



NI-IMAQ for IEEE 1394 Cameras VI Reference Help

March 2005 Edition, Part Number 370357D-01

NI-IMAQ for IEEE 1394 Cameras driver software gives you the ability to acquire images with industrial digital video cameras. This help file describes the VIs included in NI-IMAQ for IEEE 1394 Cameras.

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window.

For more information about this help file, refer to the following topics:

[Conventions](#)—formatting and typographical conventions in this help file

[Related Documentation](#)

[Important Information](#)



[Technical Support and Professional Services](#)

To comment on National Instruments documentation, refer to the [National Instruments Web site](#).

© 2001–2005 National Instruments Corporation. All rights reserved.

Conventions

This help file uses the following conventions:

- [] Square brackets enclose optional items—for example, [response].
- » The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
-  This icon denotes a tip, which alerts you to advisory information.
-  This icon denotes a note, which alerts you to important information.
- bold** Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names, emphasis, or an introduction to a key concept.
- green Underlined text in this color denotes a link to a help topic, help file, or Web address.
- italic* Italic text denotes variables or cross references. This font also denotes text that is a placeholder for a word or value that you must supply.
- monospace Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

Related Documentation

The following documents contain information you might find helpful as you use this help file:

- *NI-IMAQ for IEEE 1394 Cameras User Manual*
- *Getting Started with NI-IMAQ for IEEE 1394 Cameras*
- NI-IMAQ for IEEE 1394 Cameras Examples, located in the LabVIEW\Examples\IMAQ directory

NI-IMAQ for IEEE 1394 Cameras VIs

[Alphabetical](#)

[High-Level VIs](#)

[Low-Level VIs](#)

NI-IMAQ for IEEE 1394 Cameras VIs

[IMAQ 1394 Clear Acquisition](#)

[IMAQ 1394 Close](#)

[IMAQ 1394 Configure Acquisition](#)

[IMAQ 1394 Configure Trigger](#)

[IMAQ 1394 Get Image](#)

[IMAQ 1394 Get Image Data](#)

[IMAQ 1394 Get Interface Files](#)

[IMAQ 1394 Get Video Modes](#)

[IMAQ 1394 Grab Acquire](#)

[IMAQ 1394 Grab Setup](#)

[IMAQ 1394 Init](#)

[IMAQ 1394 Property Node](#)

[IMAQ 1394 Occurrence Config](#)

[IMAQ 1394 Read Registers](#)

[IMAQ 1394 Sequence](#)

[IMAQ 1394 Snap](#)

[IMAQ 1394 Start Acquisition](#)

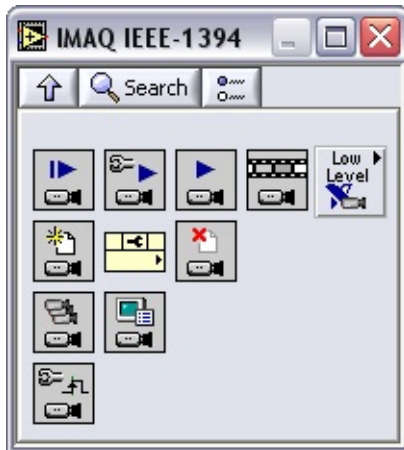
[IMAQ 1394 Stop Acquisition](#)

[IMAQ 1394 Write Registers](#)

High-Level VIs

Use high-level VIs to set up your IMAQ system and acquire images. The high-level NI-IMAQ for IEEE 1394 VIs allow you to acquire images, open and close an interface, get/set attributes, get camera features and video modes, and configure triggered acquisitions.

Click the icons for VI descriptions.



[IMAQ1394 Snap](#)

[IMAQ1394 Grab Setup](#)

[IMAQ1394 Grab Acquire](#)

[IMAQ1394 Sequence](#)

[IMAQ1394 Init](#)

[IMAQ1394 Close](#)

[IMAQ1394 Get Interface Files](#)

[IMAQ1394 GetVideoModes](#)

[IMAQ1394 Configure Trigger](#)

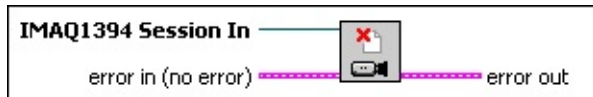
[IMAQ1394 Property Node](#)

IMAQ1394 Close

Stops an acquisition in progress, releases resources associated with the acquisition, and closes the specified IMAQ1394 Session.



Note This VI executes regardless of incoming errors. Any error generated by this VI is merged with the incoming status.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.




source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.




error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

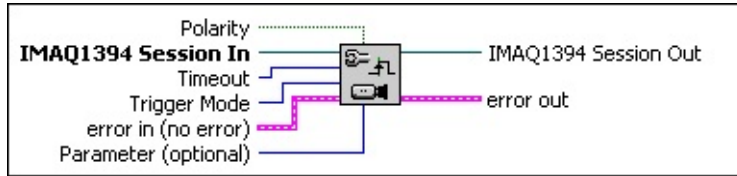
 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Configure Trigger

Configures the trigger conditions for an acquisition. Use this VI before any configuration or acquisition VI to set up a triggered image acquisition.



Polarity specifies polarity for the trigger line. FALSE sets the polarity to the falling edge. TRUE sets the polarity to the rising edge.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



Timeout specifies the time, in milliseconds, allowed for the image acquisition to complete. If the acquisition cannot complete in time, an error returns.



Trigger Mode specifies the trigger mode. The following descriptions are valid for low-active polarity. All states are inverted for high-active polarity.

Disabled—No trigger. The camera starts integration immediately.

Mode 0—The camera starts integration on every falling edge of the external trigger input. Integration time is defined in the **Shutter** attribute.

Mode 1—The camera starts integration on every falling edge of the external trigger input. Integration time is equal to the low state of the external trigger.

Mode 2—The camera starts integration on the first falling edge of the external trigger input. At the N th external trigger input falling edge, integration stops. N is defined in **Parameter (optional)**.

Mode 3—This is the internal trigger mode. The camera issues a trigger internally. The cycle time is N times the cycle time of fastest frame rate. N is defined in **Parameter (optional)**. The integration time is defined in the **Shutter** attribute.

Mode 4—The camera starts frame integration when the external trigger input changes to an active value. Each frame is exposed for a duration specified in the **Shutter** attribute. The number of frames is specified in **Parameter (optional)**, which must have a value of 1 or more.

Mode 5—The camera starts frame integration when the external trigger input changes to an active value. Each frame is exposed while the external trigger is active. The number of frames is specified in **Parameter (optional)**, which must have a value of 1 or more.



Note Some cameras may not support every trigger mode.





error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.




status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a


nonzero error code. If **status** is FALSE, **code** is zero or a warning code.


 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.


 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.


 **Parameter (optional)** defines *N* in trigger modes 2 and 3.

 **IMAQ1394 Session Out** is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.

 **error out** is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

 **status** is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Get Interface Files

Returns a list of all interface files on the host computer.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.

Interface File Array is an array of interface files that are on the host computer. This includes cameras that are currently connected or that have been connected in the past.


Type has a value of **3**. This designates an NI-IMAQ IEEE 1394 interface file.

Version is the version of the interface file. This number may increment with different versions of the driver as the format of the interface file changes.


Flags is a bitwise mask of the current interface status. If bit 0 is on (value = 1), the interface represents a camera that is currently connected. If bit 0 is off (value = 0), the interface represents a disconnected camera.


SerialNumberHi is the upper 32 bits of the interface serial


number. Every camera has a unique value for **SerialNumberHi**.


 **SerialNumberLo** is the lower 32 bits of the interface serial number. Every camera has a unique value for **SerialNumberLo**.


 **InterfaceName** is the name of the interface. Use this name when opening the interface.


 **VendorName** is the vendor name of the camera designated for this interface. **VendorName** varies from camera to camera.


 **ModelName** is the model name of the camera designated for this interface. **ModelName** varies from camera to camera.

 **CameraFileName** is the name of the camera file that this interface uses. The camera file contains all the settings for a given camera. You can configure and save these settings from Measurement & Automation Explorer (MAX).

 **error out** is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

 **status** is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

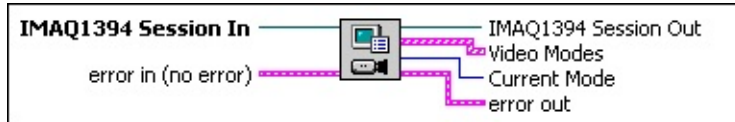
 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.













Note Refer to [Error Handling](#) for error input and output information and error codes.


IMAQ1394 GetVideoModes

Returns a list of video formats, modes, and frame rates supported by the camera.





-  **IMAQ1394 Session In** is a unique reference to the camera, which you can obtain with IMAQ1394 Init.
-  **error in (no error)** is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.
-  **status** is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.
-  **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.
-  **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.
-  **IMAQ1394 Session Out** is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.
-  **Video Modes** is an array of video modes supported by the current camera.
 -  **Format** is the format of the camera as defined by the IIDC specification. This parameter varies according to your camera.
 -  **Mode** contains the image size and type as defined by the IIDC specification. **Mode** varies according to your camera.
 -  **FrameRate** is the rate at which your camera acquires


frames. This rate varies according to your camera.


 **VideoMode Name** is the name of the video mode specified by **Format**, **Mode**, and **FrameRate**, such as "640 x 480 YUV4:2.2".


 **Current Mode** is the index into the **Video Modes** array of the current mode used by the camera.

 **error out** is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

 **status** is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

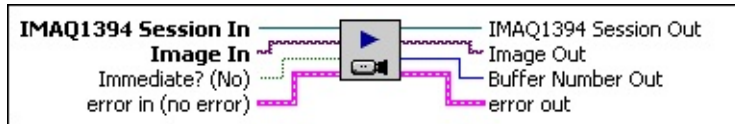
 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.











 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.

 **Note** Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Grab Acquire

Acquires the most current frame into **Image Out**. Call this VI only after calling [IMAQ1394 Grab Setup](#). If the image type does not match the video format of the camera, this VI changes the image type to a suitable format.



-  **IMAQ1394 Session In** is a unique reference to the camera, which you can obtain with IMAQ1394 Init.
-  **Image In** is the reference to the image that receives the captured pixel data.
-  **Immediate?** specifies whether the VI returns the image currently being acquired or the last completely acquired image. FALSE causes the driver to wait until the current image is completely acquired before returning it. TRUE returns the last acquired image. The default value is FALSE.
-  **error in (no error)** is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.
 -  **status** is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.
 -  **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.
 -  **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.
-  **IMAQ1394 Session Out** is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.
-  **Image Out** is the reference to the captured image.
-  **Buffer Number Out** is the buffer number of the image returned.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



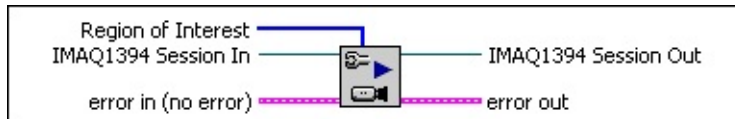
source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Grab Setup

Configures and starts a grab acquisition. A grab performs an acquisition that loops continually on three buffers. Use the grab function for high-speed image acquisition. Use [IMAQ1394 Grab Acquire](#) to copy an image out of the buffer ring. If you call this VI before calling [IMAQ1394 Init](#), IMAQ1394 Grab Setup uses cam0 by default. Use [IMAQ1394 Clear Acquisition](#) to unconfigure the acquisition.



[U32] **Region of Interest** specifies a rectangular portion of the image to be transferred into LabVIEW memory. This parameter is defined by an array of four elements: [Left, Top, Right, Bottom].



Note If your camera supports Partial Image Size Format (Format 7), you can use that setting to change the size of the image transferred over the 1394 bus. These values are coerced to the next highest multiple of the unit width or height supported by your camera.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error

occurred.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



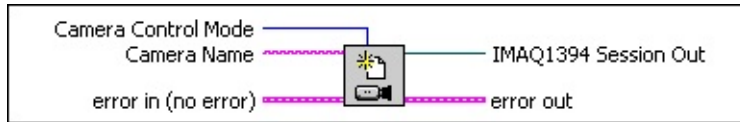
source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Init

Opens a camera, queries the camera for its capabilities, loads a camera configuration file, and creates a unique reference to the camera. Use [IMAQ1394 Close](#) when you are finished with the reference.



Camera Control Mode is the control mode of the camera used during image broadcasting. Open a camera in controller mode to actively configure and acquire image data. Open a camera in listener mode on a different host or target computer to passively acquire image data from a session that was opened in controller mode. The default value is **Controller**.

Camera Name is the name of the camera you want to open. The name (cam0, cam1,...cam N) must match the configuration file name you used to configure the camera in Measurement & Automation Explorer (MAX). You can also open a camera using its 64-bit serial number (uuid:XXXXXXXXXXXXXXXX), where the number following uuid must be a 64-bit hexadecimal number representing the internal serial number of the camera.

Note Specify "uuid:serial number in hexadecimal representation" for the camera name when opening in listening mode. The serial number must match the serial number used in MAX.

error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.

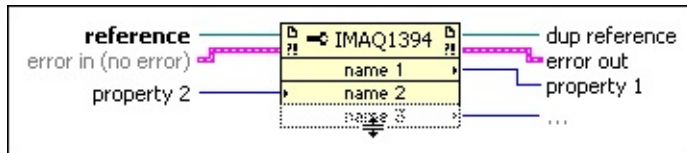


Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Property Node

Gets (reads) and/or sets (writes) properties of a reference. The Property Node automatically adapts to the class of the object that you **reference**.

Details



- reference** is the refnum associated with an IMAQ1394 session.
- error in** describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in **error out**. Use the [Simple Error Handler](#) or [General Error Handler](#) VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.
- status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.
- source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- property 2** is an example of a property you want to set (write).
- dup reference** returns **reference** unchanged.
- error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or

function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code** is the error or warning code. If **status** is TRUE, **code** is a nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.
- source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

property 1 is an example of a property you want to get (read).

Property Node Details

The node adapts to the class automatically.

Move the cursor over terminals in the Property Node to display more information about the property in the [Context Help](#) window. You also can right-click a property terminal and select **Help For Property** from the shortcut menu, where **Property** is the name of the property.

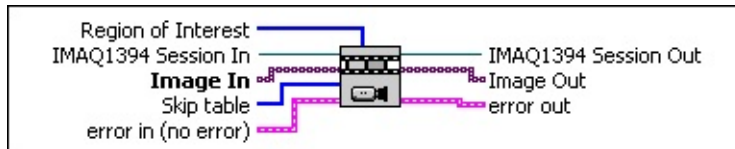
To get property information, right-click the node and select **Change to Read** from the shortcut menu. To set property information, right-click the node and select **Change to Write** from the shortcut menu. If a property is read only, **Change to Write** is dimmed in the shortcut menu. The node executes each terminal in order from top to bottom. If an error occurs on a terminal, the node stops at that terminal, returns an error, and does not execute any further terminals. You can right-click the node and select **Ignore Errors Inside Node** from the shortcut menu to ignore any errors and continue executing further terminals. The **error out** cluster reports which property caused the error.

If the small direction arrow on the property is on the right, you are getting the property value. If the small direction arrow on a property is on the left, you are setting the property value. Properties have a short or long name that you can change by right-clicking and selecting **Name Format** from

the shortcut menu. The **No Names** format displays only the data type for each property.

IMAQ1394 Sequence

Configures, starts, acquires, stops and unconfigures a sequence acquisition. Use this VI to capture multiple images. If you call this VI before calling [IMAQ1394 Init](#), IMAQ1394 Sequence uses cam0 by default.



[U32] **Region of Interest** specifies a rectangular portion of the image to be transferred into LabVIEW memory. This parameter is defined by an array of four elements: [Left, Top, Right, Bottom].



Note If your camera supports Partial Image Size Format (Format 7), you can use that setting to change the size of the image transferred over the 1394 bus. These values are coerced to the next highest multiple of the unit width or height supported by your camera.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



Image In is an array of image references that receives the captured pixel data.



Skip table is reserved for future use.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



Image Out is the array of references to the captured images.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



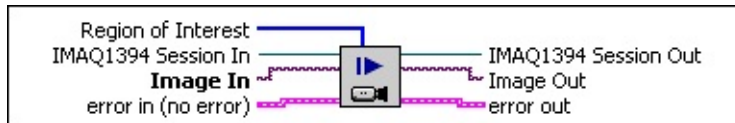
source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Snap

Configures, starts, acquires, and unconfigures a snap acquisition. Use a snap for low-speed or single-capture applications where ease of programming is essential. If you call this VI before calling [IMAQ1394 Init](#), IMAQ1394 Snap uses cam0 by default. If the image type does not match the video format of the camera, this VI changes the image type to a suitable format.



[U32] **Region of Interest** specifies a rectangular portion of the image to be transferred into LabVIEW memory. This parameter is defined by an array of four elements: [Left, Top, Right, Bottom].



Note If your camera supports Partial Image Size Format (Format 7), you can use that setting to change the size of the image transferred over the 1394 bus. These values are coerced to the next highest multiple of the unit width or height supported by your camera.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



Image In is the reference to the image that receives the captured pixel data.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any.

Typically, **source** is the name of the VI in which the error occurred.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



Image Out is the reference to the captured image.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.

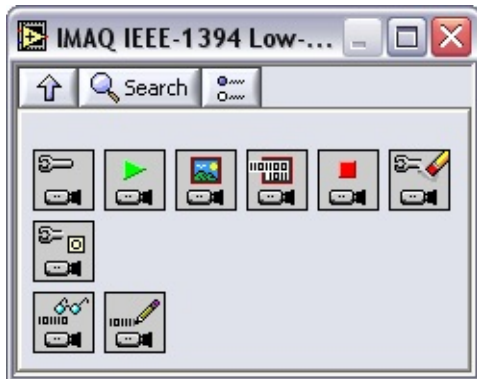


Note Refer to [Error Handling](#) for error input and output information and error codes.

Low-Level VIs

Use low-level NI-IMAQ VIs for more direct control of the IMAQ hardware.

Click the icons for VI descriptions.



[IMAQ1394 Configure Acquisition](#)

[IMAQ1394 Start Acquisition](#)

[IMAQ1394 Get Image](#)

[IMAQ1394 Get Image Data](#)

[IMAQ1394 Stop Acquisition](#)

[IMAQ1394 Clear Acquisition](#)

[IMAQ1394 Occurrence Config](#)

[IMAQ1394 Read Registers](#)

[IMAQ1394 Write Registers](#)

IMAQ1394 Clear Acquisition

Unconfigures an acquisition previously configured with [IMAQ1394 Configure Acquisition](#).



Note This VI executes regardless of incoming errors. Any error generated by this VI is merged with the incoming status.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.





error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is

FALSE, **code** is zero or a warning code.

 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Configure Acquisition

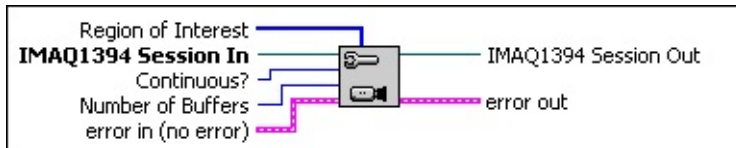
Configures a low-level acquisition previously opened with [IMAQ1394 Init](#). Specify the acquisition type with the **Continuous?** and **Number of buffers** parameters.

| | | |
|----------|----------------|-----------------------------|
| Snap | Continuous = 0 | Buffer Count = 1 |
| Sequence | Continuous = 0 | Buffer Count > 1 |
| Grab | Continuous = 1 | Buffer Count ³ 1 |



Note National Instruments recommends using three or more buffers for continuous acquisitions.

Use [IMAQ1394 Clear Acquisition](#) to unconfigure the acquisition.



Region of Interest specifies a rectangular portion of the image to be transferred into LabVIEW memory. This parameter is defined by an array of four elements: [Left, Top, Right, Bottom].



Note If your camera supports Partial Image Size Format (Format 7), you can use that setting to change the size of the image transferred over the 1394 bus. These values are coerced to the next highest multiple of the unit width or height supported by your camera.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



Continuous? specifies whether the acquisition is continuous or one-shot.



Number of Buffers For a one-shot acquisition, this parameter specifies the number of images to acquire. For a continuous acquisition, this parameter specifies the number of buffers the driver uses internally.




error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.





status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.





code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.


 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.

 **IMAQ1394 Session Out** is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.

 **error out** is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

 **status** is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

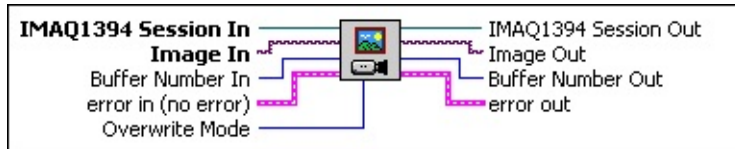
 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Get Image

Acquires the specified frame into **Image Out**. Call this VI only after calling [IMAQ1394 Configure Acquisition](#) and [IMAQ1394 Start Acquisition](#). If the type does not match the video format of the camera, this VI changes the image type to a suitable format.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



Image In is the reference to the image that receives the captured pixel data.



Buffer Number In is the cumulative image number to get. A value of -1 gets the most recently acquired buffer. A value of -2 gets the last acquired buffer.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Overwrite Mode is the overwrite policy to follow if an image is overwritten during acquisition. Specify **Get Oldest** to get the oldest valid buffer instead of the overwritten buffer. Specify **Fail** to return an error if the requested buffer is overwritten. Specify **Get Newest** to get the most recent valid buffer instead of the overwritten buffer.



Note The **Get Next Iteration** policy is not implemented by

NI-IMAQ for IEEE 1394 Cameras and is presented only to keep the API consistent with NI-IMAQ 3.x.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



Image Out is the reference to the captured image.



Buffer Number Out is the actual acquired buffer number returned.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



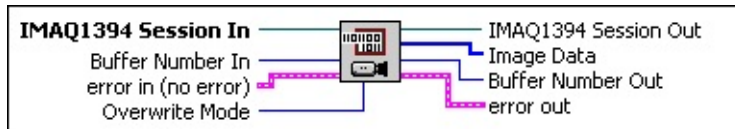
Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Get Image Data

Copies the raw data of the specified frame into **Image Data**. Call this VI only after calling [IMAQ1394 Configure Acquisition](#) and [IMAQ1394 Start Acquisition](#).



Note This VI allows you to access raw image data. For many compressed formats like YUV, **Image Data** is not compatible with IMAQ Vision functions. To use the IMAQ Vision functions, use [IMAQ1394 Get Image](#) instead of this VI.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



Buffer Number In is the cumulative image number to get. A value of -1 gets the most recently acquired buffer. A value of -2 gets the last acquired buffer.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Overwrite Mode is the overwrite policy to follow if an image is overwritten during acquisition. Specify **Get Oldest** to get the oldest valid buffer instead of the overwritten buffer. Specify **Fail** to return an error if the requested buffer is overwritten. Specify **Get Newest** to get the most recent valid buffer instead of the overwritten buffer.



Note The **Get Next Iteration** policy is not implemented by

NI-IMAQ for IEEE 1394 Cameras and is presented only to keep the API consistent with NI-IMAQ 3.x.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



Image Data is a copy of the raw image data for the current image. The data is returned as a 1D array, where each byte represents a byte of data.



Buffer Number Out is the actual acquired buffer number returned.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



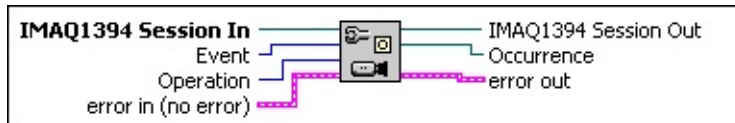
source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.









Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Occurrence Config

Creates occurrences that are set or triggered when the **Frame Done**, **Camera Attached**, or **Camera Detached** event occurs. Occurrences produced by IMAQ1394 Occurrence Config are used as inputs to the **Wait on Occurrence** LabVIEW primitive. Functions dependent on this primitive sleep until the occurrence is set or triggered. Use this VI only with low-level acquisition VIs.



-  **IMAQ1394 Session In** is a unique reference to the camera, which you can obtain with IMAQ1394 Init.
- Event** is the event that creates the occurrence. Specify **Frame Done** to receive an occurrence after the next full frame arrives. Specify **Camera Attached** to monitor when a new camera is added to the system. Specify **Camera Detached** to monitor when an existing camera is removed from the system.
- Operation** instructs the VI to create an occurrence or to clear all occurrences that have been created for the specified IMAQ1394 session.
-  **error in (no error)** is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.
 -  **status** is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.
 -  **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.
 -  **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.
-  **IMAQ1394 Session Out** is a unique reference to the camera. IMAQ1394 Session Out is the

same as IMAQ1394 Session In.



Occurrence generates an occurrence when the specified event occurs. Wire this output to the Wait on Occurrence VI.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



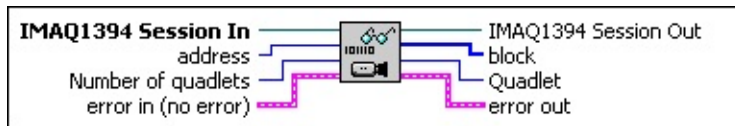
source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Read Registers

Accesses registers on the camera and reads an array of contiguous 32-bit quadlets from the camera. Data is byte-swapped for little endian alignment after transfer.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



address is the register location to access. Refer to the camera documentation for more information about camera-specific register ranges.



Note Use the Property Node **Properties»Camera Information»Base Address** to get the base address for your camera.



Number of quadlets is the number of 32-bit quadlets to read. The default is 1.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



block is an array of 32-bit quadlets returned from the camera.



Quadlet is the first element of **block**. This output is convenient when performing single-quadlet register reads.



error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



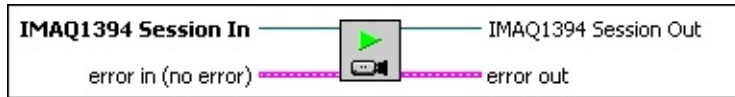
source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Start Acquisition

Starts an acquisition that was previously configured with [IMAQ1394 Configure Acquisition](#). Use [IMAQ1394 Stop Acquisition](#) to stop the acquisition.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.




IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.




error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



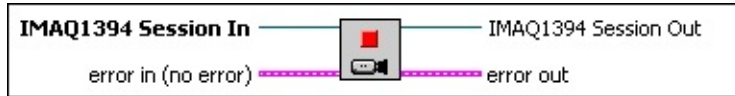
Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Stop Acquisition

Stops an acquisition previously started with [IMAQ1394 Start Acquisition](#).



Note This VI executes regardless of incoming errors. Any error generated by this VI is merged with the incoming status.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.




IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.




error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.

 **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.

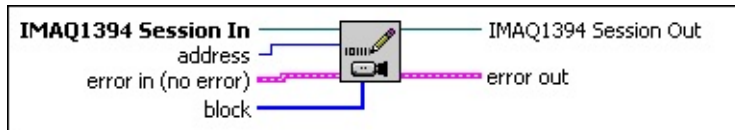
 **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Write Registers

Accesses registers on the camera and writes an array of contiguous 32-bit quadlets to the camera. Data is byte-swapped for big endian alignment before transfer.



IMAQ1394 Session In is a unique reference to the camera, which you can obtain with IMAQ1394 Init.



address is the register location to access. Refer to the camera documentation for more information camera-specific register ranges.



Note Use the Property Node **Properties»Camera Information»Base Address** to get the base address for your camera.



error in (no error) is a cluster that describes the error status before this VI executes. If **error in** indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its **error out** cluster. If no error has occurred, then this VI executes normally and sets its own error status in **error out**. Use the error handler VIs to look up the error code and to display the corresponding error message. Using the **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



block is an array of 32-bit quadlets to write to the camera.



IMAQ1394 Session Out is a unique reference to the camera. IMAQ1394 Session Out is the same as IMAQ1394 Session In.



error out is a cluster that describes the error status after this VI executes. If an error occurred

before this VI was called, **error out** is the same as **error in**. Otherwise, **error out** shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using **error in** and **error out** clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status is TRUE if an error occurred, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.



code is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.



source is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.



Note Refer to [Error Handling](#) for error input and output information and error codes.

IMAQ1394 Properties Overview

Refer to the outline below for a list of properties that are available for the NI-IMAQ for IEEE 1394 Reference.

| | |
|--------------------------------|---------------------------|
| Acquisition Attributes | |
| Bayer | |
| Bits Per Pixel | Available Color Filter |
| Bytes Per Pixel | Color Filter |
| Frame Interval | Gain B |
| Ignore First Frame | Gain G |
| Image Height | Gain R |
| Image Representation | |
| Image Width | |
| Partial Scan | |
| Region of Interest Height | Acquisition Window Height |
| Region of Interest Left | Acquisition Window Left |
| Region of Interest Top | Acquisition Window Top |
| Region of Interest Width | Acquisition Window Width |
| Shift Pixel Bits | Bytes Per Packet |
| Swap Pixel Bytes | Color Coding |
| Timeout | Max Bytes Per Packet |
| Video Format | Max Speed |
| Video Mode | Speed |
| Video Rate | Unit Bytes Per Packet |
| | Unit Height |
| | Unit Width |
| Camera Attributes | |
| Attribute | |
| Inquiry | Absolute |
| | Auto |
| | Manual |
| | Name |
| | Off |
| | One Push |
| | Present |
| | Readable |
| Mode | |
| Range | Maximum |
| | Minimum |
| Value | |
| Camera Information | |
| Base Address | |
| Model Name | |
| Unique ID High | |
| Unique ID Low | |
| Vendor Name | |
| Status Information | |
| Acquisition In Progress | |
| Last Transferred Buffer Number | |
| Lost Buffer Count | |
| Transferred Frame count | |

NI-IMAQ for IEEE 1394 Properties

Refer to the class name below for the properties associated with each NI-IMAQ for IEEE 1394 Cameras class. Refer to the *NI-IMAQ for IEEE 1394 Cameras Function Reference Help* for more information about NI-IMAQ for IEEE 1394 Cameras properties and attributes.

The following table describes the NI-IMAQ for IEEE 1394 Cameras properties.

| Property | Long Name | Data Type and Range | Access Privilege | Description |
|-------------|---|-----------------------------------|------------------|---|
| VendorName | Camera Information:Vendor Name | <input type="text" value="abcd"/> | Read Only | Returns the vendor name. |
| ModelName | Camera Information:Model Name | <input type="text" value="abcd"/> | Read Only | Returns the model number. |
| VideoFormat | Acquisition Attributes:Video Format | <input type="checkbox"/> | Read/Write | Gets/sets the video format. Must be a value between 0 and 7. |
| VideoMode | Acquisition Attributes:Video Mode | <input type="checkbox"/> | Read/Write | Gets/sets the video mode. Must be a value between 0 and 7. |
| VideoRate | Acquisition Attributes:Video Rate | <input type="checkbox"/> | Read/Write | Gets/sets the video frame rate. Must be a value between 0 and 7. |
| ImageRep | Acquisition Attributes:Image Representation | <input type="checkbox"/> | Read Only | Gets the image representation for the acquisition. Default=0 Raw=1 Mono 8=2 Mono 16=3 RGB 32=4 RGB 64=5 |
| Timeout | Acquisition Attributes:Timeout | <input type="checkbox"/> | Read/Write | Gets/sets the timeout value in milliseconds, used to abort an acquisition when the image transfer cannot be completed within the delay. |

| | | | | |
|-------------------|---|--------------------------|-----------|---|
| Format7UnitWidth | Acquisition Attributes:Partial Scan:Unit Width | <input type="checkbox"/> | Read Only | Gets the minimum width of the acquisition window. Valid only if the camera supports Partial Image Size Format (Format 7). When defining a region of interest in Format 7, the width of the ROI must be a multiple of UnitWidth . |
| Format7UnitHeight | Acquisition Attributes:Partial Scan:Unit Height | <input type="checkbox"/> | Read Only | Gets the minimum height of the acquisition window. Valid only if the camera supports Partial Image Size Format (Format 7). When defining a region of interest in Format 7, the width of the ROI must be a multiple of UnitHeight . |
| UniqueIDLow | Camera Information: Unique ID Low | <input type="checkbox"/> | Read Only | Returns the low part of the 64-bit unique node ID of the IEEE 1394 camera associated with this session. |
| UniqueIDHi | Camera Information: Unique ID High | <input type="checkbox"/> | Read Only | Returns the low part of the 64-bit unique node ID of the IEEE 1394 camera associated with this session. |
| LostBufferCount | StatusInformation:LostBufferCount | <input type="checkbox"/> | Read Only | Gets the number of lost buffers during an acquisition session. Lost buffers occur when no internal |







| | | | | |
|--------------------|---|--------------------------|------------|---|
| | | | | buffers are available when the camera is streaming video data. |
| Format7Left | Acquisition Attributes: Partial Scan: Acquisition Window Left | <input type="checkbox"/> | Read/Write | Gets/sets the left edge of the acquisition window. Valid only if the camera supports Partial Image Size Format (Format 7). |
| Format7Top | Acquisition Attributes: Partial Scan: Acquisition Window Top | <input type="checkbox"/> | Read/Write | Gets/sets the top edge of the acquisition window. Valid only if the camera supports Partial Image Size Format (Format 7). |
| Format7Width | Acquisition Attributes: Partial Scan: Acquisition Window Width | <input type="checkbox"/> | Read/Write | Gets/sets the width of the acquisition window. Valid only if the camera supports Partial Image Size Format (Format 7). |
| Format7Height | Acquisition Attributes: Partial Scan: Acquisition Window Height | <input type="checkbox"/> | Read/Write | Gets/sets the height of the acquisition window. Valid only if the camera supports Partial Image Size Format (Format 7). |
| Format7ColorCoding | Acquisition Attributes: Partial Scan: Color Coding | <input type="checkbox"/> | Read/Write | Gets/sets the color coding used for the current video format/mode if the camera supports Partial Image Size Format (Format 7). If the camera supports only standard video modes, this |






| | | | | |
|-----------------------|--|--------------------------|------------|--|
| | | | | property is Read Only. Default=0 Mono 8=1 YUV 411=2 YUV 422=3 YUV 444=4 RGB 8=5 Mono 16=6 RGB 16=7 Signed Mono 16=8 Signed Mono RGB=9 Raw 8=10 Raw 16=11 |
| ImageWidth | Acquisition Attributes: Image Width | <input type="checkbox"/> | Read Only | Gets the maximum width of the acquisition window. |
| ImageHeight | Acquisition Attributes: Image Height | <input type="checkbox"/> | Read Only | Gets the maximum height of the acquisition window. |
| BytesPerPixel | Acquisition Attributes: Bytes Per Pixel | <input type="checkbox"/> | Read Only | Gets the number of bytes used for each pixel. |
| FrameInterval | Acquisition Attributes: Frame Interval | <input type="checkbox"/> | Read Only | Gets the expected duration of a frame acquisition, in milliseconds. |
| Format7BytesPerPacket | Acquisition Attributes: Partial Scan: Bytes Per Packet | <input type="checkbox"/> | Read/Write | Gets/sets the number of bytes transmitted per isochronous IEEE 1394 serial packet. Valid only if the camera supports Partial Image Size Format (Format 7). |
| ROILeft | Acquisition Attributes: Region of Interest Left | <input type="checkbox"/> | Read/Write | Gets/sets the left edge of the region of interest. |
| ROITop | Acquisition Attributes: Region of Interest Top | <input type="checkbox"/> | Read/Write | Gets/sets the top edge of the region of interest. |
| ROIWidth | Acquisition Attributes: Region of | <input type="checkbox"/> | Read/Write | Gets/sets the |

| | | | | |
|------------------|--|-------------------------------------|------------|---|
| | Interest Width | | | width of the region of interest. |
| ROIHeight | Acquisition Attributes: Region of Interest Height | <input type="checkbox"/> | Read/Write | Gets/sets the height of the region of interest. |
| Format7Speed | Acquisition Attributes: Partial Scan: Speed | <input type="checkbox"/> | Read/Write | Gets/sets the transmission speed of the isochronous IEEE 1394 serial packet. Valid only if the camera supports Partial Image Size Format (Format 7). Default=0 100 Mbps=1 200 Mbps=2 400 Mbps=3 800 Mbps=4 1600 Mbps=5 3200 Mbps=6 |
| LastBufferNumber | Status Information: Last Transferred Buffer Number | <input type="checkbox"/> | Read Only | Gets the last transferred cumulative buffer number. |
| FrameCount | Status Information: Transferred Frame Count | <input type="checkbox"/> | Read Only | Gets the number of frames transferred. |
| AcqInProgress | Status Information: Acquisition in Progress | <input checked="" type="checkbox"/> | Read Only | Gets the current state of the acquisition. |
| IgnoreFirstFrame | Acquisition Attributes: Ignore First Frame | <input checked="" type="checkbox"/> | Read/Write | Gets/sets the video delay of one frame interval between starting the camera and receiving the video feed. |
| ShiftPixelBits | Acquisition Attributes: Shift Pixel Bits | <input checked="" type="checkbox"/> | Read/Write | Gets/sets the alignment of 16-bit cameras. Downshift the pixel bits if the camera returns most significant bit-aligned data. |
| SwapPixelBytes | Acquisition Attributes: Swap Pixel Bytes | <input checked="" type="checkbox"/> | Read/Write | Gets/sets the endianness of |

| | | | | |
|---------------------------|---|--------------------------|------------|---|
| | | | | 16-bit cameras. Swap the pixel bytes if the camera returns little endian data. |
| Format7UnitBytesPerPacket | Acquisition Attributes: Partial Scan: Unit Bytes Per Packet | <input type="checkbox"/> | Read Only | Gets the minimum value of data transfer size. Valid only if the camera supports Partial Image Size Format (Format 7). When defining data transfer size in Format 7, the value of the size must be a multiple of UnitSize . |
| BitsPerPixel | Acquisition Attributes: Bits Per Pixel | <input type="checkbox"/> | Read/Write | Gets/sets the true bit depth of 16-bit cameras. Valid values are 10, 12, 14, and 16. |
| ColorFilterInq | Acquisition Attributes: Bayer: Available Color Filter | <input type="checkbox"/> | Read Only | Gets the recommended Bayer pattern to use for a Bayer sensor. None=0 GBGB RGRG=1 GRGR BGBG=2 BGBG GRGR=3 RGRG GBGB=4 |
| ColorFilter | Acquisition Attributes: Bayer: Color Filter | <input type="checkbox"/> | Read/Write | Gets/sets the Bayer pattern to use for a Bayer sensor. None=0 GBGB RGRG=1 GRGR BGBG=2 BGBG GRGR=3 RGRG GBGB=4 |
| ColorFilterGainR | Acquisition Attributes: Bayer: Gain R | <input type="checkbox"/> | Read/Write | Gets/sets the red gain coefficient for Bayer decoding. Valid values range |

| | | | | |
|------------------|---|-------------------------------------|------------|--|
| | | | | from 0 to 3.999. |
| ColorFilterGainG | Acquisition Attributes: Bayer: Gain G | <input checked="" type="checkbox"/> | Read/Write | Gets/sets the green gain coefficient for Bayer decoding. Valid values range from 0 to 3.999. |
| ColorFilterGainB | Acquisition Attributes: Bayer: Gain B | <input checked="" type="checkbox"/> | Read/Write | Gets/sets the blue gain coefficient for Bayer decoding. Valid values range from 0 to 3.999 |
| Format7MaxSpeed | Acquisition Attributes: Partial Scan: Max Speed | <input type="checkbox"/> | Read Only | Gets the maximum value of data transfer speed. Valid only if the camera supports Partial Image Size Format (Format 7). Default=0 100 Mbps=1 200 Mbps=2 400 Mbps=3 800 Mbps=4 1600 Mbps=5 3200 Mbps=6 |
| Attr | Camera Attributes: Attribute | <input type="checkbox"/> | Read/Write | Gets/sets the current camera attribute. Brightness=0 Auto Exposure=1 Sharpness=2 White Balance U=3 White Balance V=4 Hue=5 Saturation=6 Gamma=7 Shutter=8 Gain=9 Iris=10 Focus=11 Temperature=11 Zoom=13 Pan=14 |

| | | | | |
|----------|--------------------------------------|--|------------|---|
| | | | | Tilt=15 Optical Filter=16 Trigger Delay=75 White Shading R=76 White Shading G=77 White Shading B=78 Frame Rate=79 |
| Mode | Camera Attributes: Mode | <input type="checkbox"/> | Read/Write | Gets/sets the current operation mode for the current camera attribute. Relative=0 Off=1 Auto=2 One Push=3 Absolute=4 Ignore=5 |
| Value | Camera Attributes: Value |  | Read/Write | Gets/sets the current value for the current camera attribute. |
| Min | Camera Attributes: Range: Minimum |  | Read/Write | Gets the minimum value for the current camera attribute and mode. |
| Max | Camera Attributes: Range: Maximum |  | Read/Write | Gets the maximum value for the current camera attribute and mode. |
| Present | Camera Attributes: Inquiry: Present |  | Read Only | Returns TRUE if the current camera attribute is present. |
| Absolute | Camera Attributes: Inquiry: Absolute |  | Read Only | Returns TRUE if the current camera attribute supports absolute operation mode. |
| OnePush | Camera Attributes: Inquiry: One Push |  | Read Only | Returns TRUE if the current camera attribute supports the One Push operation mode. |
| | | | | |

| | | | | |
|--------------|--------------------------------------|--|-----------|--|
| Off | Camera Attributes: Inquiry: Off |  | Read Only | Returns TRUE if the current camera attribute supports the Off operation mode. |
| Auto | Camera Attributes: Inquiry: Auto |  | Read Only | Returns TRUE if the current camera attribute supports the Auto operation mode. |
| Manual | Camera Attributes: Inquiry: Manual |  | Read Only | Returns TRUE if the current camera attribute supports the Relative operation mode. |
| Readable | Camera Attributes: Inquiry: Readable |  | Read Only | Returns TRUE if the current camera attribute is readable. |
| Name | Camera Attributes: Inquiry: Name |  | Read Only | Returns the name of the current camera attribute. |
| Base Address | Camera Information: BaseAddress | <input data-bbox="971 1020 1013 1058" type="checkbox"/> | Read Only | Returns the base address of the camera command registers. |

LabVIEW VI Error Handling

Every NI-IMAQ for 1394 VI contains an **error in** input cluster and an **error out** output cluster. The clusters contain a Boolean that indicates whether an error occurred, the code for the error, and the source or name of the VI that returned the error. If **error in** indicates an error, the VI passes the error information to **error out** and does not execute any NI-IMAQ function.

You can use the Simple Error Handler VI (**Functions»Time&Dialog**) to check for errors that occur while executing a VI. If you wire an error cluster to the Simple Error Handler VI, the VI deciphers the error information and displays a dialog box that describes the error. If no error occurred, the Simple Error Handler VI does nothing.

[Error Codes](#)

Error Codes

The following table describes the error codes used in NI-IMAQ for IEEE 1394 Cameras.

| Error Code | Status Name | Description |
|-------------|---------------------------|--|
| 0 | IMG1394_ERR_GOOD=1 | Success |
| -1074364416 | IMG1394_ERR_EMEM | Not enough memory |
| -1074364415 | IMG1394_ERR_EDRV | Cannot load the driver |
| -1074364414 | IMG1394_ERR_TIMO | Timeout |
| -1074364413 | IMG1394_ERR_NIMP | Function not implemented |
| -1074364412 | IMG1394_ERR_INTL | Internal error |
| -1074364411 | IMG1394_ERR_BMOD | Invalid mode |
| -1074364410 | IMG1394_ERR_INIT | Session not initialized |
| -1074364409 | IMG1394_ERR_BATT | Invalid attribute |
| -1074364408 | IMG1394_ERR_FTNP | Feature not present in the camera |
| -1074364407 | IMG1394_ERR_ESYS | System error |
| -1074364406 | IMG1394_ERR_HEAP | Allocation error |
| -1074364405 | IMG1394_ERR_UNINITIALIZED | Allocator not initialized |
| -1074364404 | IMG1394_ERR_ORNG | Value out of range |
| -1074364403 | IMG1394_ERR_BCAM | Invalid camera file |
| -1074364402 | IMG1394_ERR_BSID | Invalid Session ID |
| -1074364401 | IMG1394_ERR_NSUP | Attribute not supported by the camera |
| -1074364400 | IMG1394_ERR_INVF | Invalid format |
| -1074364399 | IMG1394_ERR_INVM | Invalid mode |
| -1074364398 | IMG1394_ERR_INVR | Invalid frame rate |
| -1074364397 | IMG1394_ERR_INVC | Invalid color ID |
| -1074364396 | IMG1394_ERR_NOAP | No acquisition in progress |
| -1074364395 | IMG1394_ERR_AOIP | Acquisition already in progress |
| -1074364394 | IMG1394_ERR_IRES | Insufficient resources available for the required video mode |
| -1074364393 | IMG1394_ERR_TBUF | Too many buffers used |
| -1074364392 | IMG1394_ERR_INVP | Invalid parameter |
| -1074364391 | IMG1394_ERR_NSAT | Non-writeable attribute |
| -1074364390 | IMG1394_ERR_NGAT | Non-readable attribute |
| -1074364389 | IMG1394_ERR_CMNF | Camera not found |
| -1074364388 | IMG1394_ERR_CRMV | Camera removed |

| | | |
|-------------|--|---|
| -1074364387 | IMG1394_ERR_BNRD | Buffer not ready |
| -1074364386 | IMG1394_ERR_BRST | Bus reset occurred during a transaction |
| -1074364385 | IMG1394_ERR_NLIC | No license for NI-IMAQ for IEEE 1394 Cameras |
| -1074364384 | IMG1394_ERR_NDLL | DLL could not be found (LabWindows/CVI only) |
| -1074364383 | IMG1394_ERR_NFNC | Function not found in DLL (LabWindows/CVI only) |
| -1074364382 | IMG1394_ERR_NOSR | No resource available (LabWindows/CVI only) |
| -1074364381 | IMG1394_ERR_NCFG | Session not configured |
| -1074364380 | IMG1394_ERR_IOER | I/O error |
| -1074364379 | IMG1394_ERR_CAIU | Camera already in use |
| -1074364378 | IMG1394_ERR_BAD_POINTER | Invalid pointer. The pointer may be NULL when it should be non-NULL, or non-NULL when it should be NULL. |
| -1074364377 | IMG1394_EXCEPTION | Exception occurred. Refer to the NI-PAL debug log for more information. |
| -1074364376 | IMG1394_ERR_BAD_DEVICE_TYPE | Invalid device type. Unable to create an instance. |
| -1074364375 | IMG1394_ERR_ASYNC_READ | Unable to perform asynchronous register read. Camera may be busy or broken. |
| -1074364374 | IMG1394_ERR_ASYNC_WRITE | Unable to perform asynchronous register write. Camera may be busy or broken. |
| -1074364373 | IMG1394_ERR_VIDEO_NOT_SUPPORTED | Combination of video format, mode, and rate is not supported for this camera. Refer to your camera documentation. |
| -1074364372 | IMG1394_ERR_BUFFER_INDEX | Index into the buffer list is incorrect. Reconfigure and try again. |
| -1074364371 | IMG1394_ERR_BAD_USER_ROI | Camera cannot acquire the user-defined ROI. Resize and try again. |
| -1074364370 | IMG1394_ERR_BUFFER_LIST_ALREADY_LOCKED | Buffer list already locked. Reconfigure acquisition and try again. |
| -1074364369 | IMG1394_ERR_BUFFER_LIST_NOT_LOCKED | No buffer list. Reconfigure acquisition and try again. |
| | | |

| | | |
|-------------|---|--|
| -1074364368 | IMG1394_ERR_RESOURCES_ALREADY_ALLOCATED | Isochronous resources already allocated. Reconfigure acquisition and try again. |
| -1074364367 | IMG1394_ERR_BUFFER_LIST_EMPTY | Buffer list empty. Add at least one buffer. |
| -1074364366 | IMG1394_ERR_FLAG_1 | For Format 7, combination of speed, image position, image size, and color coding is incorrect. |
| -1074364365 | IMG1394_ERR_BUFFER_NOT_AVAILABLE | Requested buffer unavailable. Contents of current buffer overwritten by the acquisition. |
| -1074364364 | IMG1394_ERR_IMAGE_REP_NOT_SUPPORTED | Requested image representation not supported for current color coding. |
| -1074364363 | IMG1394_ERR_BAD_OCCURRENCE | Invalid given occurrence. Unable to complete acquisition. |

Important Information

[Warranty](#)

[Copyright](#)

[Trademarks](#)

[Patents](#)

[Warning Regarding Use of NI Products](#)

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

Except as specified herein, National Instruments makes no warranties, express or implied, and specifically disclaims any warranty of merchantability or fitness for a particular purpose. Customer's right to recover damages caused by fault or negligence on the part of National Instruments shall be limited to the amount theretofore paid by the customer. National Instruments will not be liable for damages resulting from loss of data, profits, use of products, or incidental or consequential damages, even if advised of the possibility thereof. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action

accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about [National Instruments trademarks](#).

FireWire® is the registered trademark of Apple Computer, Inc.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, and Stateflow® are registered trademarks, and TargetBox²™, xPC TargetBox²™, and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix® and Tek are registered trademarks of Tektronix, Inc.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) National Instruments products are not designed with components and testing for a level of reliability suitable for use in or in connection with surgical implants or as critical components in any life support systems whose failure to perform can reasonably be expected to cause significant injury to a human.

(2) In any application, including the above, reliability of operation of the software products can be impaired by adverse factors, including but not limited to fluctuations in electrical power supply, computer hardware malfunctions, computer operating system software fitness, fitness of compilers and development software used to develop an application, installation errors, software and hardware compatibility problems, malfunctions or failures of electronic monitoring or control devices, transient failures of electronic systems (hardware and/or software), unanticipated uses or misuses, or errors on the part of the user or applications designer (adverse factors such as these are hereafter collectively termed "system failures"). Any application where a system failure would create a risk of harm to property or persons (including the risk of bodily injury and death) should not be reliant solely upon one form of electronic system due to the risk of system failure. To avoid damage, injury, or death, the user or application designer must take reasonably prudent steps to protect against system failures, including but not limited to back-up or shut down mechanisms. Because each end-user system is customized and differs from National Instruments' testing platforms and because a user or application designer may use National Instruments products in combination with other products in a manner not evaluated or contemplated by National Instruments, the user or application designer is ultimately responsible for verifying and validating the suitability of National Instruments products whenever National Instruments products are incorporated in a system or application, including, without limitation, the appropriate design, process and safety level of such system or application.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the patents.txt file on your CD, or ni.com/patents.

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- [Support](#)—Online technical support resources at ni.com/support include the following:
 - **Self-Help Resources**—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable [KnowledgeBase](#), [product manuals](#), step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
 - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Applications Engineers worldwide in the [NI Developer Exchange](#) at ni.com/exchange. National Instruments Applications Engineers make sure every question receives an answer.

For information about other [technical support options](#) in your area, visit ni.com/services or [contact](#) your local office at ni.com/contact.

- [Training and Certification](#)—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- [System Integration](#)—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your [local office](#) or NI corporate headquarters. You also can visit the [Worldwide Offices](#) section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Branch Offices

| Office | Telephone Number |
|---------------------------|---------------------|
| Australia | 1800 300 800 |
| Austria | 43 0 662 45 79 90 0 |
| Belgium | 32 0 2 757 00 20 |
| Brazil | 55 11 3262 3599 |
| Canada | 800 433 3488 |
| China | 86 21 6555 7838 |
| Czech Republic | 420 224 235 774 |
| Denmark | 45 45 76 26 00 |
| Finland | 385 0 9 725 725 11 |
| France | 33 0 1 48 14 24 24 |
| Germany | 49 0 89 741 31 30 |
| India | 91 80 51190000 |
| Israel | 972 0 3 6393737 |
| Italy | 39 02 413091 |
| Japan | 81 3 5472 2970 |
| Korea | 82 02 3451 3400 |
| Lebanon | 961 0 1 33 28 28 |
| Malaysia | 1800 887710 |
| Mexico | 01 800 010 0793 |
| Netherlands | 31 0 348 433 466 |
| New Zealand | 0800 553 322 |
| Norway | 47 0 66 90 76 60 |
| Poland | 48 22 3390150 |
| Portugal | 351 210 311 210 |
| Russia | 7 095 783 68 51 |
| Singapore | 1800 226 5886 |
| Slovenia | 386 3 425 4200 |
| South Africa | 27 0 11 805 8197 |
| Spain | 34 91 640 0085 |
| Sweden | 46 0 8 587 895 00 |
| Switzerland | 41 56 200 51 51 |
| Taiwan | 886 02 2377 2222 |
| Thailand | 662 992 7519 |
| United Kingdom | 44 0 1635 523545 |
| United States (Corporate) | 512 683 0100 |