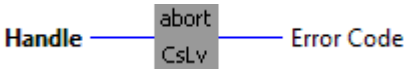
CsLv_AbortCapture.vi
This VI aborts the current acquisition by the CompuScope system identified by Handle, which is obtained by using the CsLv_GetSystem.vi. If CsLv_AbortCapture.vi is called between the acquisition of successive segments in Multiple Record Mode, data from segments acquired before CsLv_AbortCapture.vi was called may be accessed.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

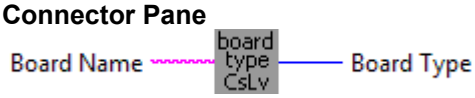
U32 **Handle**

I32 **Error Code**

CsLv_BoardNameToType.vi
This VI returns the Board Type associated with the Board Name that is passed as its input parameter. The Board Name is a string variable that has the form CSx, where x is the CompuScope model number (e.g. “CS14200” for the CompuScope 14200). The Board Type is a numeric constant used internally by the CompuScope drivers. It can be used as an optional parameter in a call to CsLv_GetSystem.vi

If the return value is 0, either the call was unsuccessful or the Board Name does not exist.

Connector Pane



Controls and Indicators

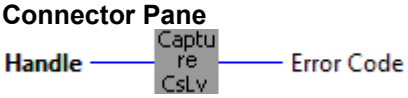
abc **Board Name**

U32 **Board Type**

CsLv_Capture.vi
Calling this VI starts an acquisition by the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. The CompuScope system will start digitizing pre-trigger data into on-board memory until a trigger event occurs. Once the trigger event occurs, the specified number of post-trigger points will be captured and then the acquisition will terminate.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

U32 **Handle**

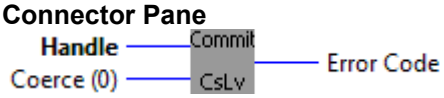
I32 **Error Code**

CsLv_Commit.vi
This VI configures a CompuScope system with the configuration settings that are set in the driver. The configuration settings may be set in the driver using the CsLv_ConfigureAcquisition, CsLv_ConfigureChannel, CsLv_ConfigureTrigger, or CsLv_ConfigureTimeStamp VIs. CsLv_Commit.vi must be called in order to actually pass any configuration settings to the CompuScope hardware. The CompuScope system is identified by Handle, which is obtained with CsLv_GetSystem.vi

If set to 1, the Coerce parameter, will coerce all configuration values to valid available settings for the current CompuScope system. The coercion procedure varies for the different types of configuration settings. For the internal sampling rate, for instance, the coercion procedure chooses the closest available sampling rate. The default value for the Coerce parameter is 0.

If the CompuScope hardware is configured correctly, a 1 is returned by the VI. If the return value is 2, the operation was successful but a value had to be coerced. Otherwise, an error code is returned. A descriptive error string can be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Coerce (0)

Error Code

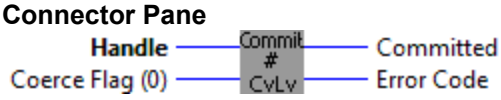
CsLv_CommitOnChange.vi
This VI configures the CompuScope system with the configuration settings that are set in the driver. Unlike CsLv_Commit.vi, this VI will only pass the configuration setting to the CompuScope hardware if they have changed since the last time CsLv_Commit.vi or CsLv_CommitOnChange.vi was called. If the configuration settings have not changed, the VI just returns. The configuration settings may be set in the driver using the CsLv_ConfigureAcquisition, CsLv_ConfigureChannel, CsLv_ConfigureTrigger, or CsLv_ConfigureTimeStamp VIs. The CompuScope system is identified by Handle, which is obtained with CsLv_GetSystem.vi.

If set to 1, the Coerce parameter will coerce all configuration values to valid available settings for the current CompuScope system. The coercion procedure varies for the different types of configuration settings. For the internal sampling rate, for instance, the coercion procedure chooses the closest available sampling rate. The default value for the Coerce parameter is 0.

If the values have changed and were committed to the driver, the Committed indicator will return a 1. Otherwise it will return a 0.

If the CompuScope hardware is configured correctly, a 1 is returned by the VI. If the return value is 2, the operation was successful but one or more values were coerced. Otherwise, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Coerce Flag (0)

Error Code

Committed

CsLv_ConfigureAcquisition.vi
This VI sends the requested acquisition configuration settings to the driver for the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. The requested settings are not actually sent to the CompuScope hardware until CsLv_Commit.vi is called.

The Acquisition input settings are Sample Rate, External Clock, Mode, Depth, Segment Size, Trigger Timeout, Trigger Delay, Trigger Holdoff, Segment Count and Extended Options. Valid values for the system identified by Handle must be supplied for all settings. If no connection is made to an input, the default value for that input will be used.

If the VI fails, an appropriate error code is returned in the Error Code indicator. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

Input Value Descriptions:

Sample Rate (default = 100000000)
This double variable value is the rate, in samples per second, at which the CompuScope system will sample input signals. If the desired sampling rate is not valid for the CompuScope system identified by the Handle, an appropriate error code is returned.

External Clock (default = 0)
This integer value should be set to 1 if external clocking is to be used. Otherwise, it should be set to 0. If an external clock is being used, the rate entered in the Sample Rate field must be set to the correct external clock signal frequency.

Mode (default = 2)
This integer value sets the operating mode for the CompuScope system. Set to 1 for Single Channel Mode, 2 for Dual Channel Mode, 4 for Quad Channel Mode and 8 for Octal Channel Mode. For digital input CompuScope cards, like the CS3200, set to 1 for 8-bit mode, set to 2 for 16-bit mode, or set to 4 for 32-bit mode. If the requested mode is invalid for the CompuScope system, an appropriate error code is returned.

Please note that the higher order bits within the Mode value may contain information that indicates alternate functionality. In order to recover the actual Mode (1, 2, 4 or 8 for Single, Dual, Quad or Octal), these alternate functionality bits must be removed by bitwise ANDing the Mode with 0x000f, which extracts only the lowest 4 bits of the Mode. Alternate functionality bits may indicate standard CompuScope functionality or optional CompuScope firmware.

Other CompuScope functionalities may also be selected by ORing the Mode with the value(s) listed below:
0x80 (128) Disable power saving mode
0x400 (1024) Use 10 MHz Reference clock
0x800 (2048) Invert sampling clock for digital input CompuScope cards

Depth (default = 8192)

This double variable value is the number of post-trigger samples that the system will acquire.

Segment Size (default = 8192)
This double variable 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 (1000000)
This double variable value is the amount of time, in microseconds, that the CompuScope system will wait for a trigger to occur before forcing a trigger. An input value of -1 indicates no trigger timeout, so that the CompuScope system will wait forever for a trigger event.

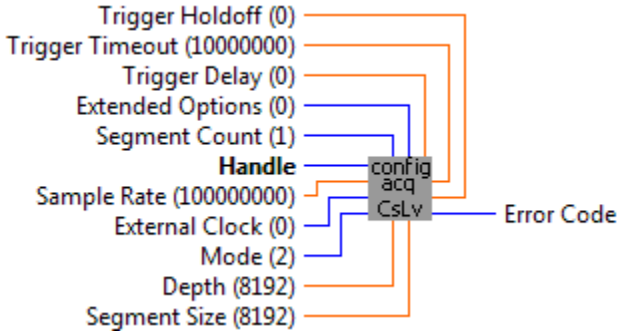
Trigger Delay (default = 0)
This double variable value is the duration, in Samples, between the time that the trigger event actually occurs and the time that the trigger event will be logged in the data record. Setting a non-zero trigger delay is useful for signals where the region of interest in the signal occurs long after the trigger event. Trigger Delay is not supported on all CompuScope models.

Trigger Holdoff (default = 0)
This double variable value is the number of samples during which the CompuScope hardware will ignore trigger events after starting an acquisition. Trigger Holdoff is useful in order to ensure the accumulation of a specified number of pre-trigger points. This is achieved by ignoring trigger events until the specified number of pre-trigger points has been acquired.

Segment Count (default = 1)
This integer value sets how many segments (or records) will be acquired in a Multiple Record acquisition. If the value of Segment Count is greater than the maximum number of Multiple Records of which the CompuScope system is capable, then the maximum amount will be acquired. For Single Record acquisitions, Segment Count must be set to 1.

Extended Options (default = 0)
This integer value sets available extended options (on-board processing firmware such as FIR or Signal Averaging). Set to 1 for FIR, 2 for Signal Averaging and 16 for FFT. Available extended options may be obtained by using CsLv_GetExtendedOptions.vi. Extended options are not supported on all CompuScope models.

Connector Pane



Controls and Indicators

U32 **Handle**

DBL **Sample Rate (100000000)**

U32 **External Clock (0)**

Mode (2)

Segment Count (1)

Depth (8192)

Trigger Timeout (10000000)

Trigger Holdoff (0)

Trigger Delay (0)

Extended Options (0)

Segment Size (8192)

Error Code

CsLv_ConfigureChannel.vi
This VI sends the requested Channel configuration settings to the driver for the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. The requested settings are not actually sent to the CompuScope hardware until CsLv_Commit.vi is called.

The Channel configuration settings are Channel, Coupling, Differential Input, Direct-to-ADC, Input Range, Impedance, and DC Offset. Valid values must be supplied for all settings. If no connection is made to an input, the default value for that input will be used. If an invalid setting is sent to the VI, an appropriate error code is returned.

If the VI fails, an appropriate error code is returned in the Error Code indicator. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel (default = 0)
This integer value selects the channel number for which the Channel configuration settings will be applied. Channel numbers in a LabVIEW CompuScope system begin at 0. (Channel numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid channel number is specified, an appropriate error code is returned.

Coupling (default = 1)
This integer value sets the current input coupling. Available values are:
1 = DC input coupling
2 = AC input coupling
An invalid value will result in an error code being returned from the VI.

Differential Input (default = 0)
This value sets the channel configuration to use differential input coupling. Available values are:
0 = use single-ended input coupling
1 = use differential input coupling
If the CompuScope system identified by Handle does not support differential input coupling and the Differential Input setting is set to 1, an appropriate error code is returned.

Direct-to-ADC (default = 0)

This integer value sets the channel configuration to the Direct-to-ADC input range. Available values are:

0 = do not use Direct-to-ADC input range
1 = use Direct-to-ADC input range

If the CompuScope system identified by Handle does not support a Direct-to-ADC input range and the Direct-to-ADC setting is set to 1, an appropriate error code is returned.

Input Range (default = 2000 millivolts)
This integer value sets the current full scale input range (in millivolts) for the current channel. For instance, for the +/-1 Volt input range, 2000 must be entered. See your CompuScope hardware manual for the input ranges available for your CompuScope model. If invalid values are sent to the VI, an appropriate error code is returned.

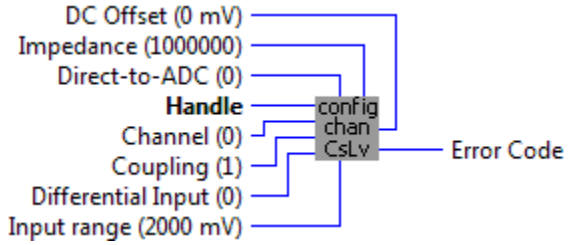
Impedance (default = 1000000)
This integer value sets the terminating input impedance for the channel. Available values are:

1000000 = 1 MOhm
50 = 50 Ohms

If the input value is not supported for the specified CompuScope system, an appropriate error code is returned.

DC Offset (default = 0 millivolts)
This integer value sets the current DC offset (in millivolts) for the specified channel. The value raises or lowers the mid point of the CompuScope input range. The value is in millivolts (plus or minus) and must be within the CompuScope channel's input range. For instance, if a +100 mV DC offset is specified in the +/-1 V range, then the effective input range becomes -0.9 Volts to 1.1 Volts. Not all CompuScope models support DC offsets. If an invalid value is entered, an appropriate error code is returned.

Connector Pane



Controls and Indicators

Handle

Channel (0)

Coupling (1)

Differential Input (0)

Input range (2000 mV)

Impedance (1000000)

DC Offset (0 mV)

Direct-to-ADC (0)

Error Code

CsLv_ConfigureChannelEx.vi
This VI sends the requested Channel configuration settings to the driver for the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. The requested settings are not actually sent to the CompuScope hardware until CsLv_Commit.vi is called.

The Channel configuration settings are Channel, Coupling, Differential Input, Direct-to-ADC, Input Range, Impedance, and DC Offset. Valid values must be supplied for all settings. If no connection is made to an input, the default value for that input will be used. If an invalid setting is sent to the VI, an appropriate error code is returned.

If the VI fails, an appropriate error code is returned in the Error Code indicator. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel (default = 0)
This integer value selects the channel number for which the Channel configuration settings will be applied. Channel numbers in a LabVIEW CompuScope system begin at 0. (Channel numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid channel number is specified, an appropriate error code is returned.

Coupling (default = 1)
This integer value sets the current input coupling. Available values are:
1 = DC input coupling
2 = AC input coupling
An invalid value will result in an error code being returned from the VI.

Differential Input (default = 0)
This value sets the channel configuration to use differential input coupling. Available values are:
0 = use single-ended input coupling
1 = use differential input coupling
If the CompuScope system identified by Handle does not support differential input coupling and the Differential Input setting is set to 1, an appropriate error code is returned.

Direct-to-ADC (default = 0)
This integer value sets the channel configuration to the Direct-to-ADC input range. Available values are:
0 = do not use Direct-to-ADC input range
1 = use Direct-to-ADC input range
If the CompuScope system identified by Handle does not support a Direct-to-ADC input range and the Direct-to-ADC setting is set to 1, an appropriate error code is returned.

Input Range (default = 2000 millivolts)
This integer value sets the current full scale input range (in millivolts) for the current channel. For instance, for the +/-1 Volt input range, 2000 must be entered. See your CompuScope hardware manual for the input ranges available for your CompuScope model. If invalid values are sent to the VI, an appropriate error code is returned.

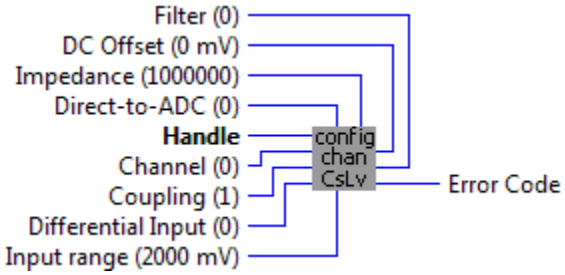
Impedance (default = 1000000)
This integer value sets the terminating input impedance for the channel. Available values are:
1000000 = 1 MOhm
50 = 50 Ohms

If the input value is not supported for the specified CompuScope system, an appropriate error code is returned.

DC Offset (default = 0 millivolts)
This integer value sets the current DC offset (in millivolts) for the specified channel. The value raises or lowers the mid point of the CompuScope input range. The value is in millivolts (plus or minus) and must be within the CompuScope channel's input range. For instance, if a +100 mV DC offset is specified in the +/-1 V range, then the effective input range becomes 0.9 Volts to -1.1 Volts. Not all CompuScope models support DC offsets. If an invalid value is entered, an appropriate error code is returned.

Filter (default = 0 (No filter))
Index of the filter to be used. The filter parameters may be extracted with the CsGetSystemCaps using the CAPS_FILTERS parameter.

Connector Pane



Controls and Indicators

Handle

Channel (0)

Coupling (1)

Differential Input (0)

Input range (2000 mV)

Impedance (1000000)

DC Offset (0 mV)

Direct-to-ADC (0)

Filter (0)

Error Code

CsLv_ConfigureTimeStamp.vi
This VI sends the Time Stamping configuration settings to the driver for the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi.

Time Stamping functionality is provided by an on-board counter whose value is latched upon each trigger event - thus time stamping the record's trigger event. The counter source may be derived from the CompuScope sampling oscillator or from a separate on-board fixed frequency oscillator.

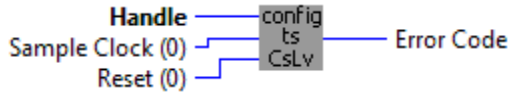
If the settings are correctly sent to the driver, a 1 is returned in the Error Code indicator. Otherwise, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Sample Clock (default = 0)
This integer value determines the source of the Time Stamping counter. If the value is set to 0, the counter source is derived from the sampling oscillator. If the value is set to 1, the on-board fixed frequency oscillator is used. If the sampling oscillator is selected, then the counter source frequency is one half of the sampling rate in Dual-channel Mode and one quarter of the sampling rate in Single-channel Mode. The actual time stamping counter clock frequency in use may be obtained by calling CsLv_TransferTimeStamp.vi.

Reset (default = 0)
This integer value determines whether the Time Stamping counter is reset after every acquisition. If the value is set to 1, the counter is reset. If the value is set to 0, the counter is not reset.

Connector Pane



Controls and Indicators

Handle

Sample Clock (0)

Reset (0)

Error Code

CsLv_ConfigureTrigger.vi
This VI sends the requested Trigger configuration settings to the driver for the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. The requested settings are not actually sent to the CompuScope hardware until CsLv_Commit.vi is called.

Many CompuScope cards are equipped with two trigger engines that may be independently configured. The outputs of the two trigger engines are bitwise ORed together so that either engine may cause a trigger event. For simple triggering, the second engine is disabled. The second engine may be used to implement more complex triggering schemes.

The Trigger configuration settings are Trigger Engine, Slope, Level, Source, External Coupling, and External Range. Valid values for the system identified by Handle must be supplied for all settings. If no connection is made to an input, the default value for that input will be used. If an error occurs, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Trigger Engine (default = 0)

This integer value specifies the number of the trigger engine in the CompuScope system for which the Trigger configuration settings will be applied. Trigger Engine numbers in a LabVIEW CompuScope system begin at 0. (Engine numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid engine number is specified, an appropriate error code is returned.

Slope (default = 1)
This integer value sets the trigger signal slope that will cause a trigger event to occur. Possible values are:

0 = configure the trigger signal slope to Falling
1 = configure the trigger signal slope to Rising

Level (default = 0)
This integer value sets the trigger level as a percentage of the input range of the trigger source. Possible values are between -100% and +100%. For example, in the +/-1 Volt range, 100% corresponds to a +1 Volt trigger level, 50% is +500 millivolts, and -100% is -1 Volt.

Source (default = 0)
This integer value sets the trigger source for the specified Trigger Engine. Available values are:

-2 = configure the external trigger input as the trigger source
-1 = disable the trigger engine
0 = configure Channel 1 as the trigger source
1 = configure Channel 2 as the trigger source
etc. for available channels

If the trigger source is a Channel, the coupling and input range of the trigger input are those of the trigger source. In these cases, therefore, the settings for the External Range and External Coupling fields are ignored. If the trigger source is set to Disabled, all other settings are ignored. Note that even if settings are ignored, they should still be set to valid values. (The values for the Trigger Source are incremented by 1 before they are sent to the driver, as required by the C API.)

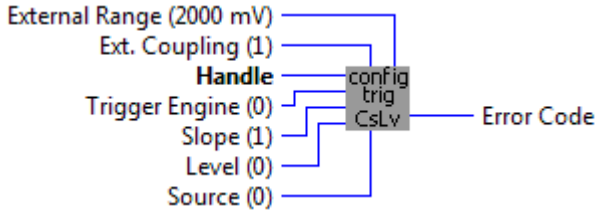
External Coupling (default = 1)
This integer value sets the coupling of the external trigger input. Possible values are:

1 = set DC external trigger coupling
2 = set AC external trigger coupling

External Range (default = 2000)
This integer value sets the input range of the external trigger input in millivolts. Possible values are:

2000 = set the +/-1 Volt external trigger input range
10000 = set the +/-5 Volt external trigger input range

Connector Pane



Controls and Indicators

U32 **Handle**

U32 **Trigger Engine (0)**

 **Slope (1)**

 **Level (0)**

 **Source (0)**

 **Ext. Coupling (1)**

 **External Range (2000 mV)**

 **Error Code**

CsLv_ConfigureTriggerEx.vi
This VI sends the requested Trigger configuration settings to the driver for the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. The requested settings are not actually sent to the CompuScope hardware until CsLv_Commit.vi is called.

Many CompuScope cards are equipped with two trigger engines that may be independently configured. The outputs of the two trigger engines are bitwise ORed together so that either engine may cause a trigger event. For simple triggering, the second engine is disabled. The second engine may be used to implement more complex triggering schemes.

The Trigger configuration settings are Trigger Engine, Slope, Level, Source, External Coupling, and External Range. Valid values for the system identified by Handle must be supplied for all settings. If no connection is made to an input, the default value for that input will be used. If an error occurs, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Trigger Engine (default = 0)
This integer value specifies the number of the trigger engine in the CompuScope system for which the Trigger configuration settings will be applied. Trigger Engine numbers in a LabVIEW CompuScope system begin at 0. (Engine numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid engine number is specified, an appropriate error code is returned.

Slope (default = 1)
This integer value sets the trigger signal slope that will cause a trigger event to occur. Possible values are:
0 = configure the trigger signal slope to Falling
1 = configure the trigger signal slope to Rising

Level (default = 0)
This integer value sets the trigger level as a percentage of the input range of the trigger source. Possible values are between -100% and +100%. For example, in the +/-1 Volt range, 100% corresponds to a +1 Volt trigger level, 50% is +500 millivolts, and -100% is -1 Volt.

Source (default = 0)
This integer value sets the trigger source for the specified Trigger Engine. Available values are:
-2 = configure the external trigger input as the trigger source
-1 = disable the trigger engine
0 = configure Channel 1 as the trigger source
1 = configure Channel 2 as the trigger source

etc. for available channels

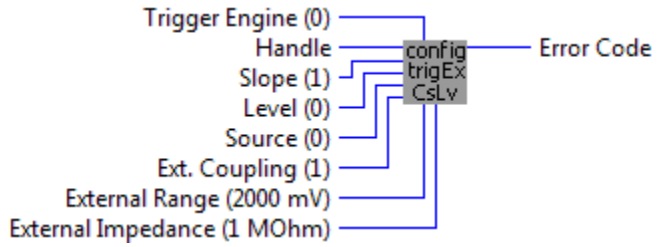
If the trigger source is a Channel, the coupling and input range of the trigger input are those of the trigger source. In these cases, therefore, the settings for the External Range and External Coupling fields are ignored. If the trigger source is set to Disabled, all other settings are ignored. Note that even if settings are ignored, they should still be set to valid values. (The values for the Trigger Source are incremented by 1 before they are sent to the driver, as required by the C API.)

External Coupling (default = 1)
This integer value sets the coupling of the external trigger input. Possible values are:
1 = set DC external trigger coupling
2 = set AC external trigger coupling

External Range (default = 2000)
This integer value sets the input range of the external trigger input in milliVolts. Possible values are:
2000 = set the +/-1 Volt external trigger input range
10000 = set the +/-5 Volt external trigger input range

External Impedance (default = 1000000)
This integer value sets the impedance of the external trigger input in Ohms. Possible values are:
50 = set the 50 Ohm external trigger input impedance
1000000 = set the HiZ external trigger input impedance

Connector Pane



Controls and Indicators

Handle

Trigger Engine (0)

Slope (1)

Level (0)

Source (0)

Ext. Coupling (1)

External Range (2000 mV)

External Impedance (1 MOhm)

Error Code

CsLv_ConvertFromSigHeader.vi

CsLv_ConvertFromSigHeader converts a GageScope signal (sig) file header passed as a parameter into a Sig Struct cluster (described below) containing the relevant information in the header. The comment and name fields of the header are returned as strings in the comment and name parameters. The sigheader parameter passed to the function is an array of 512 int8's read from the beginning of a GageScope SIG file.

If the value returned in the Error Code indicator is positive, the function succeeded. If the value is negative it represents an error code. A descriptive error string may be obtained by calling the CsLv_GetErrorString.vi.

The Input parameters are:

Sig Struct
An empty sig struct cluster is passed to the vi and filled with values converted from the SIG file header.

The Sig Struct cluster contains the following fields:

Sample Rate
A double representing the sample rate of the acquistion in Hz.

Record Start
A double representing the start address (in samples) of each segment in the file.

Record Length
A double representing the length (in samples) of each segment in the file.

Record Count
A uInt32 which represents the number of segments saved in the file.

Sample Bits
A uInt32 which represents the actual vertical resolution (in bits) of the data in the file.

Sample Size
A uInt32 representing the actual size, in bytes, of each data value in the file.

Sample Offset
An int32 which is the sample offset of the data in the file.

Sample Res
An int32 which is the actual sample resolution of the data.

Channel
A uInt32 which is the Channel number that was saved to the file. 1 represents the first channel of the system.

Input Range
A uInt32 which is the Channel full scale input range, in millivolts peak to peak.

Dc Offset
An int32 which is the Channel DC offset in millivolts.

Time Stamp
An array of 4 uInt16 values, which represent (in order) the hour, minute, second and 100th's of a second of the trigger event.

Sig Header

An array of 512 int8 values which is obtained by reading the first 512 bytes of a GageScope SIG file.

Name
an empty string that will end up containg the name field of the header of GageScope SIG file. Should be able to hold up to 14 characters

Comment
an empty string that will end up containg the comment field of the header of GageScope SIG file. Should be able to hold up to 256 characters

Connector Pane

Controls and Indicators

- **Sig Header**
  - **Numeric**
- **Error Code**
- **Sig Struct**
  - **Sample Rate**
  - **Record Start**
  - **Record Length**
  - **Record Count**
  - **Sample Bits**
  - **Sample Size**
  - **Sample Offset**
  - **Sample Res**
  - **Channel**
  - **Input Range**
  - **Dc Offset**
  - **Time Stamp**
    - **Numeric**
- **Comment**

Name

CsLv_ConvertToSigHeader.vi
CsLv_ConvertToSigHeader converts a Sig Struct cluster (described below) passed as a parameter into a GageScope SIG file header, which is put at the beginning of a GageScope SIG file. The Sig Struct cluster can be filled in from values in the AcquisitionConfig and ChannelConfig clusters obtained by calling the CsLv_QueryAcquisitonParameters and CsLv_QueryChannelParameters vi's.

If the value returned in the Error Code indicator is positive, the function succeeded. If the value is negative it represents an error code. A descriptive error string may be obtained by calling the CsLv_GetErrorString.vi.

If the vi is successful, a 512 byte array is returned which can be used as the header for a GageScope SIG file.

The Input parameters are:

Sig Struct
A sig struct cluster with valid values that is passed to the vi and used to create the SIG file header.

The Sig Struct cluster contains the following fields:

Sample Rate
A double representing the sample rate of the acquistion in Hz.

Record Start
A double representing the start address (in samples) of each segment in the file.

Record Length
A double representing the length (in samples) of each segment in the file.

Record Count
A uInt32 which represents the number of segments saved in the file.

Sample Bits
A uInt32 which represents the actual vertical resolution (in bits) of the data in the file.

Sample Size
A uInt32 representing the actual size, in bytes, of each data value in the file.

Sample Offset
An int32 which is the sample offset of the data in the file.

Sample Res
An int32 which is the actual sample resolution of the data.

Channel
A uInt32 which is the Channel number that was saved to the file. 1 represents the first channel of the system.

Input Range
A uInt32 which is the Channel full scale input range, in millivolts peak to peak.

Dc Offset
An int32 which is the Channel DC offset in millivolts.

Time Stamp
An array of 4 uInt16 values, which represent (in order) the hour, minute, second and 100th's of a second of the trigger event. This parameter can be filled with values from the LabVIEW time and date functions.

Name
A string that will be used to fill the name field of the SIG header. A maximum of 14 characters is allowed. The default is an empty string.

Comment
A string that will be used to fill the comment field of the SIG header. A maximum of 256 characters is allowed. The default is an empty string.

Connector Pane

Controls and Indicators

Sig Struct

- **Sample Rate**
- **Record Start**
- **Record Length**
- **Record Count**
- **Sample Bits**
- **Sample Size**
- **Sample Offset**
- **Sample Res**
- **Channel**
- **Input Range**
- **Dc Offset**
- **Time Stamp**
  - **Numeric**

Comment

Name

 **Error Code**

 **Sig Header**

 **Numeric**

CsLv_Extract16BitEx.vi
This VI converts an interleaved 16-bit buffer obtained from CsLv_Transfer16Ex.vi and converts it into a 3 dimensional buffer (Channels * Segment Count * Xfer Length).

This VI should only be used for converting data int16 data. The caller is responsible for passing in an 3-dimensional array of int16's (Channels * Segment Count * XFer Length), which is used to hold the extracted data.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel s(default=1)
This integer value specifies the number of channels that are interleaved in the Input Buffer data. The number of channels can be obtained from the CsLv_Transfer16Ex.vit.

Segment Count (default=1)
The number of segments contained in the interleaved Input Buffer.

Xfer Length
This double variable value specifies the amount of data, in samples, in the Input Buffer

Pattern
A constant that indicates the order of the data for different channels. For example if the Data Format is 0x11221122 , the 2 channel data will be formated as 2 samples of Channel 1, 2 samples of Channel 2, 2 sample of Channel 1, etc. Can be obtained by calling CsLv_Transfer16Ex.vi

Input Buffer
An array of int16 values containing the raw ADC values from CsLv_Transfer16Ex.vi.

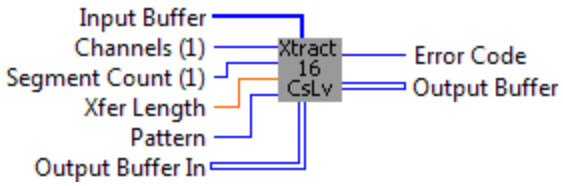
Ouput Buffer In
A 3-dimensional array of int16's. The user is responsible for ensuring the array is the proper dimension and size (Channels * Segment Count * Xfer Length). The array is filled with the appropriate values and returned in the Output Buffer.

Return Values

Output Buffer
The Output Buffer In filled with the appropriate raw ADC int16 values.

Error Code
An integer that represents the return code of the call. A positive value indicates success and a negative value indicates failure. A descriptive error string can be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Input Buffer

Numeric

Segment Count (1)

Xfer Length

Pattern

Channels (1)

Output Buffer In

Numeric

Output Buffer

Numeric

Error Code

CsLv_Extract32BitEx.vi
This VI converts an interleaved 32-bit buffer obtained from CsLv_Transfer32Ex.vi and converts it into a 3 dimensional buffer (Channels * Segment Count * Xfer Length).

This VI should only be used for converting data int32 data. The caller is responsible for passing in an 3-dimensional array of int32's (Channels * Segment Count * XFer Length), which is used to hold the extracted data.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel s(default=1)
This integer value specifies the number of channels that are interleaved in the Input Buffer data. The number of channels can be obtained from the CsLv_Transfer32Ex.vit.

Segment Count (default=1)
The number of segments contained in the interleaved Input Buffer.

Xfer Length
This double variable value specifies the amount of data, in samples, in the Input Buffer

Pattern

A constant that indicates the order of the data for different channels. For example if the Data Format is 0x11221122 , the 2 channel data will be formated as 2 samples of Channel 1, 2 samples of Channel 2, 2 sample of Channel 1, etc. Can be obtained by calling CsLv_Transfer32Ex.vi

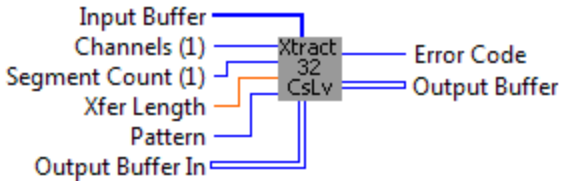
Input Buffer
An array of int32 values containing the raw ADC values from CsLv_Transfer32Ex.vi.

Ouput Buffer In
A 3-dimensional array of int32's. The user is responsible for ensuring the array is the proper dimension and size (Channels * Segment Count * Xfer Length). The array is filled with the appropriate values and returned in the Output Buffer.

Return Values
Output Buffer
The Output Buffer In filled with the appropriate raw ADC int32 values.

Error Code
An integer that represents the return code of the call. A positive value indicates success and a negative value indicates failure. A descriptive error string can be obtained by calling CsLv_GetErrorString.vi.
s

Connector Pane



Controls and Indicators

Input Buffer

Numeric

Segment Count (1)

Xfer Length

Pattern

Channels (1)

Output Buffer In

Numeric

Output Buffer

Numeric

Error Code

CsLv_Extract8BitEx.vi

This VI converts an interleaved 8-bit buffer obtained from CsLv_Transfer8Ex.vi and converts it into a 3 dimensional buffer (Channels * Segment Count * Xfer Length).

This VI should only be used for converting data uInt8 data. The caller is responsible for passing in an 3-dimensional array of uInt8's (Channels * Segment Count * XFer Length), which is used to hold the extracted data.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel s(default=1)
This integer value specifies the number of channels that are interleaved in the Input Buffer data. The number of channels can be obtained from the CsLv_Transfer8Ex.vit.

Segment Count (default=1)
The number of segments contained in the interleaved Input Buffer.

Xfer Length
This double variable value specifies the amount of data, in samples, in the Input Buffer

Pattern
A constant that indicates the order of the data for different channels. For example if the Data Format is 0x11221122 , the 2 channel data will be formated as 2 samples of Channel 1, 2 samples of Channel 2, 2 sample of Channel 1, etc. Can be obtained by calling CsLv_Transfer8Ex.vi

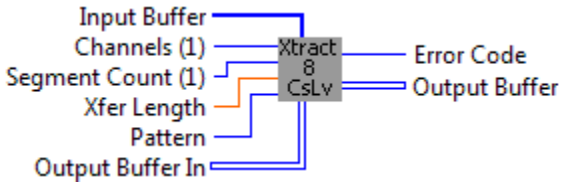
Input Buffer
An array of uInt8 values containing the raw ADC values from CsLv_Transfer8Ex.vi.

Ouput Buffer In
A 3-dimensional array of uInt8's. The user is responsible for ensuring the array is the proper dimension and size (Channels * Segment Count * Xfer Length). The array is filled with the appropriate values and returned in the Output Buffer.

Return Values
s
Output Buffer
The Output Buffer In filled with the appropriate raw ADC uInt8 values.

Error Code
An integer that represents the return code of the call. A positive value indicates success and a negative value indicates failure. A descriptive error string can be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Input Buffer

Numeric

Segment Count (1)

Xfer Length

Pattern

Channels (1)

Output Buffer In

Numeric

Output Buffer

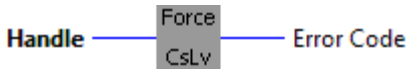
Numeric

Error Code

CsLv_ForceCapture.vi
This VI forces a trigger event on the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. This VI is useful in order to terminate acquisitions that are judged to have taken longer than acceptable in the application. The VI may also be used to create a trigger event as soon as possible. This is useful in applications where the signal of interest is associated not with any electrical signal feature, but with a software event, such as a button press.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using the CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

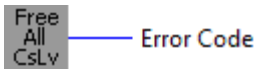
Error Code

CsLv_FreeAllSystems.vi
This VI frees all CompuScope system handles that were previously obtained by calling CsLv_GetSystem.vi.

This VI is useful as a recovery tool in the event that the CompuScope system handle is not properly released due to abnormal program termination.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

I32 **Error Code**

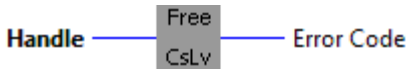
CsLv_FreeSystem.vi

This VI frees the CompuScope system identified by Handle, which should have been obtained by using CsLv_GetSystem.vi.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Until CsLv_FreeSystem.vi is called, a system that was obtained by CsLv_GetSystem.vi, will be considered to be in use by the driver and will be unavailable for other processes.

Connector Pane



Controls and Indicators

U32 **Handle**

I32 **Error Code**

CsLv_GetAdvMulRecBufferSize.vi

This VI returns the size (in bytes) of the buffer needed for advanced multiple record data on the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. This VI is used to determine the advanced multiple record buffer size so that it can be allocated before calling CsLv_GetAdvMulRecData. The Input parameters are the Handle, the Segment Start and the Segment Count. The CsLvGetAdvMulRecData.vi transfers Segment Count segments, starting from Segment Start for all channels in the system. The size of the segment is whatever was set for that CompuScope system. Included in the buffer size is room for internal information that allows the buffer to be parsed with CsLv_ParseAdvMulRecData.vi.

If the call is successful, a 1 is returned in the Error Code indicator and the size in bytes of the buffer that is needed is returned in the Size indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using the CsLv_GetErrorString.vi. Note that not all CompuScope systems support advanced multiple record. If it is not supported, an error will be returned.

Input Settings

Handle
The Handle is an integer that uniquely describes a CompuScope system. It is obtained by calling CsLv_GetSystem.vi

Segment Start (default = 1)
The multiple record Segment at which to start the transfer.

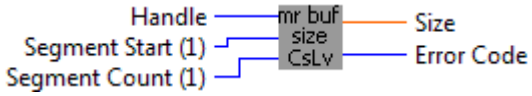
Segment Count (default = 1)
The number of multiple record segments to transfer.

Output Settings

Error Code
An integer that represents either success or an error.

Size
A double that returns the size of the required buffer in bytes.

Connector Pane



Controls and Indicators

Handle

Segment Start (1)

Segment Count (1)

Error Code

Size

CsLv_GetAdvMulRecData.vi
This VI returns an advanced multiple record buffer for the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. An advanced multiple record buffer returns multiple record data for all channels in the system, for Segment Count segments starting from Segment Start. The size of the segment is whatever was set for that CompuScope system. Included in the buffer size is room for internal information that allows the buffer to be parsed with CsLv_ParseAdvMulRecData.vi. This method of retrieving multiple record data can greatly speed transfer if the adv. mulrec buffer is saved and then parsed duing post processing.

If the call is successful, a 1 is returned in the Error Code indicator, the raw multiple record data is returned in Buffer and the size (in bytes) of the buffer that is needed is returned in the Actual Size indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using the CsLv_GetErrorString.vi. Note that not all CompuScope systems support advanced multiple record. If it is not supported, an error will be returned.

Input Settings

Handle
The Handle is an integer that uniquely describes a CompuScope system. It is obtained by calling CsLv_GetSystem.vi

Segment Start (default = 1)
The multiple record Segment at which to start the transfer.

Segment Count (default = 1)
The number of multiple record segments to transfer.

Raw Buffer
A 1 dimensional array of uInt8. The calling program must allocated an initialize the array. The necessary size (in bytes) can be obtained by calling CsLv_GetAdvMulRecBufferSize.vi. The buffer should be initialized to 0.

Size

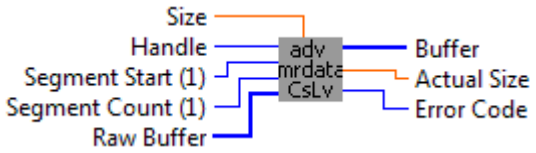
The size (in bytes) of the Raw Buffer.

Output Settings
Error Code
An integer that represents either success or an error.

Buffer
The Raw Buffer filled with the advanced multiple record values for all channels, including a header with internal information.

Actual Size
A double that returns the size of the required buffer in bytes.

Connector Pane



Controls and Indicators

Handle

Segment Start (1)

Segment Count (1)

Size

Raw Buffer

Numeric

Error Code

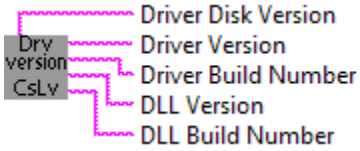
Actual Size

Buffer

Numeric

CsLv_GetDriverVersion.vi
This VI returns the version numbers of various subsystems of the CompuScope driver. If the version and build number of any subsystem cannot be determined, they will be marked as "Unknown".

Connector Pane



Controls and Indicators

Driver Version

DLL Version

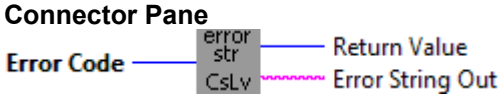
Driver Build Number

DLL Build Number

Driver Disk Version

CsLv_GetErrorString.vi
This VI inputs an error code from another CompuScope VI as input and returns a descriptive error string as Error String Out. If the call is successful, a 1 is returned as Return Value. If the error code does not correspond to a valid error string, then a negative number will be returned.

Connector Pane



Controls and Indicators

Error Code

Error String Out

Return Value

CsLv_GetExtendedOptions.vi
This VI returns the available extended options information from the driver for the CompuScope system. The extended options information tells the user which additional FPGA images are available and in which flash page they reside.

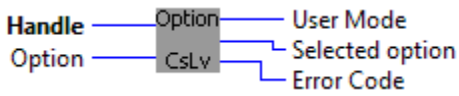
The CompuScope system is identified by Handle, which is obtained with CsLv_GetSystem.vi

The Options parameter identifies which option to check for. Currently available options are Finite Impulse Response (1), Signal Averaging (2), Peak Detection (4), Cascaded Streaming (8), Multiple Record Averaging (16), Storage Media Testing (32) , 512 FFT (64), 1K FFT (128), 2K FFT (256) and 4K FFT (512).

If the option is found, the User Mode parameter will return the value that should be OR'ed with the Mode in the call to CsLv_ConfigureAcquisition in order to activate the option. This value tells the driver to load the appropriate image for the option. If the option is not found, a 0 is returned in the User Mode parameter. The actual option selected is returned in Selection option. This allows the user to put more than 1 option in the Options input (for example, 960 for all the FFT's) and the first one chosen is returned in the Selected option field.

If the function succeeds, a 1 is returned in the Error Code indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Option

Error Code

User Mode

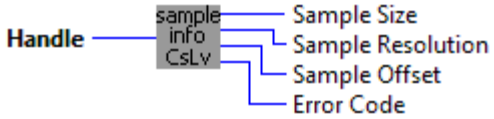
Selected option

CsLv_GetSampleInfo.vi
This VI returns information about the Samples acquired by the CompuScope system identified by Handle, which is obtained with CsLv_GetSystem.vi.

The returned values are Sample Size, Sample Resolution, and Sample Offset. Sample Size is the size of the sample of the CompuScope hardware in the current mode. For instance, for 8-bit CompuScope models, the Sample Size is 1. For 12, 14, and 16-bit CompuScope models, the Sample Size is 2. Digital input CompuScope models allow 32-bit mode, which has a Sample Size of 4. Sample Resolution is the scaling factor to convert CompuScope data to Volts; it represents a data value corresponding to full scale of the CompuScope hardware. Sample Offset is the value that represents 0 Volts. Sample Resolution and Sample Offset may be used to convert raw ADC CompuScope data into voltage values, as described in the CompuScope SDK for LabVIEW manual.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an error value is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi

Connector Pane



Controls and Indicators

Handle

Sample Size

Sample Resolution

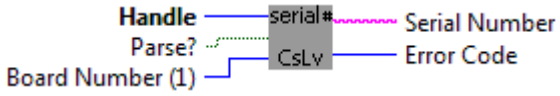
Sample Offset

Error Code

CsLv_GetSerialNumber.vi
This VI returns the serial number of the board identified by Board Number in the CompuScope system identified by Handle, which is obtained by calling CsLv_GetSystem.vi.

Board Number is the number of a CompuScope board within the system. CompuScope board numbers begin at 1. If an invalid board number is given, an error is returned.
If the call is successful, the serial number of the board is placed in the string indicator Serial Number and a 1 is returned in the Error Code indicator. If the call is not successful, an error value is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Board Number (1)

Parse?

Serial Number

Error Code

CsLv_GetSystem.vi

This VI returns a unique handle for an available CompuScope system. This Handle is an integer value that is used as an input to most other CsLv VIs. The Board Type, Channel Count, Sample Bits, and Index input fields may be used to search only for CompuScope systems with capabilities specified in these input fields. A 0 in any of the input fields means that the VI will not narrow its search using this field. If all the input fields are set to 0, the VI will return the Handle for the first available CompuScope system.

If the call is successful, a 1 is returned in the Error Code indicator and a valid Handle is returned in the Handle output. If the call is unsuccessful, a 0 is returned in the Handle field and an appropriate error code is returned in the Error Code indicator. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Valid Input Values:

The default input values for Board Type, Channel Count, Sample Bits, and Index are 0. Valid values are:
Board Type (default = 0):
A value of 0 means the board is not specified and the VI will find the first available system. CsLv_BoardNameToType.vi may be used to determine the Board Type constant for your CompuScope model.

Channel Count (default = 0):
0 = not specified
1
2
,...

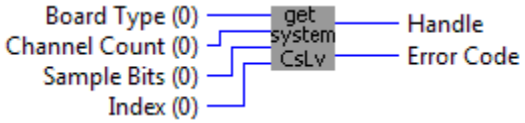
Sample Bits (default = 0):
0 = not specified
1 = 8 bits
2 = 12 bits
3 = 14 bits
4 = 16 bits
5 = 32 bits

Index (default = 0):

The index is used to get a specific CompuScope system in a computer, i.e. set to 3 to select the 3rd CompuScope system.

0 = not specified
1
2
,...

Connector Pane



Controls and Indicators

Channel Count (0)

Sample Bits (0)

Board Type (0)

Index (0)

Handle

Error Code

CsLv_GetSystemCaps.vi

This VI returns information about the CompuScope system identified by Handle, which is obtained with CsLv_GetSystem.vi. The VI can return information about different CompuScope functionalities. All information is returned as an array of double variables. The number of elements in the returned array will vary depending upon which functionality is being queried.

The functionality to be queried must be entered in the CapsID field. Available functionalities and their corresponding CapsID values are available in the file CsDefines.h (in hexadecimal), which is installed with the CompuScope driver. The value to be used is the high word shifted right by 16 bits. For example, to determine the available sample rates, the constant is 0x10000 (in hex). The value to use in in this vi is 0x10000 shifted right by 16, or 1.

Index – the channel index, which is required for some CapsID queries. The Input ranges, Input Impedances and Input Coupling queries all require a channel index. CompuScope channels for this VI begin at 1. If a channel index is not required, the value should be 0.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane

Controls and Indicators

Handle

CapsID

Index

Error Code

Buffer

Numeric

CsLv_GetSystemInfo.vi
This VI returns information about the CompuScope system identified by Handle, which is obtained with CsLv_GetSystem.vi.

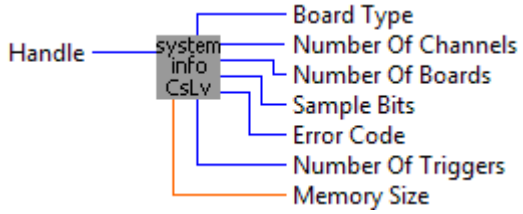
The available information is as follows:
Memory Size (of each CompuScope board in the system)
Board Type
Number Of Triggers (number of trigger engines in the system)
Number Of Channels (in the system)
Number Of Boards (in the system)
Sample Bits (vertical resolution for the CompuScope system)

Board Type is an integer constant used by the CompuScope driver to identify each type of CompuScope board. CsLv_BoardNameToType.vi may be used to determine the Board Type for any CompuScope model.

Board Type, Number Of Triggers, Number Of Channels, Number Of Boards and Sample Bits are all integer values. Memory Size is a double variable that indicates the amount of on-board CompuScope memory in samples.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Number Of Boards

Number Of Channels

Number Of Triggers

Error Code

Memory Size

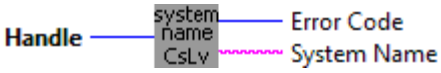
Board Type

Sample Bits

CsLv_GetSystemName.vi
This VI returns the System name for the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi.

If the call is successful, the name of the system is returned in the System Name string indicator and a 1 is returned in the Error Code indicator. If the call is unsuccessful, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

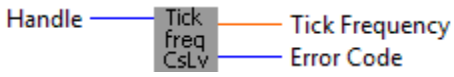
System Name

Error Code

CsLv_GetTimeStampTickFrequency.vi
This VI returns the frequency of the Trigger Time Stamp clock on the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. This frequency can be used to convert values returned from reading the Trigger Time Stamp counter (which can be done either by CsLv_TransferTmb or CsLvGetAdvMulRecData) into intervals of time.

If the call is successful, a 1 is returned in the Error Code indicato and the tick frequency is returned in the Tick Frequency indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using the CsLv_GetErrorString.vi. Note that not all CompuScope systems support trigger time stamp. If it is not supported, an error will be returned.

Connector Pane



Controls and Indicators

Handle

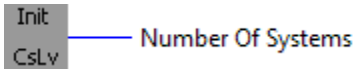
Error Code

Tick Frequency

CsLv_Initialize.vi
This VI initializes the CompuScope driver and returns the total number of CompuScope systems.

If the call is successful, a positive integer will appear in the Number Of Systems field. Otherwise, a negative integer will appear in the Number Of Systems field. The negative integer indicates an error and is the error code. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

I32 **Number Of Systems**

CsLv_ParseAdvMulRecData.vi

This VI parses an advanced multiple record buffer. An advanced multiple record buffer returns multiple record data for all channels in the system, for Segment Count segments starting from Segment Start. This method of retrieving multiple record data can greatly speed transfer if the adv. mulrec buffer is saved and then parsed duing post processing. The input parameters to this vi are the Raw Buffer, Raw Buffer Size, the Channel from which to retrieve data, the Start Segment, the Segment Count, the Start address of the data within each segment, the Length of each segment, the Tick Frequency (if time stamp data is desired) and two buffers which will be filled with data, Data Buffer and Time Stamp Data.

If the call is successful, a 1 is returned in the Error Code indicator, multiple record data is returned in Data Buffer Out and, if requested, time stamp data is returned in Time Stamp Data Out. The Actual Start and Actual Length will be returned in their respective outputs. Otherwise, an error code is returned. A descriptive error string may be obtained by using the CsLv_GetErrorString.vi. Note that not all CompuScope systems support advanced multiple record. If it is not supported, an error will be returned.

Input Settings

Raw Buffer
A 1-dimension array of uInt8 containing advanced multipe record data filled by calling CsLv_GetAdvMulRecData.vi.

Raw Buffer Size
The size of the advanced multiple record data buffer in bytes. Can be obtained by calling CsLv_GetAdvMulRecBufferSize.vi.

Channel (default=1)
The channel for which to transfer data. CompuScope channels begin at Channel 1.

Segment Start (default = 1)
The multiple record Segment at which to start the transfer.

Segment Count (default = 1)
The number of multiple record segments to transfer.

Start (default =0)
The address (in samples) in each segment at which to start the transfer.

Length
The length (in samples) of each segment to transfer.

Data Buffer
A 2-dimensional array that is allocated and initialized by the calling program in which to put the transferred data. The dimensions are the number of segments (Segment Count) by the size of each segment (Length). The buffer is filled and returned in Data Buffer Out. NOTE: It is the programs responsibility to allocate the buffer with the right data type for the CompuScope system

that it was originally produced by. 8 bit systems would require an array of uInt8's while 12, 14 and 16 bit systems would require an array of int16's.

Time Stamp Data
A 1-dimensional array (allocated and initialized by the calling program) of doubles in which to hold Trigger Time Stamp data for each segment. The array should be of size SegmentCount, as there is 1 trigger time stamp per segment. If the Tick Frequency input is 0 or less, the Time Stamp Data values are ignored.

Tick Frequency
A double value that contains the tick frequency of the trigger time stamp clock. This value can be obtained by calling CsLv_GetTimeStampTickFrequency.vi. If this value is set to 0 or less, the time stamp values are ignored.

Output Settings
Error Code
An integer that represents either success or an error.

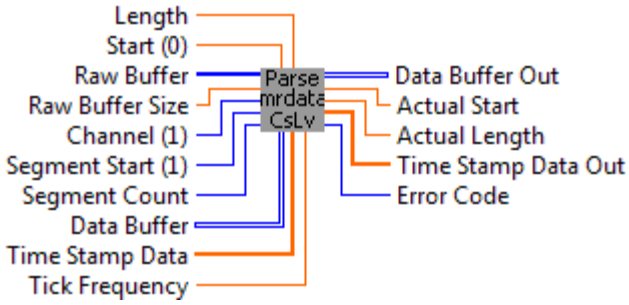
Data Buffer Out
This is the 2 dimensional Data Buffer that was passed by the calling vi, now filled with valid multiple record data for each segment.

Actual Start
A double that holds the actual start address of the data for each segment returned from the driver. In some cases this may differ from the requested start address.

Actual Length
A double that holds the actual length of the data for each segment returned from the driver. In some cases this may differ from the requested length.

Time Stamp Data Out
This is the 1-dimensional Time Stamp Data array that was passed by the calling vi. If the Tick Frequency input was greater than 0, it is now filled with valid time stamp values for each segment.

Connector Pane



Controls and Indicators

Segment Start (1)

Channel (1)

Raw Buffer Size

Raw Buffer


Numeric


Start (0)


Length


Data Buffer


Numeric


Time Stamp Data


Numeric


Segment Count


Tick Frequency


Error Code


Time Stamp Data Out


Numeric


Data Buffer Out


Numeric


Actual Start


Actual Length

CsLv_QueryAcquisitionParameters.vi
This VI returns the acquisition parameters that have been set in the driver for the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. If the call can not be completed, an appropriate error code is returned. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

The returned acquisition parameters are Sample Rate, Ext. Clock, Mode, Segment Count, Depth, Segment Size, Trigger Timeout, Trigger Holdoff, Trigger Delay, Sample Size, Sample Resolution and Sample Offset. The returned values are those that were successfully set on the CompuScope hardware by the last call to CsLv_Commit.vi. The returned values are not valid until CsLv_Commit.vi has been called at least once.

Return Value Descriptions:

Sample Rate
This double variable value is the rate at which the CompuScope system will sample input waveform. It is specified in Samples per second.

Ext. Clock
This integer value is returned as 1 if the system has been set to use external clocking. Otherwise, it is returned as 0.

Mode
This integer value indicates the current operating mode for the CompuScope system. Possible return values are:

1 = single channel mode (8-bit mode for digital input CompuScope cards)
2 = dual channel mode (16-bit mode for digital input CompuScope cards)
4 = quad channel mode (32-bit mode for digital input CompuScope cards)
8 = octal channel mode

Please note that higher order bits in the returned Mode value may be non-zero, which indicates alternate functionalities. For more details, see the CsLv_ConfigureAcquisition.vi documentation.

Segment Count
This integer value indicates how many segments (or records) will be captured in a multiple record acquisition. For Single Record acquisitions, it is always returned as 1.

Depth
This double variable value indicates the number of post-trigger samples that the CompuScope system is set to acquire.

Segment Size
This double variable indicates 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
This double variable value indicates the amount of time that the CompuScope system will wait before forcing a trigger. The value is in microseconds.

Trigger Delay
This double variable value indicates the number of Samples between the time that the trigger event actually occurs and the time that the trigger event will be logged in the data record. Setting a non-zero trigger delay is useful for signals where the region of interest in the signal occurs long after the trigger event. Trigger Delay is not supported on all CompuScope models.

Trigger Holdoff
This double variable value indicates the amount of time in Samples during which the CompuScope hardware will ignore trigger events after beginning an acquisition. Trigger Holdoff is useful in order to ensure the accumulation of specified number of pre-trigger points by ignoring trigger events until the specified number has been acquired. Not all CompuScopes cards support Trigger Holdoff.

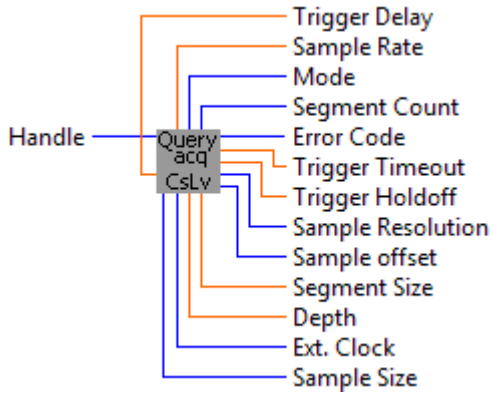
Sample Size
Sample size is the size in bytes of a data sample, i.e. 1 for 8 bit data, 2 for 12, 14 and 16 bit data and 4 for 32 bit data. Because the sample size may be affected by the loading of different FPGA images, this VI will return the appropriate value for the current sample size.

Sample Resolution
The sample resolution is the number of digitization levels between 0 and positive full scale, or 0 and negative full scale. Because the sample resolution may be affected by the loading of different FPGA images, this VI will return the appropriate value for the current sample resolution. The sign of the Sample Resolution specifies the polarity of the A/D converter.

Sample Offset

The offset is the value that represents the middle of the digitization range. Because the sample offset may be affected by the loading of different FPGA images, this VI will return the appropriate value for the current sample offset.

Connector Pane



Controls and Indicators

Handle

Sample Rate

Mode

Segment Count

Depth

Trigger Timeout

Trigger Holdoff

Trigger Delay

Error Code

Ext. Clock

Segment Size

Sample Resolution

Sample offset

Sample Size

CsLv_QueryChannelParameters.vi

This VI returns the channel parameter settings for the specified Channel identified by Handle, which is obtained by using CsLv_GetSystem.vi. If the call cannot be completed, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

This VI returns the following values for the channel identified by the Channel input: Coupling, Differential Input, Direct-to-ADC, Input Range, Impedance, and DC Offset. If an invalid channel is

specified, an appropriate error code is returned. The returned values are those that were successfully set on the CompuScope hardware by the last call to CsLv_Commit.vi. The returned values are not valid until CsLv_Commit.vi has been called at least once.

The Channel input is an integer value that indicates the channel number that is to be queried. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid channel number is specified, an appropriate error code is returned.

Return Value Descriptions:

Coupling
This integer value indicates the current input coupling. Possible values are:
1 = DC input coupling
2 = AC input coupling

Differential Input
This integer value indicates whether the channel is configured to use differential input coupling. Possible values are:
0 = differential input coupling is not activated
1 = differential input coupling is activated

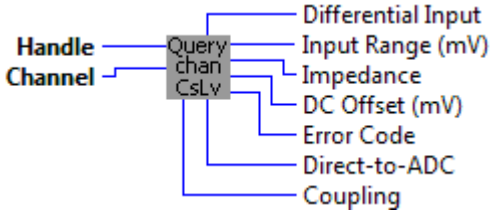
Direct-to-ADC
This integer value indicates whether the channel is configured to use Direct-to-ADC input range. Possible values are:
0 = Direct-to-ADC is not activated
1 = Direct-to-ADC is activated

Input Range
This integer value indicates the current full scale input range (in millivolts) setting for the current channel. For instance, for the +/- 1 Volt input range, the Input Range value is 2000. Values may differ depending on the CompuScope system.

Impedance
This integer value indicates the terminating input impedance for the channel. Possible values are:
1000000 = 1 MOhm input termination is activated
50 = 50 Ohms input termination is activated

DC Offset
This integer value indicates the current DC offset (in millivolts) for the specified channel. Not all CompuScope models support DC offsets.

Connector Pane



Controls and Indicators

U32 **Handle**

Channel

Error Code

Coupling

Differential Input

Input Range (mV)

Impedance

DC Offset (mV)

Direct-to-ADC

CsLv_QueryChannelParametersEx.vi
This VI returns the channel parameter settings for the specified Channel identified by Handle, which is obtained by using CsLv_GetSystem.vi. If the call cannot be completed, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

This VI returns the following values for the channel identified by the Channel input: Coupling, Differential Input, Direct-to-ADC, Input Range, Impedance, and DC Offset. If an invalid channel is specified, an appropriate error code is returned. The returned values are those that were successfully set on the CompuScope hardware by the last call to CsLv_Commit.vi. The returned values are not valid until CsLv_Commit.vi has been called at least once.

The Channel input is an integer value that indicates the channel number that is to be queried. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid channel number is specified, an appropriate error code is returned.

Return Value Descriptions:

Coupling
This integer value indicates the current input coupling. Possible values are:
1 = DC input coupling
2 = AC input coupling

Differential Input
This integer value indicates whether the channel is configured to use differential input coupling. Possible values are:
0 = differential input coupling is not activated
1 = differential input coupling is activated

Direct-to-ADC
This integer value indicates whether the channel is configured to use Direct-to-ADC input range. Possible values are:
0 = Direct-to-ADC is not activated
1 = Direct-to-ADC is activated

Input Range

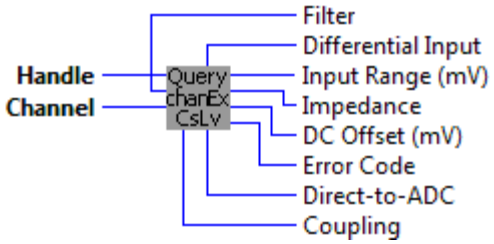
This integer value indicates the current full scale input range (in millivolts) setting for the current channel. For instance, for the +/- 1 Volt input range, the Input Range value is 2000. Values may differ depending on the CompuScope system.

Impedance
This integer value indicates the terminating input impedance for the channel. Possible values are:

1000000 = 1 MOhm input termination is activated
50 = 50 Ohms input termination is activated

DC Offset
This integer value indicates the current DC offset (in millivolts) for the specified channel. Not all CompuScope models support DC offsets.

Connector Pane



Controls and Indicators

Handle

Channel

Error Code

Coupling

Differential Input

Input Range (mV)

Impedance

DC Offset (mV)

Direct-to-ADC

Filter

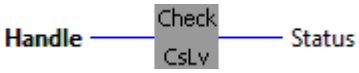
CsLv_QueryStatus.vi
This VI returns the current acquisition Status of the CompuScope system identified by Handle, which is obtained by calling CsLv_GetSystem.vi.

If the call is successful, a positive or zero status code is returned.
The possible values are:

0 = the system has finished acquiring data.
1 = the system is acquiring, but has not yet been triggered.
2 = the system is acquiring, and has been triggered.
3 = the system is transferring data.

A negative number is an error. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

U32 **Handle**

I32 **Status**

CsLv_QueryTimeStampInfo.vi
This VI returns information about the CompuScope Time Stamping functionality associated with the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi.

Time Stamping functionality is provided by an on-board counter whose value is latched upon each trigger event - thus time stamping the record's trigger event. The counter source may be derived from the CompuScope sampling oscillator or by a separate on-board fixed frequency oscillator.

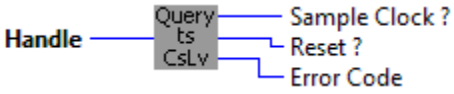
If the VI is successful, a 1 is returned in the Error Code indicator. If the call is unsuccessful, an error is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Return Value Descriptions:

Sample Clock ?
This integer value indicates which clock source the time stamping counter is using. Possible values are:
  1 = the time stamping counter source is derived from the CompuScope sampling clock.
  0 = the time stamping counter source is the on-board fixed frequency oscillator.

Reset ?
This integer value indicates whether the time stamping counter is to be reset after every acquisition. Possible values are:
  1 = the time stamping counter is to be reset after every acquisition
  0 = the time stamping counter is not reset after every acquisition.

Connector Pane



Controls and Indicators

U32 **Handle**

U8 **Sample Clock ?**

U8 **Reset ?**

I32 **Error Code**

CsLv_QueryTriggerParameters.vi
This VI returns the trigger parameter settings for the trigger engine identified by Trigger within the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. If the call is successful, a 1 is returned. If the call is unsuccessful, an appropriate error code is returned. For a descriptive error string, use CsLv_GetErrorString.vi.

This VI queries the trigger engine identified by Trigger for its Slope, Level, Source, External Coupling, and External Range values. The returned values are those that were successfully set on the CompuScope hardware by the last call to CsLv_Commit.vi. The returned values are not valid until CsLv_Commit.vi has been called at least once.

The integer Trigger input value identifies the number of the trigger engine in the system for which the parameters are to be queried. Trigger engine numbers in a LabVIEW CompuScope system start at 0. (The engine numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid engine number is specified, an appropriate error code is returned.

Return Value Descriptions:

Slope
This integer value indicates the trigger signal slope that will cause a trigger event to occur. Possible values are:
0 = falling edge is activated
1 = rising edge is activated

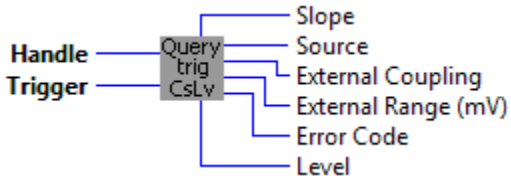
Level
This integer value indicates the trigger level as a percentage of the input range of the trigger source. Possible values between -100% and + 100%. For example, in the +/-1 Volt range, 100% corresponds to a +1 Volt trigger level, 50% is +500 millivolts, and -100% is -1 Volt.

Source
This integer value indicates the trigger source for the current trigger engine. Possible values are:
-1 = external trigger input is the trigger source
0 = trigger engine is disabled
1 = Channel 1 is the trigger source
2 = Channel 2 is the trigger source
etc. for all available channels in the system

External Coupling
This integer indicates the coupling of the external trigger input. Possible values are:
1 = external trigger coupling is set to DC
2 = external trigger coupling is set to AC

External Range
This integer indicates the input range of the external trigger input in millivolts. Possible values are:
2000 = the +/-1 Volt external trigger range is activated
10000 = the +/-5 Volt external trigger range is activated

Connector Pane



Controls and Indicators

Handle

Trigger

Error Code

Slope

Source

Level

External Coupling

External Range (mV)

CsLv_QueryTriggerParametersEx.vi
This VI returns the trigger parameter settings for the trigger engine identified by Trigger within the system identified by Handle, which is obtained by using CsLv_GetSystem.vi. If the call is successful, a 1 is returned. If the call is unsuccessful, an appropriate error code is returned. For a descriptive error string, use CsLv_GetErrorString.vi.

This VI queries the trigger engine identified by Trigger for its Slope, Level, Source, External Coupling, and External Range values. The returned values are those that were successfully set on the CompuScope hardware by the last call to CsLv_Commit.vi. The returned values are not valid until CsLv_Commit.vi has been called at least once.

The integer Trigger input value identifies the number of the trigger engine in the system for which the parameters are to be queried. Trigger engine numbers in a LabVIEW CompuScope system start at 0. (The engine numbers are incremented by 1 before being sent to the driver, as required by the C API.) If an invalid engine number is specified, an appropriate error code is returned.

Return Value Descriptions:

Slope
This integer value indicates the trigger signal slope that will cause a trigger event to occur. Possible values are:

0 = falling edge is activated
1 = rising edge is activated

Level
This integer value indicates the trigger level as a percentage of the input range of the trigger source. Possible values between -100% and + 100%. For example, in the +/-1 Volt range, 100% corresponds to a +1 Volt trigger level, 50% is +500 millivolts, and -100% is -1 Volt.

Source
This integer value indicates the trigger source for the current trigger engine. Possible values are:

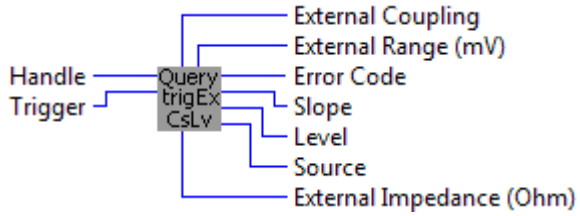
-1 = external trigger input is the trigger source
0 = trigger engine is disabled
1 = Channel 1 is the trigger source
2 = Channel 2 is the trigger source
etc. for all available channels in the system

External Coupling
This integer indicates the coupling of the external trigger input. Possible values are:
1 = external trigger coupling is set to DC
2 = external trigger coupling is set to AC

External Range
This integer indicates the input range of the external trigger input in millivolts. Possible values are:
2000 = the +/-1 Volt external trigger range is activated
10000 = the +/-5 Volt external trigger range is activated

External Impedance
This integer indicates theimpedance of the external trigger input inOhm. Possible values are:
50 = the 50 Ohm
1000000 = the HiZ impedance

Connector Pane



Controls and Indicators

Handle

Trigger

Error Code

Slope

Source

Level

External Coupling

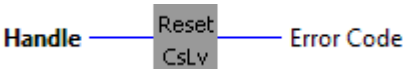
External Range (mV)

External Impedance (Ohm)

CsLv_ResetSystem.vi
This VI reset the CompuScope system identified by Handle, which should have been obtained by using CsLv_GetSystem.vi to its default state of configuration.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

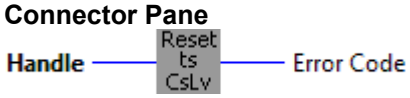
Error Code

CsLv_ResetTimeStamp.vi

This VI resets the Time Stamping counter associated with the CompuScope system identified by Handle, which is obtained by calling CsLv_GetSystem.vi. Subsequent time stamps will be measured with respect to this reset event. The time stamp should be reset at the beginning of VI execution in order to clear any initial large counter values.

If the call is successful, a 1 is returned in the Error Code indicator. Otherwise, an appropriate error code is returned. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Error Code

CsLv_Transfer16.vi

This VI transfers 16-bit data from one channel of the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. For accurate interpretation of the transferred data, the user should always use the returned Actual Start and Actual Length, and not the requested values.

This VI should only be used for transferring data from analog CompuScope models with more than 8-bit resolution. Data are transferred from on-board CompuScope memory to PC RAM using PCI Bus Mastering at rates of up to 200 MB/s.

CsLv_Transfer16.vi transfers raw integer ADC code values. The VI will not return programmatic control to LabVIEW until all of the data have been transferred.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data is available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel

This integer value specifies the channel from which data are to be transferred. The default value is 0 for the first channel in the CompuScope system. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent down to the driver, as required by the C API.) If an invalid channel number is specified, an appropriate error code is returned.

Mode

This integer value determines the current transfer mode. Currently, the only valid value for this VI is 0.

Segment
This integer value determines which Multiple Record segment is to be transferred, if the CompuScope hardware was operating in Multiple Record Mode. For Single Record acquisitions, the user must always set the Segment to 1, which is the default value. If the Segment value is too large, an error will occur.

Start
This double variable value determines the requested starting point of the data transfer from the CompuScope memory to PC RAM. Start is specified relative to the trigger address for the acquisition. A Start value of 0 will cause data transfer to begin at the trigger address. Negative Start values are for the transfer of pre-trigger data. Positive Start values may be used to begin transfer after the trigger address. The default Start value is 0.

Length
This double variable value specifies the amount of data, in samples, to be transferred. The default is 4096.

Return Value Descriptions:

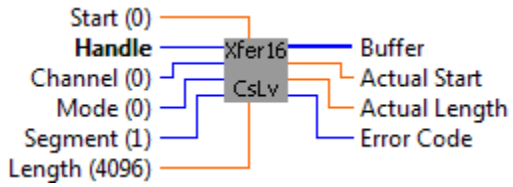
Buffer
This is an int16 LabVIEW buffer variable into which transferred data are returned. The size of the array will be at least Length samples. Data are returned as raw ADC values.

Actual Start
This double variable value returns the actual start address point of the data array. The Actual Start may be lower than the requested Start value. This is because the CompuScope memory architecture may force the drivers to download data from an earlier point than the start position. For example, if Start is input to CsLv_Transfer16.vi as 0, the Actual Start may return as -2, indicating that two extra samples were returned before the Start position.

The user must account for the discrepancy between the Start and the Actual Start. The user may choose to ignore samples transferred from before the Start position. Alternatively, the user may conserve these samples, accounting for the fact that the values begin at Actual Start.

Actual Length
This double variable value returns the actual amount of data transferred, which may be different from the requested Length. The difference, if any, results from CompuScope memory architecture.

Connector Pane



Controls and Indicators

U32 **Handle**

U16 **Channel (0)**

Segment (1)

Start (0)

Length (4096)

Mode (0)

Buffer

Numeric

Actual Length

Error Code

Actual Start

CsLv_Transfer16Ex.vi
This VI transfers 16-bit data from the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. Segment Count transfers for either all channels or 1 channel of the CompuScope system can be transferred.

This VI should only be used for transferring data from 12, 14 or 16 bit CompuScope cards. Data are transferred from on-board CompuScope memory to PC RAM using PCI Bus Mastering at rates of up to 200 MB/s.

CsLv_Transfer16Ex.vi transfers raw integer ADC code values. The VI will not return programmatic control to LabVIEW until all of the data have been transferred.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Handle
An integer value which uniquely describes a CompuScope system. The value is obtained by calling CsLv_GetSystem.vi

Channel (default=-1)
This integer value specifies the channel(s) from which data are to be transferred. The default value is -1 for all the channels in the CompuScope system. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent down to the driver, as required by the C API.) This vi can return either all the channels in the system or 1 channel in the system. If an invalid channel number is specified, an appropriate error code is returned.

Start Segment (default=1)
This integer value determines which Multiple Record segment to start transferring from, if the CompuScope hardware was operating in Multiple Record Mode. For Single Record acquisitions, the user must always set the Segment to 1, which is the default value. If the Segment value is too large, an error will occur.

Segment Count (default=1)
The number of segments to transfer, starting from Start Segment. If an invalid value is given for Segment Count, an error will be returned.

Start
This double variable value determines the requested starting point of the data transfer from the CompuScope memory to PC RAM. Start is specified relative to the trigger address for the acquisition. A Start value of 0 will cause data transfer to begin at the trigger address. Negative Start values are for the transfer of pre-trigger data. Positive Start values may be used to begin transfer after the trigger address. The default Start value is 0.

Length
This double variable value specifies the amount of data, in samples, to be transferred. The default is 4096.

Buffer In
A 1 dimensional int16 array. The size of the array must be Channels * Segment Count * Length. The user is responsible for ensuring that the buffer is at least the proper length.

Return Value Descriptions:

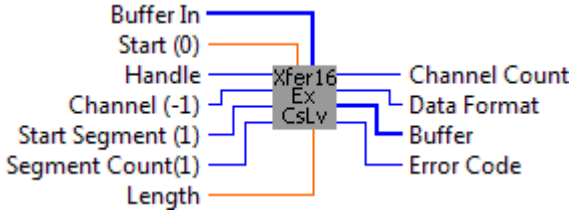
Buffer
This is the Buffer In variable filled with raw ADC values. A 3-dimensional array of Channels * Segment Count * Length can be obtained by calling CsLv_Extract16BitEx.vi.

Channel Count
The actual number of channels contained inl the buffer.

Data Format
A constant that indicates the order of the data for different channels. For example if the Data Format is 0x11221122 , the 2 channel data will be formated as 2 samples of Channel 1, 2 samples of Channel 2, 2 sample of Channel 1, etc.

Error Code
An integer that represents the return code of the call. A positive value indicates success and a negative value indicates failure. A descriptive error string can be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Channel (-1)

Start Segment (1)

Start (0)

 **Length**

 **Segment Count(1)**

 **Buffer In**

 **Numeric**

 **Buffer**

 **Numeric**

 **Error Code**

 **Data Format**

 **Channel Count**

CsLv_Transfer32.vi
This VI transfers 32-bit data from one channel of the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. For accurate interpretation of the transferred data, the user should always use the returned Actual Start and Actual Length, and not the requested values.

This VI should only be used for transferring data from 32-bit CompuScope cards. Data are transferred from on-board CompuScope memory to PC RAM using PCI Bus Mastering at rates of up to 200 MB/s.

CsLv_Transfer32.vi transfers raw integer ADC code values. The VI will not return programmatic control to LabVIEW until all of the data have been transferred.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel
This integer value specifies the channel from which data are to be transferred. The default value is 0 for the first channel in the CompuScope system. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent down to the driver, as required by the C API.) If an invalid channel number is specified, an appropriate error code is returned.

Mode
This integer value determines the current transfer mode. Currently, the only valid value for this VI is 16.

Segment
This integer value determines which Multiple Record segment is to be transferred, if the CompuScope hardware was operating in Multiple Record Mode. For Single Record acquisitions,

the user must always set the Segment to 1, which is the default value. If the Segment value is too large, an error will occur.

Start
This double variable value determines the requested starting point of the data transfer from the CompuScope memory to PC RAM. Start is specified relative to the trigger address for the acquisition. A Start value of 0 will cause data transfer to begin at the trigger address. Negative Start values are for the transfer of pre-trigger data. Positive Start values may be used to begin transfer after the trigger address. The default Start value is 0.

Length
This double variable value specifies the amount of data, in samples, to be transferred. The default is 4096.

Return Value Descriptions:

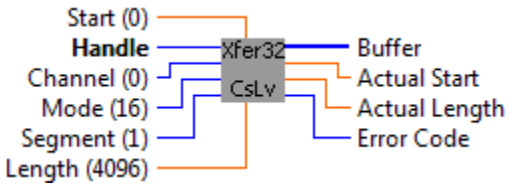
Buffer
This is an Int32 LabVIEW buffer variable into which transferred data are returned. The size of the array will be at least Length samples. Data returned are as raw ADC values.

Actual Start
This double variable value returns the actual start address point of the data array. The Actual Start may be lower than the requested Start value. This is because the CompuScope memory architecture may force the drivers to download data from an earlier point than the start position. For example, if Start is input to CsLv_Transfer32.vi as 0, the Actual Start may return as -2, indicating that two extra samples were returned before the Start position.

The user must account for the discrepancy between the Start and the Actual Start. The user may choose to ignore samples transferred from before the Start position. Alternatively, the user may conserve these samples, accounting for the fact that the values begin at Actual Start.

Actual Length
This double variable value returns the actual amount of data transferred, which may be different from the requested Length. The difference, if any, results from CompuScope memory architecture.

Connector Pane



Controls and Indicators

Handle

Channel (0)

Segment (1)

Start (0)

Length (4096)

Mode (16)

Buffer

Numeric

Actual Length

Error Code

Actual Start

CsLv_Transfer32Ex.vi
This VI transfers 32-bit data from the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. Segment Count transfers for either all channels or 1 channel of the CompuScope system can be transferred.

This VI should only be used for transferring data from 32 bit CompuScope cards. Data are transferred from on-board CompuScope memory to PC RAM using PCI Bus Mastering at rates of up to 200 MB/s.

CsLv_Transfer32Ex.vi transfers raw integer ADC code values. The VI will not return programmatic control to LabVIEW until all of the data have been transferred.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Handle
An integer value which uniquely describes a CompuScope system. The value is obtained by calling CsLv_GetSystem.vi

Channel (default=-1)
This integer value specifies the channel(s) from which data are to be transferred. The default value is -1 for all the channels in the CompuScope system. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent down to the driver, as required by the C API.) This vi can return either all the channels in the system or 1 channel in the system. If an invalid channel number is specified, an appropriate error code is returned.

Start Segment (default=1)
This integer value determines which Multiple Record segment to start transferring from, if the CompuScope hardware was operating in Multiple Record Mode. For Single Record acquisitions, the user must always set the Segment to 1, which is the default value. If the Segment value is too large, an error will occur.

Segment Count (default=1)
The number of segments to transfer, starting from Start Segment. If an invalid value is given for Segment Count, an error will be returned.

Start

This double variable value determines the requested starting point of the data transfer from the CompuScope memory to PC RAM. Start is specified relative to the trigger address for the acquisition. A Start value of 0 will cause data transfer to begin at the trigger address. Negative Start values are for the transfer of pre-trigger data. Positive Start values may be used to begin transfer after the trigger address. The default Start value is 0.

Length
This double variable value specifies the amount of data, in samples, to be transferred. The default is 4096.

Buffer In
A 1 dimensional int32 array. The size of the array must be Channels * Segment Count * Length. The user is responsible for ensuring that the buffer is at least the proper length.

Return Value Descriptions:

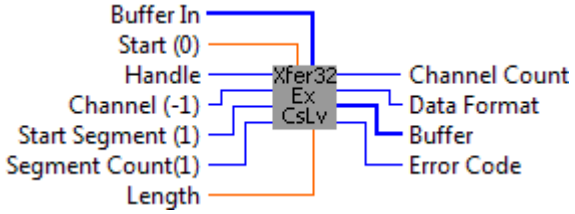
Buffer
This is the Buffer In variable filled with raw ADC values. A 3-dimensional array of Channels * Segment Count * Length can be obtained by calling CsLv_Extract32BitEx.vi.

Channel Count
The actual number of channels contained inl the buffer.

Data Format
A constant that indicates the order of the data for different channels. For example if the Data Format is 0x11221122 , the 2 channel data will be formated as 2 samples of Channel 1, 2 samples of Channel 2, 2 sample of Channel 1, etc.

Error Code
An integer that represents the return code of the call. A positive value indicates success and a negative value indicates failure. A descriptive error string can be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Channel (-1)

Start Segment (1)

Start (0)

Length

Segment Count(1)

Buffer In


Numeric


Buffer


Numeric


Error Code


Data Format


Channel Count

CsLv_Transfer8.vi
This VI transfers 8-bit data from one channel of the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. For accurate interpretation of the transferred data, the user should always use the returned Actual Start and Actual Length, and not the requested values.

This VI should only be used for transferring data from 8 bit CompuScope cards. Data are transferred from on-board CompuScope memory to PC RAM using PCI Bus Mastering at rates of up to 200 MB/s.

CsLv_Transfer8.vi transfers raw integer ADC code values. The VI will not return programmatic control to LabVIEW until all of the data have been transferred.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Channel
This integer value specifies the channel from which data are to be transferred. The default value is 0 for the first channel in the CompuScope system. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent down to the driver, as required by the C API.) If an invalid channel number is specified, an appropriate error code is returned.

Mode
This integer value determines the current transfer mode. Currently, the only valid value for this VI is 0.

Segment
This integer value determines which Multiple Record segment is to be transferred, if the CompuScope hardware was operating in Multiple Record Mode. For Single Record acquisitions, the user must always set the Segment to 1, which is the default value. If the Segment value is too large, an error will occur.

Start
This double variable value determines the requested starting point of the data transfer from the CompuScope memory to PC RAM. Start is specified relative to the trigger address for the acquisition. A Start value of 0 will cause data transfer to begin at the trigger address. Negative

Start values are for the transfer of pre-trigger data. Positive Start values may be used to begin transfer after the trigger address. The default Start value is 0.

Length
This double variable value specifies the amount of data, in samples, to be transferred. The default is 4096.

Return Value Descriptions:

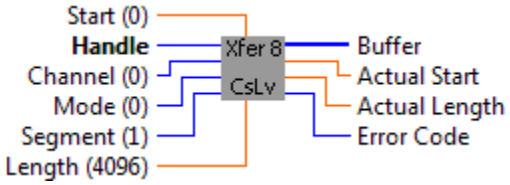
Buffer
This is a uInt8 LabVIEW buffer variable into which transferred data are returned. The size of the array will be at least Length samples. Data are returned as raw ADC values.

Actual Start
This double variable value returns the actual start address point of the data array. The Actual Start may be lower than the requested Start value. This is because the CompuScope memory architecture may force the drivers to download data from an earlier point than the start position. For example, if Start is input to CsLv_Transfer8.vi as 0, the Actual Start may return as -2, indicating that two extra samples were returned before the Start position.

The user must account for the discrepancy between the Start and the Actual Start. The user may choose to ignore samples transferred from before the Start position. Alternatively, the user may conserve these samples, accounting for the fact that the values begin at Actual Start.

Actual Length
This double variable value returns the actual amount of data transferred, which may be different from the requested Length. The difference, if any, results from CompuScope memory architecture.

Connector Pane



Controls and Indicators

Handle

Channel (0)

Segment (1)

Start (0)

Length (4096)

Mode (0)

Buffer

Numeric

Actual Length

 **Error Code**

 **Actual Start**

CsLv_Transfer8Ex.vi
This VI transfers 8-bit data from the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. Segment Count transfers for either all channels or 1 channel of the CompuScope system can be transferred.

This VI should only be used for transferring data from 8 bit CompuScope cards. Data are transferred from on-board CompuScope memory to PC RAM using PCI Bus Mastering at rates of up to 200 MB/s.

CsLv_Transfer8Ex.vi transfers raw integer ADC code values. The VI will not return programmatic control to LabVIEW until all of the data have been transferred.

If the call is successful, a 1 is returned in the Error Code indicator and the transferred data will be available in Buffer. If the call is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by using CsLv_GetErrorString.vi.

Input Value Descriptions:

Handle
An integer value which uniquely describes a CompuScope system. The value is obtained by calling CsLv_GetSystem.vi

Channel (default=-1)
This integer value specifies the channel(s) from which data are to be transferred. The default value is -1 for all the channels in the CompuScope system. Channel numbers in a LabVIEW CompuScope system begin at 0. (The channel numbers are incremented by 1 before being sent down to the driver, as required by the C API.) This vi can return either all the channels in the system or 1 channel in the system. If an invalid channel number is specified, an appropriate error code is returned.

Start Segment (default=1)
This integer value determines which Multiple Record segment to start transferring from, if the CompuScope hardware was operating in Multiple Record Mode. For Single Record acquisitions, the user must always set the Segment to 1, which is the default value. If the Segment value is too large, an error will occur.

Segment Count (default=1)
The number of segments to transfer, starting from Start Segment. If an invalid value is given for Segment Count, an error will be returned.

Start
This double variable value determines the requested starting point of the data transfer from the CompuScope memory to PC RAM. Start is specified relative to the trigger address for the acquisition. A Start value of 0 will cause data transfer to begin at the trigger address. Negative Start values are for the transfer of pre-trigger data. Positive Start values may be used to begin transfer after the trigger address. The default Start value is 0.

Length

This double variable value specifies the amount of data, in samples, to be transferred. The default is 4096.

Buffer In
A 1 dimension uInt8 array. The size of the array must be Channels * Segment Count * Length. The user is responsible for ensuring that the buffer is at least the proper length.

Return Value Descriptions:

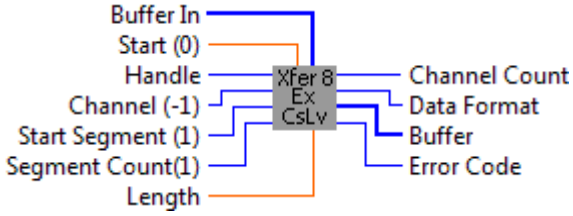
Buffer
This is the Buffer In variable filled with raw ADC values. A 3-dimensional array of Channels * Segment Count * Length can be obtained by calling CsLv_Extract8BitEx.vi.

Channel Count
The actual number of channels contained inl the buffer.

Data Format
A constant that indicates the order of the data for different channels. For example if the Data Format is 0x11221122 , the 2 channel data will be formated as 2 samples of Channel 1, 2 samples of Channel 2, 2 sample of Channel 1, etc.

Error Code
An integer that represents the return code of the call. A positive value indicates success and a negative value indicates failure. A descriptive error string can be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Channel (-1)

Start Segment (1)

Start (0)

Length

Segment Count(1)

Buffer In

Numeric

Buffer

Numeric

Error Code

Data Format

Channel Count

CsLv_TransferTimeStamp.vi
This VI transfers time stamp data from the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. Time stamps may be retrieved for any segment of a Multiple Record acquisition. The VI will not return programmatic control to LabVIEW until all of the transfer is complete. Note that not all CompuScope models support Time Stamping.

If the VI is successful, a 1 is returned in the Error Code indicator. If the VI is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

Input Value Descriptions:

Segment
This integer value selects the segment from which transfer of time stamp values will begin. The default value is 1. If you try to transfer time stamp values for an invalid segment (e.g. if it is less than 1 or greater than the number of segments acquired), an error code is returned.

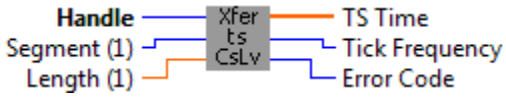
Length
This double variable value sets the number of time stamp values to be transferred, starting from the segment specified by Segment. The default value is 1. If Length is invalid (e.g. if it is less than 1 or greater than the number of segments acquired), an error code is returned.

Return Value Descriptions:

TS Time
This array of double variable values contains the time stamp readings in microseconds. Time stamp values indicate the time that has elapsed, in microseconds, between the trigger time stamp for the corresponding segment and the last reset of the time stamping counter.

Tick Frequency
This integer value indicates the time stamping counter frequency, in Hertz, that was used for the acquisition of the returned time stamps. The inverse of this frequency indicates the time for one time stamp count and is therefore the minimum possible time between two time stamp values.

Connector Pane



Controls and Indicators

Handle

Segment (1)

Length (1)

TS Time

Numeric

Error Code

Tick Frequency

CsLv_TransferTimeStampEx.vi
This VI transfers time stamp data from the CompuScope system identified by Handle, which is obtained by using CsLv_GetSystem.vi. Time stamps may be retrieved for any segment or all segments of a Multiple Record acquisition. The VI will not return programmatic control to LabVIEW until all of the transfer is complete. Note that not all CompuScope models support Time Stamping.

If the VI is successful, a 1 is returned in the Error Code indicator. If the VI is unsuccessful, an appropriate error code is returned. A descriptive error string may be obtained by calling CsLv_GetErrorString.vi.

Input Value Descriptions:

TS Values In
An Segment Count size array of doubles. The size of the array must be the same as the Segment Count that is to be transferred. This array is used as the TS Values Out array.

Start Segment
This integer value selects the segment from which to start transferring the time stamp values for. The default value is 1. If you try to transfer time stamp values for an invalid segment (e.g. if it is less than 1 or greater than the number of segments acquired), an error code is returned.

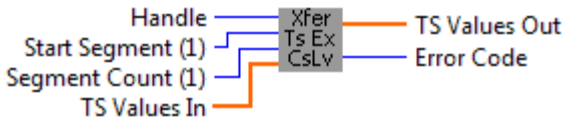
SegmentCount
This integer value determines the number of segments to transfer the time stamp data for. If the value is greater than the actual number of segments acquired, an error code is returned.

Return Value Descriptions:

TS Values Out
This array of double variable values contains the time stamp readings in microseconds. Time stamp values indicate the time that has elapsed, in microseconds, between the trigger time stamp for the corresponding segment and the last reset of the time stamping counter.

Error Code
An integer that represents the return code of the call. A positive value indicates success and a negative value indicates failure. A descriptive error string can be obtained by calling CsLv_GetErrorString.vi.

Connector Pane



Controls and Indicators

Handle

Start Segment (1)

Segment Count (1)

TS Values In

Numeric

Error Code

TS Values Out

Numeric