IMAQ™

NI-IMAQ VIs for G Reference Manual

June 1997 Edition
Part Number 371642A-01

Internet Support

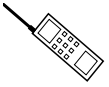
support@natinst.com
E-mail: info@natinst.com
FTP Site: ftp.natinst.com
Web Address: http://www.natinst.com

Bulletin Board Support

BBS United States: (512) 794-5422
BBS United Kingdom: 01635 551422
BBS France: 01 48 65 15 59

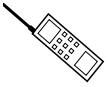
Fax-on-Demand Support

(512) 418-1111



Telephone Support (U.S.)

Tel: (512) 795-8248
Fax: (512) 794-5678



International Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20,
Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, Denmark 45 76 26 00,
Finland 09 725 725 11, France 01 48 14 24 24, Germany 089 741 31 30,
Hong Kong 2645 3186, Israel 03 5734815, Italy 02 413091, Japan 03 5472 2970,
Korea 02 596 7456, Mexico 5 520 2635, Netherlands 0348 433466, Norway 32 84 84 00,
Singapore 2265886, Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 200 51 51,
Taiwan 02 377 1200, United Kingdom 01635 523545

National Instruments Corporate Headquarters

6504 Bridge Point Parkway Austin, TX 78730-5039 Tel: (512) 794-0100

Important Information

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 manual 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

Trademarks

LabVIEW® and IMAQ™ are trademarks of National Instruments Corporation.

Product and company names listed are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Table of Contents

About This Manual

Chapter 1
How to Use NI-IMAQ VIs for G

Chapter 2
NI-IMAQ VIs for G

Figures

Tables

About This Manual

This manual describes the features, functions, and operation of the image acquisition (IMAQ) virtual instruments (VIs) for LabVIEW. To use this manual effectively, you must be familiar with image processing, LabVIEW, and your IMAQ hardware device.

Organization of This Manual

The *NI-IMAQ VIs for G Reference Manual* is organized as follows:

- Chapter 1, *How to Use NI-IMAQ VIs for G*, describes how to use National Instruments G programming and application software, such as LabVIEW and IMAQ Vision, with your IMAQ hardware and NI-IMAQ VIs.
- Chapter 2, *NI-IMAQ VIs for G*, describes the NI-IMAQ and IMAQ Vision VIs included with your NI-IMAQ software.
- Appendix A, *Attributes*, lists the attributes used with the NI-IMAQ VIs.
- Appendix B, *Error Codes*, lists the error codes for the NI-IMAQ VIs.
- Appendix C, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.
- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, mnemonics, and symbols.
- The *Index* contains an alphabetical list of key terms and topics in this manual, including the page where you can find each one.

Conventions Used in This Manual

The following conventions are used in this manual:

<>	Angle brackets enclose the name of a key on the keyboard (for example, <option>). Angle brackets containing numbers separated by an ellipsis represent a range of values associated with a bit or signal name (for example, DBIO<3..0>).
[]	Square brackets enclose optional items (for example, [`response`]).
-	A hyphen between two or more key names enclosed in angle brackets denotes that you should simultaneously press the named keys (for example, <Control-Alt-Delete>).
»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options»Substitute Fonts** directs you to pull down the **File** menu, select the **Page Setup** item, select **Options**, and finally select the **Substitute Fonts** options from the last dialog box.
bold	Bold text denotes the names of menus, menu items, parameters, dialog box, dialog box buttons or options, icons, windows, Windows 95 tabs, or LEDs.
bold italic	Bold italic text denotes a note, caution, or warning.
`bold monospace`	Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.
italic	Italic text denotes emphasis, a cross reference, an operation, or an introduction to a key concept. This font also denotes text from which you supply the appropriate word or value, as in Windows 3.*x*.
`italic monospace`	Italic text in this font denotes that you must enter the appropriate words or values in the place of these items.
`monospace`	Text in this font denotes text or characters that you should literally 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, and for statements and comments taken from programs.
paths	Paths in this manual are denoted using backslashes (\) to separate drive names, directories, folders, and files.
	The *Glossary* lists abbreviations, acronyms, metric prefixes, mnemonics, symbols, and terms.

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix C, *Customer Communication*, at the end of this manual.

How to Use NI-IMAQ VIs for G

Chapter 1

This chapter describes how to use National Instruments G programming and application software, such as LabVIEW and IMAQ Vision, with your IMAQ hardware and NI-IMAQ VIs.

Introduction

LabVIEW features interactive graphics, a state-of-the-art user interface, and a powerful graphical programming language, G. The G NI-IMAQ VI Library, a series of virtual instruments (VIs) for using LabVIEW with the PCI-1408, is included with your NI-IMAQ software.

IMAQ Vision for G is an image acquisition, processing, and analysis library that consists of more than 400 VIs for using the PCI-1408 with LabVIEW. If you have not purchased the IMAQ Vision image acquisition and analysis libraries, you can use the three IMAQ Vision VIs included with your NI-IMAQ software. If you use these basic functions, you can upgrade your programs to use IMAQ Vision without any changes to your image acquisition VIs.

Before you start building your image acquisition (IMAQ) application, you should know the following basic G IMAQ concepts:

- Location of the NI-IMAQ examples
- Location of the NI-IMAQ VIs in LabVIEW
- Common NI-IMAQ VI parameters
- Error handling
- Buffer management
- IMAQ acquisition types
- Image display
- NI-IMAQ attributes

Location of NI-IMAQ Examples

The NI-IMAQ examples for G illustrate some common applications used with NI-IMAQ in LabVIEW. You can find these examples in the `labview\examples\imaq` directory. For a brief description of any example, open the example VI and choose **Windows»Show VI Info** for a text description of the example.

Location of the NI-IMAQ VIs in LabVIEW

You can find the NI-IMAQ VIs in the **Functions** palette from your block diagram in LabVIEW. Select the IMAQ icon near the bottom of the **Functions** palette, as shown in Figure 1-1, to pop up the **IMAQ Library** palette.

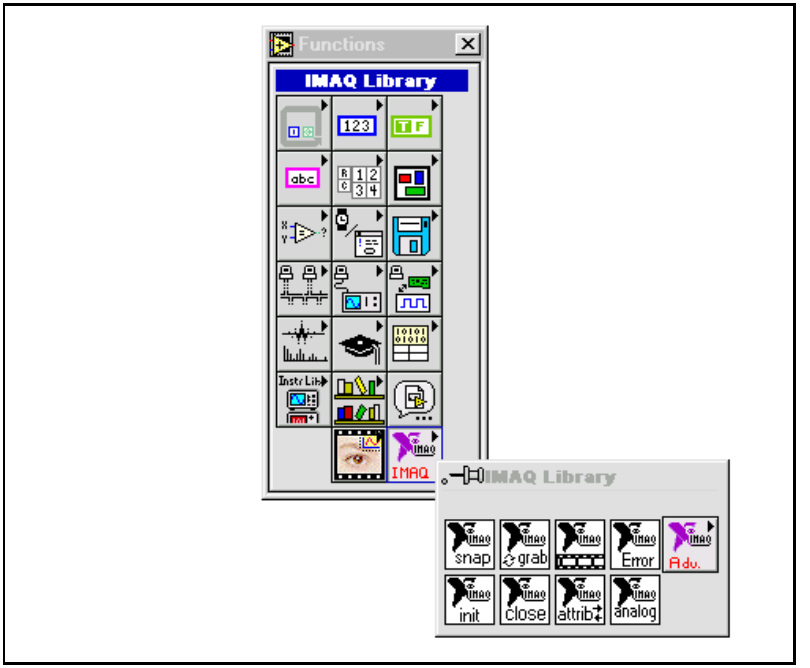


Figure 1-1. Functions Palette with IMAQ Palette

The most commonly used VIs are on the **IMAQ Library** palette. The **IMAQ Library»IMAQ Advanced** palette contains VIs for more advanced applications. See the *Advanced NI-IMAQ VIs* section in Chapter 2, *NI-IMAQ VIs for G*, for more information on the **Advanced VI** palette.

Common NI-IMAQ VI Parameters

The **Interface #** input on NI-IMAQ VIs specifies the number of the initialized interface. Use IMAQ_Init to generate the **Interface #**, which identifies this configuration in subsequent NI-IMAQ VIs. The NI-IMAQ acquisition VIs—IMAQ_Snap, IMAQ_Grab, and IMAQ_Sequence—require you to wire to **Interface #** only if you are using an interface other than the default `img0` or are using multiple boards.

All acquisition VIs require that you supply an image buffer to receive the captured image. You can create this image buffer with the Cvi Create VI. Consult the *Buffer Management* section of this chapter for more information. The input that receives the image buffer is **Image in**. The **Image out** output returns the captured image.

The acquisition VIs use the **Optional rectangle** input to specify a rectangular portion, or region of interest, of an image frame to be captured, which you can use to reduce the size of the image you want to capture. **Optional rectangle** is an array of four elements with the elements defined as Left, Top, Right, Bottom. The width [Right-Left] must be a multiple of four. If **Optional rectangle** is not wired, the entire image acquisition window is captured. You configure the acquisition window using these parameters in the `IMAQconf` configuration utility.

Error Handling

Every NI-IMAQ VI contains an **error in** input cluster and an **error out** output cluster, as shown in Figure 1-2. The clusters contain a Boolean that indicates whether an error occurred, the code for the error, and the source or the 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.

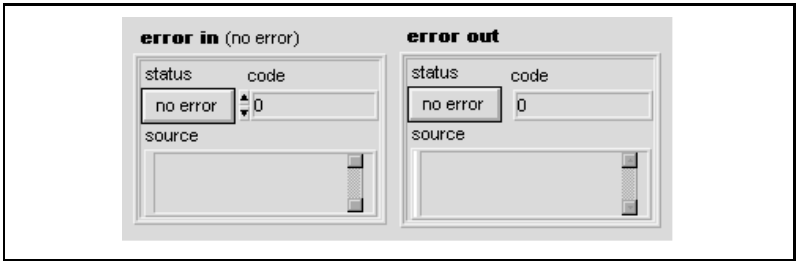


Figure 1-2. Error Clusters

You can use the IMAQ Error Handler VI to check for errors that occur while executing a VI. If you wire an error cluster to the IMAQ Error Handler VI, the VI deciphers the error information and displays a dialog box that describes the error. If no error occurred, the IMAQ Error Handler VI does nothing. Figure 1-3 shows how to wire an NI-IMAQ VI to the IMAQ Error Handler VI.

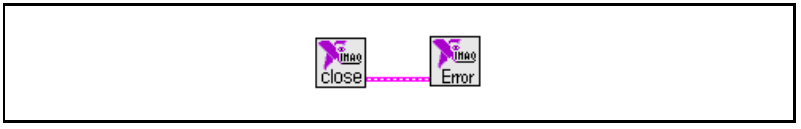


Figure 1-3. Error Checking using the IMAQ Error Handler VI

Figure 1-4 shows an example of the dialog box IMAQ Error Handler displays when an error occurs.

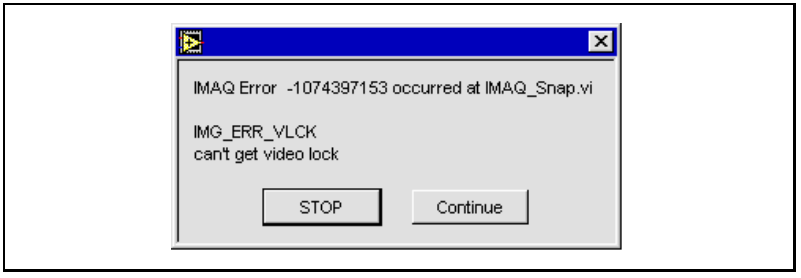


Figure 1-4. IMAQ Error Handler Dialog Box

Buffer Management

Cvi Create and Cvi Dispose manage image buffers in LabVIEW. Cvi Create allocates an image buffer. **Image Name** is a label for the buffer created. Each buffer must have a unique name. **New Image** contains pointer information to the buffer, which is initially empty. When you wire **New Image** to the **Image in** input of an image acquisition VI, the image acquisition VI allocates the correct amount of memory for the acquisition. If you are going to process the image, you might need to wire to **Border Size**. **Border Size** is the width in pixels created around an image. Some image processing functions, such as labeling or morphology, require a border.

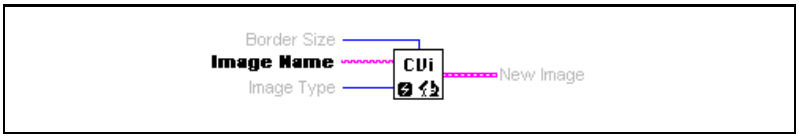


Figure 1-5. Cvi Create

CVI Dispose frees the memory allocated for the image buffer. Call this VI only after the image is no longer required for processing.

Figure 1-6. Cvi Dispose

NI-IMAQ Acquisition Types

Three image acquisition types are available in LabVIEW—snap, grab, and sequence. The following sections describe each acquisition type and give examples.

Snap

A snap acquires a single image into a memory buffer. Use this acquisition mode to acquire a single frame or field to a buffer. When you invoke a snap, it initializes the board and acquires the next incoming video frame (or field) to a buffer. A snap is appropriate for low-speed or single-capture applications.

Use the IMAQ_Snap VI for snap applications. Figure 1-7 shows a simplified block diagram for using IMAQ_Snap.

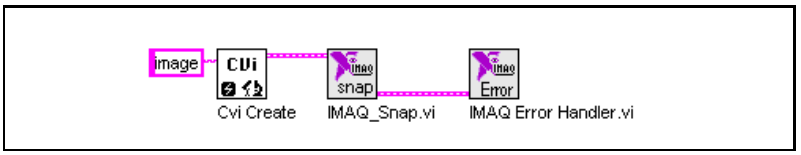


Figure 1-7. Acquiring an Image Using Snap

Grab

A grab is a continuous, high-speed acquisition of data to a single buffer in host memory. This function performs an acquisition that loops continually on one buffer. You can get a copy of the acquisition buffer by grabbing a copy to a LabVIEW image buffer.

The VI you use for grab applications, IMAQ_Grab, has three operations—*Setup*, *Acquire*, and *Release*. *Setup*, which you call only once, initializes the acquisition and starts capturing the image to an internal software buffer. *Acquire*, which you can call multiple times, copies the image currently stored in the internal buffer to a G image buffer. Call *Acquire* either synchronously or asynchronously. With a synchronous transfer, IMAQ_Grab waits for the next vertical blank signal and then transfers the image from the internal buffer to the G image buffer. With an asynchronous transfer, IMAQ_Grab immediately transfers the image from the internal buffer to the G image buffer, which could result in portions of the image transferred being acquired at different times. A typical application for an asynchronous transfer is the acquisition of images of stationary objects. The *Release* operation, which you must call at the end of your VI, shuts down the image acquisition. The **Grab operation** input determines which operation is performed in a call to IMAQ_Grab.

Figure 1-8 shows a simplified block diagram for using IMAQ_Grab. In this example, you perform a synchronous grab by wiring a TRUE to the **Synchronize transfer**. Notice that IMAQ_Grab is used multiple times with different **Grab operations**. The input to **Grab operation** is an enumerated constant. To create the enumerated constant, pop up on the **Grab operation** input and choose **Create Constant**. Click the created constant with the Operating Tool and select the desired operation. For more information on enumerated types, consult the *LabVIEW User Manual* or the *G Programming Reference Manual.*

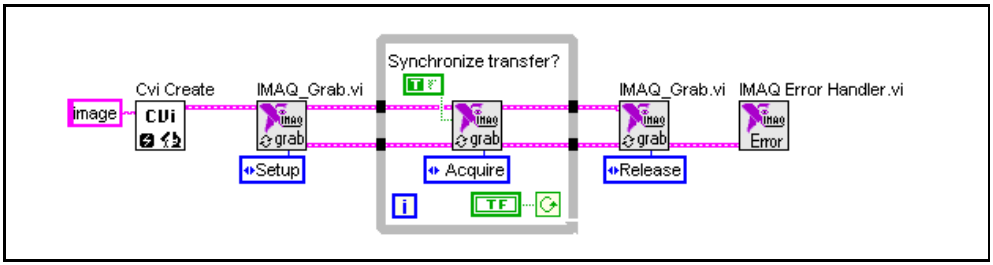


Figure 1-8. Acquiring Images Using Synchronous Grab

Sequence

A sequence initiates a variable-length and variable-delay transfer to multiple buffers. A sequence is appropriate for applications that process multiple images. You can configure a sequence to acquire every frame or skip a variable number of frames between each image.

The VI you use for sequence applications is IMAQ_Sequence. IMAQ_Sequence has four operations: *Setup*, *Start*, *Get Status*, and *Release*. *Setup*, which is called once, initializes the sequence. In the *Setup* operation, **Skip Table** is a required input. **Skip Table** is an array containing the number of frames to skip between images. *Start* starts the acquisition and can be called either synchronously or asynchronously. If called synchronously, the VI does not return until the entire sequence is acquired. If called asynchronously, the VI returns immediately. You can use *Get Status* with an asynchronous operation to determine if the acquisition is complete. Notice that an asynchronous sequence has a different result than an asynchronous grab. An asynchronous sequence forces IMAQ_Sequence to return immediately. Each image is composed of pixels acquired at one time, unlike an asynchronous grab. *Release*, which is called once, shuts down the sequence. The **Sequence operation** input determines which operation is performed in a call to IMAQ_Sequence.

Figure 1-9 shows a simplified block diagram for using IMAQ_Sequence synchronously. Cvi Create is used inside a For Loop to create an array of images for the **Images in** input to IMAQ_Sequence. To Decimal and Concatenate Strings illustrate a useful method to create a unique name for each image in the array. A synchronous operation is performed by wiring a FALSE to **Asynchronous call** with the *Start* **Sequence operation**. IMAQ_Sequence will not finish executing until the acquisition is finished. Notice that IMAQ_Sequence is used multiple times with different **Sequence operations**. The input to

Sequence operation is an enumerated constant. To create the enumerated constant, pop up on the **Sequence operation** input and choose **Create Constant**. Click the created constant with the Operating Tool and select the desired operation.

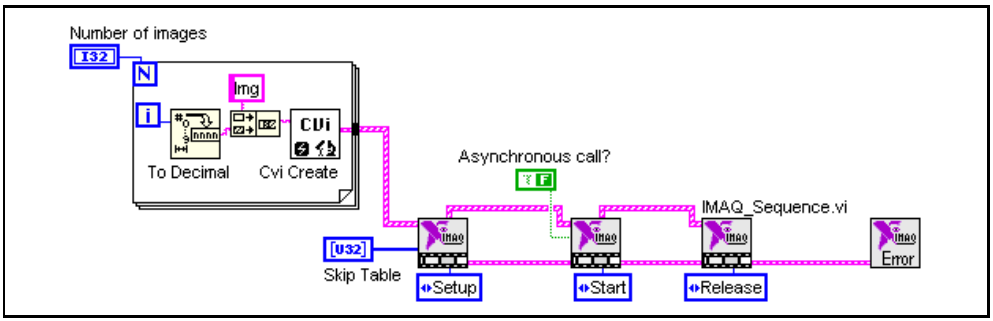


Figure 1-9. Acquiring Images Using Sequence in Synchronous Mode

Figure 1-10 shows a simplified block diagram for using IMAQ_Sequence asynchronously. An asynchronous operation is performed by wiring a TRUE to **Asynchronous call** with the *Start* **Sequence operation**. IMAQ_Sequence returns immediately. IMAQ_Sequence is called in a While Loop with the *Get Status* **Sequence operation** to determine when the acquisition is complete. **Acquiring** is wired directly to the condition terminal of the While Loop so the While Loop terminates when the acquisition is complete.

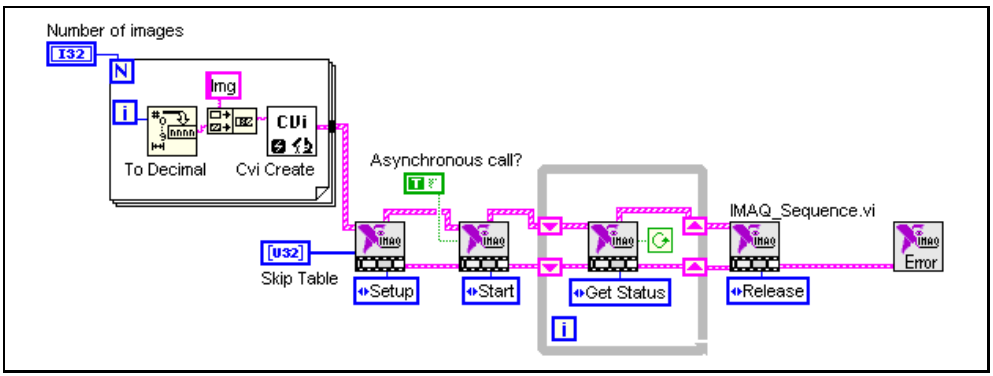


Figure 1-10. Acquiring Images Using Sequence in Asynchronous Mode

Image Display

Many image acquisition applications require that one or more images be displayed. Two options are available for displaying images in LabVIEW.

If you have IMAQ Vision for G, the image processing and analysis software for LabVIEW, Cvi WindDraw is available. Cvi WindDraw (**IMAGE»Display(basics)**) displays an image in an image window. Figure 1-11 illustrates using Cvi WindDraw to display an image acquired using IMAQ_Snap. Images can be displayed in the same way using any acquisition type. For more information on the display capabilities of IMAQ Vision, consult the *IMAQ Vision for G Reference Manual*.

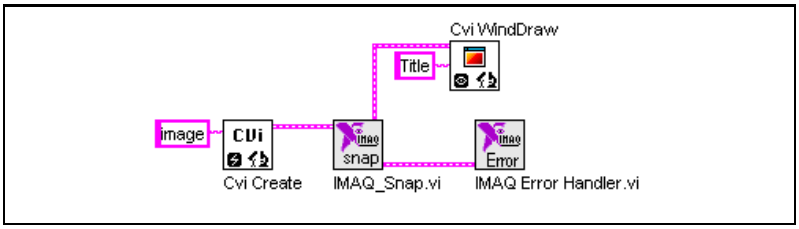


Figure 1-11. Displaying an Image Using Cvi WindDraw

If you do not have IMAQ Vision, you can display an image on a G Intensity Graph. Before you can properly display an image, you need to make some minor changes to the default properties of the Intensity Graph.

After you place the Intensity Graph on the front panel, pop up on the graph and choose **Transpose Array**. To create the correct grayscale color palette, pop up on the marker labeled `50` on the color ramp and choose **Delete Marker**. Also, change the maximum value on the color palette from 100 to 255. Next, change the Y axis so that it is inverted. You might also need to change the ranges of the X and Y axes to match the width and height of the image. Your intensity graph now should appear similar to the image shown in Figure 1-12. For more information on the Intensity Graph, consult the *LabVIEW User Manual* or the *G Programming Reference Manual*.

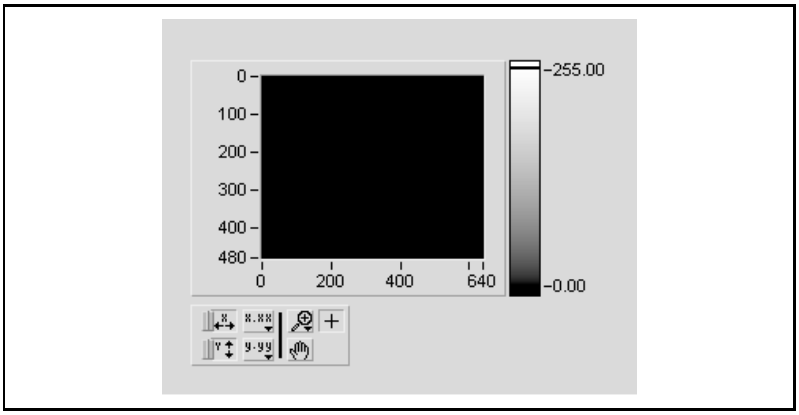


Figure 1-12. Intensity Graph for Image Display

Use the Cvi Image to Array VI to copy an image from an image buffer into a G array. Then you can wire this array directly to an Intensity Graph for display. Figure 1-13 illustrates using an Intensity Graph to display an image acquired using IMAQ_Snap.

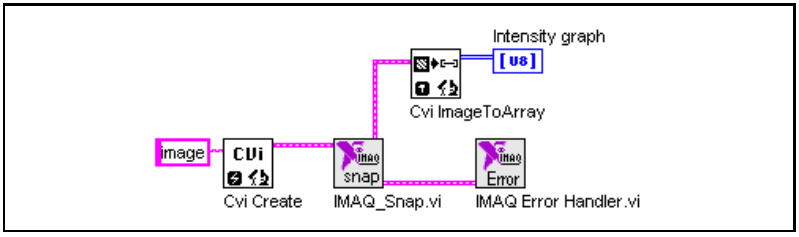


Figure 1-13. Displaying an Image Using an Intensity Graph

NI-IMAQ Attributes

Use the IMAQ_Attribute VI to get or set any attribute of the NI-IMAQ driver. The NI-IMAQ attributes are described in Appendix A, *Attributes*.

To use IMAQ_Attribute to get the current value of an attribute, first wire a FALSE to the **Get/Set** input. Then pop up on the **Attribute** input and choose **Create Constant**. Click the created constant with the Operating Tool and select the desired attribute. The current value of that attribute returns in the **Get Value** output. Figure 1-14 shows how to use IMAQ_Attribute to get the value of an attribute.

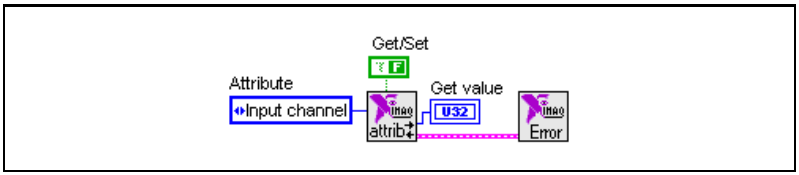


Figure 1-14. Getting the Value of an NI-IMAQ Attribute

To use IMAQ_Attribute to set the value of an attribute, first wire a TRUE to the **Get/Set** input. Then pop up on the **Attribute** input and choose **Create Constant**. Click the created constant with the Operating Tool and select the desired attribute. Wire the new value of the attribute to the **Set value** input. Figure 1-15 shows how to use IMAQ_Attribute to set the value of an attribute.

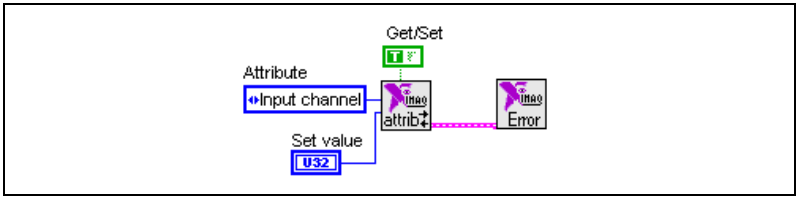


Figure 1-15. Setting the Value of an NI-IMAQ Attribute

Chapter 2

NI-IMAQ VIs for G

This chapter describes the NI-IMAQ and IMAQ Vision VIs included with your NI-IMAQ software.

NI-IMAQ VIs

All VIs dedicated to the IMAQ PCI-1408 board are in the `IMAQ.LLB` library. The NI-IMAQ VIs gives you the basic functions to:

- Load information about boards and cameras from a configuration file
- Select a video channel
- Adjust the analog parameters
- Start or stop an acquisition
- Transfer an image from PCI-1408 memory to an IMAQ Vision image buffer

The VIs described in this document are arranged according to type—Basic, Advanced, and IMAQ Vision—and then in palette order.

Basic NI-IMAQ VIs

When you choose the **Function»Image Acquisition»NI-IMAQ Library** menu, you will see the basic VI palette shown in Figure 2-1.

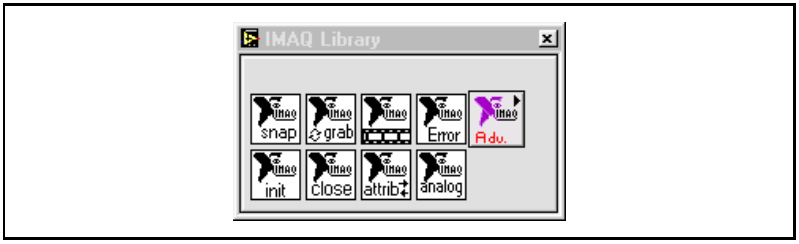
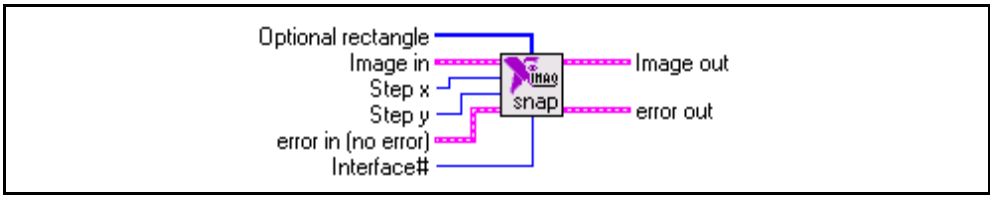


Figure 2-1. NI-IMAQ Basic VI Palette

The Basic NI-IMAQ VIs are sufficient for most applications. These VIs allow you to acquire images, open and close an interface, and get and set attributes.

IMAQ_Snap.vi



Acquires a single image into **Image out**. If necessary, this VI performs a system initialization using IMAQ_Init.

Optional rectangle specifies a rectangular portion of the image frame (the region of interest) to be captured. **Optional rectangle** is defined by an array of four elements [Left, Top, Right, Bottom]. If **Optional rectangle** is not connected or empty, the entire image acquisition window is captured. You must set the width [Right-Left] to a multiple of four.

Image in is the reference of the image that will receive the captured frame pixel data.

Step x is a horizontal sampling step or horizontal reduction factor. If it is set to its default value of 1, each column of the image is transferred. If **Step x** is set to another value *n* (≠1), only one column every *n* columns is transferred. **Step x** only accepts values of 1, 2, 4, or 8.

Step y is a vertical sampling step or vertical reduction factor. If it is set to its default value of 1, each line of the image is transferred. If **Step y** is set to another value *n* (≠1), only one line every *n* lines is transferred. **Step y** only accepts values of 1, 2, 4, or 8.

IMAQ_Snap.vi

(Continued)

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

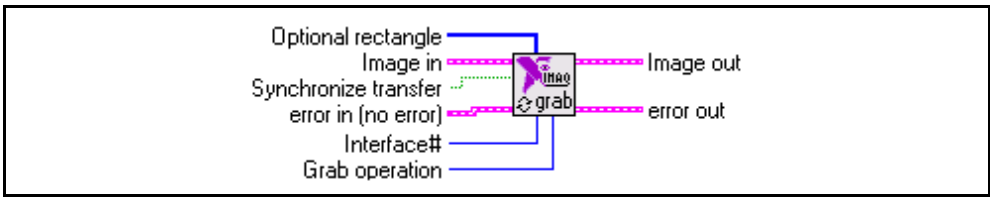
source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Image out returns the captured image.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_Grab.vi



Starts, acquires, and releases an image grab acquisition. Use the grab function for high-speed image acquisition. The grab function has three different operations.

Operation	Function
Setup	Initializes the grab function and starts capturing the image to an internal software buffer. Call this operation once to start the acquisition.
Acquire	Copies the internal buffer to **Image out**. You can call this operation multiple times, and you can copy the image synchronously or asynchronously.
Release	Shuts down image acquisition. Call this operation once to stop the acquisition.

If necessary, this VI initializes your system using the IMAQ_Init VI.

[I32] **Optional rectangle** specifies a rectangular portion of the image frame (the region of interest) to be captured. **Optional rectangle** is defined by an array of four elements [Left, Top, Right, Bottom]. If **Optional rectangle** is not connected or empty, the entire image acquisition window is captured. You must set the width [Right-Left] to a multiple of four.

Image in is the reference of the image that will receive the captured frame pixel data. The **Grab Operation»acquire** option uses the **Image in** input.

IMAQ_Grab.vi

(Continued)

Synchronize transfer determines if the acquire operation synchronizes on vertical blank or does an immediate transfer. The **Grab Operation»acquire** option uses the **Synchronize transfer** input. The default value is TRUE, synchronize on vertical blank.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

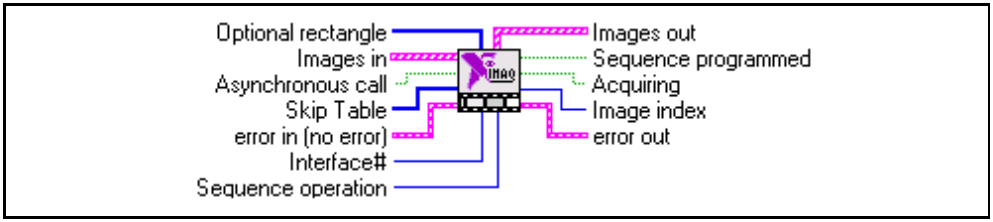
Grab operation determines which of the following operations will be performed:

0 = Setup
1 = Acquire
2 = Release

Image out returns the captured image. The image output is returned when the **Grab Operation»acquire** option is executed.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_Sequence.vi



Starts, acquires, and releases an image sequence. Use this VI to capture multiple images with fixed or variable delays between images. The sequence function has four different operations, as listed in the following table.

Operation	Function
Get Status	Returns the current status of the sequence.
Setup	Initializes the sequence. Call this operation once.
Start	Starts the image sequence acquisition.
Release	Shuts down the sequence. Call this operation once.

If necessary, this VI initializes the system using the IMAQ_Init VI.

[I32] **Optional rectangle** specifies a rectangular portion of the image frame (the region of interest) to be captured. **Optional rectangle** is defined by an array of four elements [Left, Top, Right, Bottom]. If **Optional rectangle** is not connected or empty, the entire image acquisition window is captured. You must set the width [Right-Left] to a multiple of four. The setup operation of the **Sequence Operation** parameter uses **Optional rectangle**.

IMAQ_Sequence.vi

(Continued)

Images in is an array of the images that the sequence will capture. The **Sequence Operation»Setup** option uses **Images in**.

Asynchronous call determines if the sequence is called asynchronously or synchronously. When called synchronously, the **Sequence Operation»Start** option will not return until the entire sequence is captured. The default value is synchronous.

Skip Table is an array of the frame/field delay between images. The first image in the sequence is captured when the **Sequence Operation»Start** option is selected while each following element will delay the number of frames/fields in the skip count. The **Sequence Operation»Setup** option uses the **Skip Table** input.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Sequence operation determines which of the following operations will be performed:

0 = Get Status
1 = Setup
2 = Start
3 = Release

Images out returns the captured images.

IMAQ_Sequence.vi

(Continued)

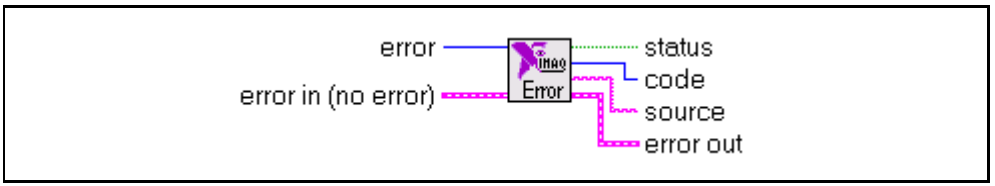
Sequence programmed returns the status of the **Sequence Operation»Setup** option. If the status is TRUE, the setup succeeded. If the status is FALSE, the setup failed.

Acquiring returns the status of the acquisition. If the status is TRUE, the board is acquiring an image. If the status is FALSE, acquisition is complete.

Image index returns the index of the last acquired buffer.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ Error Handler.vi



Informs you if an IMAQ error occurs, describes the error, and identifies where the error occurs. The information needed to return the error information is derived from the input error cluster in and from an internal error description table. The table lists all errors that can be created by the NI-IMAQ Library VIs.

error is a numeric error code. The VI ignores this value if error in indicates an error. Otherwise, this value is tested. A non-zero value signifies an error.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

- **status** is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.
- **code** is the error code identifying an error. A value of 0 generally means no error.
- **source** identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

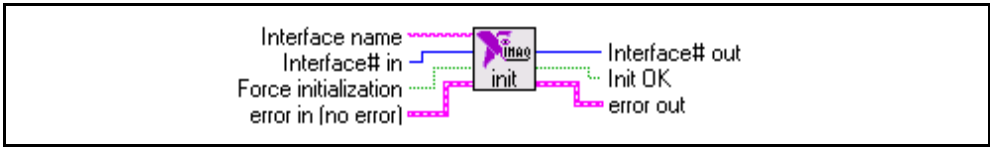
status is TRUE if the VI found an error.

code is the error code indicated by **error in** or **error**.

source indicates the source of the error.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_Init.vi



Loads an NI-IMAQ configuration file and configures the PCI-1408. If no inputs are connected, this VI automatically loads the standard configuration (`img0`). This VI is called automatically when one of the following VIs is called for the first time: IMAQ_Channel, IMAQ_Setup Analog, IMAQ_Snap, IMAQ_Grab, or IMAQ_Sequence. Call the IMAQ_Init VI only when you use more than one PCI-1408 board in the system or if you are using an interface other than the default `img0`.

Interface name is the name of the interface to be loaded. The name must match the configuration file name used in the NI-IMAQ configuration utility, `IMAQconf`. The default value is `img0`.

Interface# in is an input used to check if the system has been previously configured. Do not connect this input if you want to initialize an interface.

Force initialization forces an initialization. An initialization is done only once. If you want to initialize the interface again, you have to connect a TRUE value to this input. The default value is FALSE.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

IMAQ_Init.vi

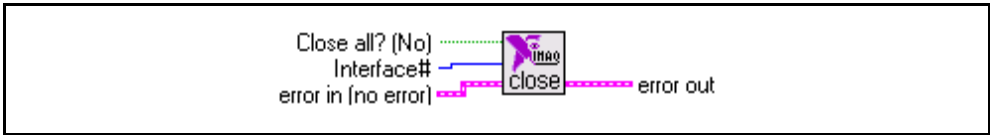
(Continued)

Interface# out contains the interface number. If the system is initialized for the first time, the newly created interface number will be output. If the system has been previously initialized, **Interface# out** will be equal to **Interface# in**.

Init OK returns the status of the initialization. TRUE indicates success; FALSE indicates an error.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_Close.vi



Closes a selected open interface or all the existing open interfaces at once.

Close all? (No) specifies whether all the open interfaces are to be closed. The default value is FALSE (No) where only the specified interface is closed.

Interface# is the ID number of the interface to be closed.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

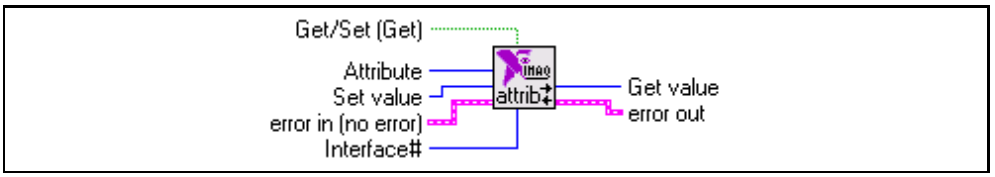
code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI. The VI-specific error is:

-107439179 IMG_ERR_PAR1 invalid interface number

IMAQ_Attribute.vi



Gets or sets any attribute of the NI-IMAQ driver. See Appendix A, *Attributes,* for a list of attributes and descriptions.

Get/Set (Get) specifies whether you want to read the current values or set new values. The default value is FALSE (get current values).

Attribute specifies the attribute to get or set.

Set value represents the value of the attribute to be set.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

- **status** is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.
- **code** is the error code identifying an error. A value of 0 generally means no error.
- **source** identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

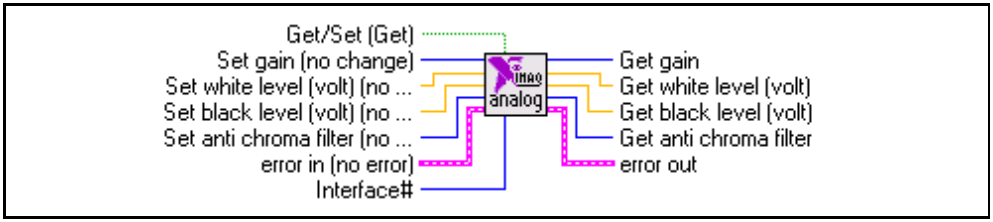
Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Get value outputs the value of the specified attribute.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI. VI-specific errors are:

-107439179	IMG_ERR_PAR1	invalid interface
-107439177	IMG_ERR_PAR3	illegal attribute value

IMAQ_1408 Setup Analog.vi



Gets or sets all the analog parameters of the PCI-1408. This function will only set the inputs that are connected; all unconnected inputs will remain unchanged.

Get/Set (Get) specifies whether you want to read the current values or set new values. The default value is FALSE.

Set gain specifies the gain to use. The following values are supported for PCI-1408 module:

0: 1.00
1: 1.33
2: 2.00
3: No Change

Set white level (volt) is the white voltage reference level in volts (at a gain of 1.00).

Set black level (volt) is the black voltage reference level in volts (at a gain of 1.00).

Set anti chroma filter specifies the following settings for the antichrominance filter:

0: Disabled
1: NTSC
2: PAL
3: No Change

IMAQ_1408 Setup Analog.vi

(Continued)

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Get gain returns the specified gain.

Get white level (volt) returns the white voltage reference level in volts.

Get black level (volt) returns the black voltage reference level in volts.

Get anti chroma filter returns the antichrominance filter setting.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

Advanced NI-IMAQ VIs

You will see the advanced VI palette shown in Figure 2-2 when you choose the **Function»Image Acquisition»Advanced» NI-IMAQ Library** menu.

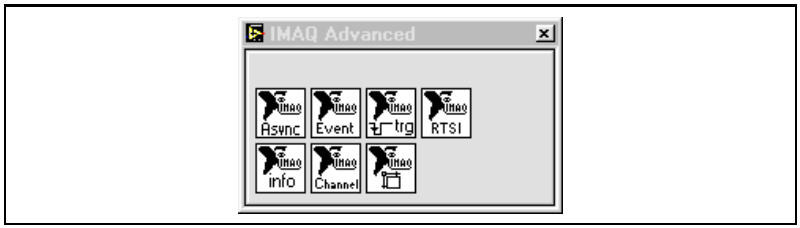
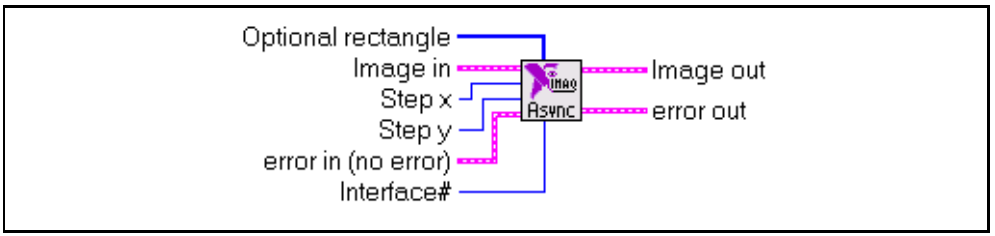


Figure 2-2. NI-IMAQ Advanced VI Palette

The Advanced NI-IMAQ VIs allow you to control the trigger lines, wait for a hardware event, and get information about the camera and interface.

IMAQ_Snap_Async.vi



Acquires a single image into **Image out**. If necessary, it performs a system initialization using IMAQ_Init. This function is asynchronous, or non-blocking. You should use it in conjunction with triggering an acquisition so that other LabVIEW activities can be executed simultaneously.

Optional rectangle specifies a rectangular portion of the image frame (the region of interest) to be captured. **Optional rectangle** is defined by an array of four elements [Left, Top, Right, Bottom]. If **Optional rectangle** is not connected or empty, the entire image acquisition window is captured. You must set the width [Right-Left] to a multiple of four. The setup operation of the **Sequence Operation** parameter uses **Optional rectangle**.

Image in is the reference of the image that will receive the captured frame pixel data.

Step x is a horizontal sampling step or horizontal reduction factor. If it is set to its default value of 1, each column of the image is transferred. If **Step x** is set to another value *n* (≠1), only one column every *n* columns is transferred. **Step x** only accepts values of 1, 2, 4, or 8.

Step y is a vertical sampling step. If it is set to its default value of 1, each line of the image is transferred. If **Step y** is set to another value *n* (≠1), only one line every *n* lines is transferred. **Step y** only accepts values of 1, 2, 4, or 8.

IIMAQ_Snap_Async.vi

(Continued)

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

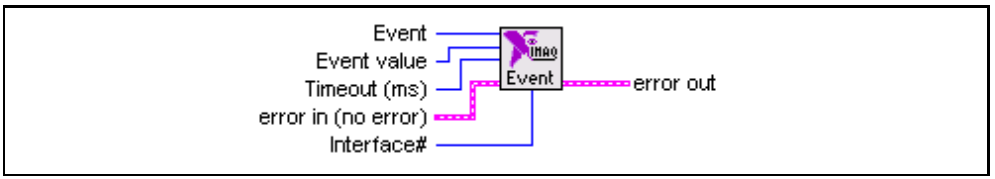
source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Image out returns the captured image.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_Wait for event.vi



Automatically waits for a selectable hardware event.

Event specifies the event for which to wait. These events include vertical blank, frame done, acquisition complete, and any trigger status.

Event value works with trigger events and specifies what value to wait for on the selected trigger line (0 = LOW, 1 = HIGH). This VI is capable of catching trigger pulses as short as 20 ns.

Timeout specifies the amount of time in milliseconds to wait for the specified event before returning a timeout error.

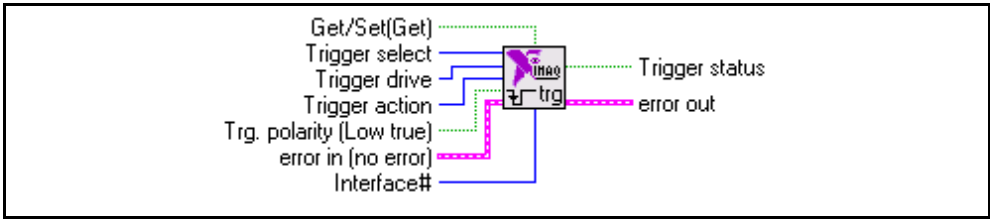
error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

- **status** is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.
- **code** is the error code identifying an error. A value of 0 generally means no error.
- **source** identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_Trigger Control.vi



Configures either the external or RTSI bus triggers. Any trigger can be driven asserted, unasserted, or with internal status signals AQ_Done, AQ_IN_PROGRESS, Pixel Clock, HSYNC, or VSYNC. In addition, the polarity of each trigger is programmable to either HIGH-TRUE or LOW-TRUE. Each trigger also can be configured to start an acquisition. If multiple triggers are configured to begin an acquisition, the image acquisition will not begin until the assertion edge of all the enabled triggers has been detected.

Get/Set (Get) specifies whether you want to read the current value on a specific trigger line or change the configuration of a trigger.

Trigger select specifies the specific trigger.

Trigger drive specifies if the signal should be driven and by which signal, or if driving the trigger should be disabled.

Trigger action specifies if an assertion edge of this trigger should start an acquisition.

Trg. polarity specifies the polarity of the trigger signal. FALSE indicates LOW-TRUE; TRUE indicates HIGH-TRUE. The default is LOW-TRUE.

IMAQ_Trigger Control.vi

(Continued)

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

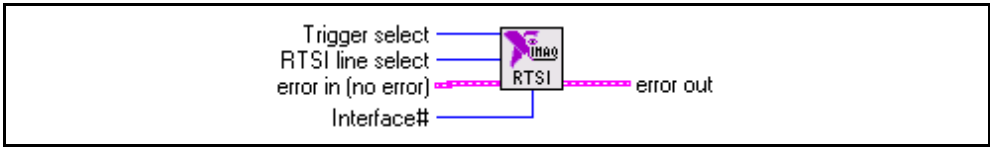
Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Trigger status specifies the current value on a trigger line. If **Get/Set** is set to Get, this output will give the current value on the specified trigger line (TRUE = HIGH, FALSE = LOW).

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI. The VI-specific error is:

-1074397179 IMG_ERR_PAR1 invalid interface number

IMAQ_Trigger RTSI map.vi



Provides a mechanism to specify to which RTSI bus wires the four device RTSI bus triggers should be mapped.The RTSI bus on the PCI-1408 is composed of seven physical wires although the device supports only four RTSI bus triggers. Call this VI before using any of the RTSI bus triggers, because the default mapping for all of the RTSI bus triggers is disabled.

Trigger select specifies the specific RTSI bus trigger.

RTSI line select specifies the physical RTSI bus wire.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

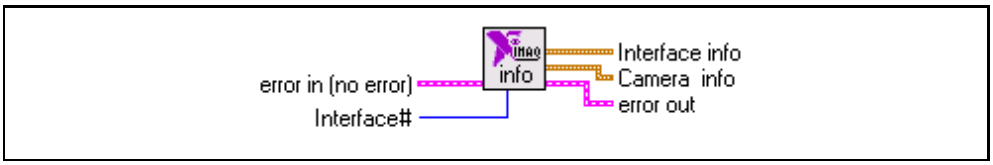
- **status** is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.
- **code** is the error code identifying an error. A value of 0 generally means no error.
- **source** identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI. The VI-specific error is:

-1074397179 IMG_ERR_PAR1 invalid interface number

IMAQ_Info.vi



Returns information about a board and a camera associated with a video channel.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Interface info is a cluster containing information about the PCI-1408 board.

Base Address is the board's physical address on PCI bus.

Interrupt Level is the interrupt level (IRQ number) assigned to the board.

Memory Size is the memory size on this board (0 for PCI-1408).

Type is the type of board (PCI-1408).

IMAQ_Info.vi

(Continued)

Camera info is a cluster containing information about the camera that is associated with the channel.

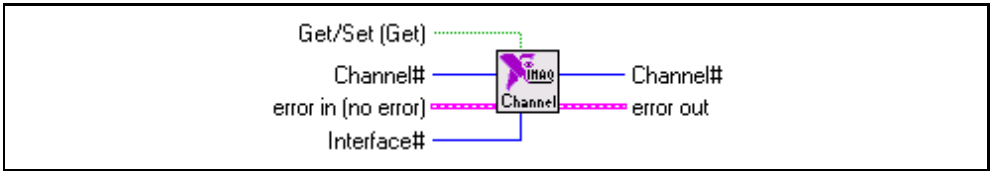
Pixel size is the pixel size (in bits) coming from the camera (8 bits only for the PCI-1408).

Horizontal pixels indicates the number of pixels in the horizontal axis of this camera.

Vertical pixels indicates the number of pixels in the vertical axis of this camera.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_Channel.vi



Selects an input channel or a camera associated with a video input channel. This VI operates with logical definitions contained in the current configuration file.

☞ **Note:** ***Changing the channel makes the software load the camera configuration corresponding to that channel as defined using the IMAQ configuration utility,*** `IMAQconf`***. This change may overwrite specific settings like*** **Analog Setup** ***or other adjustments done with the IMAQ_Attribute VI. To avoid problems, place any IMAQ_Channel VI as early as possible in the LabVIEW data flow, for example, right after the IMAQ_Init VI.***

Get/Set (Get) specifies whether you want to read the current channel or set a new channel. The default value is FALSE.

Channel# specifies the camera/channel number to select.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

- **status** is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.
- **code** is the error code identifying an error. A value of 0 generally means no error.
- **source** identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

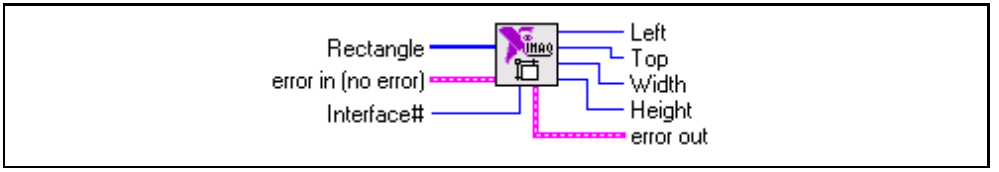
IMAQ_Channel.vi

(Continued)

Channel# returns the number of the selected channel.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ_RectToCoord.vi



Converts a rectangle array to the parameters used by the PCI-1408.

Rectangle specifies a rectangular portion of the image frame to be captured. It is defined by an array of four elements [Left, Top, Right, Bottom]. Set the width [Right-Left] to a multiple of four.

error in describes error conditions that occur before this VI executes. This cluster defaults to no error. The **error in** cluster contains three parameters.

status is TRUE if an error has occurred. If **status** is TRUE, this VI does not perform any operations.

code is the error code identifying an error. A value of 0 generally means no error.

source identifies where an error has occurred. The source string usually is the name of the VI that produced the error.

Interface# is the number of the initialized interface. If you are using the default interface, `img0`, no connection is needed. For multiple board use or to use an interface other than `img0`, connect this terminal to the **Interface# out** terminal of the corresponding IMAQ_Init VI.

Left returns the left-most pixel number.

Top returns the top pixel number.

Width returns the width of the current region of interest.

IMAQ_RectToCoord.vi

(Continued)

Height returns the height of the current region of interest.

error out contains error information. If the **error in** cluster indicates an error, the **error out** cluster contains the same information. Otherwise, **error out** describes the error status of this VI.

IMAQ Vision VIs

This section describes some basic IMAQ Vision for LabVIEW functions that you can use with NI-IMAQ VIs. The VIs support creating and disposing of images and the conversion of images to arrays. After you convert an image to an array, you can use standard LabVIEW techniques to process and display the images.

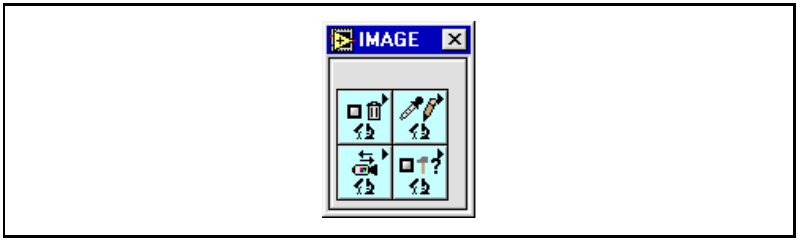
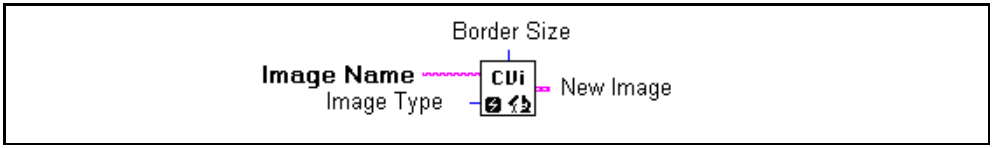


Figure 2-3. IMAQ Vision VI Palette

You can find the IMAQ Vision VIs in the **Functions** palette from your block diagram. If you have not purchased IMAQ Vision, the palette shown in Figure 2-3 will appear. If you have IMAQ Vision installed, your palette will contain more functions.

Cvi Create



Creates an image buffer that can be input into any of the acquisition functions of the PCI-1408.

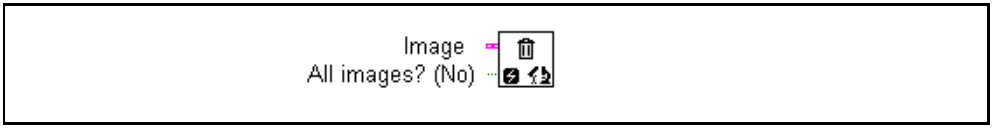
Image Name is the name that will be associated with the created image. If the application contains only a single image, then you do not have to name it.

Image Type specifies the type of image that is being created. The PCI-1408 supports only an 8-bit image (Type 0). The default value is 0.

Border Size determines the width in pixels of the border created around an image. These pixels are required for certain image processing functions.

New Image is the image structure that will be supplied as an input to all subsequent functions.

Cvi Dispose



Disposes an image and frees the memory allocated for the image. Call Cvi Dispose only when the image is no longer required for the remainder of the processing.

Image is the image to be disposed.

All images? (No) determines whether you want to dispose only the image input into this function or dispose all images. The default value is FALSE.

Cvi ImageToArray

Copies the incoming image to a LabVIEW array. This array can be used for pixel processing or displaying in a LabVIEW intensity graph.

Image is the image to be copied.

Optional Rectangle defines an array of four elements containing the coordinates [Left, Top, Right, Bottom} of the region to extract. The operation will be applied to the entire image if the input is empty or not connected.

Image Pixels returns the extracted pixel values into a 2D array [line, column]. This array is used with 8-bit image type (the only valid type for the PCI-1408).

Attributes

Appendix A

This appendix lists the attributes used with the NI-IMAQ VIs.

Attributes describe a specific property of an interface. A summary of NI-IMAQ VI attributes is listed in Table A-1.

Attribute describes the constant name of the attribute. *Immediate* describes whether the effect of setting the attribute is immediate (Yes), or whether it requires a subsequent call to IMAQ_Init to take effect (No). *R/W* describes whether the attribute is read only (R), write only (W), or both(R/W). *Description* describes what values the attribute can take and the effect the setting of the attribute has or what values are returned.

Table A-1. NI-IMAQ VI Attributes

Attribute	Immediate	R/W	Description
Acquisition in progress	Yes	R	Is an acquisition in progress on the camera associated with this interface? FALSE (0) TRUE (1)
Acquisition window height	No	R/W	Get/set the acquisition window height of the camera/channel associated with this interface
Acquisition window left	No	R/W	Get/set the acquisition window left of the camera/channel associated with this interface
Acquisition window top	No	R/W	Get/set the acquisition window top of the camera/channel associated with this interface
Acquisition window width	No	R/W	Get/set the acquisition window width of the camera/channel associated with this interface
Antichrominance filter	Yes	R/W	Set/get the antichrominance filter to be used: Disabled (0) NTSC-coded signals (1) PAL-coded signals (2)

Table A-1. NI-IMAQ VI Attributes

Attribute	Immediate	R/W	Description
Bits per pixel	Yes	R	Returns the bits per pixel value of the camera/channel associated with this session
Black reference level	Yes	R/W	The black reference value of the channel associated with this interface. Valid values are 0-63.
Buffer list locked	Yes	R	Is the interface's buffer list locked in memory? FALSE (0) TRUE (1)
Buffer X offset	No	R/W	Set/get the buffer x-left offset for image displacement. Use this attribute to acquire an image into a private buffer at a different location other than the top-left corner. You must use a private buffer when using this attribute.
Buffer Y offset	No	R/W	Set/get the buffer y-line offset for image displacement. Use this attribute to acquire an image into a private buffer at a different location other than the top-left corner. You must use a private buffer when using this attribute.
Bytes per pixel	Yes	R	Returns the bytes per pixel value of the camera/channel associated with this interface.
Callback	Yes	R	Not implemented
Color board	Yes	R	Is this board color-capable? FALSE (0) TRUE (1)
Current buffer list ID	Yes	R	Returns the BUFLIST_ID of the buffer list associated with this interface
Diagnostic restore			Not implemented
FIFO overflow			Not implemented
Flip image	No	R/W	Set/get the invert image mode: No invert (0)—image in memory is right-side up Invert (1)—image in memory is upside down

Table A-1. NI-IMAQ VI Attributes

Attribute	Immediate	R/W	Description
Frame count	Yes	R	Returns the number of frames acquired since the start of an acquisition
Frame timeout (ms)	No	R/W	Get/set the timeout value for a frame. Values are given in milliseconds.
Frame/field	Yes	R/W	Set/get the mode of the interface: Field mode (0) Frame mode (1)
Free buffers	Yes	R	Returns the number of reserved driver buffers currently left
Gain	Yes	R/W	Sets the video gain for the channel associated with this interface: 1 (0) 1.33 (1) 2 (2)
Horizontal scale	No	R/W	Set/get the horizontal hardware scaling factor for the channel associated with this session. Values are: None (0) Div2 (2) Div4 (4) Div8 (8)
Input channel	Yes	R	Returns the current channel selected on the interface (0–3)
Interface type	Yes	R	Returns the type of interface: Board interface (0) Other interface (1)
Line count	Yes	R	Returns the current line count of the frame being acquired

Table A-1. NI-IMAQ VI Attributes

Attribute	Immediate	R/W	Description
Look-up table	Yes	R/W	Programs the lookup table for the given interface. Values 0–5 indicate the LUT used: 0—Normal 1—Inverse 2—Log 3—Inverse Log 4—Binary 5—Inverse Binary
Lost frames	Yes	R	Returns the number of retries on invalid frame acquisitions
Max. horizontal size	Yes	R	Returns the maximum horizontal resolution of the interface
Max. vertical size	Yes	R	Returns the maximum vertical resolution of the interface
Number of buffers	Yes	R	Returns the number of buffers in the buffer list associated with the interface
Onboard RAM	Yes	R	Does the interface board have onboard memory? FALSE (0) TRUE (1)
Pixel depth	Yes	R	Returns the maximum pixel depth of the interface board in bytes
RAM size	Yes	R	Returns the size of the RAM on the interface board
Region of interest height	Yes	R/W	Get/set the region of interest height of the camera/channel associated with this interface
Region of interest left	Yes	R/W	Get/set the region of interest left of the camera/channel associated with this interface
Region of interest top	Yes	R/W	Get/set the region of interest top of the camera/channel associated with this interface
Region of interest width	Yes	R/W	Get/set the region of interest width of the camera/channel associated with this interface

Table A-1. NI-IMAQ VI Attributes

Attribute	Immediate	R/W	Description
Rowbytes	No	R/W	Get/set the true width of a horizontal line in memory
Start field	No	R/W	Returns the start field setting of the camera associated with this interface
Trigger mode	No	R/W	Get/set the trigger mode for the channel associated with this session: No trigger (0) Use triggered capture as specified per buffer (1) Use triggered capture as specified per buffer and repeat on last (2)
Valid buffer	Yes	R	Returns a buffer element number of the last received frame buffer
Vertical scale	No	R/W	Set/get the vertical hardware scaling factor for the channel associated with this session. Values are: None (0) Div2 (2) Div4 (4) Div8 (8)
Video type	Yes	R/W	Sets/gets the video type: Interlaced (0) Noninterlaced (progressive scan) (1)
White reference level	Yes	R/W	The white reference value of the channel associated with this interface. Valid values are 0–63.

Error Codes

Appendix B

This appendix lists the error codes for the NI-IMAQ VIs.

Table B-1. LabVIEW NI-IMAQ VI Error Codes

Error Code	Error Name	Description
-1074397137	IMG_ERR_NEPK	No external pixel clock
-1074397138	IMG_ERR_ILCK	Interface locked
-1074397139	IMG_ERR_PLCK	Partial lock—cannot perform acquisition
-1074397140	IMG_ERR_FIFO	FIFO overflow caused acquisition to halt
-1074397141	IMG_ERR_BTAC	No trigger action—acquisition will time out
-1074397145	IMG_ERR_NINF	No interface found
-1074397146	IMG_ERR_BTRG	Trigger loopback problem—cannot drive trigger with action enabled
-1074397147	IMG_ERR_HLPR	Bad parameter to low level—check attributes and high level arguments
-1074397148	IMG_ERR_ZBUF	Zero buffer size—no bytes filled
-1074397149	IMG_ERR_NBUF	No buffers available—too early in acquisition
-1074397150	IMG_ERR_TIMO	Wait timed out—acquisition not complete
-1074397151	IMG_ERR_AIOP	Cannot perform request—acquisition in progress
-1074397152	IMG_ERR_BDMA	Bad DMA transfer
-1074397153	IMG_ERR_VCLK	Cannot get video lock

Table B-1. LabVIEW NI-IMAQ VI Error Codes

Error Code	Error Name	Description
-1074397154	IMG_ERR_NAIP	No acquisition in progress
-1074397155	IMG_ERR_BBLB	A buffer list buffer is null
-1074397156	IMG_ERR_BBLE	Buffer list does contain an invalid command
-1074397157	IMG_ERR_BBLF	Buffer list does not contain a valid final command
-1074397158	IMG_ERR_NCFG	Invalid action—no buffers configured for session
-1074397159	IMG_ERR_NVBL	Not successful because of hardware limitations
-1074397160	IMG_ERR_BCMF	Bad camera file (check syntax)
-1074397161	IMG_ERR_BROI	ROI width is greater than rowbytes
-1074397162	IMG_ERR_BROW	Rowbytes is less than region of interest
-1074397163	IMG_ERR_BINT	Bad interface
-1074397164	IMG_ERR_NCAM	No camera defined for this channel
-1074397165	IMG_ERR_NCLK	Buffer list is not locked
-1074397166	IMG_ERR_BBUF	Bad buffer pointer in list
-1074397167	IMG_ERR_DISE	Error releasing the image buffer
-1074397168	IMG_ERR_ECLK	Cannot lock buffers down, no more memory
-1074397169	IMG_ERR_MXBI	Exhausted buffer id’s
-1074397170	IMG_ERR_BSIZ	Buffer size used is too small for minimum acquisition frame
-1074397171	IMG_ERR_DLLE	DLL internal error, bad logic state
-1074397172	IMG_ERR_MXBF	Too many buffers already allocated
-1074397173	IMG_ERR_PAR7	Function specific; see function description

Table B-1. LabVIEW NI-IMAQ VI Error Codes

Error Code	Error Name	Description
-1074397174	IMG_ERR_PAR6	Function specific; see function description
-1074397175	IMG_ERR_PAR5	Function specific; see function description
-1074397176	IMG_ERR_PAR4	Function specific; see function description
-1074397177	IMG_ERR_PAR3	Function specific; see function description
-1074397178	IMG_ERR_PAR2	Function specific; see function description
-1074397179	IMG_ERR_PAR1	Function specific; see function description
-1074397180	IMG_ERR_OSER	Operating system error occurred
-1074397181	IMG_ERR_EMEM	Not enough memory to perform the operation
-1074397182	IMG_ERR_OVRN	Too many interfaces open
-1074397183	IMG_ERR_NCAP	Function not implemented

Customer Communication

Appendix
C

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve your technical problems and a form you can use to comment on the product documentation. When you contact us, we need the information on the Technical Support Form and the configuration form, if your manual contains one, about your system configuration to answer your questions as quickly as possible.

National Instruments has technical assistance through electronic, fax, and telephone systems to quickly provide the information you need. Our electronic services include a bulletin board service, an FTP site, a fax-on-demand system, and e-mail support. If you have a hardware or software problem, first try the electronic support systems. If the information available on these systems does not answer your questions, we offer fax and telephone support through our technical support centers, which are staffed by applications engineers.

Electronic Services

Bulletin Board Support

National Instruments has BBS and FTP sites dedicated for 24-hour support with a collection of files and documents to answer most common customer questions. From these sites, you can also download the latest instrument drivers, updates, and example programs. For recorded instructions on how to use the bulletin board and FTP services and for BBS automated information, call (512) 795-6990. You can access these services at:

United States: (512) 794-5422
Up to 14,400 baud, 8 data bits, 1 stop bit, no parity

United Kingdom: 01635 551422
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

France: 01 48 65 15 59
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

FTP Support

To access our FTP site, log on to our Internet host, `ftp.natinst.com`, as `anonymous` and use your Internet address, such as `joesmith@anywhere.com`, as your password. The support files and documents are located in the `/support` directories.

Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at (512) 418-1111.

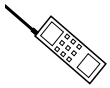
E-Mail Support (currently U.S. only)

You can submit technical support questions to the applications engineering team through e-mail at the Internet address listed below. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

`support@natinst.com`

Telephone and Fax Support

National Instruments has branch offices all over the world. Use the list below to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.



	Telephone	Fax
Australia	03 9879 5166	02 9874 4455
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Canada (Ontario)	905 785 0085	905 785 0086
Canada (Quebec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	09 725 725 11	09 725 725 55
France	01 48 14 24 24	01 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Israel	03 5734815	03 5734816
Italy	02 413091	06 57284309
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	5 520 2635	5 520 3282
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
United States	512 794 0100	512 794 8411
United Kingdom	01635 523545	01635 523154

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (___)_____ Phone (___) _____

Computer brand _____ Model _____ Processor_____

Operating system (include version number) _____

Clock speed _____MHz RAM _____MB Display adapter _____

Mouse ___yes ___no Other adapters installed _____

Hard disk capacity _____MB Brand _____

Instruments used _____

_____

National Instruments hardware product model _____ Revision _____

Configuration _____

National Instruments software product _____ Version _____

Configuration _____

The problem is: _____

_____

_____

_____

_____

List any error messages: _____

_____

_____

The following steps reproduce the problem:_____

_____

_____

_____

_____

_____

IMAQ Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

DAQ hardware _____

Interrupt level of hardware _____

DMA channels of hardware _____

Base I/O address of hardware _____

Programming choice _____

NI-IMAQ, IMAQ Vision, or LabVIEW version _____

Other boards in system _____

Base I/O address of other boards _____

DMA channels of other boards _____

Interrupt level of other boards _____

Other Products

Computer make and model _____

Microprocessor _____

Clock frequency or speed _____

Type of video board installed _____

Operating system version _____

Operating system mode _____

Programming language _____

Programming language version _____

Other boards in system _____

Base I/O address of other boards _____

DMA channels of other boards _____

Interrupt level of other boards _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: *NI-IMAQ VIs for G Reference Manual*

Edition Date: June 1997

Part Number: 371642A-01

Please comment on the completeness, clarity, and organization of the manual.

_____

_____

_____

_____

_____

_____

_____

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

Phone (___)_____ Fax (___) _____

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
(512) 794-5678

Prefix	Meaning	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}

Numbers/Symbols

%	percent
+	positive of, or plus
+5V	5 V signal
-	negative of, or minus
/	per
±	plus or minus
Ω	ohm

A

A	amperes
AC	alternating current
acquisition window	the image size specific to a video standard or camera resolution
active line region	the region of lines actively being stored; defined by a line start (relative to VSYNC) and a line count
active pixel region	the region of pixels actively being stored; defined by a pixel start (relative to HSYNC) and a pixel count
A/D	analog-to-digital
ADC	analog-to-digital converter—an electronic device, often an integrated circuit, that converts an analog voltage to a digital number
address	character code that identifies a specific location (or series of locations) in memory
ANSI	American National Standards Institute
antichrominance filter	removes the color information from the video signal
API	application programming interface
AQ_DONE	signals that the acquisition of a frame or field is completed
AQ_IN_PROGRESS	signals that the acquisition of video data is in progress
area	a rectangular portion of an acquisition window or frame that is controlled and defined by software
array	ordered, indexed set of data elements of the same type
ASIC	Application-Specific Integrated Circuit—a proprietary semiconductor component designed and manufactured to perform a set of specific functions for a specific customer
aspect ratio	the ratio of a signal's width to its height

B

b	bit—one binary digit, either 0 or 1
B	byte—eight related bits of data, an eight-bit binary number; also used to denote the amount of memory required to store one byte of data
back porch	the area of the video signal between the rising edge of the horizontal sync signal and the active video information
black reference level	the level that represents the darkest an image can get *See also* white reference level.
buffer	temporary storage for acquired data
bus	the group of conductors that interconnect individual circuitry in a computer, such as the PCI bus; typically the expansion vehicle to which I/O or other devices are connected

C

C	Celsius
cache	high-speed processor memory that buffers commonly used instructions or data to increase processing throughput
CCIR	Comite Consultatif International des Radiocommunications—a committee that developed standards for color video signals
chrominance	the color information in a video signal
CMOS	complementary metal-oxide semiconductor
compiler	a software utility that converts a source program in a high-level programming language, such as Basic, C or Pascal, into an object or compiled program in machine language. Compiled programs run 10 to 1,000 times faster than interpreted programs *See also* Interpreter.
conversion device	device that transforms a signal from one form to another; for example, analog-to-digital converters (ADCs) for analog input and digital-to-analog converters (DACs) for analog output
CPU	central processing unit

CSYNC	composite sync signal; a combination of the horizontal and vertical sync pulses
CSYNCIN	composite sync in signal
CSYNCOUT	composite sync out signal

D

D/A	digital-to-analog
DAC	digital-to-analog converter; an electronic device, often an integrated circuit, that converts a digital number into a corresponding analog voltage or current
DAQ	data acquisition—(1) collecting and measuring electrical signals from sensors, transducers, and test probes or fixtures and inputting them to a computer for processing; (2) collecting and measuring the same kinds of electrical signals with A/D or DIO boards plugged into a computer, and possibly generating control signals with D/A and/or DIO boards in the same computer
dB	decibel; the unit for expressing a logarithmic measure of the ratio of two signal levels: dB=20log10 V1/V2, for signals in volts
DC	direct current
default setting	a default parameter value recorded in the driver; in many cases, the default input of a control is a certain value (often 0) that means *use the current default setting*.
DIN	Deutsche Industrie Norme
DMA	direct memory access—a method by which data can be transferred to and from computer memory from and to a device or memory on the bus while the processor does something else; DMA is the fastest method of transferring data to/from computer memory
DRAM	dynamic RAM
drivers	software that controls a specific hardware device such as an IMAQ or DAQ device

dynamic range	the ratio of the largest signal level a circuit can handle to the smallest signal level it can handle (usually taken to be the noise level), normally expressed in dB

E

EEPROM	electrically erasable programmable read-only memory—ROM that can be erased with an electrical signal and reprogrammed
external trigger	a voltage pulse from an external source that triggers an event such as A/D conversion

F

field	For an interlaced video signal, a field is half the number of horizontal lines needed to represent a frame of video; the first field of a frame contains all the odd-numbered lines, the second field contains all of the even-numbered lines.
FIFO	first-in first-out memory buffer—the first data stored is the first data sent to the acceptor; FIFOs are used on IMAQ devices to temporarily store incoming data until that data can be retrieved. For example, an analog input FIFO stores the results of A/D conversions until the data can be retrieved into system memory, a process that requires the servicing of interrupts and often the programming of the DMA controller. This process can take several milliseconds in some cases. During this time, data accumulates in the FIFO for future retrieval.
flash ADC	an ADC whose output code is determined in a single step by a bank of comparators and encoding logic
frame	a complete image; in interlaced formats, a frame is composed of two fields
front porch	the area of a video signal between the start of the horizontal blank and the start of the horizontal sync
ft	feet

function — a set of software instructions executed by a single line of code that may have input and/or output parameters and returns a value when executed; examples of functions are:

y = COS (x)
status = `AO_config`(**board, channel, range**)

G

gamma — the nonlinear change in the difference between the video signal's brightness level and the voltage level needed to produce that brightness

genlock — circuitry that aligns the video timing signals by locking together the horizontal, vertical, and color subcarrier frequencies and phases and generates a pixel clock to clock pixel data into memory for display or into another circuit for processing

GND — ground signal

GUI — graphical user interface—an intuitive, easy-to-use means of communicating information to and from a computer program by means of graphical screen displays; GUIs can resemble the front panels of instruments or other objects associated with a computer program.

H

h — hour

hardware — the physical components of a computer system, such as the circuit boards, plug-in boards, chassis, enclosures, peripherals, cables, and so on

HSYNC — horizontal sync signal—the synchronization pulse signal produced at the beginning of each video scan line that keeps a video monitor's horizontal scan rate in step with the transmission of each new line

HSYNCIN — horizontal sync input signal

Hz — hertz—the number of scans read or updates written per second

I

IC	integrated circuit
ID	identification
IEEE	Institute of Electrical and Electronics Engineers
`IMAQconf`	a configuration and diagnostic utility included with IMAQ devices
in.	inches
INL	integral nonlinearity—A measure in LSB of the worst-case deviation from the ideal A/D or D/A transfer characteristic of the analog I/O circuitry
interlaced	a video frame composed of two interleaved fields; the number of lines in a field are half the number of lines in an interlaced frame
instrument driver	a set of high-level software functions, such as NI-IMAQ, that controls specific plug-in computer boards; instrument drivers are available in several forms, ranging from a function callable from a programming language to a virtual instrument (VI) in LabVIEW
interpreter	a software utility that executes source code from a high-level language such as Basic, C or Pascal, by reading one line at a time and executing the specified operation *See also* compiler.
interrupt	a computer signal indicating that the CPU should suspend its current task to service a designated activity
interrupt level	the relative priority at which a device can interrupt
I/O	input/output—the transfer of data to/from a computer system involving communications channels, operator interface devices, and/or data acquisition and control interfaces
IRQ	interrupt request

K

k	kilo—the standard metric prefix for 1,000, or 10^3, used with units of measure such as volts, hertz, and meters
K	kilo—the prefix for 1,024, or 2^{10}, used with B in quantifying data or computer memory
kbytes/s	a unit for data transfer that means 1,000 or 10^3 bytes/s
Kword	1,024 words of memory

L

library	a file containing compiled object modules, each comprised of one of more functions, that can be linked to other object modules that make use of these functions.
line count	the total number of horizontal lines in the picture
LSB	least significant bit
LUT	look-up table—a selection in the `IMAQconf` configuration utility that contains formulas that let you implement simple imaging operations such as contrast enhancement, data inversion, gamma manipulation, or other nonlinear transfer functions

M

m	meters
M	(1) Mega, the standard metric prefix for 1 million or 10^6, when used with units of measure such as volts and hertz; (2) mega, the prefix for 1,048,576, or 2^{20}, when used with B to quantify data or computer memory
MB	megabytes of memory
Mbytes/s	a unit for data transfer that means 1 million or 10^6 bytes/s
memory buffer	*See* buffer.

memory window	continuous blocks of memory that can be accessed quickly by changing addresses on the local processor
MSB	most significant bit
MTBF	mean time between failure
mux	multiplexer—a switching device with multiple inputs that selectively connects one of its inputs to its output

N

NI-IMAQ	driver software for National Instruments IMAQ hardware
noninterlaced	a video frame where all the lines are scanned sequentially, instead of divided into two frames as in an interlaced video frame
NTSC	National Television Standards Committee—the committee that developed the color video standard used primarily in North America, which uses 525 lines per frame. *See also* PAL.
NVRAM	nonvolatile RAM—RAM that is not erased when a device loses power or is turned off

O

operating system	base-level software that controls a computer, runs programs, interacts with users, and communicates with installed hardware or peripheral devices

P

PAL	Phase Alternation Line—one of the European video color standards; uses 625 lines per frame. *See also* NTSC.
PCI	Peripheral Component Interconnect—a high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA; it is achieving widespread acceptance as a standard for PCs and workstations and offers a theoretical maximum transfer rate of 132 Mbytes/s
PCLK	pixel clock signal—times the sampling of pixels on a video line

PCLKIN	pixel clock in signal
PFI	programmable function input
PGIA	programmable gain instrumentation amplifier
picture aspect ratio	the ratio of the active pixel region to the active line region; for standard video signals like RS-170 or CCIR, the full-size picture aspect ratio normally is 4/3 (1.33)
pixel	picture element—the smallest division that makes up the video scan line; for display on a computer monitor, a pixel's optimum dimension is square (aspect ratio of 1:1, or the width equal to the height)
pixel aspect ratio	the ratio between the physical horizontal size and the vertical size of the region covered by the pixel; an acquired pixel should optimally be square, thus the optimal value is 1.0, but typically it falls between 0.95 and 1.05, depending on camera quality
pixel clock	divides the incoming horizontal video line into pixels
pixel count	the total number of pixels between two HYSNCs; the pixel count determines the frequency of the pixel clock
PLL	phase-locked loop—circuitry that provides a very stable pixel clock that is referenced to another signal, for example, an incoming HSYNC signal
protocol	the exact sequence of bits, characters, and control codes used to transfer data between computers and peripherals through a communications channel
pts	points

R

RAM	random-access memory
real time	a property of an event or system in which data is processed as it is acquired instead of being accumulated and processed at a later time
relative accuracy	a measure in LSB of the accuracy of an ADC; it includes all nonlinearity and quantization errors but does not include offset and gain errors of the circuitry feeding the ADC

resolution	the smallest signal increment that can be detected by a measurement system; resolution can be expressed in bits, in proportions, or in percent of full scale. For example, a system has 12-bit resolution, one part in 4,096 resolution, and 0.0244 percent of full scale.
ribbon cable	a flat cable in which the conductors are side by side
ROI	region-of-interest; a hardware-programmable rectangular portion of the acquisition window
ROM	read-only memory
RS-170	the U.S. standard used for black-and-white television
RTSIbus	Real-Time System Integration Bus—the National Instruments timing bus that connects IMAQ and DAQ boards directly, by means of connectors on top of the boards, for precise synchronization of functions

S

s	seconds
scaling down circuitry	circuitry that scales down the resolution of a video signal
scatter-gather DMA	a type of DMA that allows the DMA controller to reconfigure on-the-fly
SRAM	static RAM
sync	tells the display where to put a video picture; the horizontal sync indicates the picture's left-to-right placement and the vertical sync indicates top-to-bottom placement
syntax	the set of rules to which statements must conform in a particular programming language
system RAM	RAM installed on a personal computer and used by the operating system, as contrasted with onboard RAM

T

transfer rate	the rate, measured in bytes/s, at which data is moved from source to destination after software initialization and set up operations; the maximum rate at which the hardware can operate
TRIG	trigger signal
trigger	any event that causes or starts some form of data capture
trigger control and mapping circuitry	circuitry that routes, monitors, and drives the external and RTSIbus trigger lines; you can configure each of these lines to start or stop acquisition on a rising or falling edge.
TTL	transistor-transistor logic

V

V	volts
VCO	voltage-controlled oscillator—an oscillator that changes frequency depending on a control signal; used in a PLL to generate a stable pixel clock
VI	Virtual Instrument—(1) a combination of hardware and/or software elements, typically used with a PC, that has the functionality of a classic stand-alone instrument (2) a LabVIEW software module (VI), which consists of a front panel user interface and a block diagram program
video line	a video line consists of a HSYNC, back porch, active pixel region, and a front porch
VSYNC	vertical sync signal—the synchronization pulse generated at the beginning of each video field that tells the video monitor when to start a new field
VSYNCIN	vertical sync in signal

W

white reference level	the level that defines what is white for a particular video system *See also* black reference level.

Index

L

N

O

P

R

S

T