CompuScope Software Development Kit (SDK) for LabVIEW for Windows

User's Guide

Revision 2.03 – 11/02/2021

Gage Contact Information	
Toll Free:	1-800-567-GAGE
Tel:	1-514-633-7447
Fax:	1-514-633-0770
Sales Email:	sales@gage-applied.com
Support Email:	tech-support@gage-applied.com
Web:	http://www.gage-applied.com

Gage is a Product Brand of:

Vitrek, LLC
900 North State Street
Lockport, Illinois 60441-2200
USA

Toll Free: 1-800-DATA-NOW
Tel: 1-815-838-005
Fax: 1-815-838-4424

http://vitrek.com

Changes are periodically made to the information herein; these changes will be incorporated into new editions of the publication. Vitrek, LLC may make improvements and/or changes in the products described in this publication at any time.

How to reach Gage Product Support
Toll-free phone: (800) 567-4243
Toll-free fax: (800) 780-8411

To reach Gage Product Support from outside North America
Tel: +1-514-633-7447
Fax: +1-514-633-0770

E-mail: prodinfo@gage-applied.com
Web site: http://www.gage-applied.com
On-line Support Request Form: http://www.gage-applied.com/support/support-form.php

Table of Contents

Preface

This manual is meant to serve as an aid to engineers using the CompuScope series of high-performance data acquisition cards in the LabVIEW 6.1+ for Windows environment.

Throughout this manual, it is assumed that you are familiar with the LabVIEW graphical programming environment. If you do not feel comfortable with LabVIEW, it is highly recommended that you go through the **Getting Started with LabVIEW for** Windows manual supplied to you by National Instruments, before starting any program development for the CompuScope cards.

It is also assumed that you have correctly installed the CompuScope Windows drivers.

The CompuScope LabVIEW SDK supports all GaGe CompuScope cards – including PCI, PCI Express and USB./PXI. Specific hardware features that are available from the SDK VIs, however, may not be supported by your CompuScope hardware. Please refer to the CompuScope Hardware manual for information specific to your CompuScope card in order to determine the capabilities of your CompuScope.

It is also assumed that you are familiar with PCs and Microsoft Windows. This manual will use the terms "CompuScope SDK for LabVIEW for Windows" and "CompuScope LabVIEW SDK" interchangeably.

Installation of CompuScope LabVIEW SDK

If you purchased the CompuScope LabVIEW SDK you will have been shipped a software key that allows installation of the SDK from the GaGe CompuScope CD. Simply select the installation of the CompuScope LabVIEW SDK from the CompuScope CD and enter the software key when prompted.

By default, on a 32-bit Windows machine, the CompuScope LabVIEW SDK will install itself in:

O/S system drive:\Program Files\Gage\CompuScope\CompuScope LabVIEW SDK.

By default, on a 64-bit Windows machine, the CompuScope LabVIEW SDK will install itself in:

O/S system drive:\Program Files (x86)\Gage\CompuScope\CompuScope LabVIEW SDK.

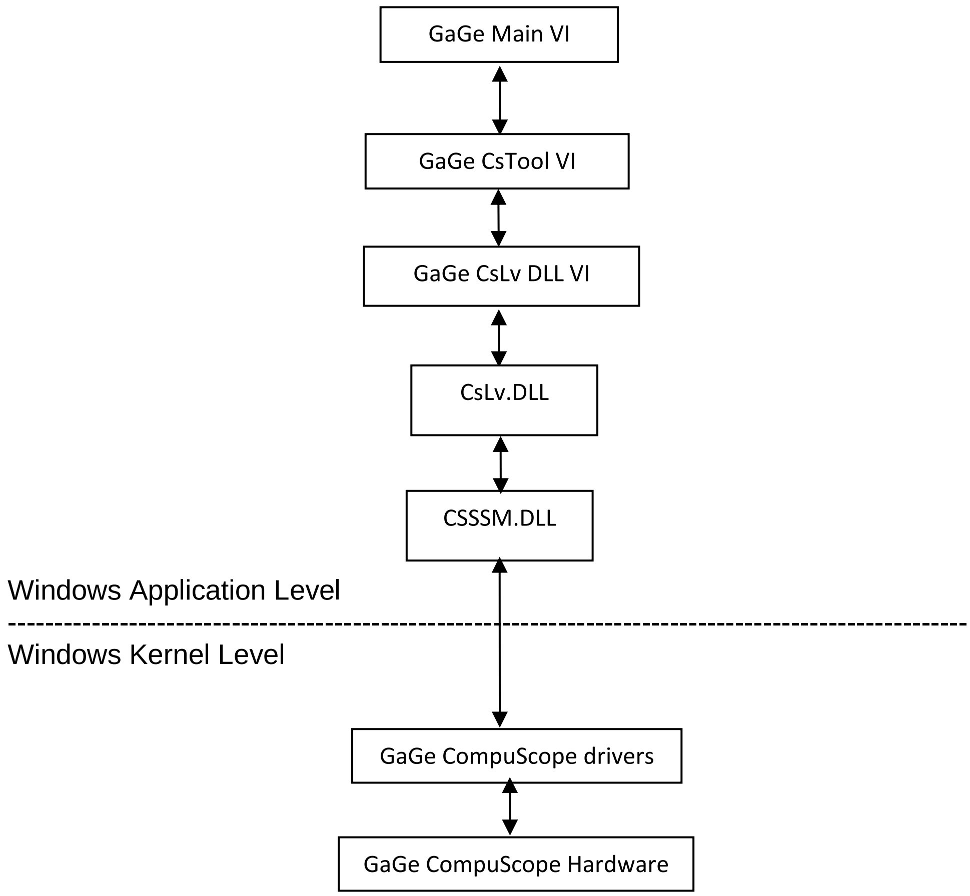
It is recommended that you use the default installation location.

If you require more detailed installation instructions, please refer to the Getting Started menu item on the Gage Software CD.

Overview of CompuScope LabVIEW SDK

Structure of CompuScope LabVIEW SDK

The overall structure of the CompuScope LabVIEW SDK and its relation to the GaGe CompuScope Hardware is best described from the bottom up with reference to the diagram below.



At the lowest level is the CompuScope hardware, which is installed within a slot that is connected to the host PC. The CompuScope hardware is directly controlled by the CompuScope Windows drivers. The drivers reside at the Windows Kernel level, which allows direct low-level access to CompuScope hardware registers and to physical PC RAM. The drivers communicate with Windows Applications through a Dynamically Linked Library (DLL) called CSSSM.DLL.

Communications through CSSSM.DLL uses the CompuScope Applications Programming Interface (API), which is a set of C subroutine calls that allows control of all CompuScope functionality. Above CSSSM.DLL is a small intermediate DLL called CsLV.DLL that packages the API calls so that they may be easily called from LabVIEW. Above CsLV.DLL sits three layers of LabVIEW VIs, called *CsLv DLL VI*s, *CsTool VI*s and *Main VIs*.

The Main VIs are top-level application VIs that can be executed directly under LabVIEW. Each Main VI illustrates usage of the CompuScope hardware in a different operating mode. The Main VIs are intended as convenient starting points for CompuScope users to develop into their own LabVIEW applications. Consequently, the controls and indicators of the Main VIs are not wired to the VI Icons and the Main VIs should not be called as sub-VIs by other VIs.

The CsTool VIs are an easy-to-use complete set of sub-VIs for calling the GaGe hardware. All Main VIs are constructed using only CsTool VIs. While they may not provide the best possible performance, CsTool VIs are extremely easy to understand and use.

The CsLv DLL VIs are simple wrapper VIs that make direct Library calls to CsLV.DLL, which is the link from LabVIEW to the CompuScope drivers. All CsTool VIs are constructed from CsLv DLL VIs. Direct usage of the CsLv DLL VIs is intended only for advanced LabVIEW users who wish to improve upon the hardware performance that is possible using the CsTool VIs. Finally, CseXpert Vis are used for support of eXpert on-board data processing images.

All three sub-VI types used by the sample Vis (CsTool VIs, CsLv VIs and CseXpert VIs) are installed within LLB libraries within LabVIEW's "instr.lib" folder. Accordingly, they may be selected from the "User Libraries" item on the LabVIEW Tools Palette. These three sub-VI types are documented within LabVIEW as "Context Help", the text of which is also dumped to PDF files that are installed with the SDK.

CompuScope Systems

A CompuScope system is defined as a single CompuScope board or a group of CompuScope boards configured as a Master/Slave multi-card CompuScope system. Since Master/Slave CompuScope systems sample, trigger and reset simultaneously on all channels, it is considered to be one multi-channel CompuScope system. For instance, a Master/Slave system composed of four two-channel CompuScope boards will be considered to be a single CompuScope system with eight available input channels.

By structuring the drivers to consider CompuScope systems, a single PC can be equipped with almost any imaginable combination of CompuScope hardware. For instance, a PC could be equipped with two separate Master/Slave systems of four channels each and then an additional single independent CompuScope board for a total of three CompuScope systems. The CompuScope Manager utility may be used to separately list all CompuScope systems in the PC.

CompuScope systems are addressed from LabVIEW sample programs by first obtaining a *Handle* for the System. After usage of the system is complete, the user must release the Handle so that it is free for usage by other processes. By obtaining handles for different systems, a single LabVIEW VI may simultaneously operate different CompuScope systems. Alternately, separate LabVIEW VI may operate independently and simultaneously operate separate CompuScope systems by calling handles for each one. Different VIs may even access the same hardware as long as one VI frees the system handle before the other VI obtains it. This is because different applications may not simultaneously access the same CompuScope system.

While most sample VIs access only a single system, understanding the CompuScope system structure allows users to easily extend these sample VIs to multiple CompuScope system operation.

Overview of CompuScope Main VIs

The Main VIs are intended to be convenient starting points around which users can develop customized LabVIEW software for their digitizer application. Users can construct more complex VIs themselves by using the existing VIs as a guide for combining their functionalities.

The basic algorithm for all of the Main VIs is the following:

1. *Initialize the CompuScope driver and the CompuScope hardware.*

2. *Obtain the handle to the first available CompuScope system in the PC.*

3. *Pass the desired configuration settings to the driver using CsTool-Configure VIs*

4. *Pass the configured settings to the CompuScope hardware using the CsTool-Commit VI.*

5. *Start the CompuScope hardware to digitize data into its on-board memory and await a trigger event.*

6. *Continuously check to see if the CompuScope hardware has finished capturing the current record.*

7. *Download the data of interest from the CompuScope hardware to LabVIEW array variables.*

8. *Plot the data.*

A list of all CompuScope Main VIs is given below with a brief description of the functionalities.

GageAcquire.vi: The simplest VI with full configuration control of CompuScope settings that illustrates usage of CompuScope Single Record Mode. An error occurs upon entry of invalid settings.

GageCoerce.vi: A VI that is just like GageAcquire.vi, except that invalid settings are coerced to the closest valid values. A panel appears that indicates the actual coerced settings.

GageMultipleRecord.vi: A VI that acquires and displays captured records when operating a CompuScope system in Multiple Record Mode.

GageDeepAcquisition.vi: A VI that acquires and manages large data captures from CompuScope Hardware (> 16 MB).

GageComplexTriggerAcquire.vi: A VI that illustrates usage of complex triggering using multiple on-board trigger engines, if available.

GageTwoIndependentSystemsAcquire.vi: A VI that operates two independent CompuScope systems, each with their own setting controls and display charts.

Not all possible CompuScope operation configurations are covered by the Main VIs. For instance, there is a VI that handles deep acquisitions and a VI that handles acquisitions from two CompuScope systems. However, there is no VI that handles deep acquisitions from two CompuScope systems. For both these functionalities, the user must study GageDeepAcquisition.vi and GageTwoIndependentSystemsAcquire.vi and then appropriately combine them to create a VI that meets the requirement.

Most SDK Vis include a “Continuous” button, which sets the VI to acquire repetitively. When this button is not activated, the VI will perform generally perform a single waveform acquisition and then terminate. When “Continuous” is activated, however, the VI should always be stopped using the STOP button provided. Stopping an SDK in another fashion will not correctly free CompuScope handles and other processes will be unable to use them. Such locked handles can be freed by invoking the CsTool-FreeSystem or CsLv_FreeSystem VIs if the handle value is known. If the locked handle value is not known, CsLv_FreeAllSystems.vi can be used to free all CompuScope systems. Finally, locked handles may be freed from outside of LabVIEW by doing a “Refresh” from the CompuScope Manager utility after you have closed down LabVIEW.

Overview of CompuScope CsTool VIs

The CsTool VIs are easy-to-use building blocks from which the Main VIs are built. A LabVIEW user is able to develop a custom LabVIEW VI for their requirement using only the CsTool VIs, with no modifications to them. All

CsTool VIs are extensively documented in the LabVIEW Context Help. For convenience, a listing of all the CsTool descriptions is also given in a file called CsTool-VI_Guide.pdf that is installed with the LabVIEW SDK.

Most CsTool VIs, require a Handle Cluster input, which identifies the CompuScope system on which the CsTool VI will operate. The Handle Cluster contains static information about the CompuScope system. All CsTool VIs return error status values, which may be translated into descriptive error message strings using CsTool-ErrorHandler.vi.

Where appropriate, CsTool VIs use descriptive string variables rather than index values for VI readability and easy programming. (By contrast, lower-level CsLv VIs generally use indices). Note that valid string values are not case-sensitive, but must the exact spelling must be used. For example, “8-Bit”, “8-BIT”, or “8-bit” are acceptable as Sample Bits inputs, however, “8 bit” is not.

CsTools VIS are divided into six different types. Each CsTool type is associated with its own distinctive LabVIEW icon. The six CsTool VI types listed below.

	Contact VIs	These CsTool VIs are used to establish or break communication with a specific CompuScope system.
	Get VIs	These CsTool VIs are used to obtain capability information about the specified CompuScope system.
	Configuration VIs	These CsTool VIs are used to adjust configuration settings in the drivers for the specified CompuScope system. The settings are not actually passed to the CompuScope System until a call to CsTool-Commit.vi is made.
	Queries VIs	These CsTool VIs are used to obtain the actual configuration settings on the specified CompuScope system.
	Action VIs	These CsTool VIs are used perform actions on the specified CompuScope system, such as loading settings, starting acquisitions and transferring data.
	Error handler VI	This VI is used to decode error number codes and display descriptive error messages.

Overview of CompuScope CsLv DLL VIs

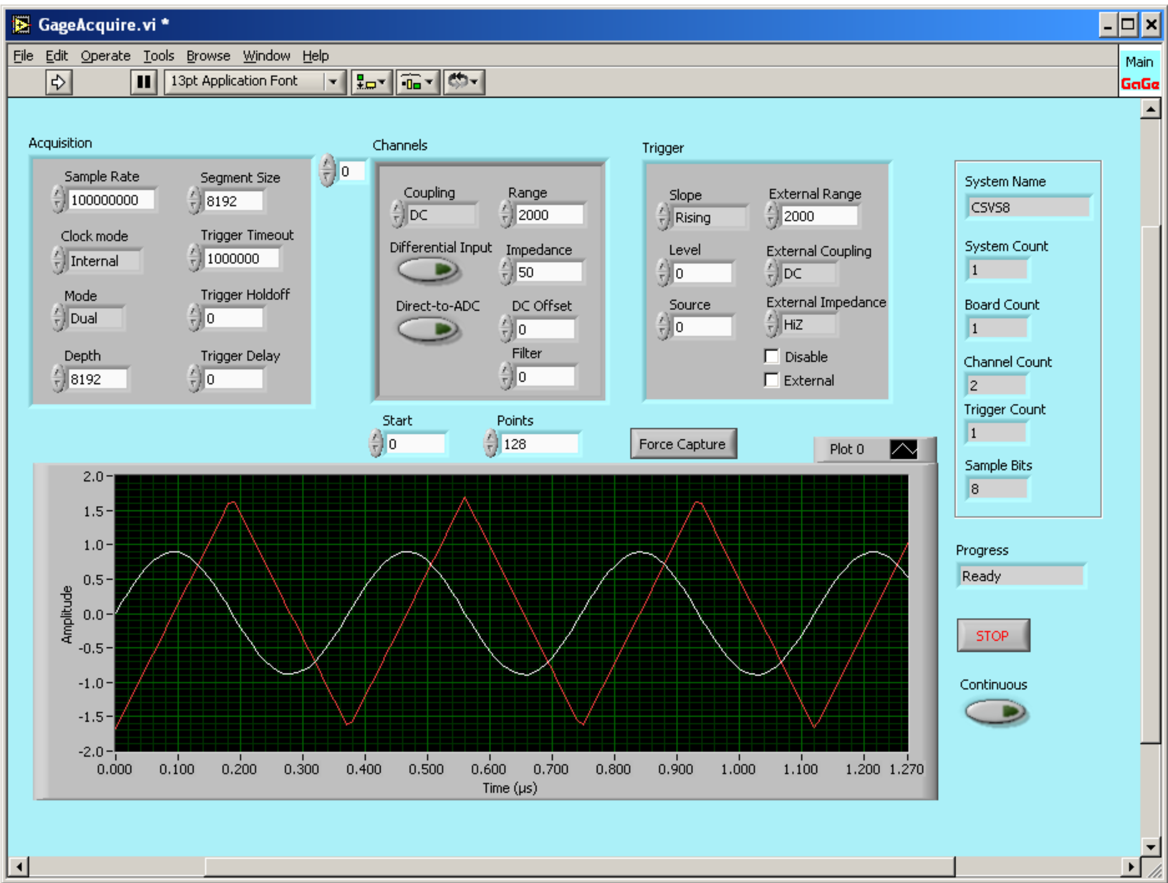
The CsLv DLL VIs are the lowest level of SDK VIs. They are simple wrapper VIs that make LabVIEW Library calls to an intermediate DLL called CsLv.DLL. This DLL simply translates calls into CompuScope C API calls that are passed along to CSSSM.DLL – the CompuScope driver DLL.

All CsLv DLL VIs are extensively documented in the LabVIEW Context Help. For convenience, a listing of all the CsLv DLL VI descriptions is also given in a file called CsLv-VI_Guide.pdf that is installed with the LabVIEW SDK.

Direct usage of CsLv DLL VIs is intended only for the advanced LabVIEW user. Advanced users can develop or by-pass CsTool VIs or directly use CsLv DLL VIs in order to achieve the best possible hardware performance under LabVIEW. For instance, an advanced user might pull out the CsLv DLL data transfer VI from the CsTool-Transfer VI and forego conversion of the raw data into Volts in order to achieve the best repetitive capture performance.

Detailed Description of CompuScope Main VIs

GageAcquire.vi



GageAcquire.vi is the Main VI for Single Record capture from a single CompuScope system. In the event of Multiple CompuScope systems, the VI only operates the first CompuScope system in the PC.

The VI accepts configuration settings from the configuration controls and uses them to operate the CompuScope system. Acquisition operations occur sequentially in a LabVIEW sequence. Data are displayed in the large display plot. The VI can be configured to operate continuously by pressing the Continuous button. Press the STOP button to terminate the VI. The Progress indicator shows the current state of the CompuScope system.

Waveform data are displayed as Amplitude in Volts against Time in microseconds. The time axis does not use Auto-Scaling since small fluctuations in the actual start of the data may lead to a fluctuating x-axis minimum.

If any of the configuration settings are determined by the VI to be invalid for the active CompuScope system, the VI will STOP and show a descriptive error string. Users who prefer the VI to continue operating upon invalid setting entry should use GageCoerce.vi, which will coerce invalid entries to valid entries and continue running.

Please note that when the Channels index is changed from the default value (0), the controls will appear grayed out. This does not mean that you cannot change the control settings. You simply have to click on any of the grayed-out controls to activate them.

The VI grabs the handle to the first available CompuScope system and displays information about the system in the boxed indicator at the top right of the front panel. The displayed information is shown below:

- System Name – A string indicating the CompuScope model name of the hardware system in use.
- System Count – The total number of detected CompuScope systems.
- Board Count – The total number of CompuScope boards in the active CompuScope system.
- Channel Count – The total number of input channels in the active CompuScope system.
- Trigger Count – The total number of trigger engines in the active CompuScope system.
- Sample bits – The vertical resolution of the hardware in the active CompuScope system.

The type and entries for all controls and indicators are listed below:

Acquisition Control: Cluster of the Acquisition Setting Controls listed below.

- Sample Rate – The Sampling rate in Hz.
- Clock Mode – A ring control that specifies the clock mode for the CompuScope system. The available modes are:
    - Internal clock (use the built-in internal sampling oscillator of the CompuScope system).
    - External clock (use a clock signal connected to the external clock input of the CompuScope system as the sampling clock)
    - Reference clock (use a 10 MHz external clock signal connected to the external clock input of the CompuScope system to synchronize the internal CompuScope sampling oscillator).
- Mode – Set to “Single”, “Dual”, “Quad” or “Octal” for single-channel, dual-channel, four-channel or eight-channel acquisition mode.
- Depth – The post-trigger depth, which is the number of samples to acquire after the trigger event.
- Segment Size – Controls the total amount of memory allocated to the acquisition. The maximum possible amount of pre-trigger data that can be acquired, therefore, is (Segment Size - Depth).
- Trigger Timeout – Trigger Timeout in microseconds, which is the amount of time that the VI will wait before forcing a trigger event.
- Trigger HoldOff – The time in samples during which trigger events will be ignored after the CompuScope hardware begins acquiring and awaiting a trigger event. The function is useful for ensuring the accumulation of a specified amount of pre-trigger data that is equal to the Trigger HoldOff value. For newer CompuScope models that support rectangular memory architecture the Trigger HoldOff must be

set equal to the difference between the Segment Size and the Depth. (See "Memory organization on CompuScope Cards" in CompuScope Hardware Manual and "Special Topics" section of this manual).

- Trigger Delay – The time in samples between the logging of the trigger event and beginning of the count-down of the post-trigger depth counter. This function is useful for waveforms in which the signal feature of interest occurs long after the trigger event.

Channel Control: An array of clusters of the Channel Setting Controls listed below. The Array index determines the channel number, starting from 0.

- Coupling – A ring control that contains "DC" and "AC" for the two input coupling modes.
- Differential – A button that activates Differential input mode, for CompuScope hardware that supports this feature.
- Direct-to-ADC – A button that activates Direct-to-ADC input mode, for CompuScope hardware that supports it.
- Input Range – The full-scale input range in millivolts. For instance, for the +/-1 Volt range, enter 2000.
- Impedance – An integer control containing 1000000 (1 MOhm) and 50 (50 Ohms) for the two possible input terminations.
- DC Offset – The DC offset in millivolts that will be subtracted from the input signal, changing the effective input range. For instance, a 200 mV DC offset will change the extrema of the ±1 Volt input range to 0.8 V and -1.2 V. DC offset is not supported by all CompuScope models.
- Filter – Many CompuScope models support independent application of a low-pass filter to each channel input in order to reduce signal noise. The filter is activated by setting the Filter control to 1, while a 0 deactivates the filter.

Trigger Control: A Cluster of Trigger Setting Controls

- Source – An integer which sets the channel trigger source if one is required. Channel trigger sources begin at 0 (for Channel 1) and can continue up to the number of channels available in the system. The Source is used if both the External and Disable buttons are not selected.
  External – A checkbox that sets the trigger source to be an external trigger. If this box is selected, the Source field is ignored and the External Range and External Coupling fields are used. Selecting both the External and Disable checkboxes will cause an error.

- Disable – A checkbox that disables the trigger source for the system. If this box is selected, the Source field is ignored and the system will not trigger unless CsTool-ForceCapture.vi is called. Selecting both the External and Disable checkboxes will cause an error.

- Level – The Trigger Level between ±100%. The percentage is with respect to the input range of the trigger source. For example, if the Trigger source is set to Channel 2 and this channel uses a ±2 V input range, then ±100% trigger level settings refer to ±2 V.

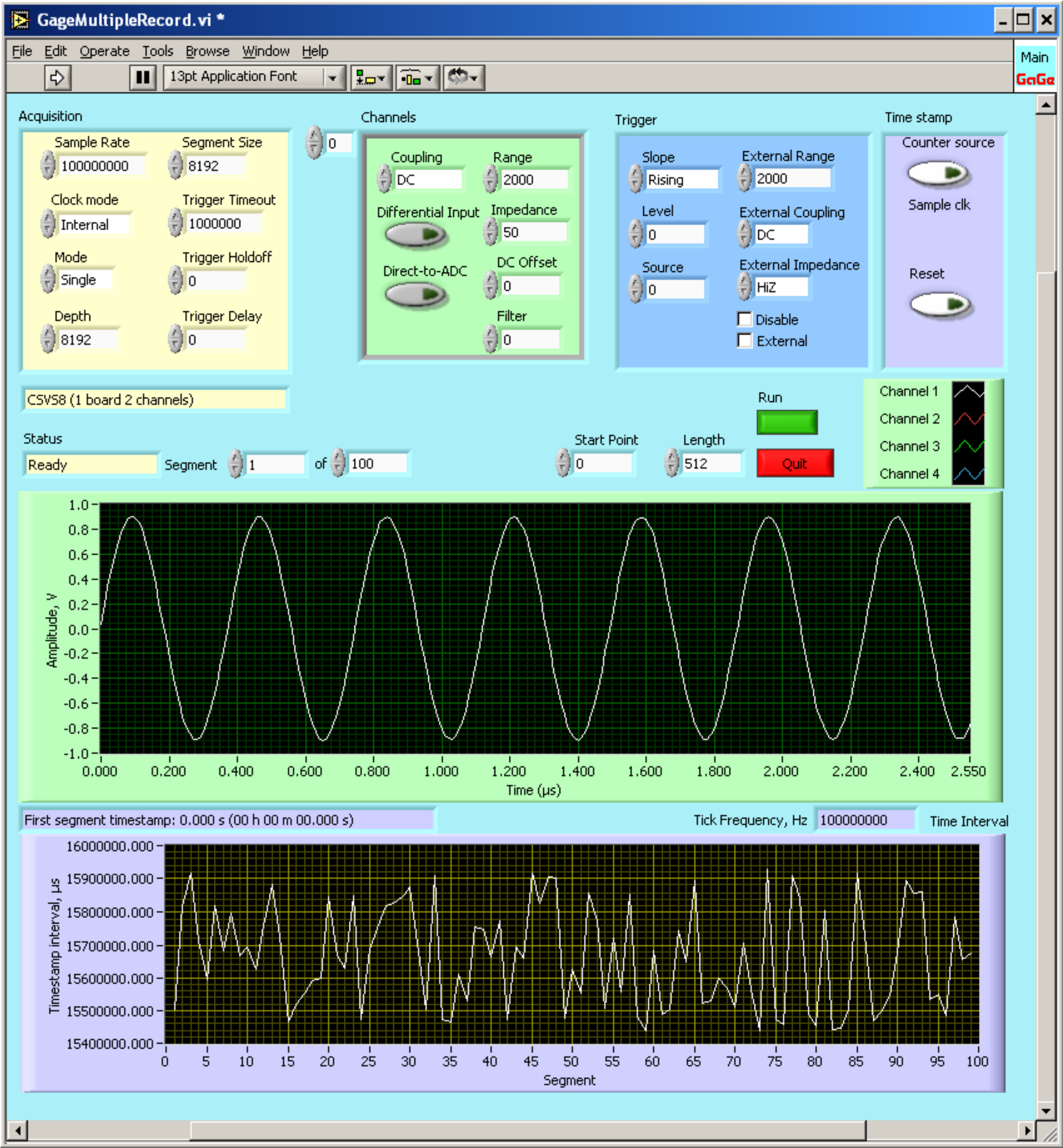
- Slope – A ring control that sets the trigger slope to "RISING" or "FALLING".

- External Range – List Box – An integer control that sets the full scale of external trigger input range in millivolts. For instance, for the ± 5 Volt range, select 10000.
- External Coupling – A ring control that contains “DC” and “AC” for the two external trigger coupling modes.
- External Impedance - A ring control that contains “HiZ” and “50 Ohms” for the two external trigger impedance settings.

Transfer Control: Two variables that determine the data to be transferred, once captured:

- Start – The point from which to start data transfer. Zero indicates the trigger address. Enter a negative value in order to download pre-trigger data.
- Points – The number of points to transfer, starting from the Start point.

GageMultipleRecord.vi



GageMultipleRecord.vi is the Main VI for Multiple Record capture from a single CompuScope system. Multiple Record mode allows multiple waveforms to be rapidly acquired and stacked in on-board CompuScope memory. For instance, a CompuScope card with 32 MegaSamples of on-board memory may acquire 16,000 records of 2000 samples each in Multiple Record mode. (Strictly speaking, depending on the CompuScope model, this relation may not exactly apply because of slight inter-record padding requirements.)

Between successive acquisitions, the CompuScope acquisition engine is re-rearmed in by the hardware with no CPU interaction required. Consequently, in Multiple Record mode, a CompuScope is capable of capturing bursts of trigger that repeat at rates of 100,000 triggers per second and more.

The Number of Records setting is used to enter the number of records desired. If this exceeds the maximum possible number of records, which is roughly equal to the available acquisition memory divided by the Depth, then an “Invalid Segment Count” error will occur. (If coercion had been used in this case, then the actual number of records would have been set equal to its maximum possible value.)

The Segment Size control is used to control the size of each Multiple Record segment. For newer CompuScope hardware, this may be set larger than the Depth so that pre-trigger data may be acquired. Older CompuScope models do not support pre-trigger data in Multiple Record mode. Consequently, setting the Segment Size to a value different from the Depth will cause an error. Please refer to your CompuScope Hardware Manual for information specific to your CompuScope model’s capabilities.

When executed, GageMultipleRecord.vi first initializes and configures the CompuScope system, as in GageAcquire.vi. Next, a Multiple Record Acquisitions are performed.

During acquisition, the user may display any record acquired in Multiple Record Mode. Multiple records are displayed in a fashion that is identical to the MulRec Playback feature in GageScope. The front panel contains a string that reads “Record XX of YY”, where YY is the Number of Records. XX is the Segment control value that may be used to select the record to be downloaded and displayed. Since, with each change of XX, the VI downloads and displays a new record, only one record is resident in the VI at a time. The user may easily modify the VI so that the records are analyzed and/or stored, instead of being displayed.

Most CompuScope models support Time-Stamping, where each Multiple Record trigger event stores a counter value that indicates the trigger occurrence time. If the CompuScope system is equipped with on-board Time-Stamping, GageMultipleRecord.vi displays the Time Stamp for the displayed record in microseconds. Otherwise, 0’s are displayed as the Time Stamp value.

Time zero is the time at which the time-stamping counter was last reset. For convenience, the Time Interval display shows the difference between the Time Stamps for the current record and the previous record.

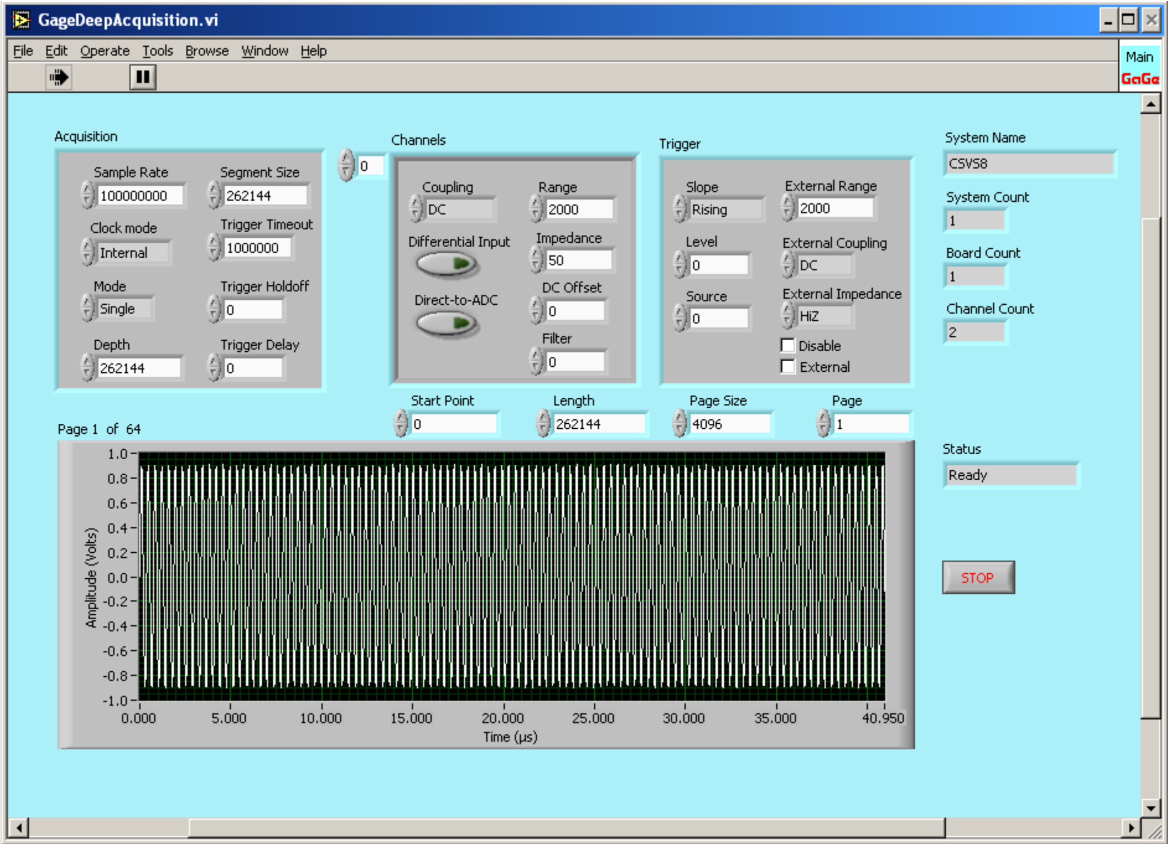
The Time stamp Reset button, when activated, will reset the time stamp counter at the beginning of each Multiple Record acquisition. Otherwise, the counter is never reset.

The Time stamp Clock source button selects the clock source of the time stamp counter: Fixed or Sample. “Fixed” selects an internal fixed oscillator that is asynchronous to the sampling. “Sample” selects a source that is synchronous to the sampling clock. “Sample” generally provides the greatest accuracy.

All remaining controls and indicators for GageMultipleRecord.vi are identical to those for GageAcquire.vi.

If your CompuScope model supports rectangular memory architecture, then you can use the advanced sample VI called GageAdvMulRecEx.vi to download all Multiple Records in a single data transfer. Please see the “CompuScope Memory Organization” section in the CompuScope hardware manual for more information.

GageDeepAcquisition.vi



GageDeepAcquisition.vi is the Main VI for large acquisitions from a single CompuScope system. The definition of large varies with System configuration but is roughly of order 16 MegaBytes. For small acquisitions, LabVIEW is fully capable of downloading the entire acquisition into PC RAM. For larger sizes, the host PC will begin to use "virtual RAM", which is a section of the hard drive that Windows treats like PC RAM. Usage of virtual RAM causes the hard drive to spin and slows down execution. For extremely large enough data sets, LabVIEW will complain that it cannot allocate sufficient memory to hold all the downloaded data.

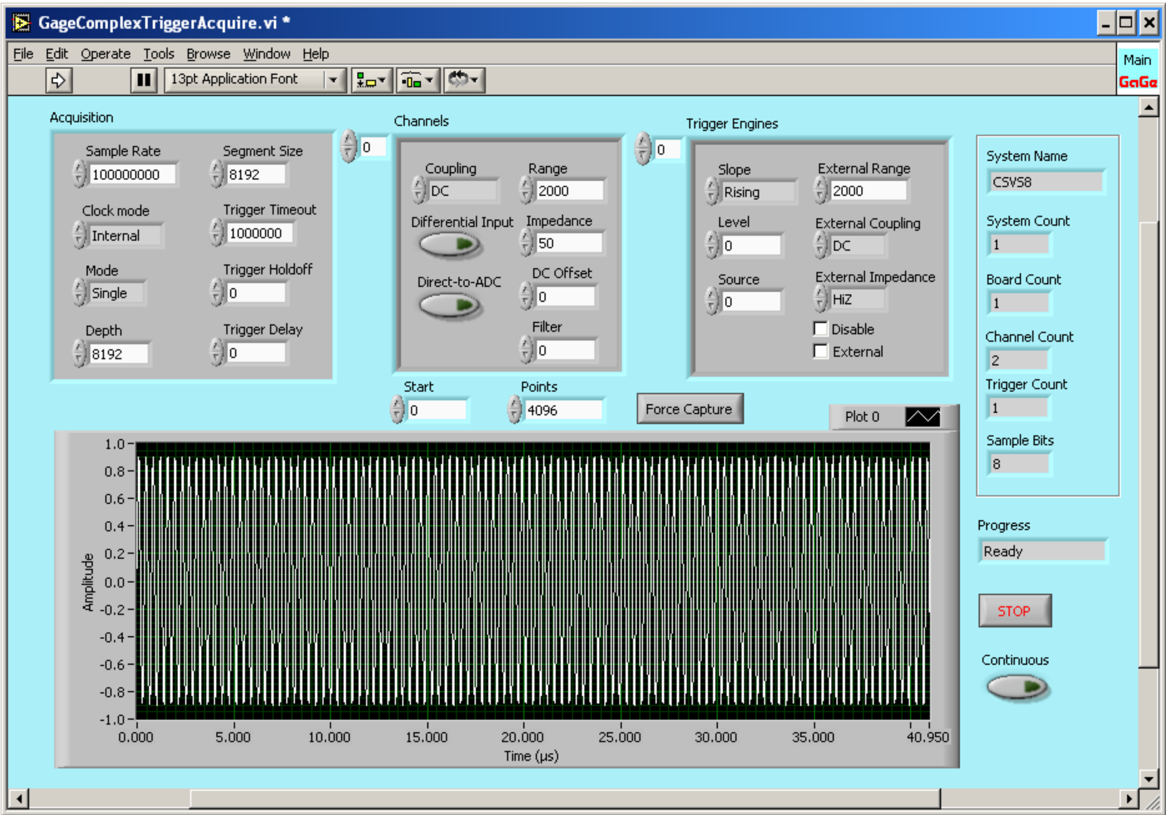
In order to avoid keeping large data sets in LabVIEW at one time, GageDeepAcquisition.vi manages deep acquisitions by dividing them up into more manageable data *pages*. The VI does initialization, configuration setting and acquisitions as usual. After the acquisition, however, only a single data page of the acquisition is downloaded and displayed.

After performing an acquisition, the VI goes into a loop that allows the user to select and display any page from the large acquisition. There is a control/indicator pair that shows "Page XX of YY", where XX is the value in the adjustable Page control. YY is the total number of Pages that is equal to (Length/Page Size). With each change of Page control value, the VI downloads and displays a new page, so that only one page is resident in the VI at a time. The adjustable Page Size default is 32768. The time x-axis is offset to indicate the page position in time. For example, if the user displays Page 11 of a capture taken at 100 MS/s, the lower x-axis limit would be 10 X 32k / 100 MS/s = 326.78 milliseconds.

The user is able to flip up and down through the pages until the STOP button is pressed. The user can easily modify the VI so that the pages are analyzed and/or stored instead of being displayed.

All remaining controls and indicators for GageDeepAcquisition.vi are identical to those for GageAcquire.vi.

GageComplexTriggerAcquire.vi



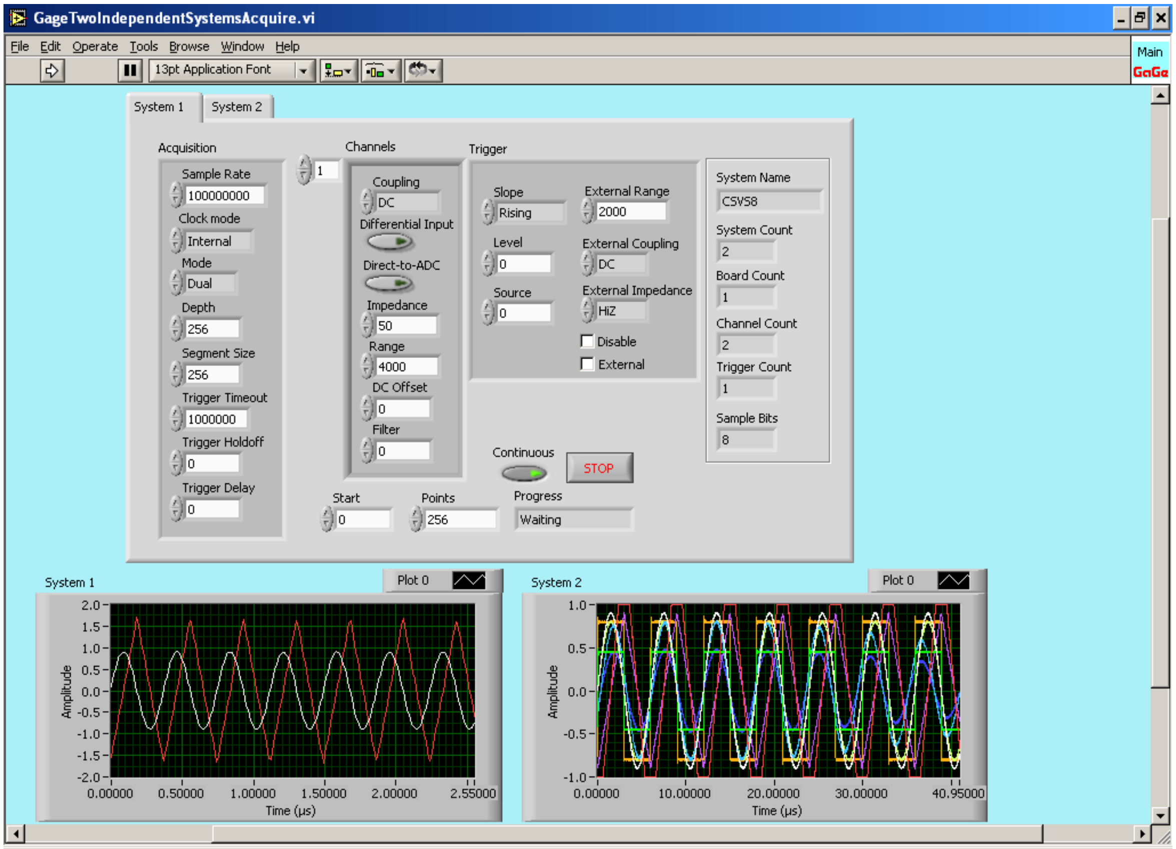
GageComplexTriggerAcquire.vi is a main VI that illustrates complex triggering. Some CompuScope models are equipped with two on-board trigger engines that can be used for complex triggering. On these models, the two engines can be configured independently and their outputs are Boolean ORed together so that either engine may cause a trigger event. For simple triggering, the second engine is disabled.

The Triggers array control allows the separate configuration of each trigger engine. For instance, by setting the first two engine sources to 0 and 1, the user may configure the system to trigger on a pulse that occurs on either channel. (In CompuScope LabVIEW systems, channels start at 0). Alternately, by setting the first two engine sources to 0 but selecting different levels and slopes for each, the user may configure the system to do windowed triggering, where the system triggers if the input level leaves a specified voltage range.

Please note that when the Trigger Engines index is changed from the default value (0), the controls will appear grayed out. This does not mean that you cannot change the control settings. You simply have to click on any of the grayed-out controls in order to activate them.

All remaining controls and indicators for GageComplexTriggerAcquire.vi are identical to those for GageAcquire.vi.

GageTwoIndependentSystemsAcquire.vi



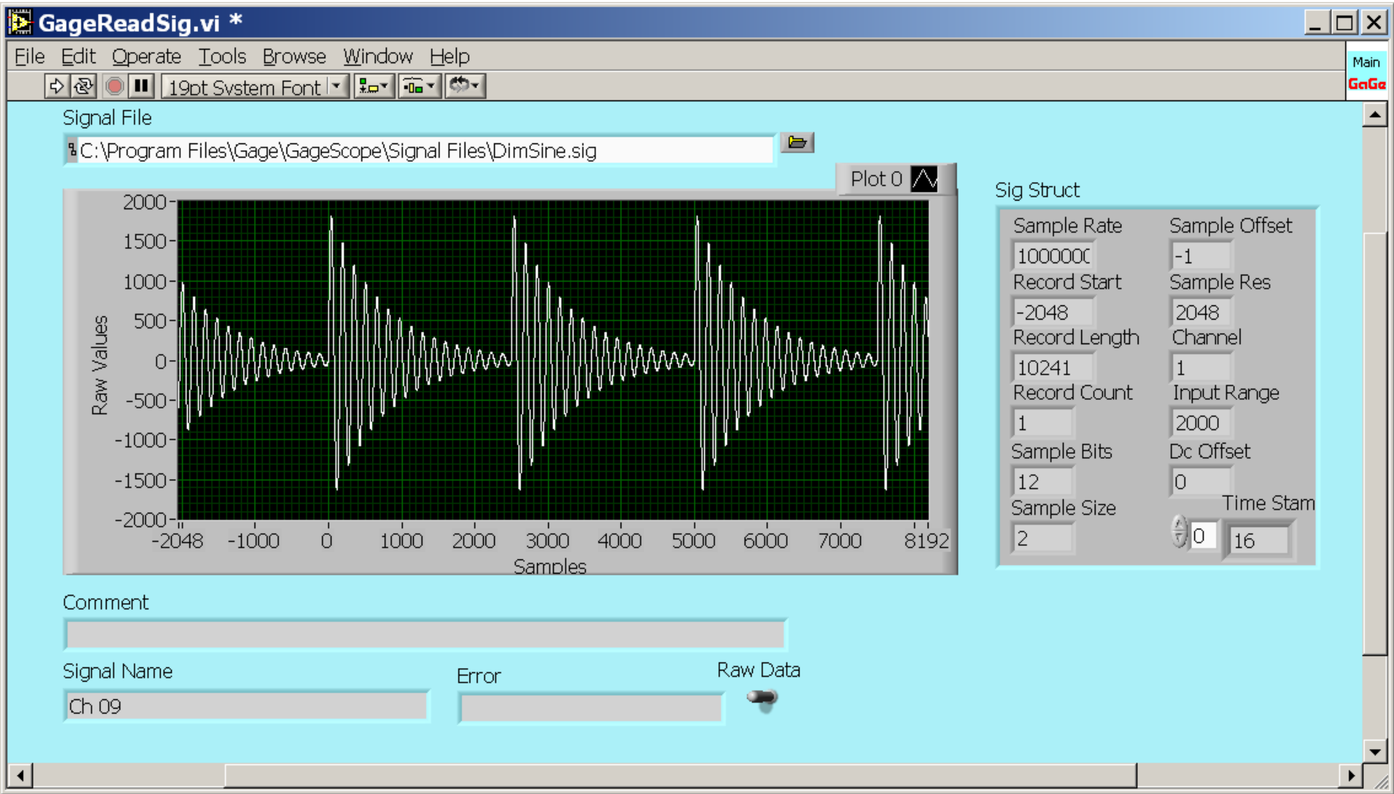
GageTwoIndependentSystemsAcquire.vi is the main VI for Multiple Independent CompuScope Systems. The VI is identical to GageAcquire.vi, except that there are two identical sets of controls and indicators for the two CompuScope systems. The two sets of controls and indicators for the two systems may be accessed by selecting the "System 1" or the "System 2" tabs. The VI operates two systems, but can be easily extendable to three or more Systems by the user.

GageTwoIndependentSystemsAcquire.vi acquires waveforms from the two CompuScope systems in a completely asynchronous fashion. Waveform acquisition on the two systems is controlled from two independent loops between which LabVIEW's built-in multi-tasking engine toggles in an indeterminate fashion. The VI, therefore, is designed to operate two CompuScope systems that are acquiring unrelated signals and triggers. In order to trigger both CompuScope systems simultaneously, the user must connect the same external trigger signal to both systems. Furthermore, the user must ensure that external trigger signals are inhibited until both cards are capturing. Otherwise, an external trigger may trigger one system and not the other.

Note: This VI will coerce all configuration values to be valid available settings for all CompuScope systems, as does GageCoerce.vi. Unlike GageCoerce.vi, however, GageTwoIndependentSystemsAcquire.vi does not display the actual coerced settings. Consequently, the user should bear in mind that the settings that were requested may not actually have been implemented.

All remaining controls and indicators for GageComplexTriggerAcquire.vi are identical to those for GageAcquire.vi

GageReadSig.vi and GageWriteSig.vi



GageScope, Gage's flagship stand-alone oscilloscope software, is ideal software from which to operate CompuScope hardware, since it provides an easy-to-use graphical environment and lots of display and analysis tools for recording and exploring high-speed waveforms. Users who do not require control of other devices, on-line application-specific analysis or the best repetitive acquisition performance may decide to always use GageScope to operate their CompuScope, rather than developing their own software application from an SDK. These users may, however, still want to analyze waveforms that they have acquired with their CompuScope under GageScope from LabVIEW

By default, GageScope stores waveform data in Gage's compact binary SIG file format. This format efficiently stores waveform data as raw binary values with a compact header that holds configuration settings (e.g. sampling rate, input range). The LabVIEW SDK provides a sample VI called GageReadSig.vi that reads a SIG file created by GageScope, displays the configuration settings and plots the waveform in graphical form. This VI may be easily modified by the user to analyze waveforms from stored SIG files in order to extract relevant measurements. The LabVIEW SDK also includes GageWriteSig.VI, which allows users to acquire and store SIG files from LabVIEW for later viewing and analysis from GageScope.

Advanced Main Vis

The LabVIEW SDK contains "Advanced" VIs in addition to the documented Main VIs. Usage of some of these files may require special CompuScope hardware options, on-board eXpert firmware processing images or special driver versions. These VIs are provided as-is with limited documentation in the form of on-line LabVIEW Help and/or an accompanying explanatory text file.

Special Topics

Converting from CompuScope ADC Code to Voltages

SDK sample programs are configured to save or display CompuScope waveform data that have been scaled so that the sample values are in Volts. The user may want to bypass the voltage conversion step, however, in order to achieve the best repetitive capture performance. Voltage conversion may then be done at leisure in post-processing or not at all.

Raw ADC code waveform data may be converted to voltage data for all CompuScope models using the following equation:

$$Voltage = \frac{Offset - ADC\_Code}{Resolution} \times \frac{(Full\ Scale\ Input\ Voltage)}{2} + DC\_Offset$$

The *Offset* and *Resolution* for the CompuScope system may be obtained programmatically using various SDK tools. The DC offset settings must be added. (Do not confuse "DC offset:" and the "Offset" used in the above equation.

For instance, for the 12-bit CompuScope 12501 and 12502, *Offset*=16 and *Resolution* = -32768 Let us assume that the user has selected the ±1 Volt Input Range using the CS12501, for a Full Scale Input Range of 2 Volts. Let us further assume that the user has applied a 200 mV offset. For this example, therefore, the voltage conversion equation becomes:

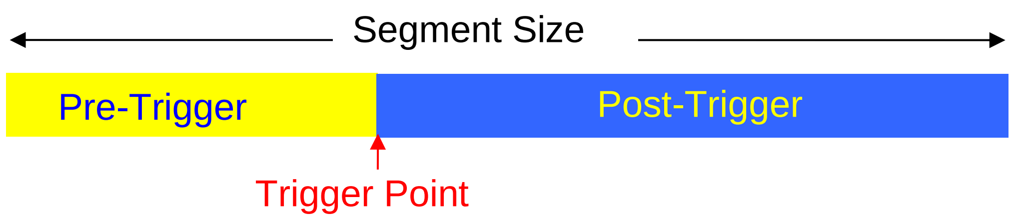
$$Voltage = \frac{16 - ADC\_Code}{-32768} \times \frac{2\ \text{Volts}}{2} + 0.2\ \text{Volts}$$

The Offset and Resolution for the CompuScope system may be obtained using the CsGet() API method using the CS_ACQUISITION Index. The resolution and offset values are returned in the CSACQUISITIONCONFIG structure as the values of i32SampleRes and i32SampleOffset, respectively. If the user has applied a DC Offset voltage to the signal, then this voltage may be obtained by calling CsGet() with the CS_CHANNEL index. The DC offset voltage is returned as the value of i32DcOffset within the CSCHANNELCONFIG structure.

Depth and Segment Size

The end of any CompuScope waveform acquisition is always initiated by the trigger event. After the trigger event occurs, the CompuScope acquires the requested number of post-trigger data points, called the post-trigger *Depth,* is acquired and then the acquisition terminates.

The total amount of CompuScope acquisition memory dedicated to the waveforms called the *Segment Size*. The difference between the Segment size and the Depth is equal to the amount of memory available for the acquisition of pre-trigger data. The image below illustrates the relationship between the Segment Size and the pre- and post-trigger data.



As an example, consider a user who wants to acquire 4 kiloSamples of post-trigger depth and 2 kiloSamples of pre-triggers data. In this case, the user must set the Segment Size to 6k so that 6k-4k = 2k are available for pre-trigger data accumulation.

Trigger HoldOff

Trigger HoldOff is a feature that is useful for ensuring the accumulation of a specified amount of pre-trigger data. The Trigger HoldOff setting specifies the amount of time, in Samples during which the CompuScope hardware will ignore trigger events after acquisition has begun and pre-trigger data are being acquired.

Without a zero Trigger HoldOff value, there is no guarantee that a given number of pre-trigger samples will be acquired. This is because a trigger event may occur immediately after acquisition, leading to a very small number of pre-trigger points. By ignoring trigger events for a time equal to the specified Trigger HoldOff value, the accumulation of a number of pre-trigger points equal to the Trigger HoldOff setting is guaranteed.

The downside of ensuring pre-trigger data with a non-zero Trigger HoldOff value is that triggers are ignored, so that important trigger events may be missed. For instance, in lightning monitoring applications, researchers usually want to acquire pre-trigger data. These data give information about signal behavior immediately preceding the lightning strike, which triggers the CompuScope hardware. Lightning strikes may occur very close together in time, however, and missing a lightning pulse is much worse than missing pre-trigger data. Consequently, lightning testers should not use Trigger HoldOff but should simply accept acquisitions of only the amount of pre-trigger data that naturally occur between triggers. Generally speaking, therefore, the user must decide whether to use Trigger HoldOff, based on the application.
Most modern CompuScopes support rectangular memory architecture (see “Memory organization on CompuScope Cards” in CompuScope Hardware Manual). For these models, the Trigger HoldOff must be set to be equal to the amount of acquisition memory that is available for pre-trigger data, which is (Segment Size – Depth). Any other value for the Trigger HoldOff will cause an error.

Trigger Delay

All CompuScopes support the Trigger Delay feature. This feature is useful for situations where the signal portion of interest occurs long after the trigger event. Trigger Delay allows suppression of storage of samples that occur during the Trigger Delay so that memory is not needlessly consumed.

Normally, with a Trigger Delay of zero, the trigger event activates count-down of the post-trigger depth counter, which was preloaded with the post-trigger depth. The counter is decremented by one count for each sample acquired after the trigger event. When the counter value reaches 0, the acquisition is terminated. In this way, the CompuScope acquires a number of samples equal to the depth after the trigger event.

A non-zero Trigger Delay value is used to delay the beginning of the countdown of the post-trigger depth counter. The Trigger Delay value sets the number of samples that the CompuScope hardware will wait after the trigger event occurs before beginning the count-down of the depth counters. When the Trigger Delay value is non-zero, pre-trigger data may not be acquired. Consequently, when the Trigger Delay value is non-zero, the user must set the Segment Size equal to the Depth.

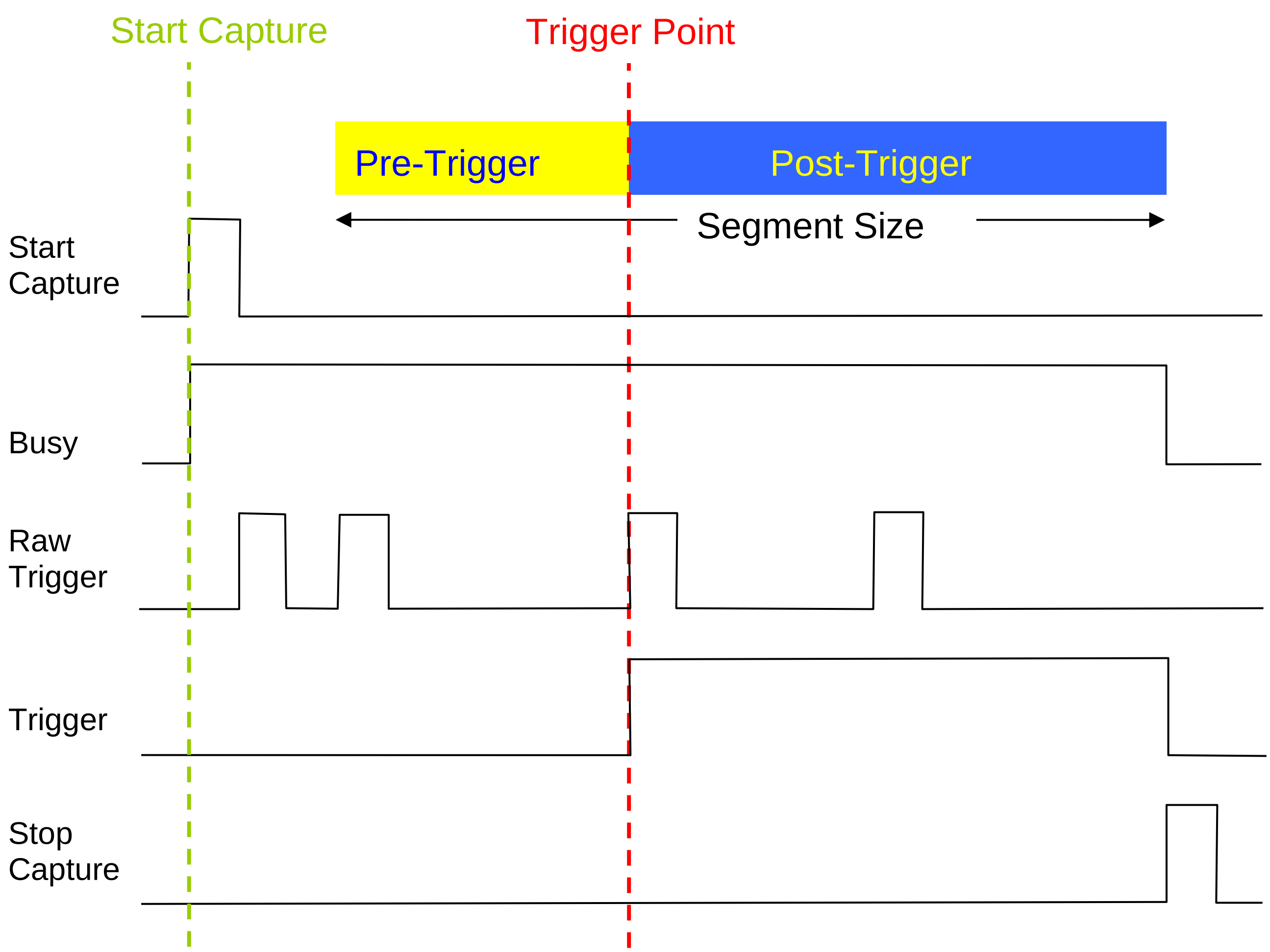
As an example, consider a signal where the feature of interest is 20,000 samples long but begins 100,000 samples after the trigger event. In this case, the SegmentSize and Depth should be set to 20,000. The Trigger Delay value should be set to 100,000. Without the Trigger Delay feature, the user would be forced to set the Depth to 120,000 Samples and waste 100,000 Samples of memory, even though only the last 20,000 were downloaded. Using the Trigger Delay feature, however, only the data of interest are retained in CompuScope memory.

CompuScope Acquisition Timing Diagram

The timing diagram below is provided as an aid for understanding the timing of events during the acquisition of a segment or record. The pseudo-signals are indicated as HIGH when the labeled function is active. The diagrams pertain to CompuScope acquisitions using rectangular memory architecture (see “CompuScope Memory Organization” in CompuScope Hardware Manual). In this case, the Trigger HoldOff must be set equal to (Segment Size – Depth) so that triggers are ignored until the memory allocated for pre-trigger data is filled. The Trigger Delay setting is assumed to be 0.

As discussed, the user sets the Segment Size and post-trigger Depth, which leaves (Segment Size – Depth) for the acquisition of pre-trigger data. The Segment is illustrated above with pre- and post-trigger data respectively shown in yellow and blue.

The Start Capture signal starts the CompuScope acquiring pre-trigger data and awaiting a trigger event. When this occurs, the card begins acquiring post-trigger data and terminates when Depth post-trigger points have been acquired. The Busy signal above shows the time during which the board is acquiring data.



The Raw Trigger signal shows that 4 raw trigger pulses occurred during the acquisition. The first two raw triggers were ignored because, when each occurred, the CompuScope had not yet acquired the requested number of pre-trigger points. The fourth raw trigger is ignored because it occurred during the acquisition of post-trigger data that was initiated by the third raw trigger, which is the only raw trigger that registered as a true trigger event. That this third raw trigger is the true trigger event is illustrated above by the Trigger signal above. The Stop Capture signal occurs when the post-trigger depth counter decrements to zero and terminates the acquisition.

Notice that not all pre-trigger data were retained in memory. The pre-trigger data fills the (Segment Size – Depth) available memory like a FIFO (First In First Out) so that only the most recent (Segment Size – Depth) points are retained. Consequently, the data that were acquired between the vertical green line and the yellow box were overwritten and lost in the above acquisition.

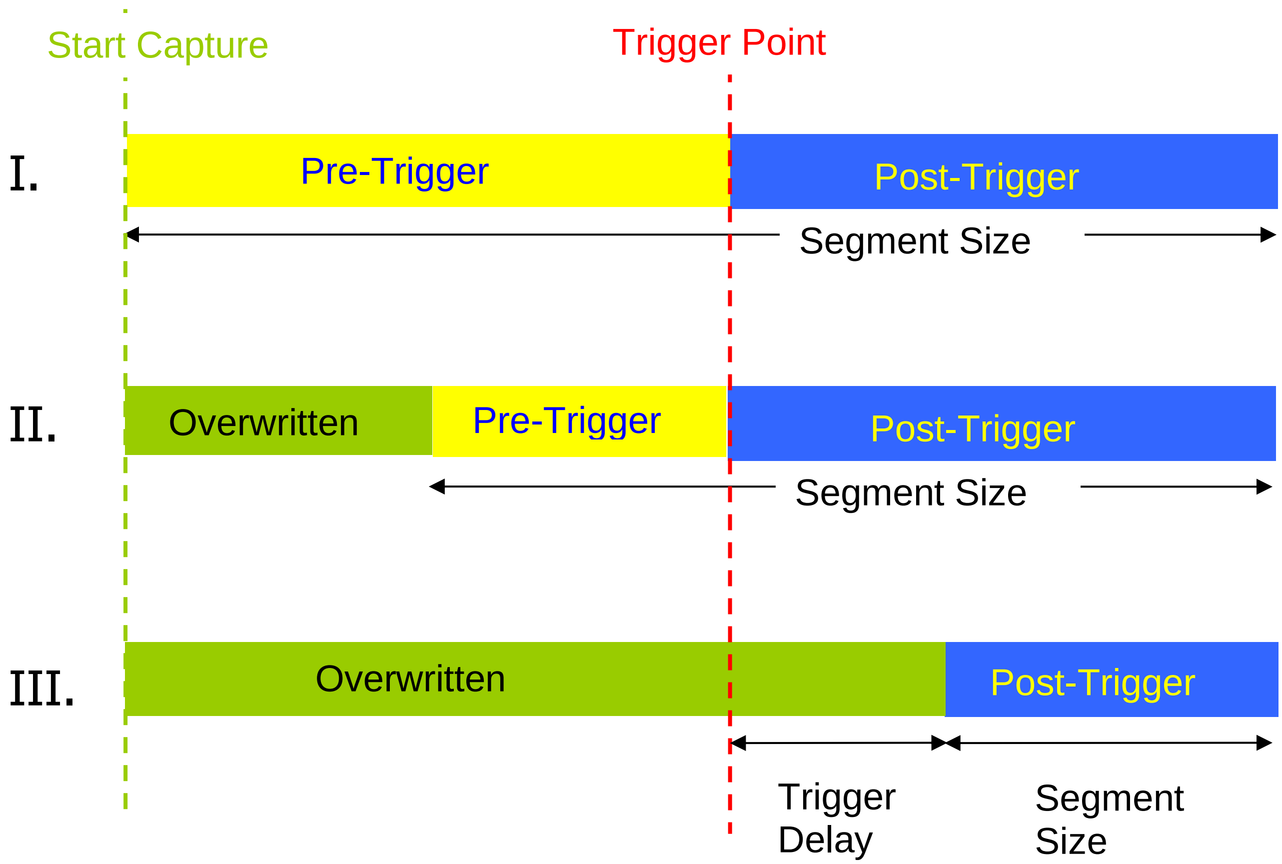
Representative Acquisition Sequences

The diagrams below illustrate three key types of CompuScope waveform acquisition. The diagrams pertain to CompuScope acquisitions using rectangular memory architecture (see “CompuScope Memory Organization” in CompuScope Hardware Manual). In this case, the Trigger HoldOff must be set equal to (Segment Size – Depth) so that triggers are ignored until the memory allocated for pre-trigger data is filled.

Acquisition I shows a situation where all pre-trigger data since the start of acquisition have been acquired. This may be because the time interval between Start Capture and the trigger Event was exactly equal to the amount of memory available for pre-trigger data (which is (Segment Size – Depth)) More likely, the triggers are very rapid and these triggers are all ignore until the pre-trigger memory is filled, after which it triggers on the next trigger.

Acquisition II shows a situation where not all pre-trigger data since the start of acquisition have been acquired. In this case, the amount of pre-trigger data is relatively small compared with the trigger occurrence frequency. From the diagram, only about the most recent half of the pre-trigger data are retained. Earlier pre-trigger data, shown in green, are overwritten and lost.

Acquisition III shows an acquisition where the Trigger Delay feature is in use. As required, with Trigger Delay active, no memory for the acquisition of pre-trigger may be allocated and Segment Size must be made equal to the Depth. In this case, all data before the trigger event are overwritten and lost. The same is true for the first Trigger Delay samples after the trigger event. Only data that occur at least Trigger Delay samples after the trigger event are retained.


Start Capture
Trigger Point
I.
Pre-Trigger
Post-Trigger
Segment Size
II.
Overwritten
Pre-Trigger
Post-Trigger
Segment Size
III.
Overwritten
Post-Trigger
Trigger Delay
Segment Size

Technical Support

We offer technical support for all our Software Development Kits.

In order to serve you better, we have created a web-based technical support system that is available to you 24 hours a day.

By utilizing the internet to the fullest, we are able to provide you better than ever technical support without increasing our costs, thereby allowing us to provide you the ***best possible product at the lowest possible price.***

To obtain technical support, simply visit:

www.gage-applied.com/support/support_form.php

Please complete this form and submit it. Our form processing system will intelligently route your request to the Technical Support Specialist (TSS) most familiar with the intricacies of your product. This TSS will be in contact with you within 24 hours of form submittal.

In the odd case that you have problems submitting the form on our web site, please e-mail us at

tech-support@gage-applied.com

As opposed to automatic routing of technical support requests originating from the GaGe web site, support requests received via e-mail or telephone calls are routed manually by our staff. Providing you with high quality support may take an average of 2 to 3 days if you do not use the web-based technical support system.

Please note that Technical Support Requests received

via e-mail or by telephone will take an average of 2 to 3 days to process.

It is faster to use the web site!

When calling for support we ask that you have the following information available:

1. Version and type of your CompuScope SDK and drivers.
   (The version numbers are indicated in the About CD screen of the CompuScope CD. Version numbers can also be obtained by looking in the appropriate README.TXT files)

2. Type, version and memory depth of your CompuScope card.

3. Type and version of your operating system.

4. Type and speed of your computer and bus.

5. If possible, the file saved from the Information tab of the CompuScope Manager utility.

6. Any extra hardware peripherals (i.e. CD-ROM, joystick, network card, etc.)

7. Were you able to reproduce the problem with standalone GaGe Software (e.g. GageScope, CsTest)?