

# NI-IMAQdx<sup>™</sup> Help

#### June 2008, 372214B-01

This help file describes the NI-IMAQdx driver software, lists the supported application development environments (ADEs), describes the fundamentals of creating applications using NI-IMAQdx, describes the files used to build these applications, and explains where to find sample programs. NI-IMAQdx gives you the ability to use GigE Vision cameras and IEEE 1394 industrial digital video cameras to acquire images.

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

#### Using Help

**Related Documentation** 

**Important Information** 

**Technical Support and Professional Services** 

To comment on National Instruments documentation, refer to the <u>National</u> <u>Instruments Web site</u>.

© 2007–2008 National Instruments Corporation. All rights reserved.

# **Activating Your Software**

#### How do I activate my software?

Use the NI Activation Wizard to obtain an activation code for your software. You can launch the NI Activation Wizard two ways:

- Launch the product and choose to activate your software from the list of options presented.
- Launch NI License Manager by selecting Start»All Programs»National Instruments»NI License Manager. Click the Activate button in the toolbar.
- Note If your software is a part of a Volume License Agreement (VLA), contact your VLA administrator for installation and activation instructions.

### What is activation?

Activation is the process of obtaining an activation code to enable your software to run on your computer. An *activation code* is an alphanumeric string that verifies the software, version, and computer ID to enable features on your computer. Activation codes are unique and are valid on only one computer.

### What is the NI Activation Wizard?

The NI Activation Wizard is a part of NI License Manager that steps you through the process of enabling software to run on your machine.

### What information do I need to activate?

You need your product serial number, user name, and organization. The NI Activation Wizard determines the rest of the information. Certain activation methods may require additional information for delivery. This information is used only to activate your product. Complete disclosure of National Instruments licensing privacy policy is available at ni.com/activate/privacy. If you optionally choose to register your software, your information is protected under the National Instruments privacy policy, available at ni.com/privacy.

### How do I find my product serial number?

Your serial number uniquely identifies your purchase of NI software. You can find your serial number on the Certificate of Ownership included in

your software kit. If your software kit does not include a Certificate of Ownership, you can find your serial number on the product packing slip or on the shipping label.

If you have installed a previous version using your serial number, you can find the serial number by selecting the **Help**»**About** menu item within the application or by selecting your product within NI License Manager (**Start**»**All Programs**»**National Instruments**»**NI License Manager**). You can also contact your local National Instruments <u>branch</u>.

#### What is a Computer ID?

The computer ID contains unique information about your computer. National Instruments requires this information to enable your software. You can find your computer ID through the NI Activation Wizard or by using NI License Manager, as follows:

- 1. Launch NI License Manager by selecting **Start»All Programs»National Instruments»NI License Manager**.
- 2. Click the **Display Computer Information** button in the toolbar.

For more information about product activation and licensing refer to ni.com/activate.

## **Related Documentation**

NI-IMAQdx manuals also are available as PDFs. You must have Adobe Reader with Search and Accessibility 5.0.5 or later installed to view the PDFs. Refer to the <u>Adobe Systems Incorporated Web site</u> at www.adobe.com to download Adobe Reader. Refer to the <u>National</u> <u>Instruments Product Manuals Library</u> at ni.com/manuals for updated documentation resources.

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

- *NI Vision Acquisition Software Release Notes*—Contains information about new functionality, minimum system requirements, and installation instructions for NI-IMAQdx driver software.
- *NI-IMAQdx VI Reference Help*—Contains reference information about the LabVIEW palettes and VIs for NI-IMAQdx driver software.
- *NI-IMAQdx Function Reference Help*—Contains reference information about the LabWindows™/CVI™ functions for NI-IMAQdx driver software.

## **Using Help**

<u>Conventions</u> <u>Navigating Help</u> <u>Searching Help</u> <u>Printing Help File Topics</u>

# Conventions

This help file uses the following formatting and typographical conventions:

»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence <b>File»Page Setup»Options</b> directs you to pull down the <b>File</b> menu, select the <b>Page Setup</b> item, and select <b>Options</b> from the last dialog box.	
	This icon denotes a note, which alerts you to important information.	
	This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.	
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.	
<u>green</u>	Underlined text in this color denotes a link to a help topic, help file, or Web address.	
italic	Italic text denotes variables, emphasis, cross-references, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.	
monospace	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.	
monospace bold	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.	
monospace italic	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.	

# **Navigating Help (Windows Only)**

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window or use the following toolbar buttons located above the tabs:

- **Hide**—Hides the navigation pane from view.
- Locate—Locates the currently displayed topic in the Contents tab, allowing you to view related topics.
- **Back**—Displays the previously viewed topic.
- Forward—Displays the topic you viewed before clicking the **Back** button.
- **Options**—Displays a list of commands and viewing options for the help file.

# **Searching Help (Windows Only)**

Use the **Search** tab to the left of this window to locate content in this help file. If you want to search for words in a certain order, such as "related documentation," add quotation marks around the search words as shown in the example. Searching for terms on the **Search** tab allows you to quickly locate specific information and information in topics that are not included on the **Contents** tab.

### Wildcards

You also can search using asterisk (\*) or question mark (?) wildcards. Use the asterisk wildcard to return topics that contain a certain string. For example, a search for "prog\*" lists topics that contain the words "program," "programmatically," "progress," and so on.

Use the question mark wildcard as a substitute for a single character in a search term. For example, "?ext" lists topics that contain the words "next," "text," and so on.



**Note** Wildcard searching will not work on Simplified Chinese, Traditional Chinese, Japanese, and Korean systems.

### **Nested Expressions**

Use nested expressions to combine searches to further refine a search. You can use Boolean expressions and wildcards in a nested expression. For example, "example AND (program OR VI)" lists topics that contain "example program" or "example VI." You cannot nest expressions more than five levels.

### **Boolean Expressions**

Click the **•** button to add Boolean expressions to a search. The following Boolean operators are available:

- **AND** (default)—Returns topics that contain both search terms. You do not need to specify this operator unless you are using nested expressions.
- **OR**—Returns topics that contain either the first or second term.
- **NOT**—Returns topics that contain the first term without the second term.
- **NEAR**—Returns topics that contain both terms within eight words of each other.

### **Search Options**

Use the following checkboxes on the **Search** tab to customize a search:

- **Search previous results**—Narrows the results from a search that returned too many topics. You must remove the checkmark from this checkbox to search all topics.
- Match similar words—Broadens a search to return topics that contain words similar to the search terms. For example, a search for "program" lists topics that include the words "programs," "programming," and so on.
- Search titles only—Searches only in the titles of topics.

# **Printing Help File Topics (Windows Only)**

Complete the following steps to print an entire book from the **Contents** tab:

- 1. Right-click the book.
- 2. Select **Print** from the shortcut menu to display the **Print Topics** dialog box.
- 3. Select the **Print the selected heading and all subtopics** option.
  - Note Select Print the selected topic if you want to print the single topic you have selected in the **Contents** tab.
- 4. Click the **OK** button.

### **Printing PDF Documents**

This help file may contain links to PDF documents. To print PDF documents, click the print button located on the Adobe Acrobat Viewer toolbar.

## About the NI-IMAQdx Software

You can use cameras with the following output formats.

- Monochrome (8–16-bits/pixel)
- RGB (24-48-bits/pixel)
- YUV 4:1:1 (12-bits/pixel)
- YUV 4:2:2 (16-bits/pixel)
- YUV 4:4:4 (24-bits/pixel)
- Bayer (8–16-bits/pixel)

The cameras may operate at various resolutions and frame rates, depending on camera capabilities.

NI-IMAQdx complies with the Automated Imaging Association GigE Vision specification and the 1394 Trade Association Industrial and Instrumentation specification for Digital Cameras (IIDC), and controls all available modes and features of the digital camera.



**Note** Refer to the *NI Vision Acquisition Software Release Notes* for the specific version of the GigE Vision or the IIDC specification specification to which this driver complies.

# **Application Development Environments**

This release of NI-IMAQdx supports the following Application Development Environments for Windows Vista/XP/2000:

- LabVIEW version 7.1.1 and later
- LabVIEW Real-Time Module 7.1.1 and later
- LabWindows™/CVI™ 7.0 and later
- Microsoft Visual C/C++ 6.0 and later
- Microsoft Visual Basic 6.0 and later
- Microsoft Visual Studio .NET 2003 and later
- Note Although the NI-IMAQdx software has been tested and found to work with these ADEs, other ADEs may also work.

# **Configuring Your Camera**

Use National Instruments Measurement & Automation Explorer (MAX) to configure your camera. Refer to *Measurement & Automation Explorer Help for NI-IMAQdx* for information about configuring your camera. You can access *Measurement & Automation Explorer Help for NI-IMAQdx* from within MAX by selecting **Help \*Help Topics\*NI-IMAQdx**.

The camera configuration is saved in a camera file, which the NI-IMAQdx VIs and functions use to configure a camera and supported attributes.

### **Camera Session**

To acquire images using the high-level or low-level functions, you first must open a camera session. A camera session is a <u>Process-safe</u> <u>Handle</u> to a camera.

The driver uses a camera session to identify the camera to which further NI-IMAQdx functions apply. You can simultaneously open as many camera sessions as there are cameras connected to your system.

### **Camera files**

Camera files store all the configurable attributes. Camera files can be shared between identical cameras. Use MAX to configure the default state of a particular camera.

# Fundamentals of Building Applications with NI-IMAQdx

## Architecture

The following figure illustrates the NI-IMAQdx driver architecture:



The architecture uses a hardware abstraction layer, which separates software API capabilities, such as general acquisition and control functions, from hardware-specific information. This layer lets you run your application on different operating systems and use updated versions of the driver without having to recompile your application.

### **NI-IMAQdx Libraries**

The NI-IMAQdx function libraries are dynamic link libraries (DLLs), which means that NI-IMAQdx routines are not linked into the executable files of applications. Only the information about the NI-IMAQdx routines in the NI-IMAQdx import libraries is stored in the executable files.

Import libraries contain information about their DLL-exported functions. They indicate the presence and location of the DLL routines. Depending on the development tools you use, you can give the DLL routines information through import libraries or through function declarations. Your NI-IMAQdx software contains function prototypes for all routines.

## **Example Programs**

You can find NI-IMAQdx code examples in the following directories.

**Note** If you installed NI-IMAQdx in the default location, you can find the following example directories within C:\Program Files\National Instruments.

• LabVIEW—LabVIEW\examples\imaq. For a brief description of any example VI, open the VI, and select Windows»Show VI Info for a text description of the example



- **CVI**—CVI\samples\imaqdx
- **C**—NI-IMAQdx\examples\MSVC
- Visual Basic— NI-IMAQdx\examples\VB
- Microsoft Visual Studio .NET 2003—NI-IMAQdx\examples\ MSVB.NET. The .NET examples are converted from the NI-IMAQdx for Visual Basic examples. The .NET examples are written in Visual Basic .NET and demonstrate use of the NI-IMAQdx 3.0 Assemblies and the IMAQ Vision 8.2 Viewer control.

Refer to the readme.rtf file located in your target installation directory for the latest details about the example programs.

## **Basic Acquisition with NI-IMAQdx**

This section contains an overview of the NI-IMAQdx library, a description of the acquisition flow of NI-IMAQdx, and generic programming examples. The chapter also contains flowcharts of high-level and lowlevel snap, grab, and sequence operations.

## **Application Programming Interface**

The NI-IMAQdx application programming interface (API) is divided into two main function groups:

- High-Level Functions
- Low-Level Functions

Both high-level and low-level functions support snap, grab, sequence, and triggered acquisitions. Using high-level functions, you can write programs quickly without having to learn the details of the low-level API and driver. The low-level functions give you finer granularity and control over the image acquisition process, but you must understand the API and driver in greater detail to use these functions.



**Note** The high-level functions call low-level functions and use certain attributes that are listed in the high-level function description of the *NI-IMAQdx Function Reference Help*. Changing the value of these attributes while using low-level functions affects the operation of the high-level functions.

# **High-Level Functions**

High-level functions are used to capture images quickly and easily. If you need more advanced functionality, you can mix high-level functions with low-level functions. Using high-level functions, you can write programs quickly without having to learn the details of the low-level API and driver.

The different types of high-level functions are:

- **Snap Functions**—Capture all or a portion of a single image to the user buffer.
- **Grab Functions**—Perform an acquisition that loops continually on one or more internal buffers. You can copy the last acquired buffer to a separate user buffer for processing or analysis.
- **Sequence Functions**—Acquire a specified number of internal buffers and then stops.

## **Low-Level Functions**

Low-level functions are used when you require more direct control of the image acquisition. The low-level functions give you finer granularity and control over the image acquisition process, but you must understand the API and driver in greater detail to use these functions.

- Acquisition Functions—After configuring and starting your acquisition, the camera sends data to the internal buffers. To process the acquired image data, you must copy the data from the internal buffer into your user buffer.
- Attribute Functions—Attribute functions examine and change the acquisition or camera attributes.
- Event Functions—Event functions register events.
- **Register Functions**—Register functions access registers. Refer to <u>Register-level Programming</u> for more information about programming registers.
- **Session Functions**—Session functions open, close, or enumerate sessions.
- **Utility Functions**—Utility functions provide detailed error information.

# **Acquisition Flow**

This section describes the basic steps of performing an acquisition with the NI-IMAQdx software. The basic steps are initialization, configuration, and acquisition.

## **Opening a Camera Session**

To acquire images using the high-level or low-level functions, you first must open a camera session. A camera session is a process-safe handle to a camera. A process-safe handle is a handle that allows only one process to access a camera at any given time. The driver uses a camera session to identify the camera to which further NI-IMAQdx functions apply. You can simultaneously open as many camera sessions as there are cameras connected to your system.

When opening the camera session, you need to specify two parameters: camera name and camera control mode.

#### **Camera Name**

NI-IMAQdx references all camera sessions by a name. The driver creates default names for each camera in your system in the order that the cameras are connected. The names observe the convention shown in the following table.

Camera Name	<b>Camera Installed</b>
cam0	Device 0
cam1	Device 1
cam <i>n</i>	Device <i>n</i>

Every camera has an .iid interface file and an .icd camera file.

- **Interface files**—Interface files store information about which physical camera is associated with a camera name. Each interface file can be used by only a single camera.
- **Camera files**—Camera files store all the configurable attributes. Camera files can be shared between identical cameras. Use MAX to configure the default state of a particular camera.

### **Camera Control Mode**

The camera control mode parameter has two options: **controller** and **listener**. The default option—**controller**—controls the camera and receives video data. The listener only receives video data. Use the listener option in broadcasting applications.

Refer to Broadcasting in Advanced Programming with NI-IMAQdx for

more information about broadcasting.

When an application is finished with the camera, call the Close Camera function to close the camera session.

# **Configure the Acquisition**

After opening the camera, configure the camera for acquisition by specifying the following parameters: whether the acquisition is one-shot or continuous, and the number of internal buffers to use.

While configuring the camera, the driver validates all the userconfigurable attributes. If any attributes are invalid or out of range, the driver returns an error and does not configure the acquisition.

If you want to reconfigure the acquisition, call the Unconfigure Acquisition function before calling the Configure function again.



**Note** National Instruments recommends that you do not configure an acquisition in a loop because doing so is time-intensive.

### **One-Shot Acquisition**

Use a one-shot acquisition to start an acquisition, perform the acquisition, and stop the acquisition using a single function. The number of images acquired is equal to the number of images in the images collection.

With a one-shot acquisition, you specify a certain number of internal buffers. The camera transfers each image up to and including the specified number of buffers. The driver acquires every image during a one-shot acquisition. National Instruments recommends one-shot acquisition for applications that do not require real-time acquisition or processing.

## **Continuous Acquisition**

Use a continuous acquisition to start an acquisition, continuously acquire images into the internal buffers, and explicitly stop the acquisition. With continuous acquisition, the driver acquires video data continuously from the camera and enables you to examine the most current buffer. National Instruments recommends continuous acquisition for real-time acquisition and processing.



**Note** If CPU activity increases during a continuous acquisition, the driver might miss subsequent images. Check the buffer number output to determine if you have missed any images.

### **Number of Buffers**

Another aspect of configuration is specifying the number of internal buffers into which you want to acquire image data. During configuration, buffers are allocated from system memory and page-locked. Once the acquisition starts, the camera transfers video data over the IEEE 1394 bus to the IEEE 1394 interface card FIFO. Then, video data is directly transferred to the internal buffer. This transfer requires negligible CPU resources. For the GigE Vision bus, CPU resources are used to pass network packets. For ethernet, ethernet packets are evaluated by software and copied into an internal buffer.

Each internal buffer you allocate is the exact size of the raw data being transmitted by the camera. For continuous acquisitions, allocate three or more buffers. Allocating a single buffer for a continuous acquisition may result in a high number of lost images. For one-shot acquisitions, specify the number of buffers that the application requires. For example, if the application runs for two seconds, and the camera acquires at 30 frames per second, allocate 60 buffers to capture each image.

Having more buffers available for the GigE Vision bus can increase the reliability of data transfer, especially when there are adverse network conditions.

### **Region of Interest**

The region of interest (ROI) specifies a rectangular portion of the image to be captured. If the camera supports scalable video modes, the ROI defines the portion of the image to transfer from the camera to system memory.

If the camera does not support scalable video modes, the entire image is transferred from the camera to system memory. In all video modes, the ROI specifies the amount of data decoded by the driver while acquiring into a user buffer.

By default, the driver transfers the entire image. Specify a smaller ROI for the following reasons.

- To acquire only the necessary subset of data
- To increase the acquisition speed by reducing the amount of data transferred and/or decoded
- To allow for multiple simultaneous acquisitions by reducing bandwidth usage
  - Note Although you can specify an ROI of any size, the NI-IMAQdx software coerces the ROI into one that is more compatible for the given camera. Refer to the section Advanced Programming with NI-IMAQdx, for more information about defining an ROI for scalable images.

### **Pixel Format**

The pixel format specifies the source type of the image format. Different pixel formats decode into different image types. Refer to the <u>Acquisition</u> section for more information on decoding.
# Relationship between Cameras, Interface Files, and Camera Files

The following figure shows the relationship between cameras, interface files, and camera files.



When you open a camera session with the Camera Open function, the camera with the unique serial number described by the interface file cam*n*.iid opens, where *n* is the reference to the camera. If the camera is not present and a camera of the same make and model is present, as described in the interface file, the driver opens the available camera. The interface file updates to use the new camera. The camera file described by the interface file opens, and all the user attributes are set in the driver. If no camera of the same make and model is present, the Camera Open function returns an error.

Note Use the Enumerate function to query the number and names of available cameras. Use the Discover Ethernet Cameras function to find ethernet cameras on the network with a remote subnet.

# Acquisition

After configuring and starting your acquisition, the camera sends data to the internal buffers. To process the acquired image data, you must copy the data from the internal buffer into your user buffer.

## **User Buffer**

Before starting the acquisition, you must allocate a user buffer in addition to configuring internal buffers. The driver copies or decodes image data from the internal buffer into the user buffer during acquisition. Then, process and analyze the image in the user buffer.

When acquiring data into an image, the driver resizes and casts the image as needed. However, if you acquire data into a user buffer, you must allocate enough space for one decoded image.



**Note** Unlike internal buffers, you are responsible for destroying user buffers.

#### **Internal Buffer**

Memory page that is marked as non-pageable by the virtual file system. Internal buffers remain in physical memory and do not cause page faults. They are also known as a page-locked buffer.

#### **Page-Locked Buffers**

Memory page that is marked as non-pageable by the virtual file system. Page-locked buffers remain in physical memory and do not cause page faults. Also known as internal buffers.

## **Buffer Number Mode**

Specify one of the following options for the buffer number mode.

- **Buffer number**—Gets the exact buffer number specified in the Buffer Number parameter.
- **Next**—Gets the next incoming buffer.
- Last—Gets the most recently acquired buffer.

## **Buffer Number**

Buffer number gets the exact buffer number specified in the buffer number parameter.

A buffer number is a zero-based index that represents the cumulated transferred image count. For example, during a continuous acquisition with three internal buffers, the buffer number is updated as follows: 0, 1, 2, 3, 4, 5, and so on. Buffer numbers 0 and 3 refer to the same internal buffer in the buffer ring.

For a one-shot acquisition, you can request only one of the available buffer numbers. For a continuous acquisition, you can request any present or future buffer number. You can also request the next logical buffer or the buffer containing the most recently acquired data. With highlevel grab acquisitions, the buffer number defaults to the next transferred buffer.

When you complete the buffer acquisition step, the driver returns the actual buffer number with the image.

#### **Overwrite Mode**

Ideally, a continuous acquisition acquires and processes every image that is transferred from the camera. However, because of processing time fluctuations, some images from the camera may not be processed before the camera transfers the next image. Using multiple internal buffers in a continuous acquisition allows for a small amount of jitter. However, if a delay becomes too long, the camera overwrites the requested buffer with new image data.

#### Timeout

A timeout is the length of time, in milliseconds, that the driver waits for an image from the camera before returning an error. A timeout error usually occurs if the camera has been removed from the system or when the camera did not receive an external trigger signal.

## Decoding

Except for 8-bit monochrome images, all video modes require decoding before you can interpret the image data. For example, many color cameras output images of type YUV 4:2:2. However, NI Vision does not natively support the YUV mode. To process and display the image, the driver automatically decodes the YUV image into a 32-bit RGB image.

The following table lists common video modes and their corresponding image types after being decoded by NI-IMAQdx.

Raw Camera Output	Image_U8
8-bit monochrome	Image_U8
10–16-bit monochrome	Image_I16
YUV 4:1:1	Image_RGB
YUV 4:2:2	Image_RGB
YUV 4:4:4	Image_RGB
24-bit RGB	Image_RGB
30–48-bit RGB	Image_RGB_U64
8-bit Bayer	Image_RGB
10–16-bit Bayer	Image_RGB

Decoding images requires CPU resources. However, many of the decoding algorithms have been optimized in the driver. If you do not want decoded image data, you can use NI-IMAQdx to get a copy of the raw camera output.

## **High-level Function Programming Examples**

This section contains examples of high-level and low-level image acquisitions. Refer to <u>Fundamentals of Building Applications with NI-</u>IMAQdx section for directory paths to the code examples discussed in this section.

## **High-Level Snap Functions**

Capture all or a portion of a single image to the user buffer. The following figure illustrates the typical programming order of a high-level snap acquisition.



Use a snap for low-speed or one-shot applications where ease of programming is essential. When you invoke a snap, the driver opens a session on a camera and initializes the camera. Opening a session sets the region of interest (ROI) to the size of the settings you configured in MAX.



**Note** If you do not have a valid session, a temporary session is created using cam0.

Then, the snap acquires the next incoming image into a user buffer. After the image is acquired, the program calls image processing and analysis functions. When the processing and analysis functions are finished, the program calls the Close Camera function using the camera handle. This function instructs NI-IMAQdx to free all of the resources associated with this camera, which releases the session.

## **High-Level Grab Functions**

Grab functions perform an acquisition that loops continually on one or more internal buffers. You can copy the last acquired buffer to a separate user buffer for processing or analysis. The following figure illustrates the typical programming order of a high-level grab acquisition.



Use a grab for high-speed applications during which you need to process only one image at a time. You can copy the last acquired buffer to a separate user buffer for processing or analysis. To use these functions, you must have a valid session. If you do not have a valid session, the Configure Grab function creates a session using cam0.

Calling the Configure Grab function opens a session for a grab acquisition. During acquisition, each successive grab copies the last acquired internal buffer into a user buffer where you can process the image.

## **High-Level Sequence Functions**

A sequence acquires a specified number of internal buffers and then stops.

The following figure illustrates the programming order of a low-level sequence acquisition.



Use a sequence in applications where you need to process a series of consecutive images. Sequence acquisitions are synchronous. If you do not specify a session, a temporary session is created using cam0.

# Low-Level Function Programming Examples

Use low-level functions for more advanced programming techniques. In general, low-level functions have more parameters than high-level functions.

## **Low-Level Snap Functions**

The low-level snap examples set up a one-shot, single-image acquisition and start the acquisition. The program acquires an image and processes it. Finally, the program stops the acquisition, unconfigures the acquisition, and closes the session.

The following figure illustrates the programming order of a low-level snap acquisition.



#### **Low-Level Grab Functions**

The low-level grab examples demonstrate how to perform a grab acquisition using low-level function calls. The program sets up a continuous acquisition into three internal buffers and starts the acquisition. The main loop iterates continuously. In the main processing loop, the program acquires an image and processes it. After the loop, the program stops the acquisition, unconfigures the acquisition, and closes the session.

The following figure illustrates the programming order of a low-level grab acquisition.



#### **Low-Level Sequence Functions**

The low-level sequence examples demonstrate how to perform a sequence acquisition using low-level calls. The program sets up a one-shot, multi-image acquisition and starts the acquisition. The main loop iterates once for each internal buffer. In the main processing loop, the program acquires an image and processes it. After the loop, the program stops the acquisition, unconfigures the acquisition, and closes the session.

The following figure illustrates the programming order of a low-level sequence acquisition.



## Advanced Programming with NI-IMAQdx

This section contains information about setting camera attributes, broadcasting acquired images to multiple machines, using scale to define the size of transferred images, and triggering.

## **Camera Attributes**

After opening a camera, configure the camera attributes by specifying the following parameters: the attribute name, the attribute type, and the attribute value. Specify the same attributes when querying the camera attributes.

Use the Set Attribute function to set an attribute value. Use the Get Attribute function to get an attribute value.

The driver returns an error if an attribute is not accessible. Query the attribute access before accessing an attribute. When setting an attribute, the driver returns an error if the value is out of range. Query the attribute range before setting an attribute.

Visit the following topics to learn more about camera attributes.

Attribute Name

Attribute Type

<u>Attribute Value</u>

Attribute Access

Attribute Range



**Note** To modify camera attributes in LabVIEW, use the IMAQdx Property Node. Every camera attribute has two parameters: Attribute Name and Attribute Value.

## **Attribute Name**

The attribute name is a string constant. Each camera has a different set of attributes. Use the Enumerate Attributes function to list all available attributes for a given camera. The attribute name contains several keywords separated by a double colon namespace marker. AcquisitionAttributes::Timeout and AcquisitionAttributes::Bayer::Pattern are two examples of attribute names.

The namespace marker separates different levels in the attribute tree as described by the driver and the camera. Use the fully qualified attribute name such as AcquisitionAttributes::Timeout, or use the short version of the attribute name such as Timeout, when specifying the attribute name.

Specify the attribute name with the attribute property node.

# Attribute Type

The attribute type determines how a camera attribute is stored in the driver and the camera. Use the Get Attribute Type function to query the type of a given attribute. Each attribute is represented as one of the following types.

Attribute Type	Description
U32	32-bit unsigned integer
1164	64-bit signed integer
F64	Double precision floating point
String	String
Enum	Name and value pair
Bool	Boolean (true or false)
Command	Action

## **Attribute Value**

The attribute value represents the active value of an attribute in the driver and the camera. The following value types are supported. Enter a value type for the attribute. The value type must be compatible with the attribute type. Refer to the following table for a list of attribute value types. Use the **Range** property nodes to find the valid range for the current camera.

Value Type	Description
U32	32-bit unsigned integer
F64	Double precision floating point
String	String
EnumItem	Name and value pair
Bool	Boolean (true or false)

The following value types are compatible for any given attribute type.

Value Type	Description
U32	U32, 164, F64, String
164	U32, <b>I64</b> , F64, String
F64	U32, I64, <b>F64</b> , String
String	String
Enum	U32, I64, F64, String, EnumItem
Bool	U32, I64, F64, String, <b>Bool</b>
Command	U32, I64, F64, String, <b>Bool</b>

The **bolded** compatible value type indicates the native value type. For example, use a 32-bit unsigned integer value type when dealing with a 32-bit unsigned integer attribute type. Optionally, use a signed 64-bit integer, double precision floating point, or string when dealing with a 32-bit unsigned integer attribute type.

## **Attribute Access**

Each attribute has a read access and a write access. Use the Is Attribute Readable function to query read access for a given attribute. Use the Is Attribute Writable function to query write access for a given attribute. Some attributes are only settable before configuring the acquisition, while other attributes can be changed at any time.

# Attribute Range

Most attributes have a specific range of valid values. Use the Get Attribute Minimum, Get Attribute Maximum, and Get Attribute Increment functions to query the range of numeric attribute types. Use the Enumerate Attribute Values function to query the range of enumerated types. String, bool, and command attributes do not have a range.

Attribute Type	Compatible Range
U32	Minimum, Maximum, Increment
164	Minimum, Maximum, Increment
F64	Minimum, Maximum, Increment
String	N/A
Enum	Attribute Values
Bool	N/A
Command	N/A

The following range is applicable for any given attribute.

## Broadcasting

Many machine vision applications involve a single host computer acquiring data from a single industrial camera. Other machine vision applications acquire data from multiple industrial cameras concurrently. With the broadcasting feature, a machine vision application can run on multiple host computers while acquiring data from a single camera, as shown in the following figure.



The camera broadcasts video data on the camera bus and all the connected host computers receive the same image data. In this scenario, one host computer is designated as the controller. The controller is responsible for starting/stopping the camera feed. There can be only one controller per camera. The listeners obtain image data from the camera bus. The listeners do not control the camera in any way. There may be one or more listeners per camera.

Broadcasting has many uses. Computationally intensive tasks can be spread across different machines, thus effectively distributing computations. Multiple host computers can also perform redundancy checks. Additionally, listeners can monitor the current status of a headless system.

## Implementation

Usage for the controller is unchanged from a stand-alone application. Open your camera interface with the default interface name (for example, cam0) configured in Measurement & Automation Explorer (MAX). Configure and start your acquisition.

For GigE Vision cameras, you can configure the camera to broadcast or multicast image data to all nodes on the network. Broadcast is not routable, and everyone on the same network sees the data, even if they are not listening. Multicast is routable if the network is configured properly. A multicast configuration is preferred if supported by the camera because only interested parties can see the data traffic.

Next, start the listener(s). On the listening computer, open your camera interface with the 64-bit unique identifier of the target camera, which you can find in the General tab in MAX. The controller can get a unique ID and send it to the listener sessions. Additionally, you must set the listener camera control mode parameter.

At this point, both the controller and listener systems are acquiring the same live data from the same camera. When running as a listener, most acquisition attributes are read-only. No camera feature or control is accessible when running as a listener system. Attempts to set these attributes result in the following error: **Attribute not writable**.

There is no synchronization between the controller and the listener host computers provided by the low-level driver. The user must start the controller before starting the listener. If the camera is not transmitting data when the listener initializes, the session returns the following error: **No acquisition in progress**. If the controller stops the video feed of the camera, the listener times out.

## Scalable Image Size

Digital cameras support a predefined set of image sizes, which you can select through the **Acquisition** attributes in Measurement & Automation Explorer (MAX). Refer to your camera documentation for a list of supported formats.

If you are using LabVIEW, the NI-IMAQdx software recognizes the predefined formats and automatically allocates enough memory to accommodate the image. The image is resized as necessary.

Some cameras support user-configurable Region of Interest (ROI), which allows you to define the size of the acquired image. If you use this format, you must input the image size using the **Region of Interest** attributesoffset x, offset y, width, and height. The size and position of the sub-image you are acquiring must be a multiple of the **Increment** attribute, as shown in the following figure, or the driver acquires the smallest sub-image that contains the ROI you defined.



The **Increment** values are camera-specific. Refer to the camera documentation or query the **Increment** attributes for width and height to obtain the actual values.

The IIDC 1.31 and GigE Vision 1.0 specifications provide several external triggering modes for cameras. A camera may support one or more of the triggering modes. Additionally, a camera may support triggering modes that are vendor specific. Refer to your camera documentation to find out which standard modes are supported.

Configure triggers before configuring and starting the acquisition. Use the **Camera Attribute** functions to configure the triggers.

The following sections describe how to configure your triggers for different cameras.

Trigger Modes for GigE Vision Cameras

Trigger Modes for IIDC Cameras

# Trigger Modes for IIDC Cameras

Configure triggers before configuring and starting the acquisition. Use the **Camera Attribute** functions to configure the triggers.

All IIDC cameras that support triggering have the following attributes.

- **TriggerMode**—Specifies one of the trigger modes—Mode0, Mode1...Mode5—described in the following sections. A value of Off indicates that the camera is not externally triggered.
- **TriggerActivation**—Specifies when the trigger input is active. A value of **LevelHigh** indicates that the trigger is considered active when the signal is high. A value of **LevelLow** indicated that the trigger is considered active when the signal is low.
- **TriggerParameter**—Certain trigger modes require an additional parameter. Refer to the following sections to see if the optional parameter is required.

Trigger Mode 0 Trigger Mode 1 Trigger Mode 2 Trigger Mode 3 Trigger Mode 4 Trigger Mode 5

With trigger mode 0, the camera starts frame integration when the external trigger input changes to an active value. The frame is exposed for a duration specified by the shutter attribute before the camera transfers the image to the host computer. No optional parameter is required.



With trigger mode 1, the camera starts frame integration when the external trigger input changes to an active value. The frame is exposed while the external trigger is active. When the trigger becomes inactive, the camera stops frame integration and transfers the image to the host computer. No optional parameter is required.



With trigger mode 2, the camera starts frame integration when the external trigger input changes to an active value. The same frame is exposed for multiple triggers. The number of triggers is specified by the optional parameter, which must have a value of 2 or more.



With trigger mode 3, the camera triggers continuously internally. The frame is exposed for a duration specified by the shutter attribute before the camera transfers the image to the host computer. The next internal trigger becomes active after a set cycle time. The cycle time is *N* times the cycle time of the fastest frame rate. *N* is specified by the optional parameter, which must have a value of 1 or more.



With trigger mode 4, the camera starts frame integration when the external trigger input changes to an active value. Multiple frames are exposed before the camera transfers the image to the host computer. Each frame is exposed for a duration specified by the shutter attribute. The number of frames is specified by the optional parameter, which must have a value of 1 or more.



With trigger mode 5, the camera starts frame integration when the external trigger input changes to an active value. Multiple frames are exposed before the camera transfers the image to the host computer. Each frame is exposed while the external trigger is active. The number of frames is specified by the optional parameter, which must have a value of 1 or more.


## **Trigger Modes for GigE Vision Cameras**

Most GigE Vision cameras that support triggering have the following attributes.



**Note** All triggering modes and parameters for GigE Vision cameras are subject to camera vendor implementation. Refer to your camera documentation for triggering modes and parameters available for your camera.

- **TriggerActivation**—Specifies the type of signal change. Examples of signal changes are RisingEdge, FallingEdge, LevelHigh, and LevelLow.
- **TriggerMode**—Specifies the operation mode of the trigger for acquisition. Examples of operation modes are Off, Hardware, and Software.
- **TriggerSelector**—Specifies the type of trigger to control. Examples of triggers to control are FrameStart, FrameEnd, FrameActive, AcquisitionActive, LineStart, ExposureStart, ExposureEnd, ExposureActive.
- **TriggerSoftware**—Generates a software trigger to start the acquisition in software trigger mode.
- **TriggerSource specifies**—The input line signal. Examples of input line signals are Off, Line1, and Line2.

## Using NI-IMAQdx in LabVIEW

The NI-IMAQdx VI librarypart of the NI-IMAQdx software is a group of virtual instruments (VIs) that enable you to use LabVIEW with your camera.

NI Vision for LabVIEW is the National Instruments image processing and analysis library, which consists of more than 400 VIs. Some of the basic NI Vision VIs are shared with NI-IMAQdx. If you do not have NI Vision, you can use the NI Vision VIs included with NI-IMAQdx to create an image acquisition application. When you use these basic VIs, you can upgrade your application later to use additional NI Vision VIs without making changes to your initial image acquisition application.

NI-IMAQdx adds a subpalette of VIs to the Vision and Motion Functions palette and an Image Display control to the Controls palette.

Create NI-IMAQdx applications as you would any other LabVIEW or LabVIEW Real-Time (RT) application. Place VIs onto the block diagram to create the program, and use the front panel to design the user interface. Click Run to compile and run the application.

Before you start building an image acquisition application, familiarize yourself with the basic knowledge and concepts contained in the following sections.

### Location of the NI-IMAQdx VIs

You can find the NI-IMAQdx VIs in the LabVIEW **Functions** palette. From the LabVIEW block diagram, select **Vision and Motion»NI-IMAQdx**.

The most commonly used, high-level VIs are on the **NI-IMAQdx** palette. You can find VIs for basic acquisition and changing attributes. The **Vision and Motion»NI-IMAQdx»NI-IMAQdx Low-Level** palette contains VIs for more advanced applications.

Refer to the *NI-IMAQdx VI Reference Help* for more information about using these VIs.

### **Common VI Parameters**

#### **Image Buffer**

Many acquisition VIs require an image buffer to receive the captured image. You can create this image buffer using the IMAQ Create VI. Refer to <u>Buffer Management</u> for more information about using buffers. **Image In** receives the image buffer. **Image Out** returns the captured image.

#### **Session In and Session Out**

**Session In** is a unique identifier that specifies which interface file to use for the acquisition. An NI-IMAQdx Session is produced by the IMAQdx Open Camera VI and used as an input to all other NI-IMAQdx VIs. The NI-IMAQdx VIs also use **Session Out**, which is identical to **Session In**, to simplify dataflow programming. **Session Out** is similar to the duplicate file sessions provided by the file I/O VIs. The high-level acquisition VIsIMAQdx Snap, IMAQdx Configure Grab, and IMAQdx Sequencerequire you to wire **Session In** only in the following instances.

- If you are using an interface other than the default cam0
- If you are using multiple cameras
- If you need to set NI-IMAQdx properties before the acquisition

To get and set properties of the acquisition and camera, wire the **Session** In to the LabVIEW Property Node.

## **Acquisition VIs**

Two types of acquisition VIs are available in LabVIEW

#### **High-Level Acquisition VIs**

Use the high-level acquisition VIs for basic image acquisition applications. VIs are included for snap, grab, and sequence, as described in the <u>Acquisition Types</u> section of this chapter.

#### **Low-Level Acquisition VIs**

Use the low-level acquisition VIs for more advanced image acquisition applications. The low-level VIs configure an acquisition, start an acquisition, retrieve the acquired images, and stop an acquisition. You can use these VIs to construct advanced Vision applications.

Complete the following general steps to perform a low-level acquisition.

- 1. Call IMAQdx Open Camera to initialize the board and create an NI-IMAQdx Session.
- 2. Call IMAQdx Configure Acquisition to allocate resources for the acquisition.
- 3. Call IMAQdx Start Acquisition to start transferring data from the camera.
- 4. Call IMAQdx Get Image to obtain a copy of the requested image data.
- 5. After an acquisition, call IMAQdx Stop Acquisition to stop transferring data from the camera.
- 6. Call IMAQdx Unconfigure Acquisition to release the resources associated with the acquisition.
- 7. Call IMAQdx Close Camera to close the camera session.
- Note If an acquisition is in progress and you call IMAQdx Close Camera, the driver automatically stops the acquisition and releases resources associated with the acquisition.

## **Acquisition Types**

The following sections describe snap, grab, and sequence acquisitions in LabVIEW and give examples.

**Snap Acquisition** 

Grab Acquisition

**Sequence Acquisition** 

## **Snap Acquisition**

Use the IMAQdx Snap VI for snap applications in LabVIEW. The following figure shows a simplified block diagram for using IMAQdx Snap.



## **Grab Acquisition**

Use two VIsIMAQdx Configure Grab and IMAQdx Grabfor a grab acquisition in LabVIEW. Call IMAQdx Configure Grab once to open the acquisition and start capturing the image to an internal software buffer. You can call IMAQdx Grab multiple times to copy the image currently stored in the internal buffer to a LabVIEW image buffer. After the program finishes copying images, call IMAQdx Close Camera once to shut down the acquisition.

The following figure shows a simplified block diagram for using IMAQdx Configure Grab and IMAQdx Grab.



## **Sequence Acquisition**

Use the IMAQdx Sequence VI for sequence applications. IMAQdx Sequence starts, acquires, and releases a sequence acquisition. IMAQdx Sequence does not return until the entire sequence is acquired.

The following figure shows a simplified block diagram for using IMAQdx Sequence. Place the IMAQ Create VI inside a For Loop to create an array of images for the Image Array In input to IMAQdx Sequence. The Number to Decimal String VI and Concatenate String VI create a unique name for each image in the array.



## Image Display

Many image acquisition applications require that one or more images be displayed. You have several options for displaying images in LabVIEW.

You can display an image directly on the front panel using an Image Display control, which is available on the Vision Controls palette. To display an image on an Image Display control, place the control on the front panel of your VI. On the block diagram, wire **Image Out** from an acquisition VI to the **Image Display** control terminal.

The following figure illustrates using an image control to display an image using an Image Display control. For more information about the Image Display control, refer to the *NI Vision for LabVIEW Help*.



### IMAQ WindDraw VI

If you have NI Vision for LabVIEW, you can display an image in an external window using IMAQ WindDraw VI, located at **Vision and Motion»Vision Utilities»External Display**. Use IMAQ WindDraw when you need more image size and location control.

The following figure illustrates using IMAQ WindDraw to display an image acquired using IMAQdx Snap. You can display images in the same way using any acquisition type. For more information about the display capabilities of NI Vision, refer to the *NI Vision for LabVIEW Help*.



### IMAQ RT Video Out VI

If you have LabVIEW RT, you can use IMAQ RT Video Out VI, located at **Vision and Motion**»**Vision Utilities**»**IMAQ RT**, to display an image on the monitor connected to your RT device. Use IMAQ Video Out Display Mode, located at **Vision and Motion**»**Vision Utilities**»**IMAQ RT**, to configure the monitor for display. The following figure illustrates configuring the monitor and displaying an image acquired with IMAQdx Snap.



Note The IMAQ RT Video Out VI is available only on RT devices with Intel i815 or i845 video chipsets. These devices include NI CVS-1450 Series devices, PXI-817*x* controllers, and PXI-818*x* controllers.

## **Camera Attributes in LabVIEW**

To modify camera attributes in LabVIEW, use the IMAQdx Property Node. Every camera attribute has two parameters: Attribute Name and Attribute Value.

#### Attribute Name in IMAQdx Property Node

Specify the attribute name with the attribute property node. The attribute name is a string constant containing several keywords separated by a double colon namespace marker. The namespace marker separates different levels in the attribute tree as described by the driver and the camera. Use the fully qualified attribute name, for example AcquisitionAttributes::Timeout, or the short version of the attribute name, for example Timeout, when specifying the attribute name.

#### Attribute Value in IMAQdx Property Node

Enter a value type for the attribute. The value type must be compatible with the attribute type. Refer to <u>Attribute Value</u> for a list of attribute value types. Use the Range property nodes to find the valid range for the current camera.

The following figure shows how to set camera attributes with the property nodes in NI-IMAQdx.



Visit the <u>Camera Attributes</u> section of this help file for more information regarding various camera attributes.

### **Buffer Management**

The IMAQ Create VI and IMAQ Dispose VIs manage image buffers in LabVIEW.

IMAQ Create, as shown in the following figure, allocates an image buffer. Image Name is a label for the buffer created. Each buffer must have a unique name. Image Type specifies the type of image being created. Use Grayscale (U8) for 8-bit monochrome images, Grayscale (I16) for 16-bit monochrome images, and RGB (U32) for RGB color images.



**Note** If **Image Type** is set to a value incompatible with the current video mode, NI-IMAQdx automatically changes the value to a compatible one when acquiring images.

**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 provide a value for **Border Size**. **Border Size** is the width, in pixels, of a border created around an image. Some image processing functions, such as labeling or morphology, require a border.



IMAQ Dispose, as shown in the following figure, frees the memory allocated for the image buffer. Call this VI only after the image is no longer required for processing.



#### **Buffers for One-shot Acquisitions**

For a one-shot acquisition, you can request only one of the available buffer numbers. For a continuous acquisition, you can request any present or future buffer number. You can also request the next logical buffer or the buffer containing the most recently acquired data. With highlevel grab acquisitions, the buffer number defaults to the next transferred buffer.

When you complete the buffer acquisition step, the driver returns the actual buffer number with the image.

#### **Detecting Overwritten Internal Buffers**

NI-IMAQdx is able to detect overwritten internal buffers. You can configure the driver to manage an overwritten buffer in one of the following ways.

- Get newest valid buffer
- Get oldest valid buffer
- Fail and return an error

In all cases, the camera continues to transfer data when a buffer is overwritten.

The default overwrite mode for all types of acquisition is to get the newest valid buffer. This option, which National Instruments recommends for most applications, enables you to process the most recent image. If you need to get the image closest in time to a requested buffer, configure the driver to get the oldest valid buffer. If your application requires that every image be processed, configure the driver to fail when a buffer is overwritten so that you are alerted.

## **Error Handling**

Every NI-IMAQdx VI contains an **error in** input cluster and an **error out** output cluster. The clusters, shown in the following figure, contain a Boolean value 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-IMAQdx function.



You can use the Simple Error Handler VI, located on the **Functions»Dialog & User Interface** palette, to check for errors that occur while executing a VI. If you wire an error cluster to the Simple Error Handle VI, the VI deciphers the error information and displays a dialog box that describes the error. If no error occurred, the Simple Error Handler VI does nothing. The following figure illustrates wiring an NI-IMAQdx VI to the Simple Error Handler VI.



## Using NI-IMAQdx in C and .NET

This section briefly describes how to use NI-IMAQdx functions in Microsoft Visual C and Microsoft Visual Studio .NET.

## Using NI-IMAQdx for C

This section outlines the process for developing NI-IMAQdx applications using C. Detailed instructions about creating project and source files are not included. For information about creating and managing project files, refer to the documentation included with your particular development environment.



**Note** The generic and high-level functions appear within each function class in the logical order you might need to use them. The low-level functions appear within each function class in alphabetical order.

When programming, use the following guidelines:

- Include the niimaqdx.h header file in all C source files that use NI-IMAQdx functions. Add this file to the top of your source files.
- Add the niimaqdx.lib import library to your project. In some environments, you can add import libraries simply by inserting them into your list of project files. In other environments, you can specify import libraries under the linker settings portion of the project file.
- When compiling, indicate where the compiler can find the NI-IMAQ header files and shared libraries. You can find most of the files you need for development under the NI-IMAQ target installation directory. If you choose the default directory during installation, the target installation directory is C:\Program Files\National Instruments\NI-IMAQdx. You can find the include files under the include subdirectory. The import libraries for Microsoft Visual C++ are located under the lib\msvc subdirectory.

You can use the additional Image functions installed with NI-IMAQdx. These functions use the NI Vision memory management feature, which automatically allocates the memory for your image. To use these Image functions, first create an image using imaqCreate, and then pass that image to an acquisition function.

## Using NI-IMAQdx for Microsoft Visual Studio .NET

NI-IMAQdx installs the following assemblies that enable .NET languages to interact with the driver software:

- NationalInstruments.CWNIIMAQDX.Interop.dll
- NationalInstruments.AxCWNIIMAQDXControlsLib.Interop.dll

Use NI Vision to display images with the included Viewer control

The CWIMAQdx assembly is installed in the NI-IMAQdx\dotNET\ Assemblies\Current directory. The AxCWIMAQControlsLib assembly is installed in the Vision\dotNET\Assemblies\Current directory. Refer to the *NI-IMAQdx Function Reference Help* for information about the properties, methods, and events available with these assemblies.

### **Creating a New .NET Application**

You first must add a reference to the NI-IMAQdx assembly in your project when creating a new application. Complete the following steps to add a reference to the NI-IMAQdx assembly in Microsoft Visual Studio .NET 2003 or later:

- 1. Create a new application, or open an existing one.
- 2. Select Project»Add Reference.
- 3. Under the .NET Framework Components tab, select NI-IMAQdx.

If you need to display acquired images, you also must add an NI Vision Viewer control to your toolbox and to your form. Complete the following steps to add the NI Vision Viewer control to the Microsoft Visual Studio .NET toolbox.

- 1. With your project open, open a form in Design View.
- 2. Open the Toolbox (View»Toolbox).
- 3. Select the category in which you want the NI Vision controls to appear (General, Components, and so on).
- 4. Select Tools»Add/Remove Toolbox Items.
- 5. Under the **.NET Framework Components** tab, select the CWIMAQViewer control.

When the NI Vision Viewer control is in the toolbox, you can add it to your forms by clicking on the tool and drawing an area on the form. References to the NI Vision Interop Assemblies are automatically added to your project.

## **Register-Level Programming**

All cameras communicate to the host computer through register maps. The register map reflects the system memory located on the camera. The register map allows the host computer to read and write information with minimal overhead.

The host computer sends asynchronous messages over the host bus to the connected camera. When the data is written into memory on the camera, the camera processes the incoming request. If possible, the camera responds immediately. Otherwise, a pending transaction message is returned. When the pending request is completed, the camera returns the results of the request.



The following image illustrates an example of a split transaction.

NI-IMAQdx supports the GigE Vision 1.0 specification and the IEEE 1394 Trade Association IIDC 1.31 register specification for industrial cameras. Most of the intricacies of register-level programming are abstracted by the driver. The driver is responsible for manipulating camera features and activating/deactivating the video data stream.

Some cameras implement additional registers that are not contained in

the IIDC 1.31 or GigE Vision 1.0 specifications. These advanced camera features are not natively supported by the camera driver. To use these advanced features, you must use the low-level, register-level access tools to communicate with the camera.

GigE Vision cameras have all features defined in an XML file which may eliminate the need for direct register programming.

### **Register-Level Primitives**

The NI-IMAQdx software provides the following register-level primitives.

- **Read Register**—Reads 32-bits of data from a specified memory location.
- Write Register—Writes 32-bits of data to a specified memory location.
- **Read Memory**—Reads an array of bytes from a specified memory location and range.
- Write Memory—Writes an array of bytes to a specified memory location.

### Usage

To perform a register-level access, specify a memory location (or offset) and data storage. IEEE 1394 memory locations are specified as 48-bit values. The upper 20 bits are filled in by the driver. The low-level register primitives accept the lower 28-bit offset. The memory storage contains the result/desired data when transferring. GigE Vision memory locations are specified as 32-bit values.

### **Basic Example of Usage**

The isonchronous enable register indicates active video transmission. To read the ISO\_EN register (0x614), calculate the memory offset by adding the specified offset to the base register. The base register is 0xF0F00000 for most IEEE 1394 cameras.

0xF0F00000 + 0x614 = 0xF0F00614

The value is read, and the result is placed in the specified memory location.

read register (0xF0F00614) = <iso\_en>

where <iso\_en> = (0x8000000 or 0x0000000).

If bit 0 has a value of 0x80000000, the bit is on, and the camera is transmitting video data. If bit 0 has a value of 0x00000000, the camera is not currently transmitting data.

### **Advanced Example of Usage**

The advanced feature described in this example is specific to Basler IEEE 1394 cameras. The advanced feature replaces the live video feed with a static test pattern.

According to the user documentation for the Basler A601f camera, the TEST\_IMAGE register is located at advanced offset 0x0098. You can enable a static test pattern by setting bit 17 of the TEST\_IMAGE register. To get the advanced base register, first read ADVANCED\_FEATURE\_INQ register (0x480). Add the specified offset to the base register—0xF0F00000 for most IEEE 1394 cameras.

 $0 \times F0F00000 + 0 \times 480 = 0 \times F0F00480$ 

Read the value into storage.

read register (0xF0F00480) = <advanced\_feature\_inq>

where <advanced\_feature\_inq> = 0x800000.

Now, calculate the offset to the advanced feature offset. You need to multiply the previous result by 4 to convert the quadlet offset value to byte offset.

(0xF0F00000 + ( <advanced feature offset> x 4) + 0x98) = newly calculated offset byte swap (1 < < 17) = newly calculated register mask write register (0xF2F00098, 0x00002000) Now the camera is set to the test pattern.

## Caveats

This following are caveats to consider when programming registers using the NI-IMAQdx software.

**Endianness**—Data that spans multiple bytes, such as a quadlet, may be written left-to-right or right-to-left. The method with which data is written is called *endianness*. Two types of endianness exist: *big endian* and *little endian*.

**Byte Array**—Many cameras allow register-level access to more than 32 bits of data per communication request. In most cases, you can safely write and read a large, contiguous block of data to and from the connected camera. Some cameras fail when trying to access large payloads. If the camera does not successfully transfer an array of bytes, attempt to transfer the smaller packets of data one at a time.

The ethernet and IEEE 1394 bus transports data using the big endian method. However, Windows and LabVIEW RT host machines accept little endian data. To correct for this discrepancy, NI-IMAQdx byte-swaps every quadlet that is read or written with low-level register primitives.

**Timing**—Many cameras are responsive to successive register accesses. In most cases, you can safely read and write registers as quickly as possible. Some cameras lock up under stressed conditions. The camera driver inserts an artificial delay between register accesses. You can change this artificial delay in the registry under the following registry key: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\niimaqdxk\F

The key specifies the millisecond value to delay before each transaction. After changing the value, reboot the host computer to enable the changes.

Note Changing this delay affects the entire driver, not just register-level access.

**Invalid Memory Location**—The NI-IMAQdx software allows access to register locations that do not exist. If an error occurs while accessing the register, check the validity of the register location.

# Glossary A B C D E F G H I L M N O P Q R S T U V W Y

### Α

A/D	Analog-to-digital.
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 vertical sync signal) and a line count.
active pixel region	The region of pixels actively being stored; defined by a pixel start (relative to the horizontal sync signal) and a pixel count.
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.
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 picture or image's width to its height.
asynchronous	(1) Independent in time from any other event. (2) Communication mechanism on the IEEE 1394 bus, which guarantees delivery of the message but does not guarantee timing.
## В

The area of the video signal between the rising edge of the back horizontal sync signal and the active video information. porch Method to produce color images with a single imaging Bayer encoding sensor, as opposed to three individual sensors for the red, green, and blue components of light. Color filter array pattern that can appear in four variations, Bayer pattern depending on the current left and top offsets of the acquisition window: GBGB GRGR BGBG RGRG RGRG BGBG GRGR GBGB biq Describes computers that store bytes of memory by placing the most significant byte at the memory location with the endian lowest address, the next significant byte at the next memory location, and so on. black The level that represents the darkest an image can get. See reference **also** white reference level. level **BMP** Bitmap. Image file format commonly used for 8-bit and color images (extension .bmp). 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.

## С

cache	High-speed processor memory that buffers commonly used instructions or data to increase processing throughput.	
camera session	A process-safe handle to a camera.	
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.	
CompactPCI	Refers to the core specification defined by the PCI Industrial Computer Manufacturer's Group (PICMG).	
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. <b>See also</b> <u>interpreter</u> .	
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.	

## 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 devices plugged into a computer, and possibly generating control signals with D/A and/or DIO devices in the same computer.
- DC Direct current.

default A default parameter value recorded in the driver; in many

- setting cases, the default input of a control is a certain value (often 0) that means use the current default setting.
- DLL Dynamic link library. A software module in Microsoft Windows containing executable code and data that can be called or used by Windows applications or other DLLs; functions and data in a DLL are loaded and linked at run time when they are referenced by a Windows application or other DLLs.
- 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.
- driver Software that controls a specific hardware device such as an image acquisition device.

dynamic 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 decibels.

#### Ε

- EEPROM Electrically erasable programmable read-only memory. ROM that can be erased with an electrical signal and reprogrammed.
- endianness The convention describing the ordering of bytes in memory or the sequence in which bytes are transmitted.
- external 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 of the odd-numbered lines, and 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; FIFO buffers are used on image acquisition devices to temporarily store incoming data until that data can be retrieved.
- flash An ADC whose output code is determined in a single step by ADC a bank of comparators and encoding logic.
- frame A complete image; in interlaced formats, a frame is composed of two fields.
- front The area of a video signal between the start of the horizontal porch blank and the start of the horizontal sync.
- 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.

## G

- gain Applied value to compensate for discrepancies in the filter for a particular color.
- 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.

Gigabit Describes technologies which transmit Ethernet packets at a Ethernet rate of a gigabit per second.

- GigEA camera interface standard developed using the GigabitVisionEthernet communication protocol.
- grab Performs an acquisition that loops continually on one buffer. You obtain a copy of the acquisition buffer by grabbing a copy to a separate buffer that can be used for analysis.
- 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.

#### Η

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

hardware Separates software API capabilities, such as general abstraction acquisition and control functions, from hardware-specific layer information.

- 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.
- hue Represents the dominant color of a pixel. The hue function is a continuous function that covers all the possible colors generated using the R, G, and B primaries. **See also** <u>RGB</u>.

#### I

- I/O Input/output. The transfer of data to/from a computer system involving communications channels, operator interface devices, or data acquisition and control interfaces.
- IEEE Institute of Electrical and Electronics Engineers.
- 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.

instrument A set of high-level software functions, such as NI-IMAQ,

- driver 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.
- 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.

internal A page-locked buffer. **See also** <u>page-locked buffer</u>.

buffer

- interpreter A software utility that executes source code from a highlevel language, such as Java or Basic, by reading one line at a time and executing the specified operation. In contrast, a compiler converts all source code to executable machine code before execution. Compiled languages give significantly higher performance than interpreted languages. Examples of compiled languages are C, C++, and LabVIEW, while Java and Basic are generally interpreted languages. **See also** compiler.
- interrupt A computer signal indicating that the CPU should suspend its current task to service a designated activity.

interrupt The relative priority at which a device can interrupt. level

- IRE A relative unit of measure (named for the Institute of Radio Engineers). 0 IRE corresponds to the blanking level of a video signal, 100 IRE to the white level. Note that for CIR/PAL video the black level is equal to the blanking level or 0 IRE, while for RS-170/NTSC video, the black level is at 7.5 IRE.
- IRQ Interrupt request. See also interrupt.

# L

- library A file containing compiled object modules, each comprised of one or 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.
- little Describes computers that store bytes of memory by placing endian the least significant byte at the memory location with the lowest address, the second least significant byte at the next memory location, and so on.
- LSB Least significant bit.
- luminance The brightness information in the video picture. The luminance signal amplitude varies in proportion to the brightness of the video signal and corresponds exactly to the monochrome picture.
- LUT Lookup table. A selection in Measurement & Automation Explorer (MAX) for Vision that contains formulas that let you implement simple imaging operations such as contrast enhancement, data inversion, gamma manipulation, or other nonlinear transfer functions.

#### Μ

MAX Measurement & Automation Explorer. The National Instruments Windows-based graphical configuration utility you can use to configure NI software and hardware, execute system diagnostics, add new channels and interfaces, and view the devices and instruments you have connected to your computer. MAX is installed on the desktop during the National Instruments driver software installation.

memory See <u>buffer</u>.

buffer

memory Continuous blocks of memory that can be accessed quickly window 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.

## Ν

NI-IMAQ	Driver software for National Instruments image acquisition hardware.	
NI-IMAQdx	National Instruments driver software for IEEE 1394 and GigE Vision cameras.	
noninterlaced	A video frame where all the lines are scanned sequentially, rather than being 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. <b>See also</b> PAL.	
NVRAM	Nonvolatile RAM. RAM that is not erased when a device loses power or is turned off.	

## 0

one- Applies to pulse generation and acquisitions. A one-shot pulse shot or acquisition happens only once.

### Ρ

page- Memory page that is marked as non-pagable by the virtuallocked file system. Page-locked buffers remain in physical memorybuffer and do not cause page faults.

- PAL Phase Alternation Line. One of the European video color standards; uses 625 lines per frame. **See also <u>NTSC</u>**.
- 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 133 Mbytes/s.
- PCIe PCI Express. A high-performance expansion bus architecture originally developed by Intel to replace PCI. PCIe offers a theoretical maximum transfer rate that is dependent upon lane width. A x1 link theoretically provides 250 MB/s in each direction—to and from the device. Once overhead is accounted for, a x1 link can provide approximately 200 MB/s of input capability and 200 MB/s of output capability. Increasing the number of lanes in a link increases maximum throughput by approximately the same factor.
- PCLK Pixel clock signal. Times the sampling of pixels on a video line.
- PGIA Programmable gain instrumentation amplifier.

picture The ratio of the active pixel region to the active line region;

aspect for standard video signals such as RS-170 or CCIR, the fullratio size picture aspect ratio typically 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 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; however, typically it falls between 0.95 and 1.05, depending on camera quality.

- pixel Divides the incoming horizontal video line into pixels. clock
- pixel The total number of pixels between two horizontal sync

# Q

quadlet A 32-bit (four-byte) word.

quadrature An encoding technique for a rotating device where two encoder tracks of information are placed on the device, with the signals on the tracks offset by 90 degrees from each other. The phase difference indicates the position and direction of rotation.

## 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 A measure in LSB of the accuracy of an ADC; it includes all accuracy 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.
- RGB Red, green, and blue. The three primary colors used to represent a color picture. An RGB camera is a camera that delivers three signals, one for each primary.

#### ribbon A flat cable in which the conductors are side by side.

cable

- ring Performs an acquisition that loops continually on a specified number of buffers.
- ROI Region of interest. (1) An area of the image that is graphically selected from a window displaying the image. This area can be used focus further processing; (2) 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.
- RTSI bus Real-Time System Integration Bus. The National Instruments timing bus that connects image acquisition and DAQ devices directly, by means of connectors on top of the devices, for precise synchronization of functions.

## S

saturation	The richness of a color. A saturation of zero corresponds to no color, that is, a gray pixel. Pink is a red with low saturation.	
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.	
sequence	Performs an acquisition that acquires a specified number of buffers, then stops.	
snap	Acquires a single frame or field to a buffer.	
SRAM	Static RAM.	
StillColor	A post-processing algorithm that allows the acquisition of high-quality color images generated either by an RGB or composite (NTSC or PAL) camera using a monochrome video acquisition device.	
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	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.	

## Т

timeout	Length of time, in milliseconds, that the driver waits for an image from the camera before returning an error	
transfer rate	The rate, measured in bytes/s, at which data is moved from source to destination after software initialization and setup operations; the maximum rate at which the hardware can operate.	
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 RTSI bus trigger lines; you can configure each of these lines to start or stop acquisition on a rising or falling edge.	
TTL	Transistor-transistor logic. A digital circuit composed of bipolar transistors wired in a certain manner. A typical medium-speed digital technology. Nominal TTL logic levels	

are 0 and 5 V.

#### U

user Memory buffer created by the user as a destination for the buffer image. In LabVIEW, this is created with the IMAQ Create VI.

UV **See <u>YUV</u>**. plane

#### V

- 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 A video line consists of a horizontal sync signal, back porch, line 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.

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

#### W

## Υ

YUV A representation of a color image used for the coding of NTSC or PAL video signals. The luminance information is called Y, while the chrominance information is represented by two components, U and V, that represent the coordinates in a color plane.

# **Important Information**

Warranty Copyright Trademarks Patents Warning Regarding Use of NI Products

# Warranty

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

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

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

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

# Copyright

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

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

# Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about <u>National Instruments trademarks</u>.

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

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

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

The Bluetooth  $\ensuremath{\mathbb{R}}$  word mark is a registered trademark owned by the Bluetooth SIG, Inc.

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

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

## Patents

For patents covering National Instruments products, refer to the appropriate location: **HelpPatents** in your software, the patents.txt file on your media, or <u>ni.com/patents</u>.

# WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS. SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE. INJURY. OR DEATH. THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES. INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR

APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

# **Technical Support and Professional Services**

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

- <u>Support</u>—Technical support resources at ni.com/support include the following:
  - Self-Help Resources—For answers and solutions, visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-bystep troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.
  - Standard Service Program Membership—This program entitles members to direct access to NI Applications Engineers via phone and email for one-toone technical support, as well as exclusive access to on demand training modules via the <u>Services Resource</u> <u>Center</u>. NI offers complementary membership for a full year after purchase, after which you may renew to continue your benefits.

For information about other <u>technical support options</u> in your area, visit ni.com/services or <u>contact</u> your local office at ni.com/contact.

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

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

# **Branch Offices**

Office	Telephone Number
Australia	1800 300 800
Austria	43 662 457990-0
Belgium	32 (0) 2 757 0020
Brazil	55 11 3262 3599
Canada	800 433 3488
China	86 21 5050 9800
Czech Republic	420 224 235 774
Denmark	45 45 76 26 00
Finland	358 (0) 9 725 72511
France	33 (0) 1 57 66 24 24
Germany	49 89 7413130
India	91 80 41190000
Israel	972 0 3 6393737
Italy	39 02 41309277
Japan	0120-527196 / 81 3 5472 2970
Korea	82 02 3451 3400
Lebanon	961 (0) 1 33 28 28
Malaysia	1800 887710
Mexico	01 800 010 0793
Netherlands	31 (0) 348 433 466
New Zealand	0800 553 322
Norway	47 (0) 66 90 76 60
Poland	48 22 3390150
Portugal	351 210 311 210
Russia	7 495 783 6851
Singapore	1800 226 5886
Slovenia	386 3 425 42 00

South Africa	27 0 11 805 8197
Spain	34 91 640 0085
Sweden	46 (0) 8 587 895 00
Switzerland	41 56 2005151
Taiwan	886 02 2377 2222
Thailand	662 278 6777
Turkey	90 212 279 3031
United Kingdom	44 (0) 1635 523545
United States (Corporate)	512 683 0100