridge from Intel (bus 0, slot 0, function 0)

Vendor ID 0x8086 | Rev. ID 0x05 | Latency 0x00
Device ID 0x591F | IntLine 0x00 | Min Grant 0x00
SubVendID 0x1028 | IntPin 0x00 | Max Lat. 0x00
SubsystID 0x06B7 | Line size 0x00 | Class Code 0x060000

Hex dump: ● 8-bit ○ 16-bit ○ 32-bit

```
addr    0
0x00> 0x86 0x80 0x1F 0x59
0x04> 0x06 0x01 0x90 0x20
0x08> 0x05 0x00 0x00 0x06
0x0C> 0x00 0x00 0x00 0x00
0x10> 0x00 0x00 0x00 0x00
```

Command 0x0106

Header type 0x00

Status 0x2090

BIST 0x00

Base address registers: 0 ---- | 1 ---- | 2 ---- | 3 ---- | 4 ---- | 5 ----

Expansion ROM ----

PCI-PCI bridge: Primary Bus ---- | Second. Bus ---- | Subord. Bus ---- | Bridge Ctrl ----

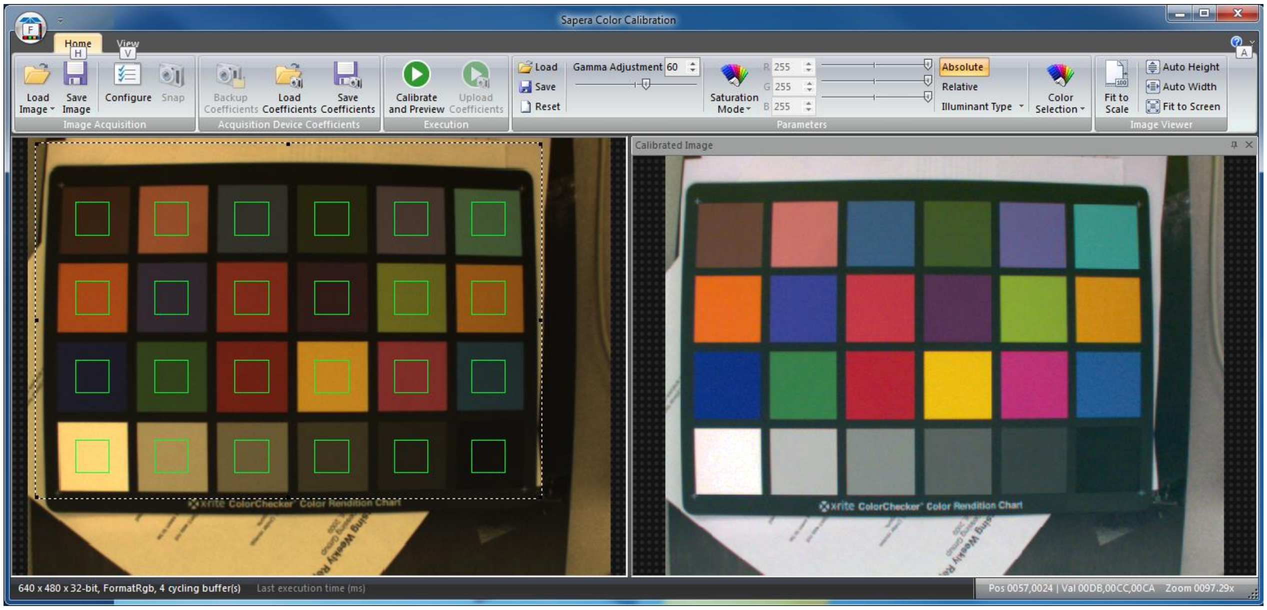
PCIe Device Capability

Maximum payload size supported (bytes) ----
Maximum payload size (bytes) ----
Maximum read request size (bytes) ----
Link Speed ----
Negotiated Link Width ----

Sapera Color Calibration Tool

The Sapera LT Color Calibration Tool 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).



640 x 480 x 32-bit, FormatRgb, 4 cycling buffer(s)   Last execution time (ms)

Pos 0057,0024 | Val 00DB,00CC,00CA   Zoom 0097.29x

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 2010 to 2017.

Source code for Sapera LT ++ based demos and examples can now be compiled as Unicode instead of ANSI. Project files provided by Sapera for Visual Studio 2012 now support both character sets. However, project files for earlier versions of Visual Studio still support ANSI only. Project files for Visual Studio 2013/2015/2017 support Unicode only.

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 more complete applications 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 in order 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 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\Example directory.

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

Demos and Examples for Frame Grabbers

The following demo programs and corresponding source code is available:

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 the Acquiring with Grab Demo section. The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital camera. The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2013.
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. The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2013.
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. The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2013.
BayerDemo (deprecated)	This program demonstrates the Bayer conversion functionality 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 Bayer conversion parameters. You may optionally apply Bayer filtering to acquired images. The minimum requirements to run this demo are a Sapera-compatible frame grabber with a digital Bayer filter camera. The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2013. This demo has been replaced by the Color Conversion demo which uses the SapColorConversion class (the SapBayer class is now deprecated).

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. The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2013.
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 to create flat field coefficients, see the Using the Flat field Demo section. The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital monochrome camera. The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2013.
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. The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2013. This example is *not available in .NET*.

The following compiled console examples and source code is available:

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.
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).
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.

The following compiled console examples and source code is available:

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.
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.
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).
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.

Generic Sapera LT Examples

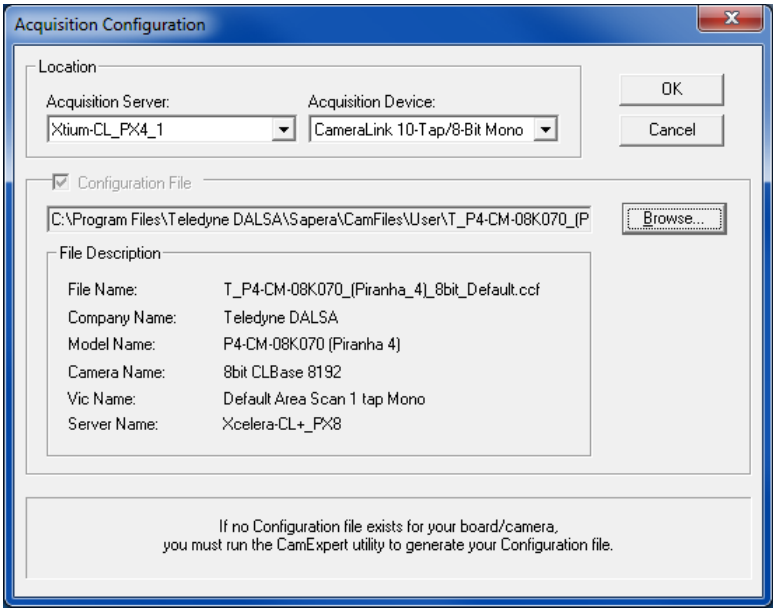
The following compiled console examples and source code is available:

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 Grab Demo:



Acquisition Configuration

Location

Acquisition Server: Xtium-CL_PX4_1

Acquisition Device: CameraLink 10-Tap/8-Bit Mono

☑ Configuration File

C:\Program Files\Teledyne DALSA\Sapera\CamFiles\User\T_P4-CM-08K070_(P

File Description

File Name:	T_P4-CM-08K070_(Piranha_4)_8bit_Default.ccf
Company Name:	Teledyne DALSA
Model Name:	P4-CM-08K070 (Piranha 4)
Camera Name:	8bit CLBase 8192
Vic Name:	Default Area Scan 1 tap Mono
Server Name:	Xcelera-CL+_PX8

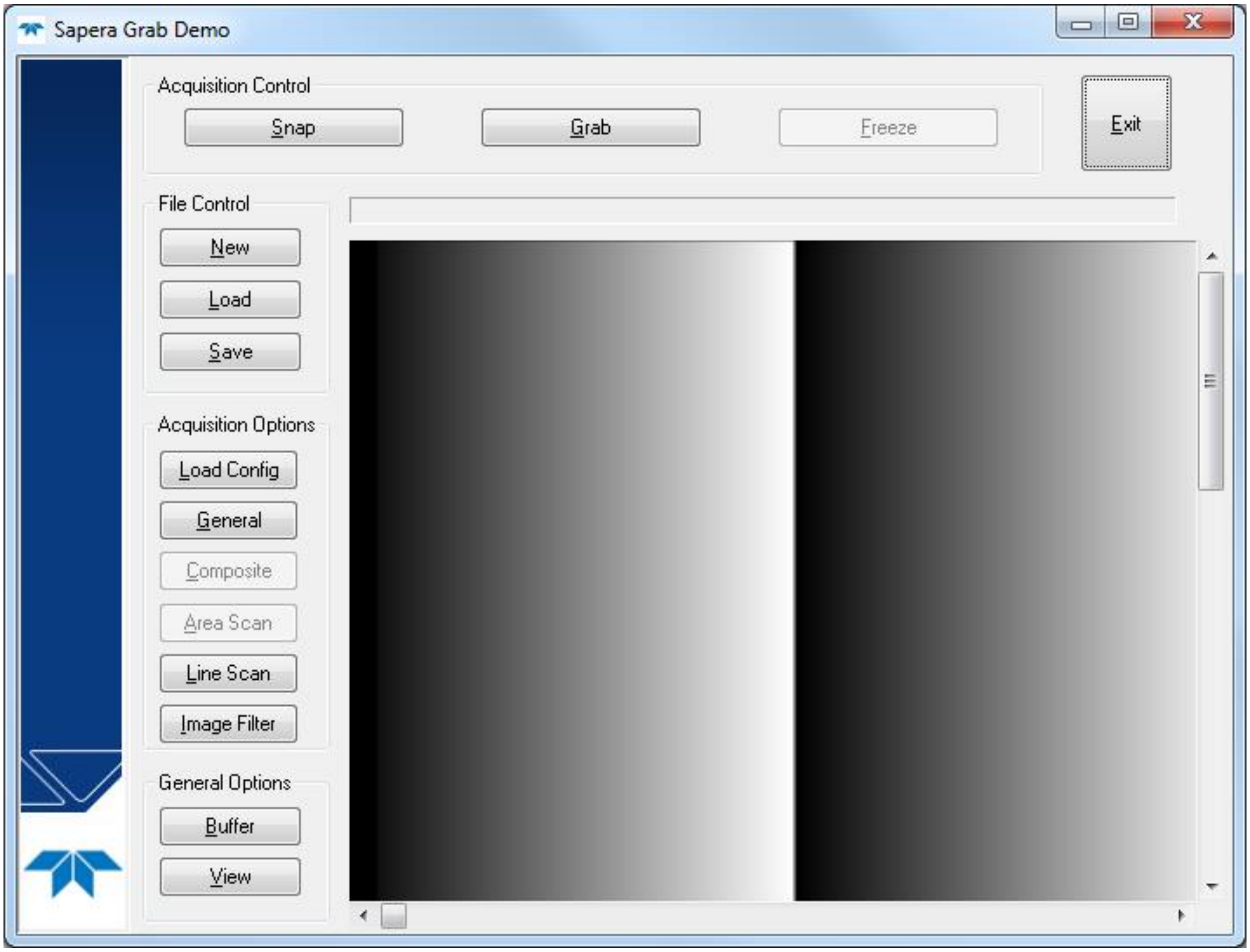
If no Configuration file exists for your board/camera,
you must run the CamExpert utility to generate your Configuration file.

You must 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).

You must then select an acquisition configuration file (CCF File) compatible with your camera from the list of available files. CamExpert must be used to generate CCF files (for example, external trigger, cropping window, and so forth).

⚠	**Note:** No frame grabber was found within your system if this dialog box does not appear. 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. It, however, only allows you to load the file from the disk.

Click OK to start the demo.



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.
- Reload the CCF file using the **Load Config** button (this overwrites all the parameters modified in step 3).
- The **Buffer** button allows you to change the number of buffers used for internal cycling and the type of buffer used (contiguous, scatter-gather, off-screen, or overlay).
- Adjust the hardware-based acquisition **Image Filter** (if supported).

Using the Flat field Demo

Program	**Start•Programs•Teledyne DALSA•Sapera LT•Demos•Flat Field Demo**
Program file	**\Program Files\Teledyne DALSA\Sapera\Demos\Classes\VC\ FlatFieldDemo\Release\FlatfieldDemo.exe**
Description	This program demonstrates Flat Field (2D Shading Correction) or Flat Line processing (1D Shading Correction), either performed by supporting Teledyne DALSA hardware or performed on the host system via the Sapera library. The program allows you to acquire a flat field or flat line reference image, and then do real time correction either in continuous or single acquisition mode. The program code may be extracted for use within your own application.
Remarks	The executable provided in the Sapera LT installation for this demo is built using Visual C++ 2013 using the MFC library. It is based on Sapera C++ classes. See the Sapera User's and Reference manuals for more information.

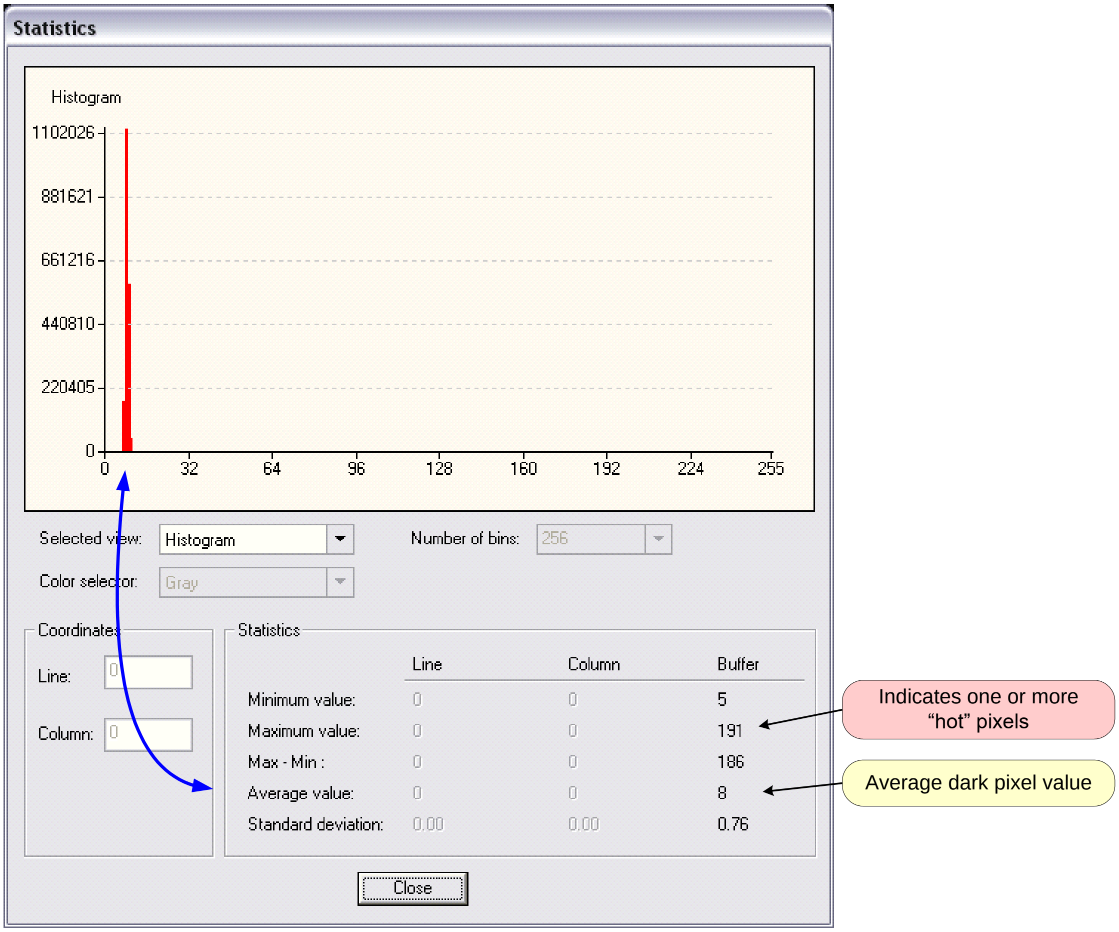
Flat Field Correction is the process of eliminating small gain differences between pixels in a CCD array. When calibrated flat field correction is applied to the image, the CCD exposed to a uniformly lighted field will have no gray level differences between pixels. The Flat Field demo automatically functions both with hardware supporting flat field processing or performs the processing via the Sapera library on the host system processor.

Set up Dark and Bright Acquisitions with the Histogram Tool

Before performing calibration, verify acquisition with a live grab. Also, at this time make preparations to grab a flat light gray level image, required for the calibration, 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.

Verify a Dark Acquisition

Close the camera lens iris and cover the lens with a lens cap. Using CamExpert, click on the grab button and then the histogram button. The following figure shows a typical histogram for a Genie grabbing a very dark image.



Selected view: Histogram

Number of bins: 256

Color selector: Gray

Coordinates

Line: 0

Column: 0

Statistics

	Line	Column	Buffer
Minimum value:	0	0	5
Maximum value:	0	0	191
Max - Min :	0	0	186
Average value:	0	0	8
Standard deviation:	0.00	0.00	0.76

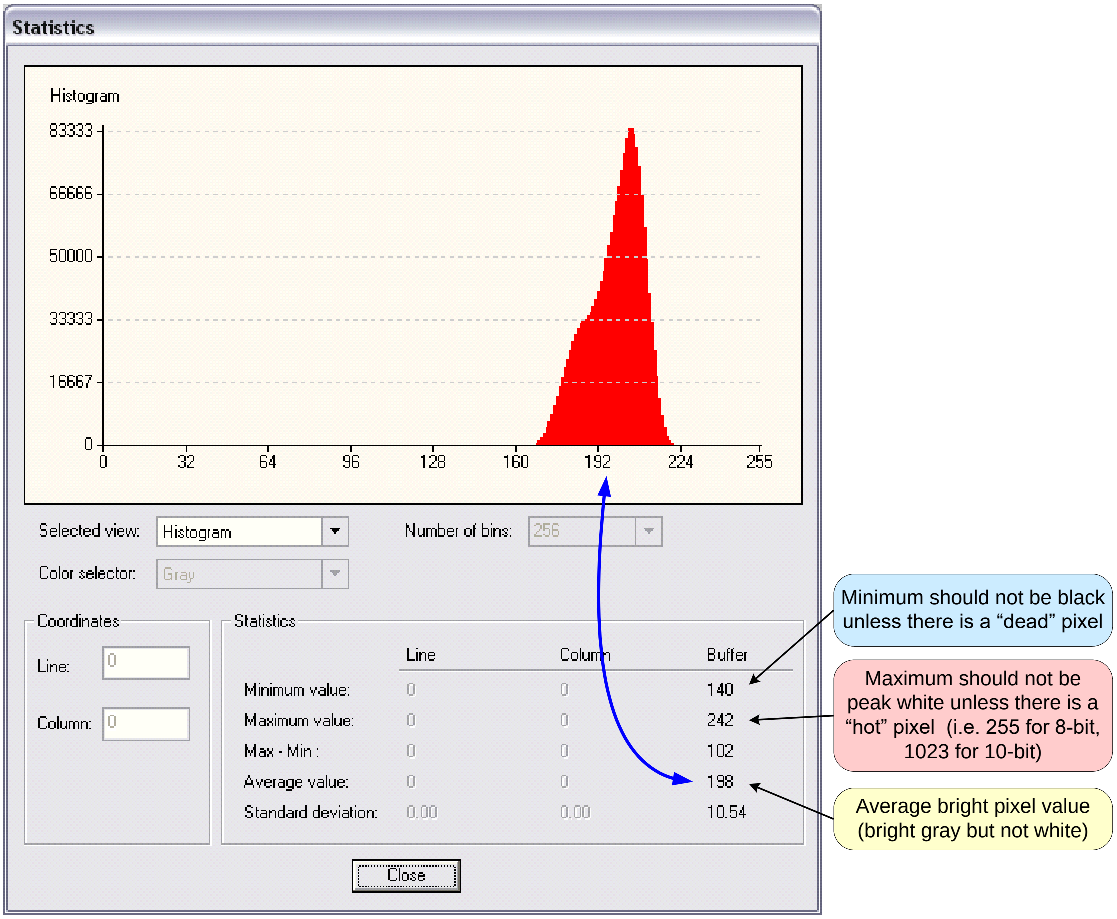
Indicates one or more “hot” pixels

Average dark pixel value

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).

Verify a Bright Acquisition

Aim the camera at a diffused light source or evenly lit white wall with no shadows falling on it. Using CamExpert, click on the grab button and then the histogram button. 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.



Selected view: Histogram

Number of bins: 256

Color selector: Gray

Coordinates

Line: 0

Column: 0

Statistics

	Line	Column	Buffer
Minimum value:	0	0	140
Maximum value:	0	0	242
Max - Min :	0	0	102
Average value:	0	0	198
Standard deviation:	0.00	0.00	10.54

Minimum should not be black unless there is a “dead” pixel

Maximum should not be peak white unless there is a “hot” pixel (i.e. 255 for 8-bit, 1023 for 10-bit)

Average bright pixel value (bright gray but not white)

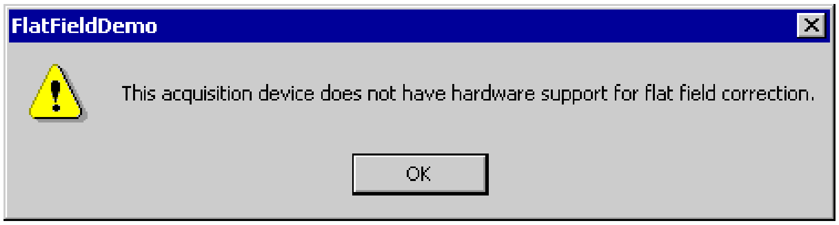
Important: In this example, the **average** pixel value for the frame is bright gray. Also note that sensors may show a much higher maximum or a much lower minimum pixel value due to one or more "hot or dead pixels". The sensor specification accounts for a small number of hot, stuck, or dead pixels (pixels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

After the bright gray acquisition setup is done, note the camera position and lens iris position to be able to repeat it during the calibration procedure.

Program Start and Server Selection

Run the demo via the Windows start menu shortcut **Start•All Programs•Teledyne DALSA•Sapera LT•Demos (GUI-based)• Sapera ++ Demos (Executables)**. This opens the C:\Program Files\Teledyne DALSA\Sapera\Demos\Binaries folder where the Flat Field Demo executable is located. The demo program first displays the acquisition configuration menu to select the board acquisition server and acquisition device.

If the selected board does not support onboard flat field processing, a message is displayed (see following figure) stating that processing will be on the host system.

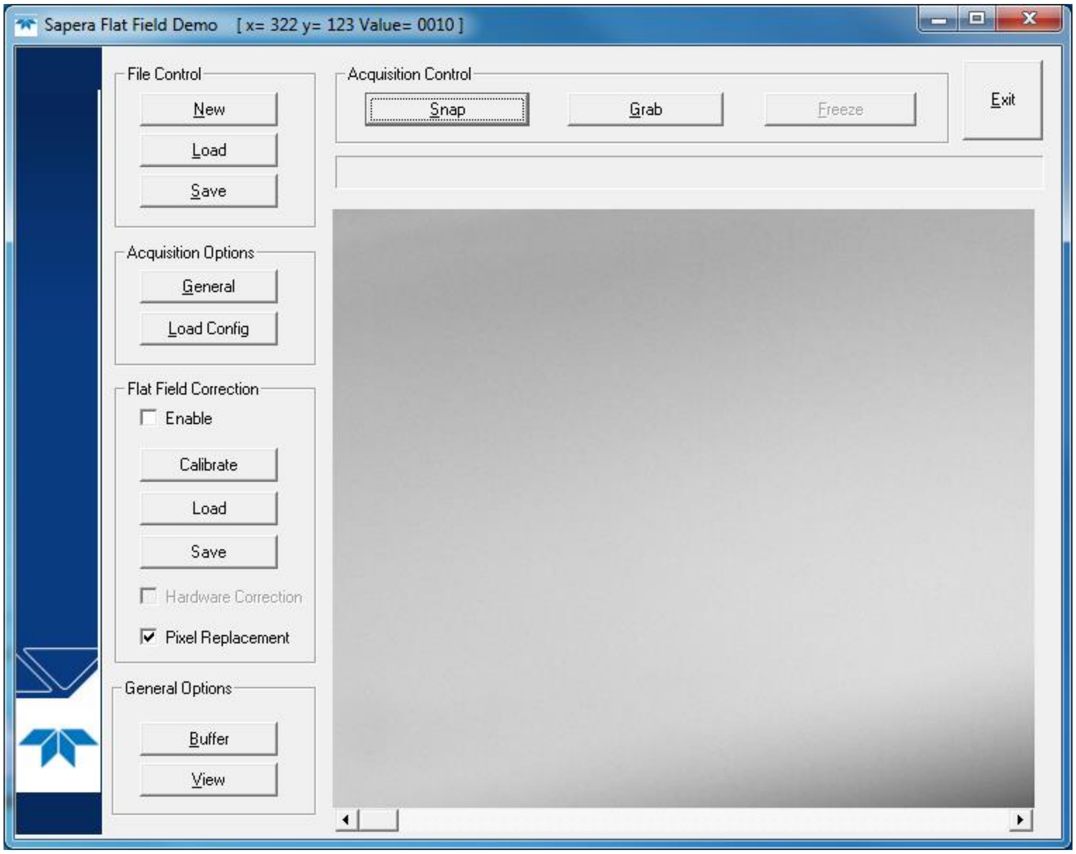


This acquisition device does not have hardware support for flat field correction.

This menu is also used to select the required camera configuration file for the connected camera. For a more detailed description of the acquisition configuration menu see "Acquiring with Grab Demo". Sapera camera files contain timing parameters and video conditioning parameters.

Flat Field Demo Main Window

The demo main window provides 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.



Sapera Flat Field Demo [x= 322 y= 123 Value= 0010]

☐ Enable

☐ Hardware Correction

☑ Pixel Replacement

Using Flat Field Correction

The demo has typical file and acquisition controls as previously described for the Grab Demo. What is different is the Flat Field Correction control section which has three buttons and a check box. Follow the procedure described below to setup and use 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 provides acquisition controls to confirm image capture.

Enable

Before activating flat field or flat line correction, follow the calibration procedure described in this section (see "Flat Field Calibration "). 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

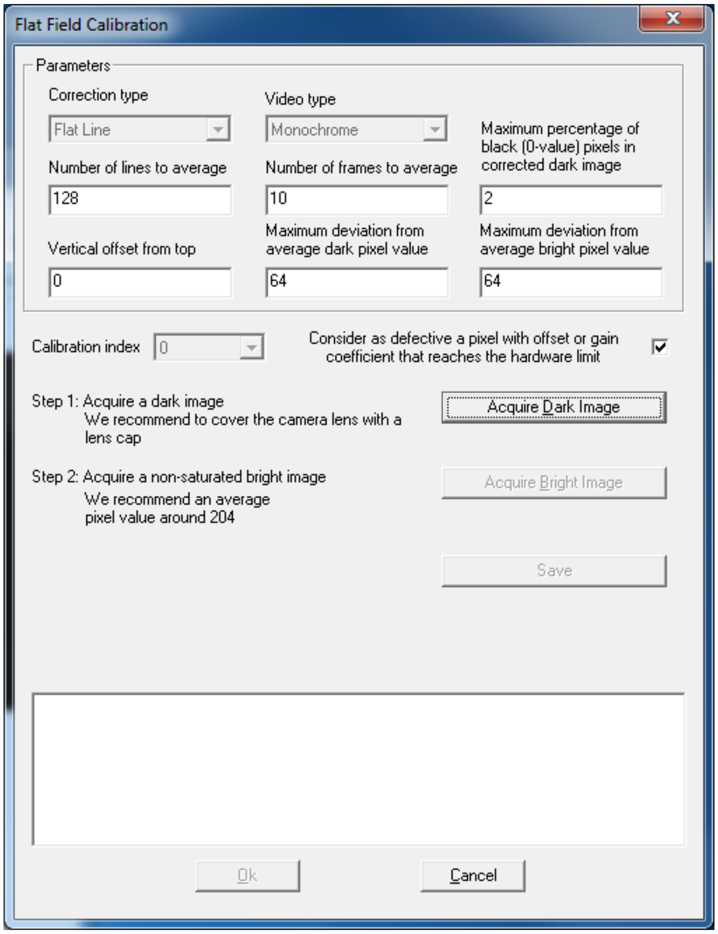
Click on the **Load** button to retrieve files with the flat field gain and offset data gathered with the calibration procedure.

Flat Field Calibration

Calibration is the process of taking two reference images, one of a dark field – one of a light gray field (not saturated), to generate correction data for images captured by the CCD. 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.

Click the **Calibrate** button. The calibration window opens as shown.



Flat Field Calibration

Parameters

Correction type
Flat Line

Video type: Monochrome

Maximum percentage of black (0-value) pixels in corrected dark image: 2

Number of lines to average: 128

Number of frames to average: 10

Vertical offset from top: 0

Maximum deviation from average dark pixel value: 64

Maximum deviation from average bright pixel value: 64

Calibration index: 0

Consider as defective a pixel with offset or gain coefficient that reaches the hardware limit ☑

Step 1: Acquire a dark image
We recommend to cover the camera lens with a lens cap

Step 2: Acquire a non-saturated bright image
We recommend an average pixel value around 204

Setup Before Calibration

- 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.

Calibration Procedure

- 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.
- Click on **Acquire Black 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.
- 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.

Appendix: File Locations

The table below lists the different file groups and locations:

Directory	**Contents**
Network Interface	Sapera Network Imaging Package (optional)
Sapera	Sapera LT Readme and version history documents
	Third-party software licenses (for example, GenICam)
Sapera\Bin	Utility programs
Sapera\CamExpert	CamExpert frame-grabber and camera configuration utility
Sapera\Expert	Z-Expert 3D sensor 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	On-line documentation (Compiled HTML and PDF formats)
Sapera\Help\VisualStudio2010	Integrated C++ and .NET help for Visual Studio 2010, 2012, 2013, 2015, and 2017
Sapera\Images	Images files used by demos and examples
Sapera\Include	Header files for C libraries
Sapera\Lib	Import libraries for Microsoft and Borland (Embarcadero) compilers
Teledyne DALSA\GenICam	GenICam CL Protocol run-time support for Teledyne DALSA CameraLink cameras
Windows\system32 directory	Dynamic Link Libraries (DLLs)
Windows\system32\drivers directory	Device drivers

Contact Information

The following sections provide sales and technical support 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 Diagnostic PciDiag.txt file (for frame grabbers)
- The Device Manager BoardInfo.txt file (for frame grabbers)

⚠	Note, the Sapera Log Viewer and PCI Diagnostic tools are available from the Windows start menu shortcut **Start•All Programs•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 shortcut **Start•All Programs•Teledyne DALSA•<Device Name>•Device Manager**.