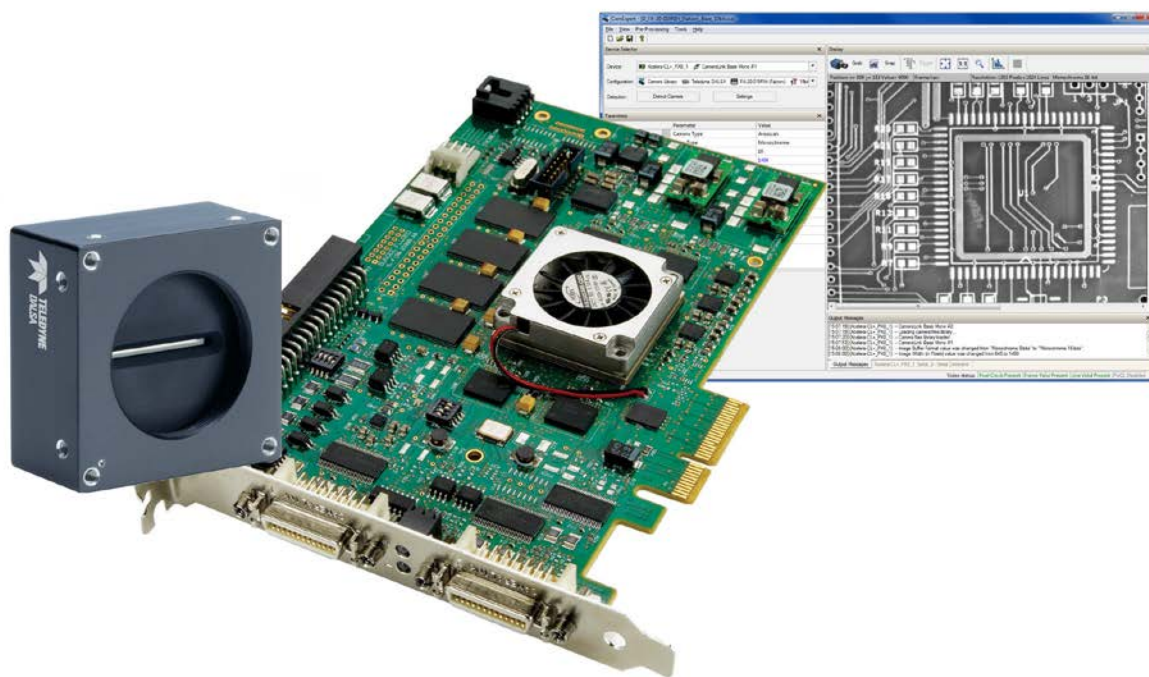


# Sapera LT™ 8.30

## Getting Started Manual For Frame Grabbers

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



**P/N: OC-SAPM-INTR0**  
[www.teledynedalsa.com](http://www.teledynedalsa.com)

 **TELEDYNE DALSA**  
Everywhere you look™

## **NOTICE**

© 1998-2017 Teledyne DALSA, Inc. All rights reserved.

This document may not be reproduced nor transmitted in any form or by any means, either electronic or mechanical, without the express written permission of TELEDYNE DALSA. Every effort is made to ensure the information in this manual is accurate and reliable. Use of the products described herein is understood to be at the user's risk. TELEDYNE DALSA assumes no liability whatsoever for the use of the products detailed in this document and reserves the right to make changes in specifications at any time and without notice.

Microsoft® is a registered trademark; Windows®, Windows® 7, Windows® 8 and Windows® 10 are trademarks of Microsoft Corporation.

All other trademarks or intellectual property mentioned herein belongs to their respective owners.

Printed on April 24, 2017

Document Number: OC-SAPM-INTRO  
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 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

<b>INTRODUCTION TO SAPERA LT</b>	<b>5</b>
SAPERA LT LICENSING.....	5
THE SAPERA LT APIS.....	5
SAPERA TOOLS AND UTILITIES .....	5
SAPERA LT MANUAL DESCRIPTIONS .....	6
SUPPORTED OPERATING SYSTEMS .....	7
SUPPORTED SAPERA LT DEVELOPMENT ENVIRONMENTS .....	7
INSTALLATION TYPES .....	8
<i>Upgrading Previous Versions of Sapera LT</i> .....	9
START MENU SHORTCUTS.....	9
SAPERA LT ARCHITECTURE OVERVIEW .....	11
<i>Application Architecture</i> .....	11
<i>Library Architecture</i> .....	12
<i>Configuration Files</i> .....	13
<b>QUICK START GUIDE</b>	<b>14</b>
USING SAPERA LT WITH A TELEDYNE DALSA GENCP CAMERALINK CAMERA AND FRAME	
GRABBER .....	16
<i>Sapera LT Classes For GenCP CL Cameras and Frame Grabber Applications</i> ..	16
ANY ASCII TEXT-BASED (OR OTHER) CAMERALINK CAMERA AND TELEDYNE DALSA FRAME	
GRABBER.....	19
USING THE CAMEXPERT TOOL .....	21
<b>SAPERA LT UTILITIES</b>	<b>23</b>
SAPERA MONITOR .....	23
<i>The Sapera Monitor Window</i> .....	23
<i>Sapera Monitor Menu Commands</i> .....	25
<i>Using Sapera Monitor</i> .....	27
SAPERA LOG VIEWER .....	28
SAPERA CONFIGURATION UTILITY.....	29
<i>Configuring Frame Grabber Board Serial Ports</i> .....	30
<i>Configuring Contiguous Memory</i> .....	31
PCI DIAGNOSTIC.....	31
SAPERA COLOR CALIBRATION TOOL .....	32
<b>DEMOS AND EXAMPLES</b>	<b>33</b>
DEMO SOURCE CODE .....	33
EXAMPLE SOURCE CODE.....	33
DEMOS AND EXAMPLES FOR FRAME GRABBERS.....	34
DEMOS AND EXAMPLES FOR GENCP CAMERALINK CAMERAS.....	37
GENERIC SAPERA LT EXAMPLES .....	38
ACQUIRING WITH GRAB DEMO .....	39
USING THE FLAT FIELD DEMO.....	41
<i>Set up Dark and Bright Acquisitions with the Histogram Tool</i> .....	41
<i>Verify a Dark Acquisition</i> .....	42
<i>Verify a Bright Acquisition</i> .....	43
<i>Program Start and Server Selection</i> .....	44
<i>Flat Field Demo Main Window</i> .....	45
<i>Using Flat Field Correction</i> .....	45
<i>Flat Field Calibration</i> .....	46
<i>Setup Before Calibration</i> .....	47
<i>Calibration Procedure</i> .....	48

<b>APPENDIX: FILE LOCATIONS</b>	<b>49</b>
<b>CONTACT INFORMATION</b>	<b>50</b>
SALES INFORMATION.....	50
TECHNICAL SUPPORT.....	50

# Introduction to Sopera LT

Sopera™ LT is a software API for controlling image acquisition devices such as frame grabbers and camera. Sopera LT libraries support Teledyne DALSA cameras and frame grabbers as well as hundreds of 3rd party camera models across all common interfaces 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, Sopera Essential, a full-featured image processing library, is available as a separate software package. For more information see [www.teledynedalsa.com/imaging/products/software/](http://www.teledynedalsa.com/imaging/products/software/).

The manual provides an introduction to the Sopera LT API and is designed to help programmer's with installation, and quickly perform hardware setup and validation.

---

## Sopera LT Licensing

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

---

## The Sopera LT APIs

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

- Sopera LT++ classes (based on C++ language)
- Sopera LT .NET classes (based on .NET languages)
- Sopera 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 Sopera LT compatible device. Sopera 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 Sopera LT.

---

## Sopera Tools and Utilities

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

- [CamExpert](#): acquisition and device configuration, including image display
- [Sopera Monitor](#): real-time event viewing of applications.
- [Sopera Log Viewer](#): error and other message log for applications and hardware
- [Sopera 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	<b>Sapera LT Getting Started With GigE Vision Cameras Manual</b>	Provides a general overview using Sapera LT APIs with GigE Vision cameras, possible hardware configurations, and a quick overview of Sapera programming.
General	<b>Sapera LT User's Manual</b>	Introduces Sapera LT ++ API and Sapera .NET API programming procedures, including sample code for typical operations in C++, C# and VB .NET.
GigE Vision	<b>GigE Vision Module User's Manual</b>	Describes the GigE Vision Host Control features and overview of available GigE demos.
GigE Vision	<b>Network Imaging User's Manual</b>	Network setup and optimization guide for GigE Vision cameras as well as troubleshooting guidelines.
.NET	<b>Sapera LT .NET Programmer's Manual</b>	Provides a complete reference of the Sapera .NET Framework for Visual Studio. Sapera .NET reflects the underlying low-level Sapera LT architecture.
C++	<b>Sapera LT ++ Programmer's Manual</b>	Provides a complete reference of all the Sapera LT ++ classes. Sapera LT ++ is based on the C++ language.
C++	<b>Sapera LT GUI Classes Reference Manual</b>	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++	<b>Sapera LT Legacy Classes Reference Manual</b>	Describes the obsolete C++ classes that continue to be supported but have been replaced or retired.
C	<b>Sapera LT Acquisition Parameters Reference Manual</b>	Describes the Sapera LT low level acquisition parameters and capabilities (based on the C language).
C	<b>Sapera LT Basic Modules Reference Manual</b>	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 7® (32-bit and 64-bit)
- Windows 8® (32-bit and 64-bit)
- Windows 8.1® (32-bit and 64-bit)
- Windows 10® (32-bit and 64-bit)

(**Note:** Windows NT 4.0, Windows 2000 and Windows Vista are no longer supported.)

---

## 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 2005 (with Service Pack 1)
  - Microsoft Visual Studio 2008 (with Service Pack 1)
  - Microsoft Visual Studio 2010
  - Microsoft Visual Studio 2012
  - Microsoft Visual Studio 2013
  - Microsoft Visual Studio 2015

For 32-bit development using Embarcadero (previously Borland):

- Embarcadero C++Builder XE1 and up (classic C++ compiler)
- Embarcadero C++Builder XE10.1 and up (Clang based C++ compiler)

For 64-bit development using Embarcadero:

- Embarcadero C++Builder XE5 and up



If you are using Visual Studio 2005 or 2008 to develop Sapera LT applications, then you should have the latest Service Pack plus all security patches installed. You can check if these are installed through the Help->About menu item in Visual Studio.

---

## 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, or all components.
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 8.30; the same directory structure and component configuration are used.

### Sapera LT 7.20 and Lower

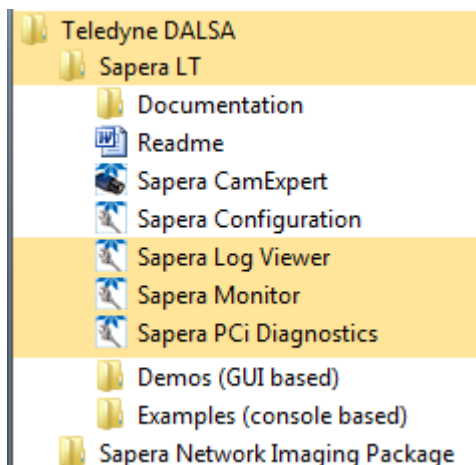
Before installing Sapera LT 8.30, uninstall Sapera LT from the Windows Control Panel > Programs and Features.

---

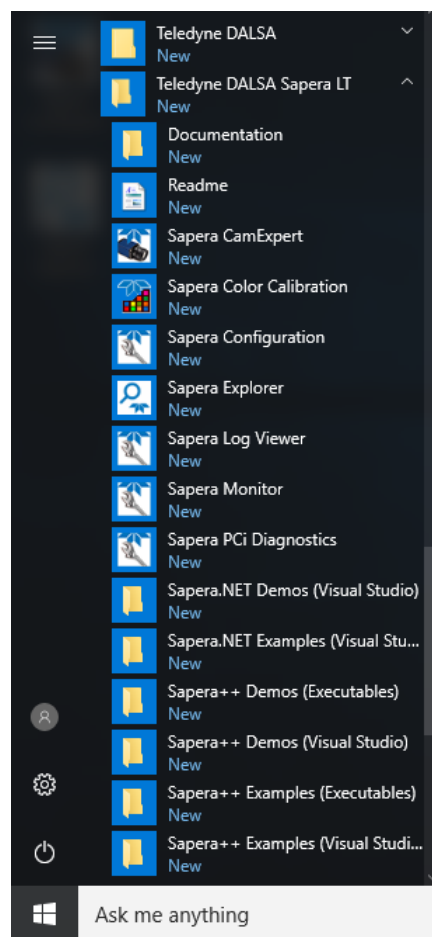
## Start Menu Shortcuts

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.

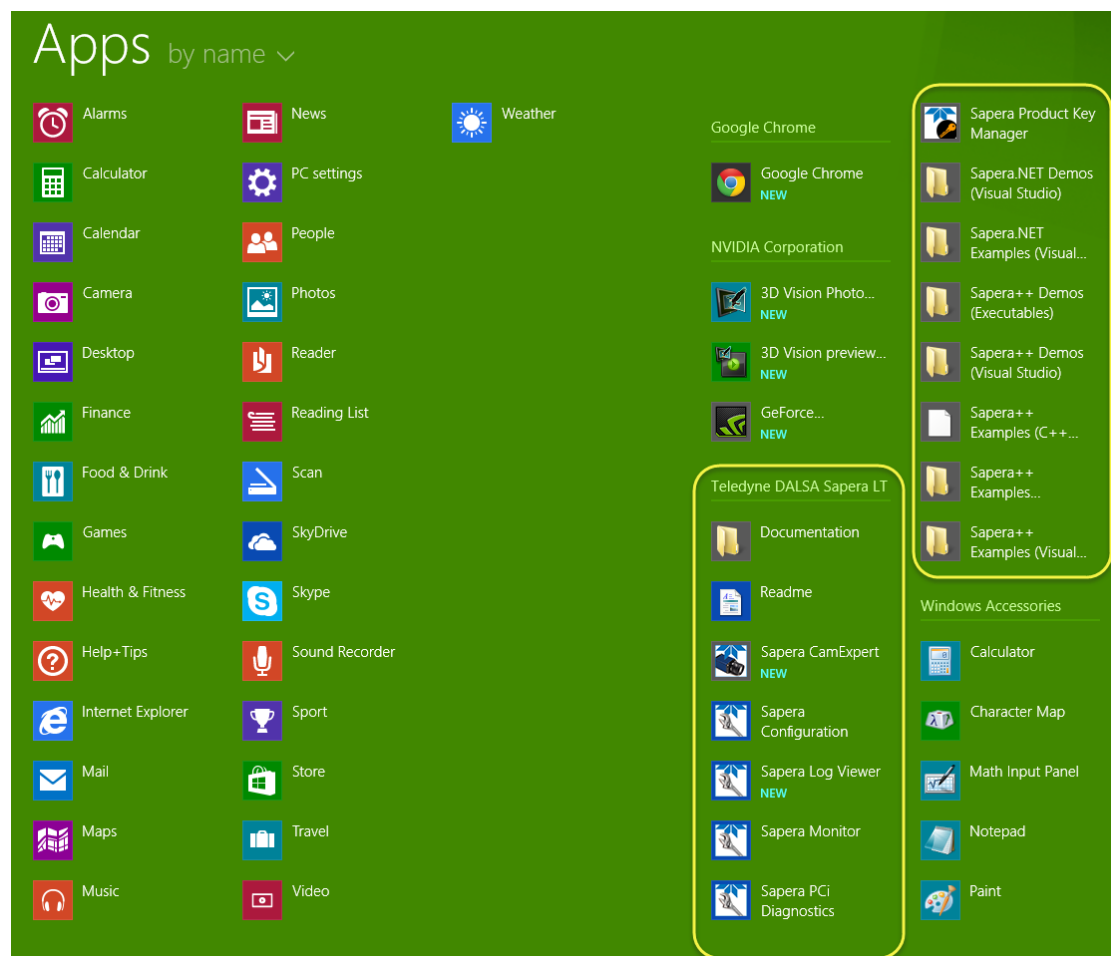


Windows 7



Windows 10

For Windows 8, shortcuts are presented as tiles on the desktop through the Apps menu.



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

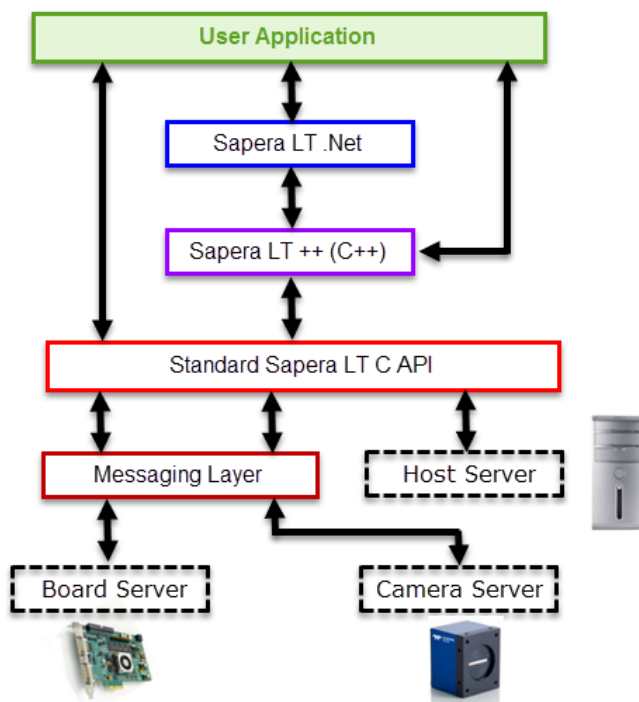
---

# Sapera LT Architecture Overview

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

## Application Architecture

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



### What is a server?

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

A server allows Sapera applications to interact with the server's resources. The server name consists of the product model name and an index. For example, the Xtium CLHS PX8 frame grabber has a server name "Xtium-CLHS\_1". If more than one device of the same type is available, the index differentiates them.

A server may also have different resources available, identified by a resource index. For example, the Xtium\_CLHS\_1 server has two types of acquisition resource available to select, depending on the configuration required: Camera Link HS Mono or Camera Link RGB.

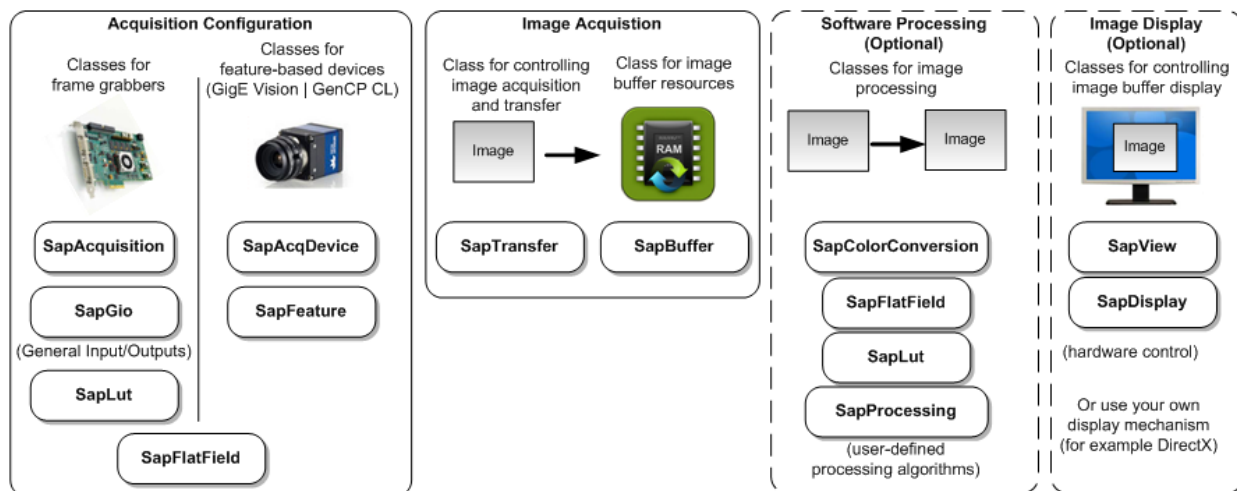
## Library Architecture

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

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

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



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

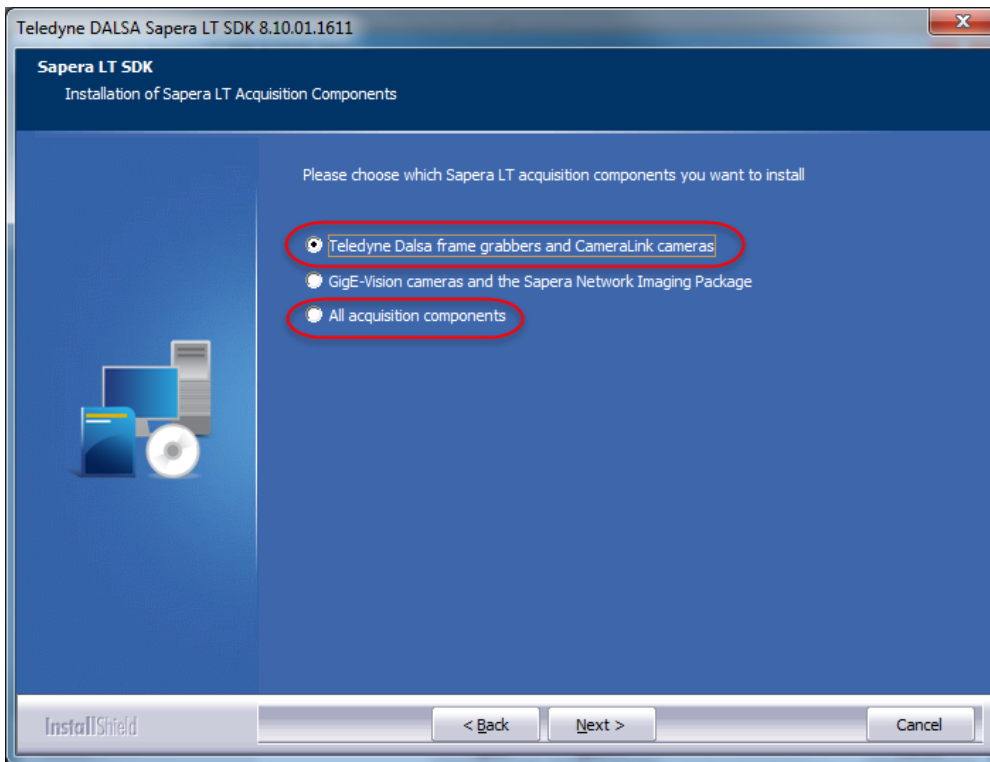
Sapera LT applications need to load the appropriate acquisition configuration file before acquiring images from a camera. The camera configuration file has the extension **.ccf**. For applications that use both a frame grabber and camera, both of the device's parameter settings are saved in this file.

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 a large number of 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 either the option to install Sapera LT for Teledyne Dalsa frame grabbers and Camera Link cameras or All acquisition components.

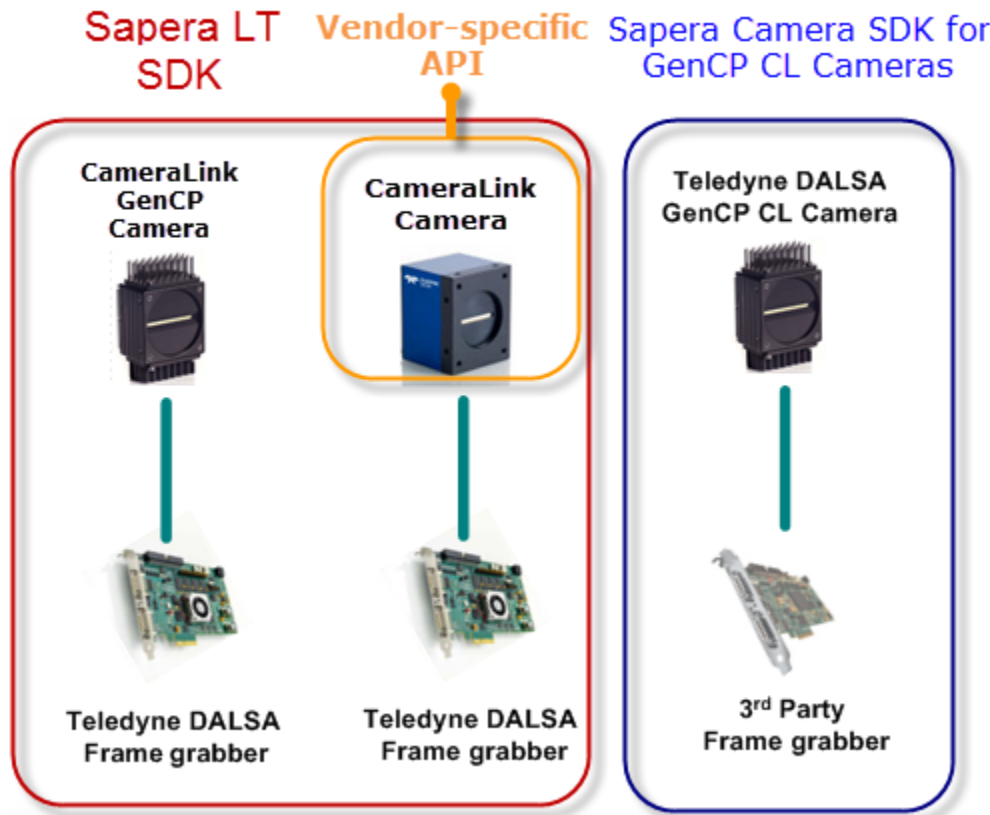


Note that if you try to install Sapera LT from a network location you will not be able to install the Sapera Network Imaging Package; copy the installation executable locally.

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

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 Spera Camera SDK for GenCP CL Cameras in this scenario instead of the full Spera LT installation: this provides access to the camera features, but does not provide acquisition since this must be handled by the frame grabber API)



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



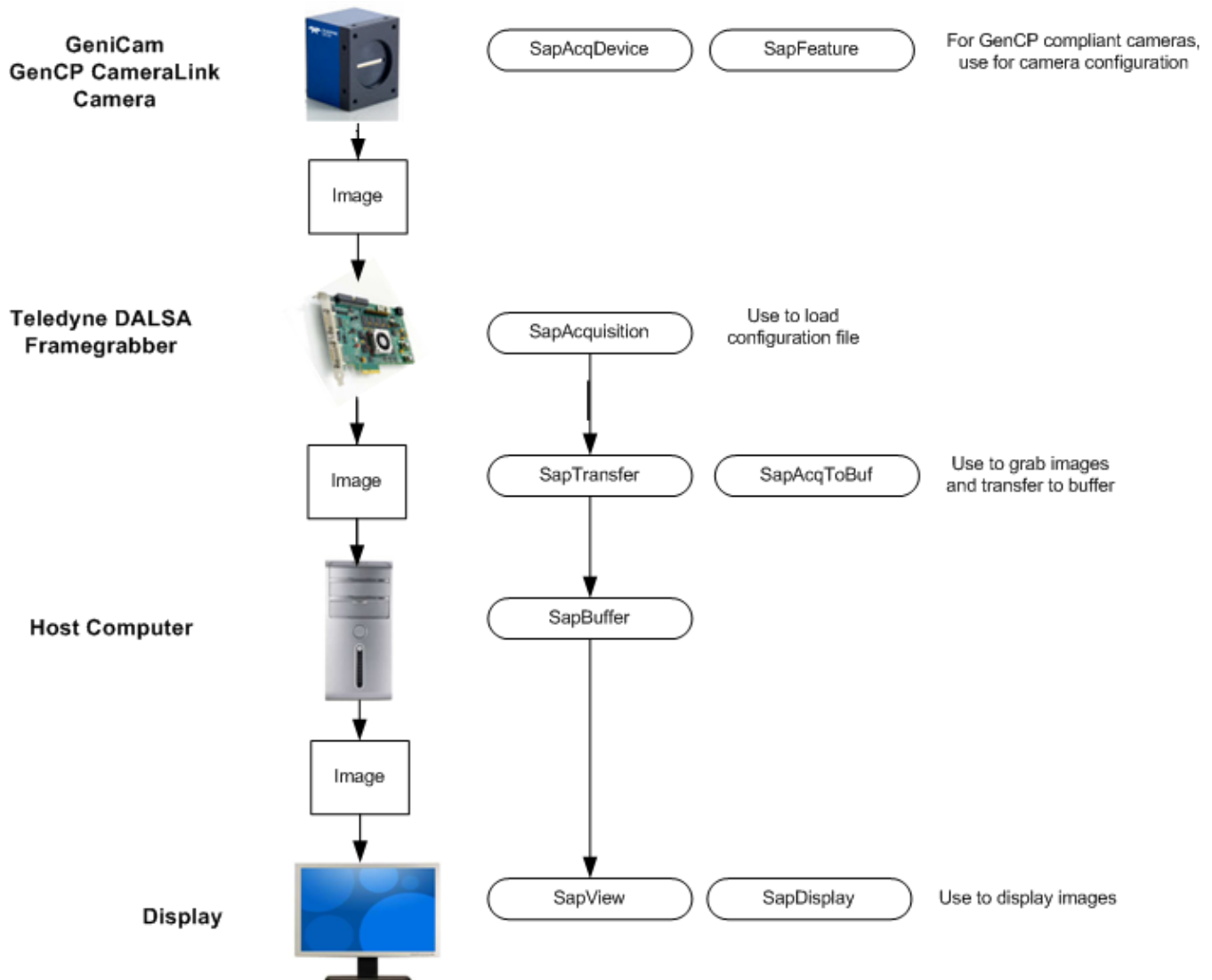
When using CameraLink cameras it is important to verify if the camera complies with the GenCP protocol or requires the vendor-specific API. For more information on the Sapera LT implementation of GenICam, refer to either of the Sapera LT ++ or .NET Programming Manuals.

---

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

## Sapera LT Classes For GenCP CL Cameras and Frame Grabber Applications

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

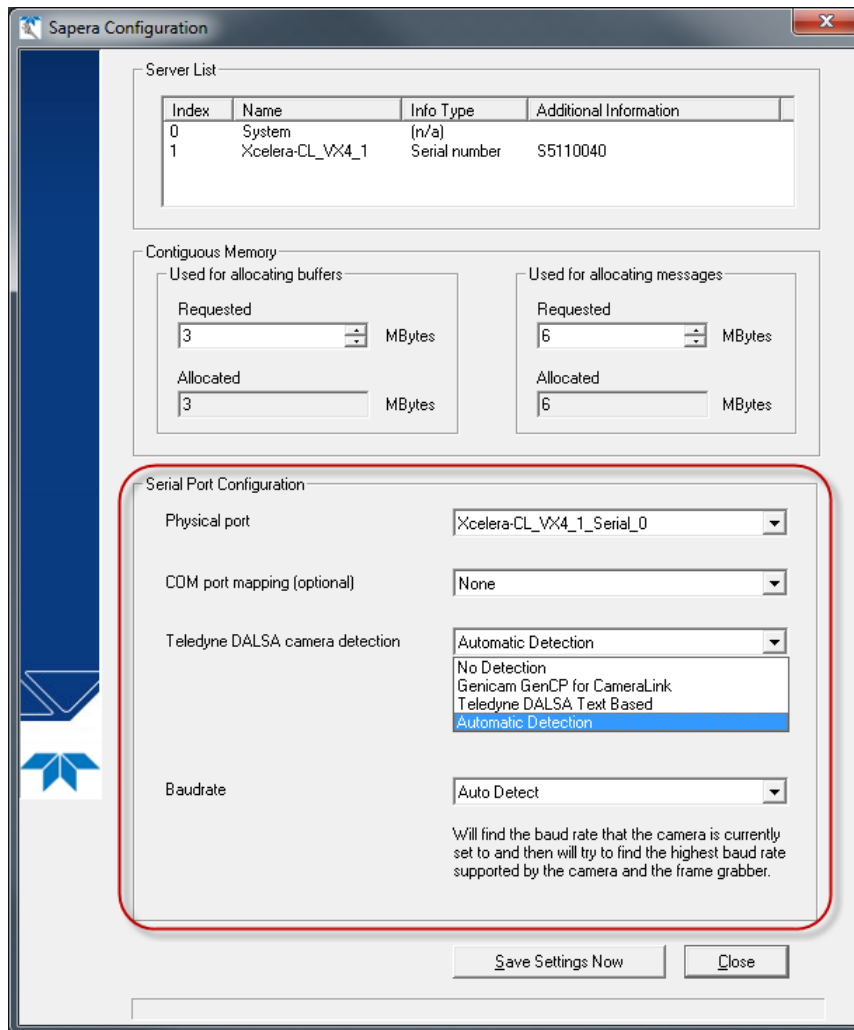


Teledyne DALSA CameraLink cameras either support the GenCP standard, or can be programmed using ASCII commands on CameraLink serial ports; refer to the camera documentation to determine which method the camera supports.

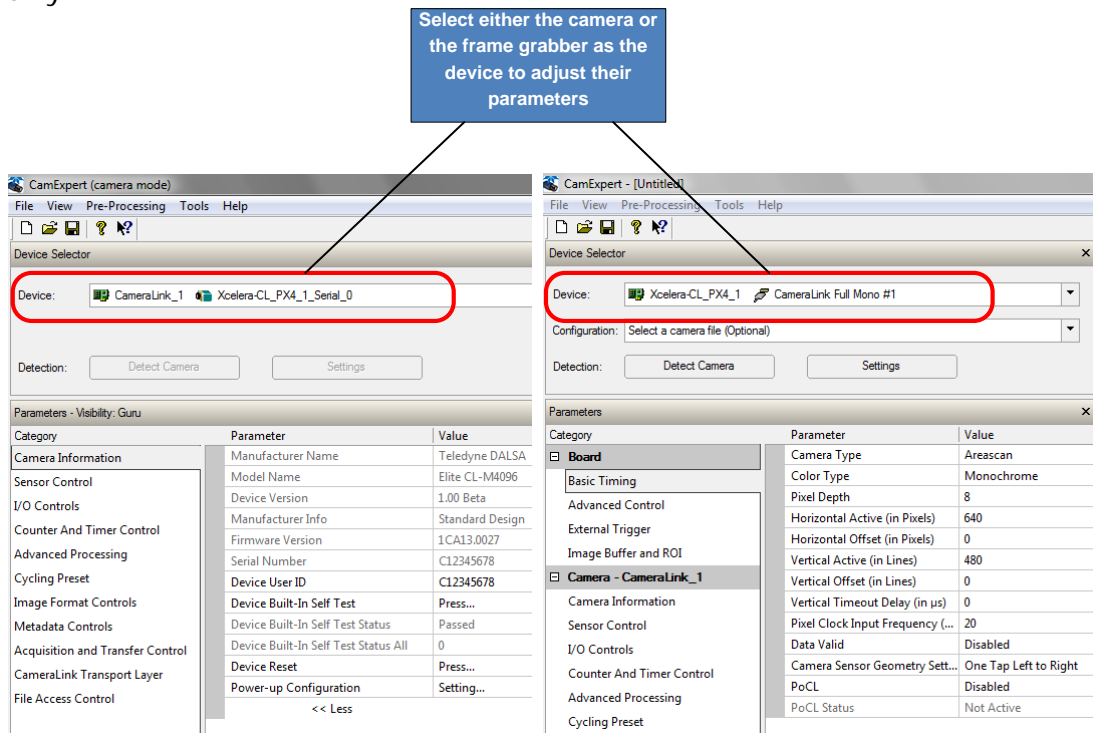


## To configure Teledyne DALSA GenCP CameraLink Cameras:

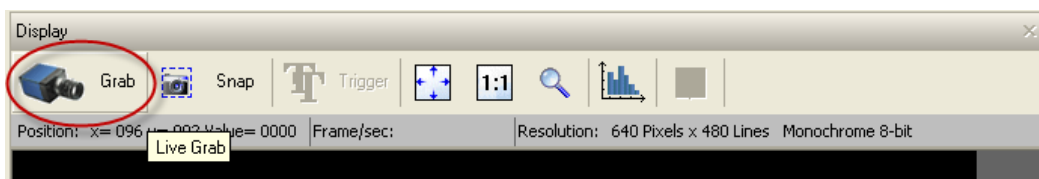
1. Run the [Sapera Configuration](#) utility, select the frame grabber serial port connected to the camera, and set the **Teledyne DALSA camera detection** parameter to one of the three following options:
  - **GenICam GenCP for CameraLink:** CamExpert will try to detect Teledyne DALSA cameras that support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



2. Start 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.



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



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

---

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

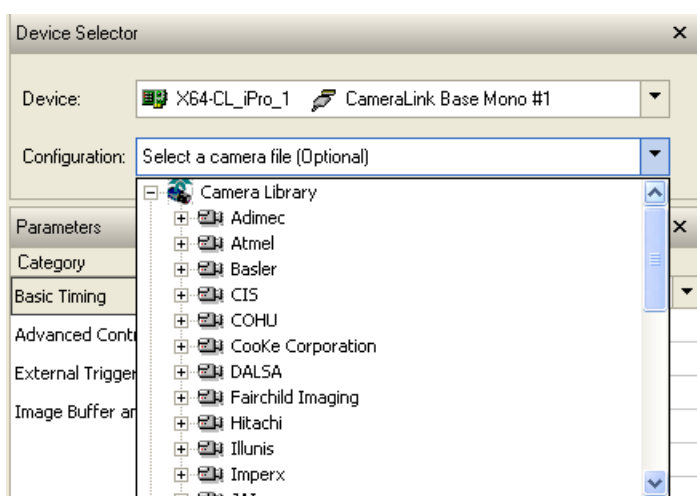
The Teledyne DALSA frame grabber requires a camera configuration file (\*.ccf) to program the frame grabber to match the characteristics of the ASCII text-based (non-GenCP) CameraLink Teledyne DALSA or 3<sup>rd</sup> party camera (for example, cameras that use three letter serial port commands). Teledyne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

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

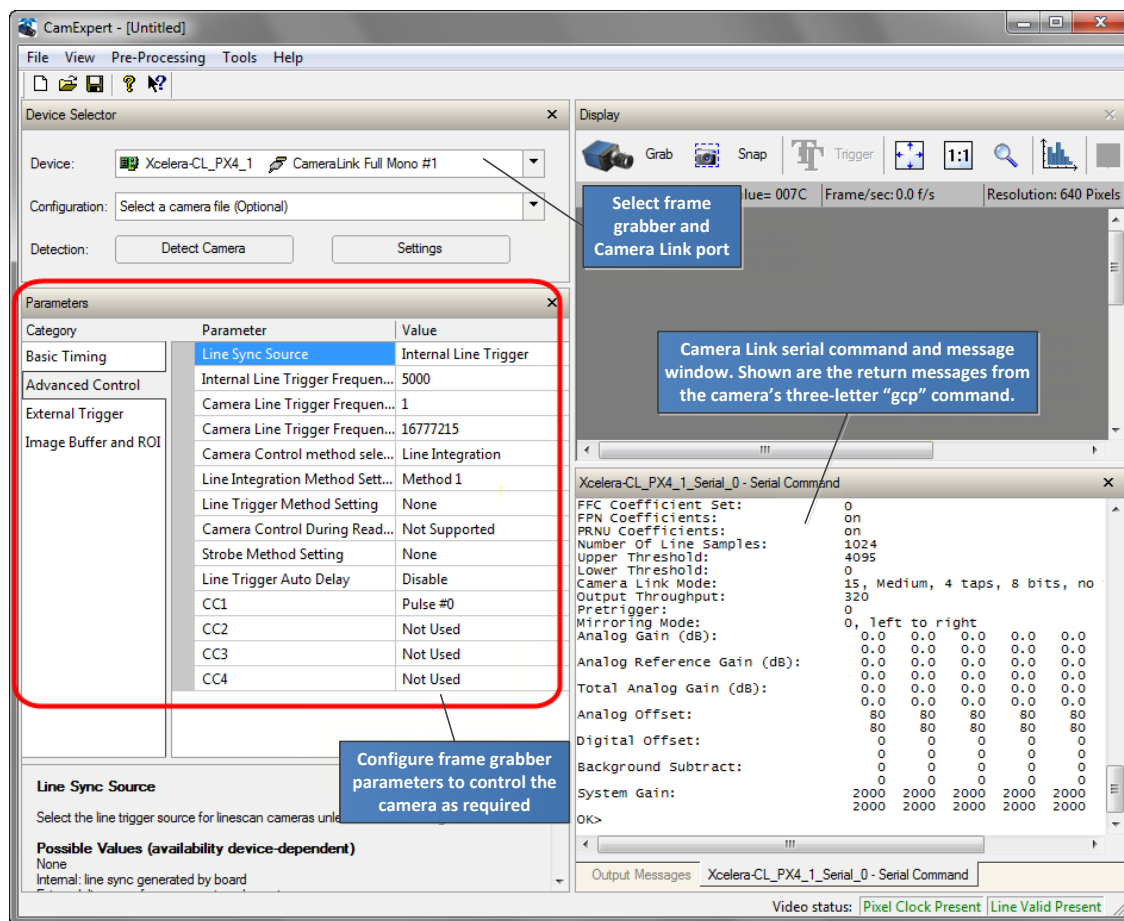
For certain 3<sup>rd</sup> party applications, it may be necessary to map frame grabber serial ports to Windows COM ports. The [Sapera Configuration](#) utility can be used to do so; refer to the Serial Port Mapping section.

### To configure cameras which support ASCII commands only:

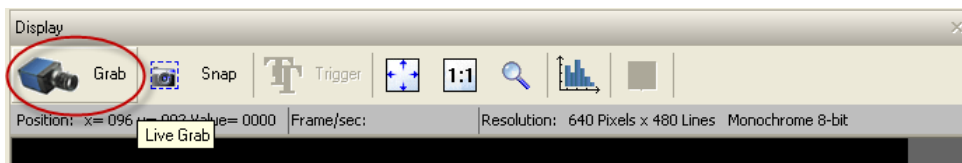
1. Start the CamExpert application. In the Device tab, select the required frame grabber, so that you can adjust its parameters.
2. If available, select the camera configuration file for the camera.



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



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

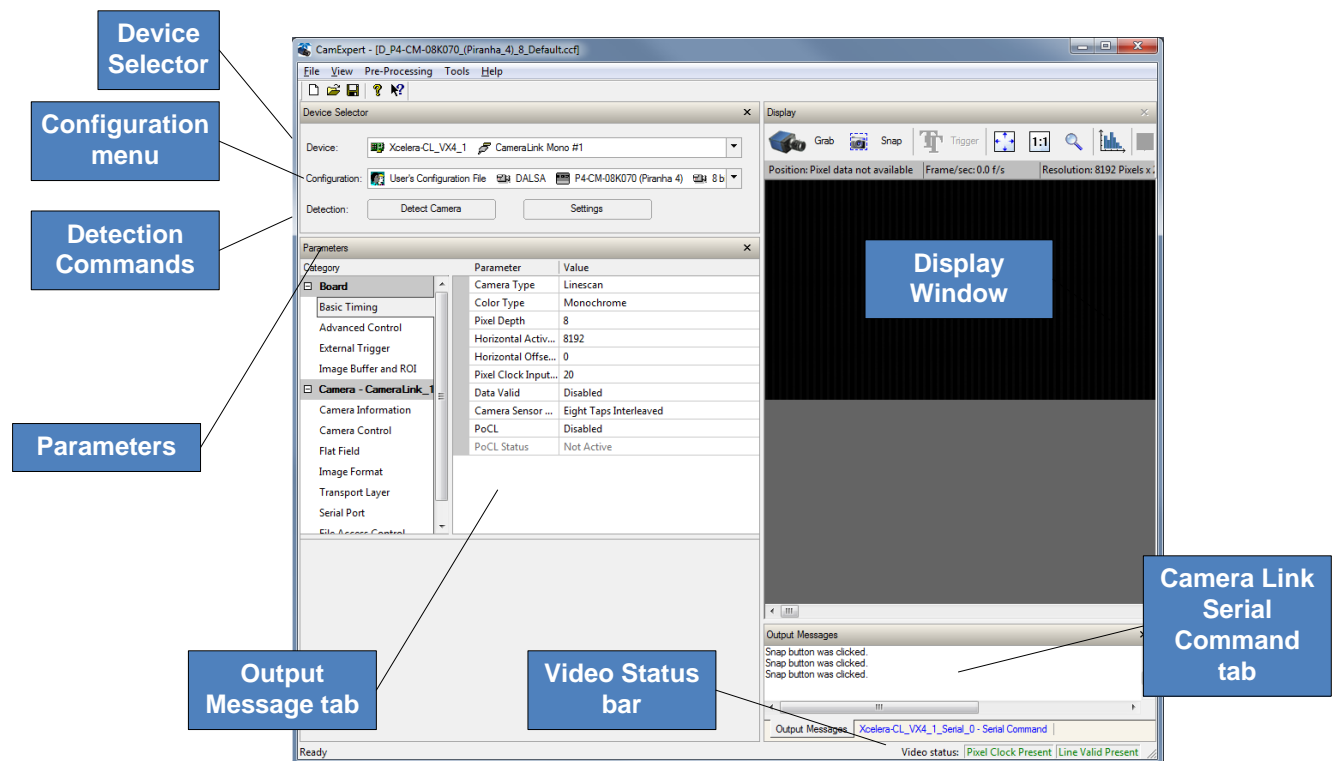


- Save the frame grabber configuration to a new \*.ccf file.









# 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 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	<b>Acquisition control button:</b> Click once to start live grab, click again to stop.
 Snap	<b>Single frame grab:</b> Click to acquire one frame from device.
 Trigger	<b>Software trigger button:</b> With the I/O control parameters set to Trigger Enabled / Software Trigger type, click to send a single software trigger command.
  	<b>CamExpert display controls:</b> (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.
	<b>Histogram / Profile tool:</b> 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:

<a href="#">Sapera Monitor</a>	Monitors user selected events generated by a Sapera LT application.
<a href="#">Sapera Log Viewer</a>	Displays error and other messages generated by Sapera LT applications and Teledyne DALSA hardware.
<a href="#">Sapera Configuration</a>	Configuration utility for Teledyne DALSA frame grabbers.
<a href="#">Sapera PCI Diagnostics</a>	Low-level diagnostic utility for Teledyne DALSA frame grabbers.
<a href="#">Sapera Color Calibration Tool</a>	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 applications 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.

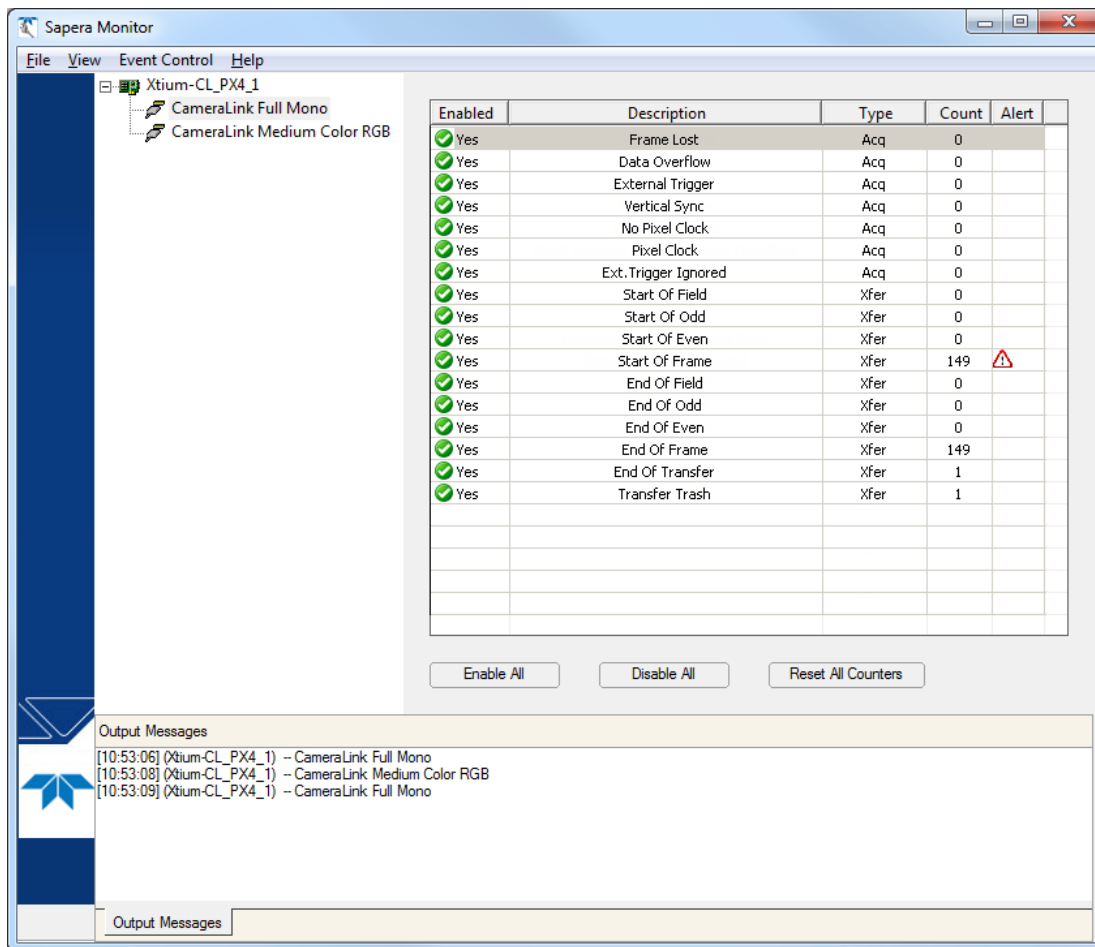


Note: older driver versions of Teledyne DALSA devices may not support Sapera Monitor. Check the Teledyne DALSA website for updated drivers for your device that support Sapera Monitor. In addition, when using Teledyne DALSA Genie devices, you must start a Sapera application, such as CamExpert, that uses the device you want to monitor, before launching Sapera Monitor.

## The Sapera Monitor Window

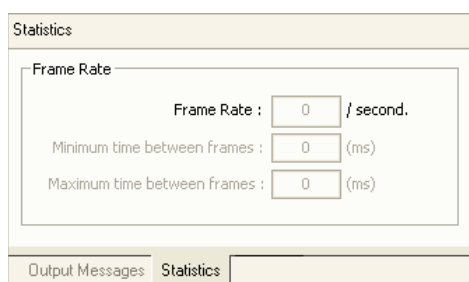
The Sapera Monitor Window is divided into three panes:

- 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.



## Statistic Tab

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



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



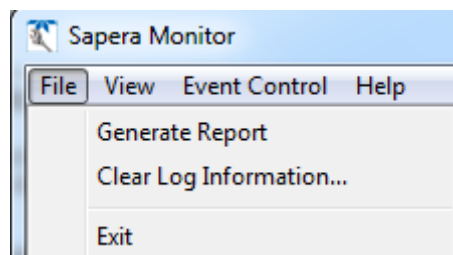
## Sapera Monitor Menu Commands

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

### File Menu Commands

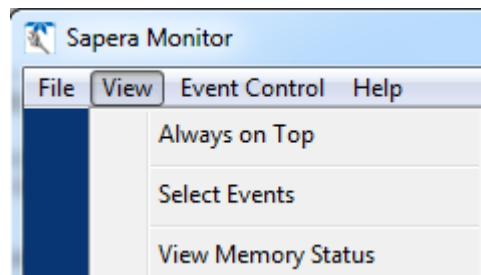
The **File•Generate Report** command generates a text file report that includes all event settings and messages included in the current Output Messages pane.

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

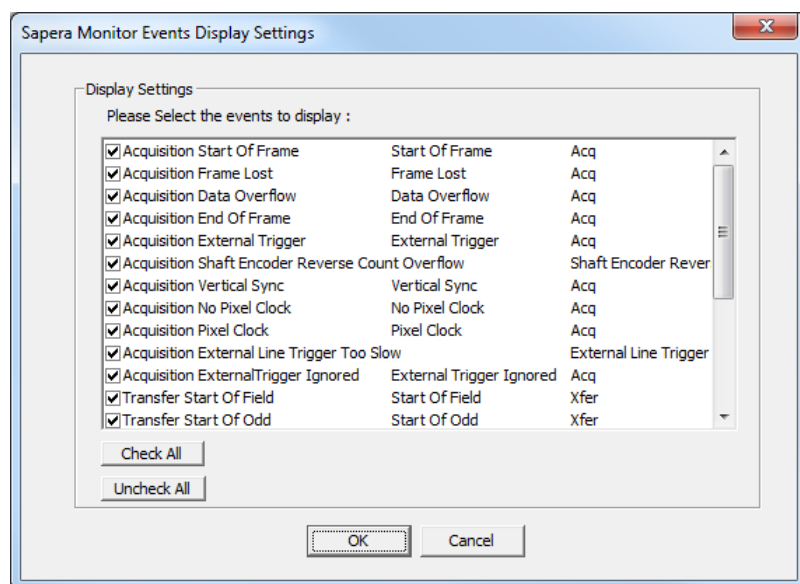


### View Menu Commands

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



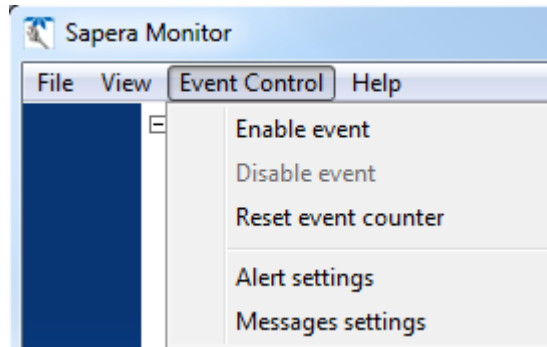
The **View•Select Events** command opens the Sapera Monitor Events Display Settings dialog which allows you to specify the events to display in the Event table.



## Event Control Menu Commands

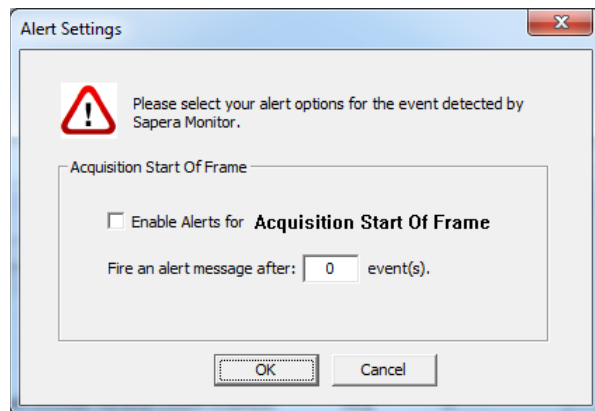
The **Event Control•Enable** and **Disable** commands enable or disable the currently selected event in Event table.

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.



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.

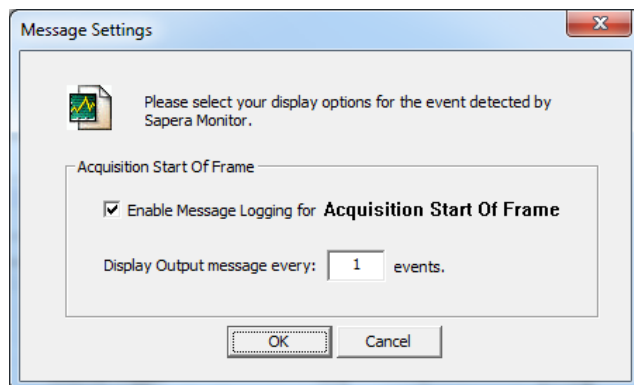
```

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.



## 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.



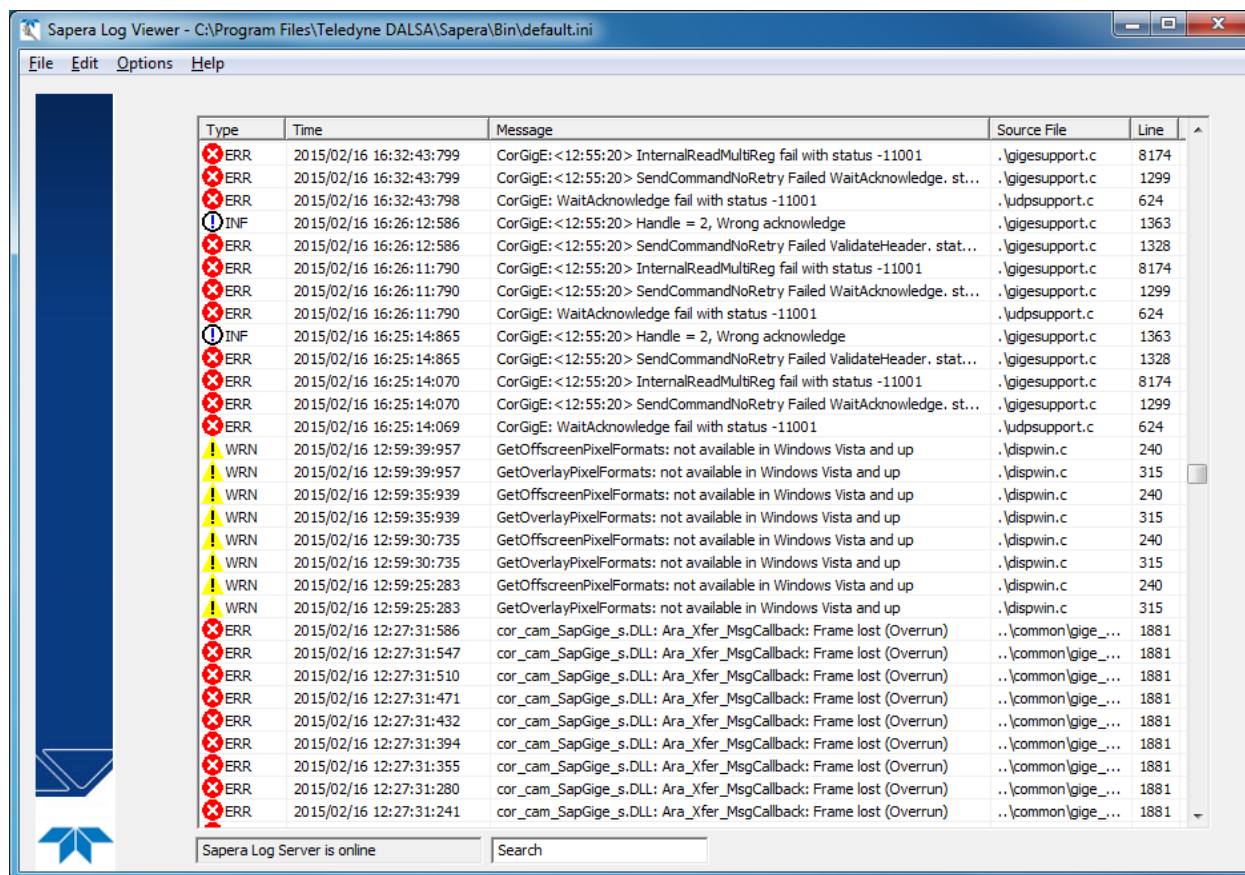
These commands are also available through the Event Control Menu. Alternatively, you can use the Enable All and Disable All buttons to quickly clear or select events.

## 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.

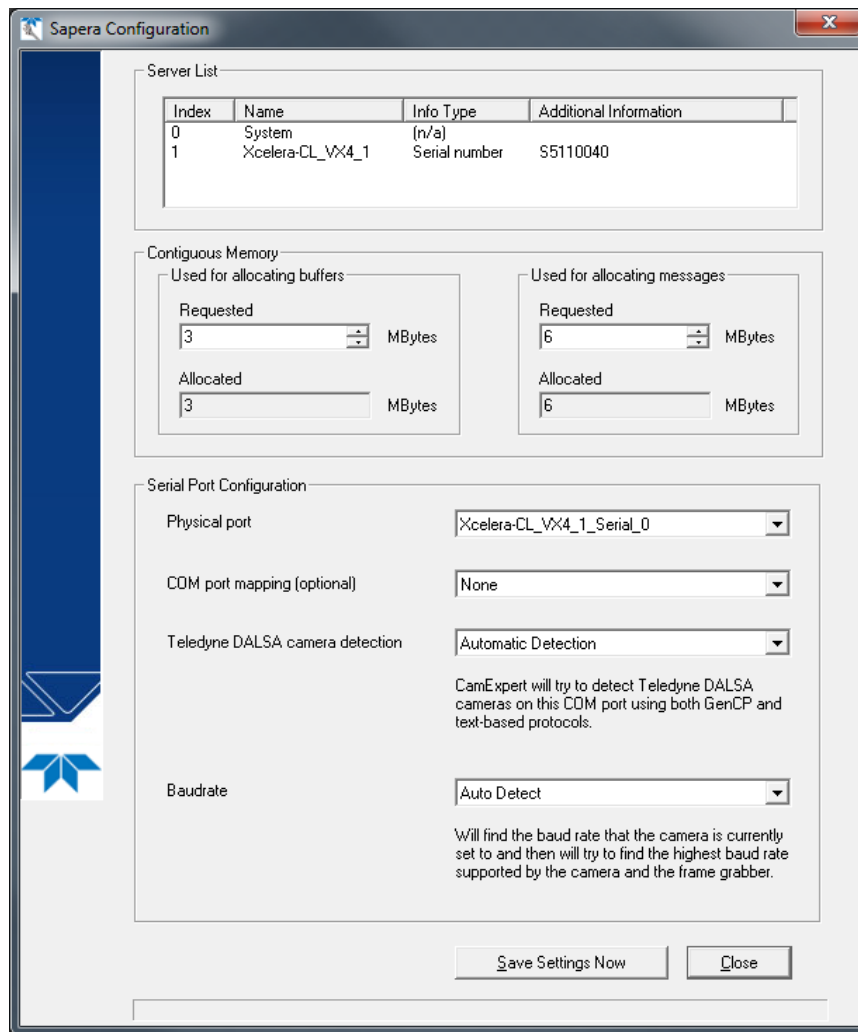
During development it is recommended to start the Sapera Log Viewer before your application and then let it run so it can be referred to any time a detailed error description is required. However, errors are also stored by a low-level service (running in the background), even if the utility is not running.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



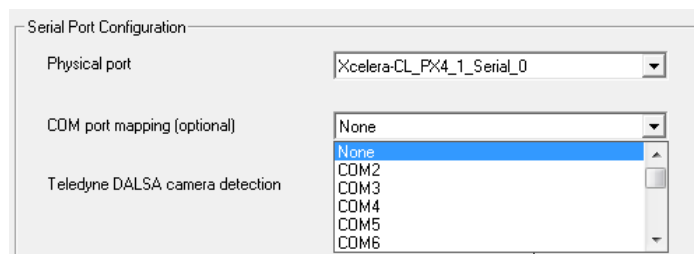
- The **System** entry represents the system server. It corresponds to the host machine (your computer), and is the only server that should always be present. The other servers correspond to the device's present within the system.
- The **Serial Port Configuration** section allows you to select the frame 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.

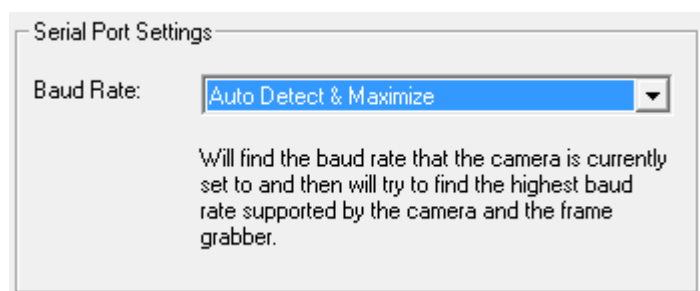
### Serial Port Mapping

For certain 3<sup>rd</sup> 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.

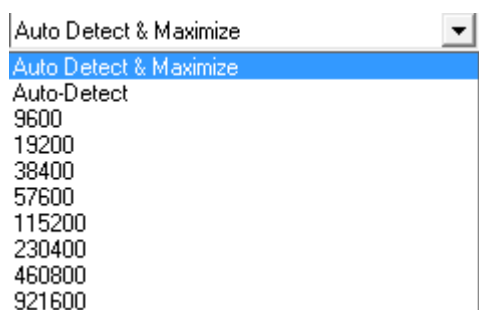


### Baud Rate

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



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



## Configuring Contiguous Memory

The **Contiguous Memory** section lets you specify the total amount of contiguous memory to be reserved for allocating **buffers** and **messages**.



**Note:** All Sapera LT demos and examples **do not use contiguous memory**. Therefore, you should not modify these settings unless your application requires contiguous memory.

The **Requested** value displays what was requested. The **Allocated** value displays the amount of contiguous memory that was allocated successfully. The current value for **buffers** 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. The current value for **messages** 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.

## 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 2.2

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

Device disabled | Rescan devices | Refresh

Vendor ID: 0x8086 Rev. ID: 0x22 Latency: 0x00  
Device ID: 0x3405 IntLine: 0x00 Min Grant: 0x00  
SubVendID: 0x15D9 IntPin: 0x00 Max Lat: 0x00  
SubsysID: 0xF580 Line size: 0x00 Class Code: 0x060000

Hex dump: 8-bit 16-bit 32-bit  
addr: 0  
0x00> 0x86 0x80 0x05 0x34  
0x04> 0x00 0x00 0x10 0x00  
0x08> 0x22 0x00 0x00 0x06  
0x0C> 0x00 0x00 0x00 0x00  
0x10> 0x00 0x00 0x00 0x00

Command: 0x0000 FBB SERR Wait PE VGA MW SpC BM Mem IO  
Header type: 0x00 Multi-func.

Status: 0x0010 PE SE MA TA SA fast DPE B2B user 66 MHz  
BIST: 0x00 BIST capable

Base address registers:  
0: I/O Pre View  
1: I/O Pre View  
2: I/O Pre View  
3: I/O Pre View  
4: I/O Pre View  
5: I/O Pre View

Expansion ROM: Enabled

PCI-PCI bridge:  
Primary Bus: Second. Bus: Subord. Bus: Bridge Ctrl: Diagnostic Save Help OK

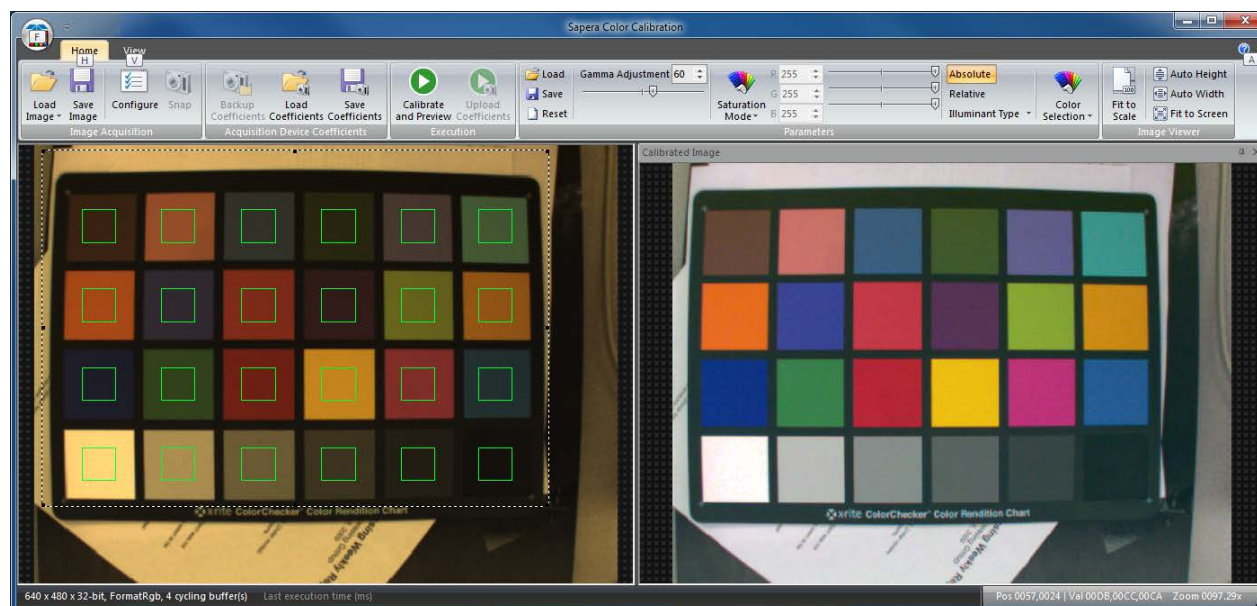
PCIe Device Capability:  
Maximum payload size supported (bytes): 128  
Maximum payload size (bytes): 128  
Maximum read request size (bytes): 128  
Link Speed: Gen 1  
Negotiated Link Width: 4 lanes

---

## 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).





# 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 2005 to 2015.

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 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.



**Note:** You can open the C++ Builder XE1 project file supplied with Sapera LT (Sapera\Examples\Classes\SapExamples.groupproj (32-bit) or the C++ Builder XE5 project file SapExamples64.groupproj (64-bit) or the C++ Builder XE10.1 project file SapExamples\_Clang.groupproj (32-bit) from more recent versions of C++ Builder XE.

---

## Demos and Examples for Frame Grabbers

The following demo programs and corresponding source code is available:

Demo Name	Description
<b>GrabDemo</b>	<p>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.</p> <p>For a tutorial on using this demo, see the Acquiring with Grab Demo section.</p> <p>The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital camera.</p> <p>The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2005.</p>
<b>SeqGrabDemo</b>	<p>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.</p> <p>The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital camera.</p> <p>The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2005.</p>
<b>MultiBoardSyncGrabDemo</b>	<p>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.</p> <p>The minimum requirements for this demo are two Sapera-compatible frame grabbers, each with an analog or digital camera.</p> <p>The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2005.</p>
<b>BayerDemo (deprecated)</b>	<p>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.</p> <p>The minimum requirements to run this demo are a Sapera-compatible frame grabber with a digital Bayer filter camera.</p> <p>The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2005.</p> <p>This demo has been replaced by the Color Conversion demo which uses the SapColorConversion class (the SapBayer class is now deprecated).</p>

<b>ColorConvDemo</b>	<p>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.</p> <p>The minimum requirements to run this demo are a Sapera-compatible frame grabber with a digital Bayer filter or other supported color format camera.</p> <p>The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2005.</p>
<b>FlatFieldDemo</b>	<p>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.</p> <p>For a tutorial on using this demo to create flat field coefficients, see the Using the Flat field Demo section.</p> <p>The minimum requirements to run this demo are a Sapera-compatible frame grabber with an analog or digital monochrome camera.</p> <p>The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2005.</p>
<b>IODemo</b>	<p>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.</p> <p>The minimum requirements to run this demo are a Sapera-compatible board with General I/O capabilities.</p> <p>The supplied executable is built using Sapera LT ++ plus the MFC library under Visual Studio 2005. This example is <i>not available in .NET</i>.</p>

The following compiled console examples and source code is available:

Example Name	Description
<b>GrabLut</b>	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.
<b>GrabMFC</b>	Same as GrabCPP but additionally uses the GUI Spera LT ++ classes to allow for board and camera selection using dialog boxes. This example is <i>not available in .NET</i> .
<b>GrabCameraLink</b>	This example shows how to acquire images using a Teledyne DALSA frame grabber (using the SapAcquisition API class) together with a Teledyne DALSA CameraLink camera which is accessible through the SapAcqDevice API class (such as Falcon2 GenCP camera).
<b>GrabCPP</b>	Grab Console. This example shows how to grab an image from a selected camera into a Spera buffer and then display it. The buffer is created according to the camera settings. Any Spera compatible frame grabber can be used. This example is named Grab in .NET.
<b>GrabConsoleMultiformat</b>	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 <i>not available in .NET</i> .

---

## 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.
GrabMFC	Same as GrabCPP but additionally uses the GUI Samera LT ++ classes to allow for board and camera selection using dialog boxes. This example is <i>not available in .NET</i> .
GrabCameraLink	This example shows how to acquire images using a Teledyne DALSA frame grabber (using the SapAcquisition API class) together with a Teledyne DALSA CameraLink camera which is accessible through the SapAcqDevice API class (such as Falcon2 GenCP camera).
GrabCPP	Grab Console. This example shows how to grab an image from a selected camera into a Samera buffer and then display it. The buffer is created according to the camera settings. Any Samera 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:

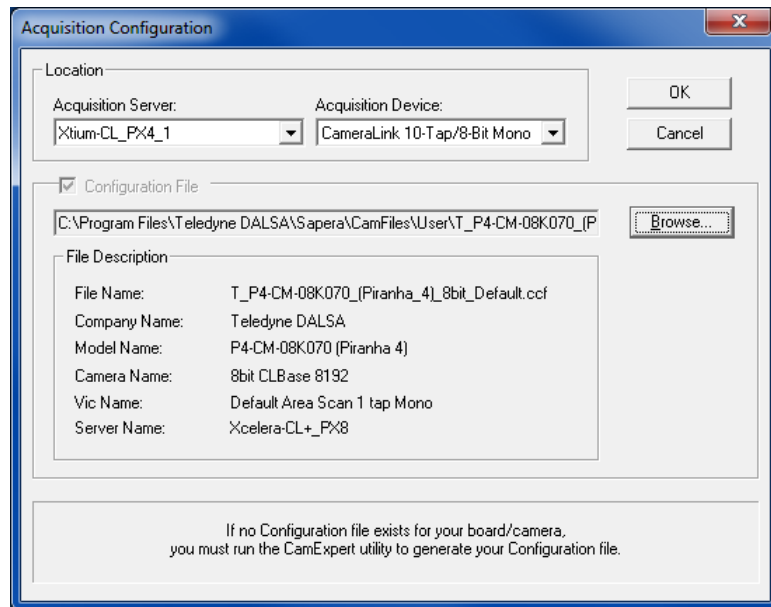
Example Name	Description
<b>ColorSplit</b>	Shows how to split and merge color images into single monochrome components. An RGB image is loaded, split into three monochrome components, then a simple processing is applied to the three components before they are merged back to RGB as output.
<b>FileLoadCPP</b>	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.
<b>FileLoadMFC</b>	Same as the FileLoadCPP Example, but additionally uses the Sapera LT ++ GUI classes allowing file browser usage. This example is <i>not available in .NET</i> .

---

## 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:



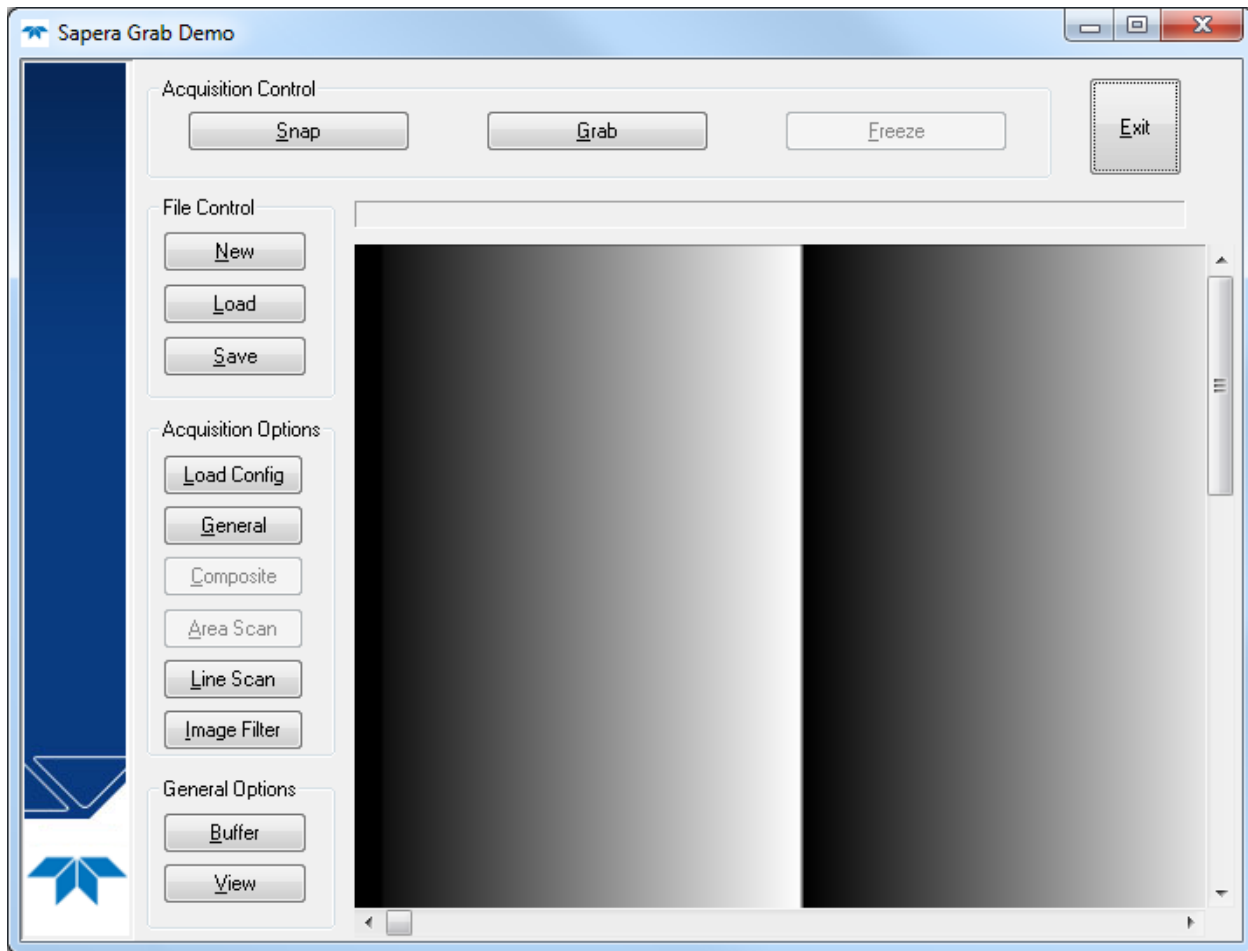
You must select the **Acquisition Server** and the **Acquisition Device**. The first one corresponds to the board you want to grab from; the second represents the acquisition device on this board (some boards may have more than one).

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

<b>Program</b>	<b>Start • Programs • Teledyne DALSA • Sopera LT • Demos • Flat Field Demo</b>
<b>Program file</b>	<b>\\Program Files\\Teledyne DALSA\\Sopera\\Demos\\Classes\\VC\\FlatFieldDemo\\Release\\FlatfieldDemo.exe</b>
<b>Description</b>	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 Sopera 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.
<b>Remarks</b>	This demo is built using Visual C++ 2005 using the MFC library. It is based on Sopera C++ classes. See the Sopera User's and Reference manuals for more information.

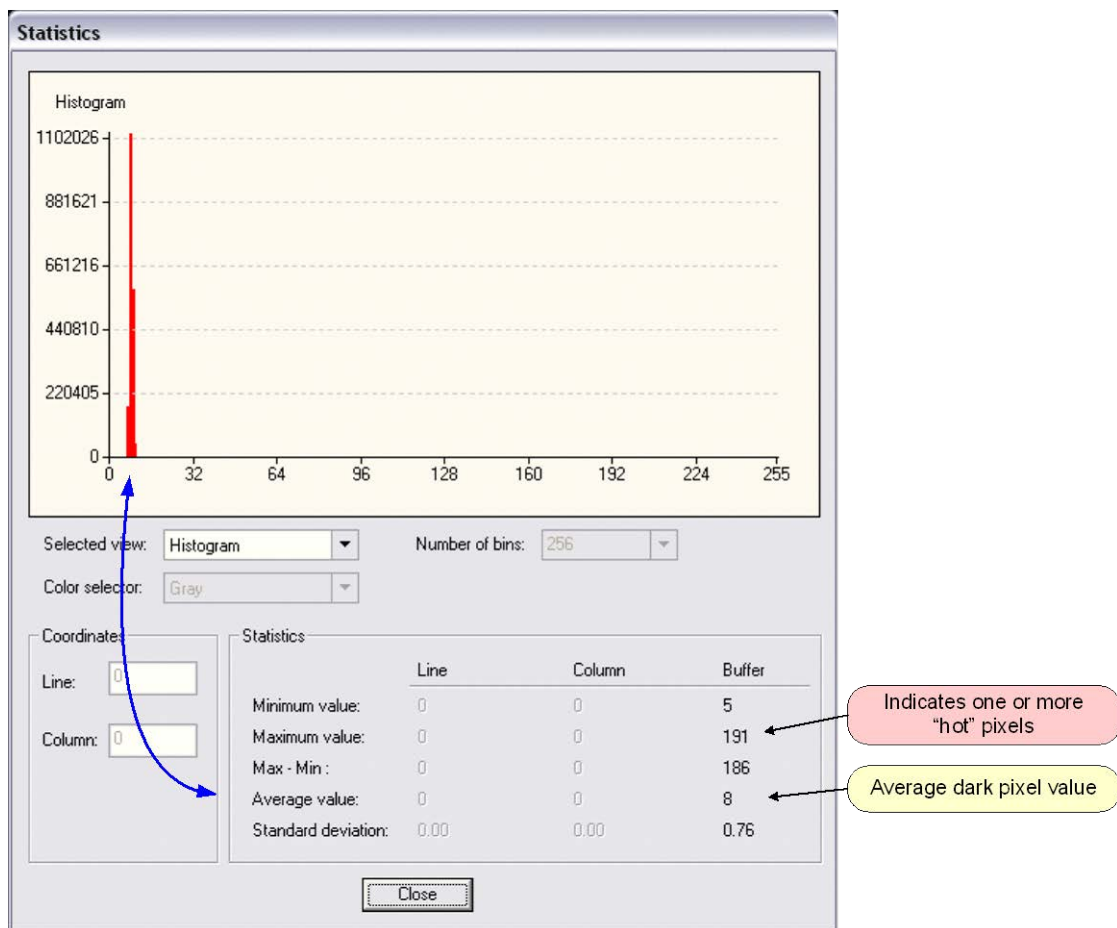
Flat Field Correction is the process of eliminating small gain differences between pixels in a CCD array. When calibrated flat field correction is applied to the image, the CCD exposed to a uniformly lighted field will have no gray level differences between pixels. The Flat Field demo automatically functions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

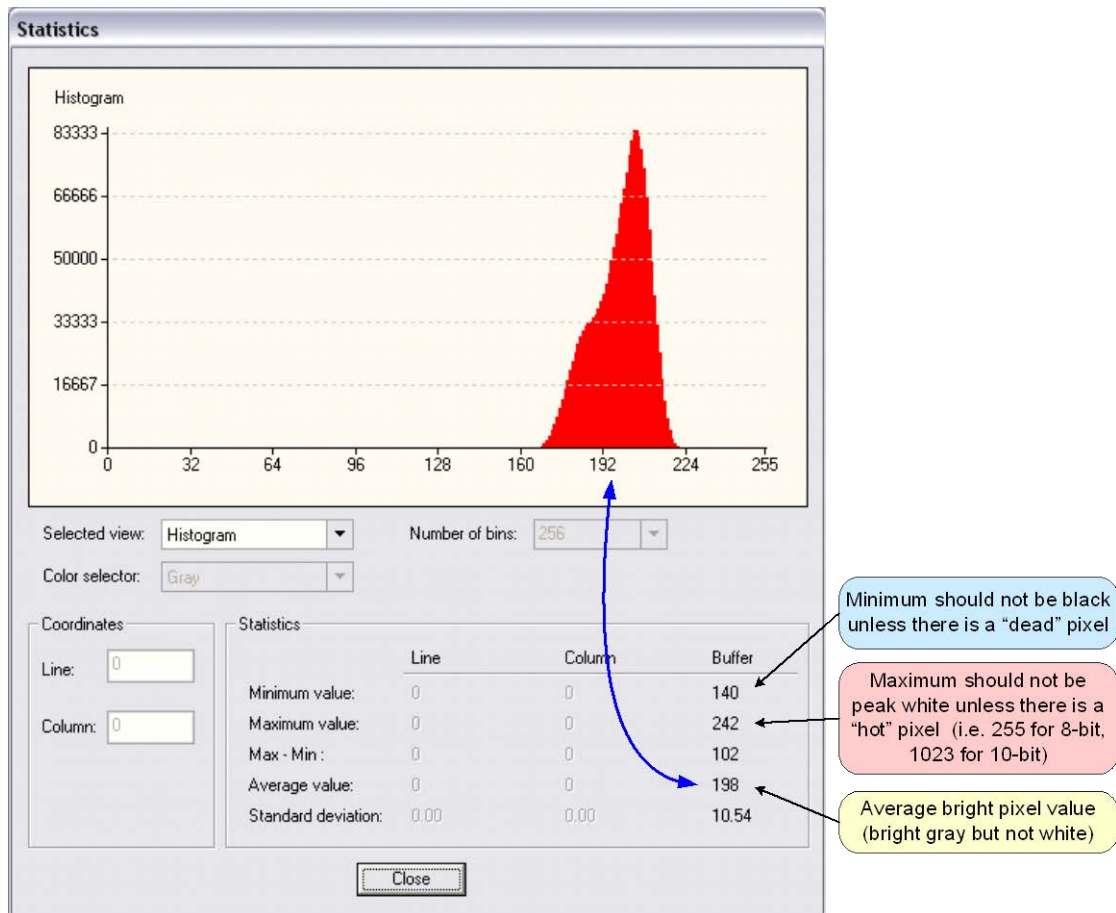
Close the camera lens iris and cover the lens with a lens cap. Using CamExpert, click on the grab button and then the histogram button. The following figure shows a typical histogram for a Genie grabbing a very dark image.



**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.



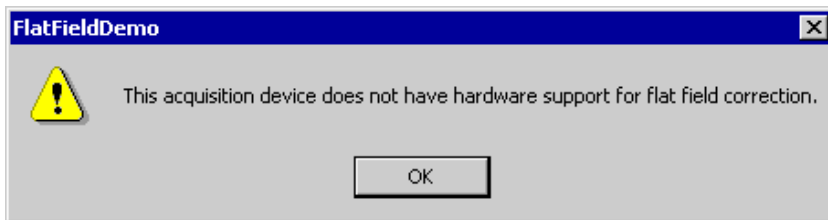
**Important:** In this example, the **average** pixel value for the frame is bright gray. Also note that sensors may show a much higher maximum or a much lower minimum pixel value due to one or more "hot or dead pixels". The sensor specification accounts for a small number of hot, stuck, or dead pixels (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 so as 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 • Sopera LT • Demos (GUI-based) • Sopera ++ Demos (Executables)**. This opens the C:\Program Files\Teledyne DALSA\Sopera\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.

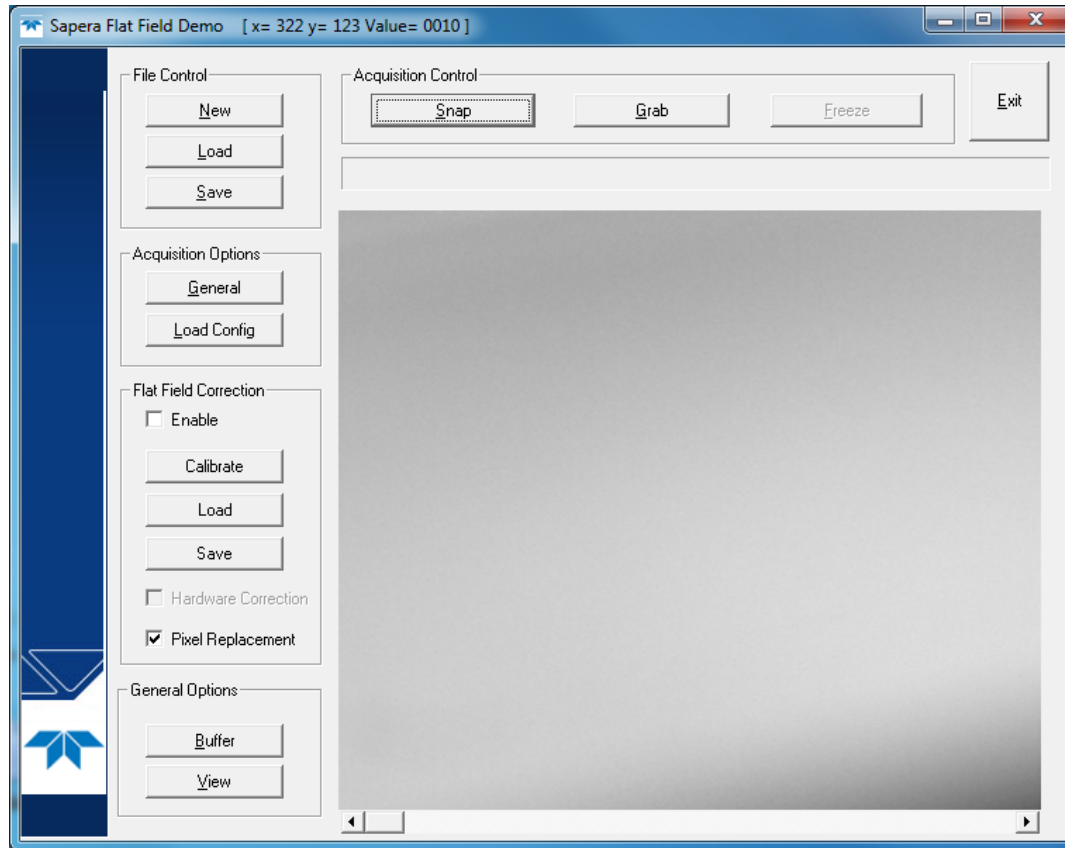
If the selected board does not support onboard flat field processing, a message is displayed (see following figure) stating that processing will be on the host system.



This 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". Sopera 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.



## 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

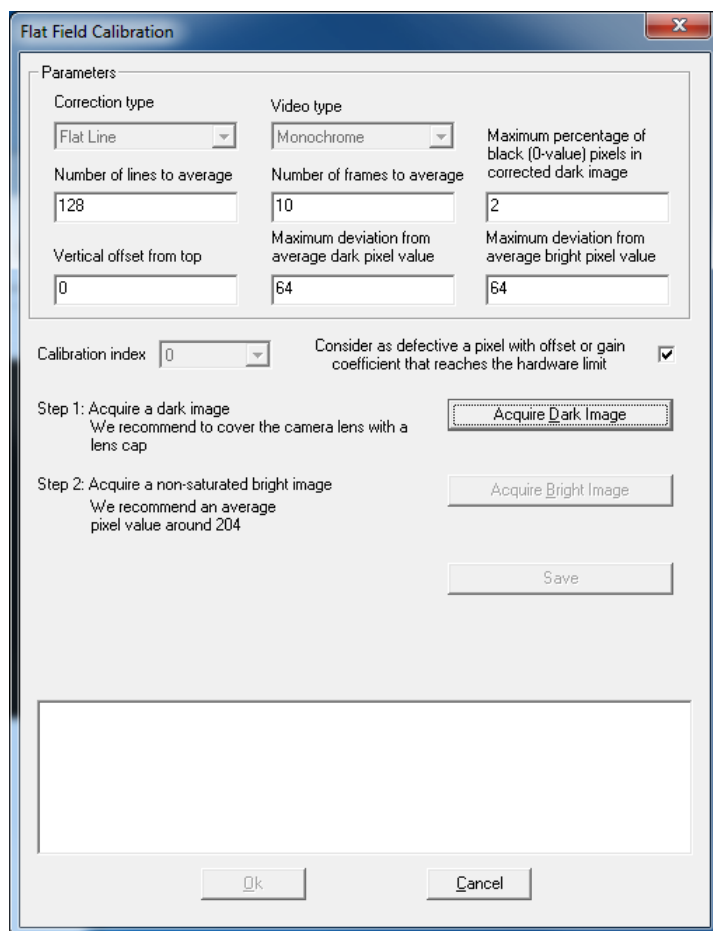
Click on the **Load** button to retrieve files with the flat field gain and offset data gathered with the calibration procedure.

## 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.



The image shows a 'Flat Field Calibration' dialog box with a blue title bar and a close button (X) in the top right corner. The dialog is divided into several sections. The 'Parameters' section contains three columns of controls: 'Correction type' (Flat Line), 'Video type' (Monochrome), and 'Maximum percentage of black (0-value) pixels in corrected dark image' (2). Below these are 'Number of lines to average' (128), 'Number of frames to average' (10), and 'Maximum deviation from average dark pixel value' (64). Further down are 'Vertical offset from top' (0) and 'Maximum deviation from average bright pixel value' (64). A 'Calibration index' dropdown is set to 0, and a checkbox 'Consider as defective a pixel with offset or gain coefficient that reaches the hardware limit' is checked. Below the parameters are two steps: 'Step 1: Acquire a dark image' with a button 'Acquire Dark Image' and a note 'We recommend to cover the camera lens with a lens cap'; and 'Step 2: Acquire a non-saturated bright image' with a button 'Acquire Bright Image' and a note 'We recommend an average pixel value around 204'. A 'Save' button is located below step 2. At the bottom of the dialog are 'Ok' and 'Cancel' buttons.

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	Maximum deviation from average dark pixel value 64
Vertical offset from top 0	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  
[Acquire Dark Image]

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

[Save]

[Ok] [Cancel]

## 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\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\GenICam_2_4	GenICam CLProtocol run-time support for Teledyne DALSA CameraLink cameras
Sapera\Help	On-line documentation (Compiled HTML and PDF formats)
Sapera\Help\VisualStudio	Integrated C++ and .NET help for Visual Studio 2005, 2008, 2010, 2012, 2013 and 2015
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
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:

[www.teledynedalsa.com/corp/contact/](http://www.teledynedalsa.com/corp/contact/)

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	<a href="http://www.teledynedalsa.com/imaging/support">http://www.teledynedalsa.com/imaging/support</a>
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 Sopera Log Viewer .txt file
- The PCI Diagnostic PciDiag.txt file (for frame grabbers)
- The Device Manager BoardInfo.txt file (for frame grabbers)



Note, the Sopera Log Viewer and PCI Diagnostic tools are available from the Windows start menu shortcut **Start • All Programs • Teledyne DALSA • Sopera 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**.