

## **NI Vision for LabVIEW™ Help**

June 2008, 370281J-01

NI Vision for LabVIEW adds multipurpose image processing and machine vision capabilities to LabVIEW. This help file contains reference information for the NI Vision for LabVIEW virtual instruments (VIs).

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

**Using Help** 

**Related Documentation** 

**Glossary** 

**Important Information** 

**Technical Support and Professional Services** 

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

## **Examples**

The examples included in NI Vision show you how to use the VIs. Use the **NI Example Finder (Help»Find Examples**) to find and launch NI Vision examples.

© 2000–2008 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** You do not need to activate your software if it is managed by NI Volume License Manager as a part of a Volume License Agreement.

#### 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?

You can find your serial number on the proof-of-ownership and registration card that you received with your product, as shown in the

### following example.



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.

### 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 <u>product activation and licensing</u> refer to ni.com/activate.

### **Related Documentation**

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

The following documents contain information that you may find helpful as you use this help file. You can access NI Vision documents by selecting Start»All Programs»National Instruments»Vision»Documentation»NI Vision.

- NI Vision Development Module Readme—Contains information about new functionality, minimum system requirements, installation instructions, and descriptions of the documentation for the following: NI Vision for LabVIEW, NI Vision for LabWindows/CVI, NI Vision for Visual Basic, and Vision Assistant.
- NI Vision Concepts Manual—Describes the basic concepts of image analysis, image processing, and machine vision. This document also contains in-depth discussions about imaging functions for advanced users.
- NI OCR Training Interface Help—Contains information about how to use the OCR Training Interface to train characters, save character sets, and verify characters by comparing them to a reference character.
- *NI Classification Training Interface Help*—Contains information about how to use the NI Classification Training Interface to train and classify binary samples.
- NI Vision Template Editor Help—Contains information about how to use the NI Vision Template Editor to learn and edit template images that you can use with pattern matching, geometric matching, and golden template comparison functions.

# **Using Help**

**Conventions** 

**Navigating Help** 

Searching Help

Printing Help File Topics

### **Conventions**

This help file uses the following conventions:

< > Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name–for example, DBIO<3..0>.

[] Square brackets enclose optional items—for example, [response].

The » symbol leads you through nested menu items and dialog box options to a final action. The sequence File»Page Setup»Options directs you to pull down the File menu, select the Page Setup item, and select Options from the last dialog box.

This icon denotes a tip, which alerts you to advisory information.

This icon denotes a note, which alerts you to important information.

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 on in the software, such as menu items and dialog box options. Bold text also denotes parameter names, emphasis, or an introduction to a key concept.

dark red Text in this color denotes a caution.

green Underlined text in this color denotes a link to a help topic, help file, or Web address.

italic Italic text denotes variables or cross references. This font also denotes text that is a placeholder for a word or value that you must supply.

monospace Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names,

functions, operations, variables, filenames, and extensions.

# **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.

## **Supported Image Types**

NI Vision VIs that can take an image reference input have image-type icons at the top of their help topics. The icons indicate which image types you can pass in as source/input images.

The following table lists every NI Vision VI that can take an image reference input and indicates which image types the VI supports.

VI Name	B U16	<b>I16</b>	SGL	CSG	RGB U32	HSL U32	RGB U64				
IMAQ 3DView				~				~			
IMAQ Absolute Difference				~		~	~		1		
IMAQ Add				~		~	~		~		
IMAQ Add Classifier Sample				~							
IMAQ Add Custom Sample				~							
IMAQ And				~		~			~	~	
IMAQ ArrayToColorImage									~	~	~
IMAQ ArrayTolmage				~	~	~	~				
IMAQ AutoBThreshold 2				~							
IMAQ AutoMThreshold				~		~	1				
IMAQ AVI Write Frame				~					1		
IMAQ BCGLookup				~							
IMAQ Browser Delete									~		
IMAQ Browser Focus									1		
IMAQ Browser Focus Setup									1		
IMAQ Browser Insert				~		~	1		1	1	~
IMAQ Browser Replace				~		~	~		1	1	~
IMAQ Browser Setup									1		
IMAQ Caliper Tool				~		~	1				
IMAQ CannyEdgeDetection				-							
IMAQ Cast Image				~	~	~	~	~	•	~	~
IMAQ Centroid				~		-	-				
IMAQ Clamp Horizontal Max				~		-	-				

IMAQ Clamp Horizontal Min	~		~	•				
IMAQ Clamp Vertical Max	~		٧	~				
IMAQ Clamp Vertical Min	~		1	~				
IMAQ Classify	~							
IMAQ Clear Overlay	~	>	~	~	~	~	~	~
IMAQ ClipboardToImage	~					~		
IMAQ ColorBCGLookup						~		
IMAQ ColorEqualize						~	~	
IMAQ ColorHistogram						~	~	
IMAQ ColorHistograph						~	~	
IMAQ ColorImageToArray						~	~	~
IMAQ ColorLearn						~	~	
IMAQ ColorMatch						~	~	
IMAQ ColorThreshold						~	~	
IMAQ ColorUserLookup						~	~	
IMAQ Compare Golden Template	~							
IMAQ Concentric Rake 2	~		١	٧				
IMAQ ConstructROI	~	>	٧	1	1	1	1	~
IMAQ Convert Pixel to Real World	~	-	1	1		1	1	
IMAQ Convert Real World to Pixel	~	~	>	1		~	~	
IMAQ Convex Hull	~		٧	1				
IMAQ Convolute	~		~	~				
IMAQ Copy	~	>	~	~	~	~	~	~
IMAQ Copy Overlay	~		>	~	~	~	~	~
IMAQ Correct Calibrated Image	~		~	~		~	~	
IMAQ Correlate	~							
IMAQ Count Objects 2	~		~	~				
IMAQ Danielsson	~		~					
IMAQ Detect Circles	~							
IMAQ Detect Ellipses	-							

IMAQ Detect Lines	~							
IMAQ Detect Rectangles	~							
IMAQ Detect Shapes	-							
IMAQ Dispose	-	>	~	~	~	~	~	1
IMAQ Distance	-							
IMAQ Divide	-		-	~	~	~		
IMAQ Draw	-	>	-	~		~	~	
IMAQ Draw Text	-					~		
IMAQ Edge Tool 2	-		~	~				
IMAQ EdgeDetection	-		~	~				
IMAQ Equalize	~		1	1				
IMAQ Expand	-		~	~		~	~	
IMAQ Extract	-		~	~		~	~	
IMAQ Extract Curves	-							
IMAQ ExtractColorPlanes						~	~	~
IMAQ ExtractSingleColorPlane						1	1	1
IMAQ FillHole	~							
IMAQ FillImage	~	٧	-	-	-	-	-	١
IMAQ Find Circular Edge	~							
IMAQ Find Concentric Edge	~							
IMAQ Find CoordSys (2 Rects) 2	~		١	١				
IMAQ Find CoordSys (Pattern) 2	~							
IMAQ Find CoordSys (Rect) 2	~		1	1				
IMAQ Find Edge	~		1	1				
IMAQ Find Pattern 2	~							
IMAQ Find Straight Edges	~		-	-				
IMAQ Find Straight Edges 2	~		-	-				
IMAQ Flatten Image to String	~	٧	-	~	~	~	>	~
IMAQ Get Calibration Info	~	~	-	-		~	>	
IMAQ Get Classifier Sample Info	~							

IMAQ Get Custom Keys	~	~	~	~	•	~	~	•
IMAQ Get LCD ROI	~		1	~				
IMAQ Get Meter	~		~	~				
IMAQ Get Meter 2	~		~	~				
IMAQ Get Overlay Properties	~	~	~	~	~	~	٧	~
IMAQ GetColorPixelLine						1	~	~
IMAQ GetColorPixelValue						٧	١	1
IMAQ GetImageInfo	1	١	١	١	٧	٧	١	>
IMAQ GetImagePixelPtr	1	١	١	١	٧	٧	١	>
IMAQ GetImageSize	1	١	١	١	٧	٧	١	>
IMAQ GetOffset	1	١	١	١	٧	٧	١	>
IMAQ GetPixelLine	1	١	١	١				
IMAQ GetPixelValue	1	١	١	١				
IMAQ GetPointsOnContour	1							
IMAQ GetRowCol	~	1	٧	٧				
IMAQ Grade Data Matrix Barcode AIM	1							
IMAQ GrayMorphology	~		١	١				
IMAQ Histogram	1		1	1				
IMAQ Histograph	1		٧	١				
IMAQ Image Bit Depth		~	~					~
IMAQ ImageBorderOperation	~	1	٧	٧	1	1	~	~
IMAQ ImageBorderSize	~	~	~	~	~	~	٧	~
IMAQ ImageToArray	~	~	~	~				
IMAQ ImageToClipboard	~					~		
IMAQ ImageToImage 2	~	>	1	~	~	~	٧	~
IMAQ Interlace	~	~	~	~	~	~	٧	~
IMAQ Interpolate 1D	~		~	~				
IMAQ Inverse	~		~	~				
IMAQ Is Vision Info Present 2	~	~	~	~	-	~	~	~
IMAQ Label	~		>	~				

IMAQ Learn Calibration Template	~		-	•		•	~	
IMAQ Learn Color Pattern						~	~	
IMAQ Learn Geometric Pattern	~							
IMAQ Learn Golden Template	~							
IMAQ Learn Multiple Geometric Patterns	1							
IMAQ Learn Pattern 2	~							
IMAQ Light Meter (Line)	~		>	~				
IMAQ Light Meter (Point)	~		٧	~				
IMAQ Light Meter (Rectangle)	~		>	~				
IMAQ Line Gauge	~		>	~				
IMAQ LinearAverages	~		>	~				
IMAQ LineProfile	1		>	~				
IMAQ Local Threshold	~		>					
IMAQ LowPass	1		>	~				
IMAQ MagicWand	1		>	-				
IMAQ MaskToROI	~							
IMAQ Match Color Pattern						1	~	
IMAQ Match Geometric Pattern	~							
IMAQ Match Multiple Geometric Patterns	~							
IMAQ Match Pattern 2	~							
IMAQ MathLookup	٧		~	1				
IMAQ Merge Overlay	~	~	>	~	~	~	~	~
IMAQ Morphology	~							
IMAQ Multiply	~		>	~		~		
IMAQ MultiThreshold	1		>	~				
IMAQ NthOrder	~		>	~				
IMAQ OCR Read Text 3	~					~	~	
IMAQ OCR Train	~					~	~	
IMAQ OCR Verify Text	~					~	~	
IMAQ Or	~		>			~	~	

IMAQ Overlay Arc	~	~	~	~	~	~	~	~
IMAQ Overlay Bitmap	٧	>	٧	~	٧	>	٧	~
IMAQ Overlay Line	~	~	~	~	~	~	~	~
IMAQ Overlay Multiple Lines	~	~	~	~	~	~	~	~
IMAQ Overlay Oval	1	1	1	~	1	1	1	~
IMAQ Overlay Points	~	~	~	~	~	~	~	~
IMAQ Overlay Rectangle	~	~	~	~	~	~	~	~
IMAQ Overlay ROI	~	~	~	~	~	~	~	~
IMAQ Overlay Text	~	~	~	~	~	~	~	~
IMAQ Particle Analysis	~		~	~				
IMAQ Particle Analysis Report	~		~	~				
IMAQ Particle Filter 2	~		~	~				
IMAQ Particle Filter 3	~		~	~				
IMAQ Quantify	1		1	~				
IMAQ Rake 2	٧		٧	~				
IMAQ Read Barcode	~							
IMAQ Read Custom Data	~	~	~	~	~	~	~	~
IMAQ Read Data Matrix Barcode 2	~							
IMAQ Read Image And Vision Info	~	~	~			~	~	~
IMAQ Read LCD	1		1	~				
IMAQ Read Meter	~		~	~				
IMAQ Read Multiple Geometric Template	1							
IMAQ Read PDF417 Barcode	1							
IMAQ Read Single Digit	٧		٧	~				
IMAQ ReadFile	1	1	1	~	1	1	1	~
IMAQ RejectBorder	1							
IMAQ Remove Custom Data	1	1	1	~	1	1	1	~
IMAQ Remove Vision Info 2	~	~	~	~	~	~	~	~
IMAQ RemoveParticle	~							
IMAQ ReplaceColorPlane						>	>	~

IMAQ Resample	~		-	-		-	-	
IMAQ ROIProfile	~		٧	~				
IMAQ ROIToMask	~							
IMAQ Rotate	~		٧	~		~	~	
IMAQ Rotation Detect	~		٧	~				
IMAQ RT Video Out	-	~	>	~	~	~	~	~
IMAQ Select Annulus	-	~	>	~	~	~	~	~
IMAQ Select Line	-	~	1	~	~	~	~	~
IMAQ Select Point	-	~	>	~	~	1	~	~
IMAQ Select Rectangle	-	~	>	~	~	1	~	~
IMAQ Separation	~							
IMAQ Set Calibration Info	-	~	>	~		1	~	
IMAQ Set Overlay Properties	-	~	1	~	~	~	~	~
IMAQ Set Simple Calibration	-	~	>	~		1	~	
IMAQ SetColorPixelLine 2						~	~	~
IMAQ SetColorPixelValue						~	~	~
IMAQ SetImageSize	-	~	>	~	~	1	~	~
IMAQ SetOffset	~	>	٧	1	١	1	١	~
IMAQ SetPixelLine	-	~	>	~				
IMAQ SetPixelValue	-	~	1	~				
IMAQ SetRowCol	~	1	٧	1				
IMAQ Shape Match Tool	~		٧					
IMAQ Shift	~		٧	1		1	٧	
IMAQ Simple Edge	~		1	1				
IMAQ Skeleton	-							
IMAQ Spoke 2	-		1	~				
IMAQ Subtract	~		١	~		~		
IMAQ Symmetry	~		٧	~		~	~	
IMAQ Threshold	~		٧	~				
IMAQ Unwrap	~		~	-		-		

IMAQ UserLookup	•		~					
IMAQ Watershed Transform	~		-					
IMAQ WindDraw	~	~	~	~	~	~	~	~
IMAQ Write BMP File 2	~					~		
IMAQ Write BMP String	-					~		
IMAQ Write Custom Data	-	-	-	-	~	~	~	~
IMAQ Write File 2	-	-	~			~	~	~
IMAQ Write Image And Vision Info	~	~	~			~	~	~
IMAQ Write Image And Vision Info File 2	-	~	~			~	~	~
IMAQ Write Image And Vision Info String	-	~	~			~	~	~
IMAQ Write JPEG2000 File 2	-	-	~			~		~
IMAQ Write JPEG File 2	-					~		
IMAQ Write JPEG String	~	~				~		
IMAQ Write PNG File 2	-	~	~			~		~
IMAQ Write PNG String	~	~	~			~		~
IMAQ Write String	-	~	~			~	~	~
IMAQ Write TIFF File 2	~	•	~			~		~
IMAQ Write TIFF String	~	~	-			~		~

## **Vision Utilities**

Owning Palette: Vision and MotionInstalled With: NI Vision

Development Module

Use the Vision Utilities VIs to create and manipulate images in NI Vision.

Subpalette	Description
<u>Image</u> <u>Management</u>	Use the Image Management VIs to create and dispose of images, set and read attributes of an image, and copy one image to another.
Files	Use the Files VIs to read, write, and retrieve image file information. The file formats NI Vision supports are BMP, TIFF, JPEG, JPEG2000, PNG, AIPD (internal file format), and AVI (for multiple images).
External Display	Use the External Display VIs to do the following in an external display window: display an image, get and set ROIs, manipulate the floating ROI tools window, configure an ROI constructor window, and set up and use an image browser.
Region of Interest	Use the Region of Interest VIs to focus processing and analysis on part of an image. The Region of Interest Conversion subpalette contains VIs convert machine vision point, line, rectangle, and annulus datatypes to or from a Region of Interest descriptor.
Image Manipulation	Use the Image Manipulation VIs to modify an image. Use these VIs to resample an image; extract parts of an image; and rotate, shift, and unwrap images.
Pixel Manipulation	Use the Pixel Manipulation VIs to modify specific areas of an image. Use these VIs to read and set pixel values in an image, read and set values along a row or column in an image, and fill the pixels in an image with a particular value.
Overlay	Use the Overlay VIs to overlay figures, text, and bitmaps onto an image without destroying the image data. Use these VIs to overlay the results of your inspection application onto the images you inspected.
<b>Calibration</b>	Use the Calibration VIs to spatially calibrate an image.

	Spatial calibration converts pixel coordinates to real-world coordinates while compensating for potential perspective errors or nonlinear distortions in your imaging system.
Color Utilities	Use the Color Utilities VIs to manipulate the colors and color planes of an image. Use these VIs to extract different color planes from an image, replace the planes of a color image with new data, convert a color image into a 2D array and back, read and set pixel values in a color image, and convert pixel values from one color space to another.
Vision RT	Use the Vision RT VIs to configure and control display settings for your RT system and to setup time-bounded operations within your application.

# **Image Management**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development

Module

Use the Image Management VIs to create and dispose of images, set and read attributes of an image, and copy one image to another.

Palette Object	Description
IMAQ Create	Creates a temporary memory location for an image.
IMAQ Dispose	Destroys an image and frees the space it occupied in memory. This VI is required for each image created in an application to free the memory allocated to the IMAQ Create VI. Execute IMAQ Dispose only when the image is no longer needed in your application. You can use IMAQ Dispose for each call to IMAQ Create or just once for all images created with IMAQ Create.
IMAQ Image Bit Depth	Gives information regarding the bit depth of an image or modifies the bit depth of an image. The bit depth of an image determines how NI Vision displays the image, casts the image to another image type, and writes the image to a PNG file. Refer to the <a href="IMAQ WindDraw">IMAQ WindDraw</a> VI, the <a href="IMAQ Cast Image">IMAQ Write PNG File</a> 2 VI for more information.
IMAQ GetImageSize	Gives information regarding the size (resolution) of the image.
IMAQ SetImageSize	Modifies the resolution of an image.
IMAQ GetImageInfo	Gives different characteristics of the image.
IMAQ Copy	Copies the specifications and pixels of one image into another image of the same type. You can use this function to keep an original copy of an image (for example, before processing an image). The full definition of the source image as well as the pixel data are copied to the destination image. The border size of the destination image also is modified to be equal to that of

	the source image. If the source image contains additional information, such as calibration information, overlay information, or information for pattern matching, this information is also copied to the destination image.
IMAQ ImageToImage 2	Copies a small image into part of another larger image. This VI is useful for replacing a portion of an image with another image.
IMAQ GetOffset	Returns the position of an image mask in relation to the origin of the coordinate system (0, 0). The default offset value [0, 0] is established when the image is initially created by the <u>IMAQ Create</u> VI. The offset is used only for mask images. By setting different offset values with the <u>IMAQ SetOffset</u> VI, you can move the mask to any location in the image without having to create a new image for each mask.
IMAQ SetOffset	Defines the position of an image mask in relation to the origin of the coordinate system (0, 0).
IMAQ Cast Image	Converts the current image type to the image type specified by <b>Image Type</b> . If you specify a lookup table, the IMAQ Cast Image VI converts the image using a lookup table. If converting from a 16-bit image to an 8-bit image, the VI executes this conversion by shifting the 16-bit pixel values to the right by the specified number of shift operations and then truncating to get an 8-bit value.
IMAQ Is Vision Info Present 2	Checks for the presence of extra vision information associated with the image. Extra vision information can include overlay information, color and grayscale pattern matching templates, calibration information, and custom data.
IMAQ Remove Vision Info 2	Removes specified extra vision information associated with the image. Extra vision information can include overlay information, color and grayscale pattern matching templates, calibration information, and custom data.
IMAQ Read Custom Data	Reads the data associated with a key from an image.

IMAQ Write Custom Data	Associates data with a key in an image. If the key does not exist, it is created. If it does exist, its current value is overwritten.
IMAQ Get Custom Keys	Retrieves all the keys from an image that have data associated with them.
IMAQ Remove Custom Data	Removes a key and its data from an image.
IMAQ Remote Display Options	Sets or gets the image compression options for remote display. These compression options affect remotely acquired images that are transferred and displayed on a host machine.
IMAQ Flatten Image Options	Sets or gets the image compression options for flattening an image to string. These compression options affect images flattened using the LabVIEW Flatten To String primitive.
IMAQ Flatten Image to String	Returns the string representation of an image using the provided options.
IMAQ Multi- Core Options	Sets the number of available processor cores to use for NI Vision applications.

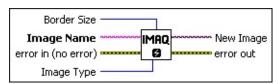
Subpalette	Description
External Library Support	Use the External Library Support VIs for additional external library functionality that is not provided by LabVIEW.  Caution These functions are intended for advanced users. Improper use may result in hanging your application.

## **IMAQ Create VI**

Owning Palette: <a href="mage-Management-Installed With">Image Management</a>Installed With: NI Vision

**Development Module** 

Creates a temporary memory location for an image. Use IMAQ Create in conjunction with the <u>IMAQ Dispose</u> VI to create or dispose of NI Vision images in LabVIEW.

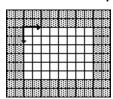


Border Size determines the width, in pixels, of the border to create around an image. These pixels are used only for specific VIs. Create a border at the beginning of your application if an image is to be processed later using functions that require a border (for example, labeling and morphology). The default border value is 3. With a border of three pixels, you can use kernels up to 7 × 7 with no change. If you plan to use kernels larger than 7 × 7 in your process, specify a larger border when creating your image.

The following graphic illustrates an  $8 \times 6$  image with a border equal to 0.



In the following  $8 \times 6$  image, the border equals 2, allowing the use of kernels up to  $5 \times 5$ .





**Note** The border of an image is taken into account only when the image is processed. It is never displayed or stored in a file.

Image Name is the name associated with the created image. Each image created must have a unique name.

error in (no error) describes the error status before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Type specifies the image type. Choose from the following values:

Grayscale (U8) (0)	8 bits per pixel (unsigned, standard monochrome)
<b>Grayscale (16)</b> (1)	16 bits per pixel (signed)
Grayscale (SGL) (2)	32 bits per pixel (floating point)
Complex (CSG) (3)	2 × 32 bits per pixel (floating point)
<b>RGB (U32)</b> (4)	32 bits per pixel (red, green, blue, alpha)
<b>HSL (U32)</b> (5)	32 bits per pixel (hue, saturation, luminance, alpha)
<b>RGB (U64)</b> (6)	64 bits per pixel (red, green, blue, alpha)
<b>Grayscale (U16)</b> (7)	16 bits per pixel (unsigned, standard monochrome)

- New Image is the Image reference that is supplied as input to all subsequent (downstream) functions used by NI Vision. Multiple images can be created in a LabVIEW application.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Dispose VI**

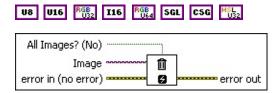
Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Destroys an image and frees the space it occupied in memory. This VI is required for each image created in an application to free the memory allocated to the <a href="IMAQ Create">IMAQ Create</a> VI. Execute IMAQ Dispose only when the image is no longer needed in your application. You can use IMAQ Dispose for each call to IMAQ Create or just once for all images created with IMAQ Create.



**Note** When a LabVIEW application is aborted, allocated images remain in memory.



- All Images? (No) specifies whether to destroy a single image or all previously created images. Giving a TRUE value on input destroys all images previously created. The default is FALSE. Be sure to use this function at the end of an application to free the memory occupied by the images.
- Image specifies the reference to the image to destroy.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

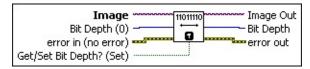
## **IMAQ Image Bit Depth VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Gives information regarding the bit depth of an image or modifies the bit depth of an image. The bit depth of an image determines how NI Vision displays the image, casts the image to another image type, and writes the image to a PNG file. Refer to the <a href="IMAQ WindDraw">IMAQ WindDraw</a> VI, the <a href="IMAQ Cast">IMAQ Write PNG File 2</a> VI for more information.





- **Image** is a reference to the source image.
- Bit Depth (0) specifies the new bit depth of Image. Use this input only when Get/Set Bit Depth? (Set) is TRUE (Set). The value must be from 8 to 15 for signed 16-bit images, from 8 to 16 for unsigned 16-bit images and 64-bit RGB images, or 0. A value of 0 indicates that NI Vision should use the entire range of the image datatype.

**Note** For color images, this parameter specifies the bit depth for each plane.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Bit Depth? (Set) specifies whether to determine the bit depth of the image or modify the bit depth of the image. The default is TRUE (Set).
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Bit Depth returns the bit depth of Image. Use this output only when Get/Set Bit Depth? (Set) is FALSE (Get).
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ GetImageSize VI**

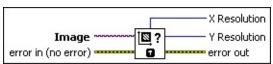
Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Gives information regarding the size (resolution) of the image.

#### **Examples**





- Image is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- X Resolution specifies the horizontal resolution of the image.
- Y Resolution specifies the vertical resolution of the image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

# **Examples**

Refer to the following for an example of how to use this VI.

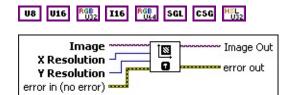
• LabVIEW\examples\Vision\3. Applications\Rotating Part Example.llb

# IMAQ SetImageSize VI

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Modifies the resolution of an image. This VI does not preserve existing image data. If you want to resample an image and preserve the image data, use the <u>IMAQ Resample</u> VI.



- **Image** is a reference to the source image.
- X Resolution specifies the horizontal resolution of the image.
- Y Resolution specifies the vertical resolution of the image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to Image.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ GetImageInfo VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Gives different characteristics of the image.

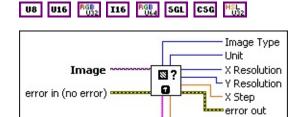


Image is a reference to the source image.

Y Step Image Name

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Type specifies the type of the image. Choose from the following values:

Grayscale (U8) (0)	8 bits per pixel (unsigned, standard monochrome)
<b>Grayscale (16)</b> (1)	16 bits per pixel (signed)
Grayscale (SGL) (2)	32 bits per pixel (floating point)
Complex (CSG) (3)	2 × 32 bits per pixel (floating point)
<b>RGB (U32)</b> (4)	32 bits per pixel (red, green, blue, alpha)
<b>HSL (U32)</b> (5)	32 bits per pixel (hue, saturation, luminance, alpha)
<b>RGB (U64)</b> (6)	64 bits per pixel (red, green, blue, alpha)
<b>Grayscale (U16)</b> (7)	16 bits per pixel (unsigned)

**Unit** is the measuring unit associated with the image. Choose from the following values:

Undefined (0)	Units are undefined.
Angstrom (1)	Units are measured in angstroms.
Micrometer (2)	Units are measured in micrometers.
Millimeter (3)	Units are measured in millimeters.
Centimeter (4)	Units are measured in centimeters.
Meter (5)	Units are measured in meters.
Kilometer (6)	Units are measured in kilometers.
Microinch (7)	Units are measured in microinches.
Inch (8)	Units are measured in inches.
Feet (9)	Units are measured in feet.
Nautical miles (10)	Units are measured in nautical miles.
Standard miles (11)	Units are measured in standard miles.
<b>Steps</b> (12)	Units are measured in steps.

X Resolution specifies the horizontal resolution of the image.

Y Resolution specifies the vertical resolution of the image.

X Step specifies the horizontal distance separating two adjacent

pixels in the specified **Unit**.

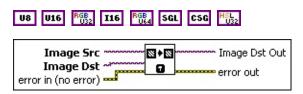
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Y Step specifies the vertical distance separating two adjacent pixels in the specified **Unit**.
- **Image Name** is the name of the image.

## **IMAQ Copy VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Copies the specifications and pixels of one image into another image of the same type. You can use this function to keep an original copy of an image (for example, before processing an image). The full definition of the source image as well as the pixel data are copied to the destination image. The border size of the destination image also is modified to be equal to that of the source image. If the source image contains additional information, such as calibration information, overlay information, or information for pattern matching, this information is also copied to the destination image.



- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ ImageTolmage 2 VI**

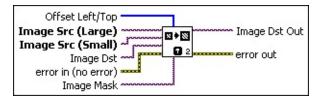
Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Copies a small image into part of another larger image. This VI is useful for replacing a portion of an image with another image.

#### **Details**



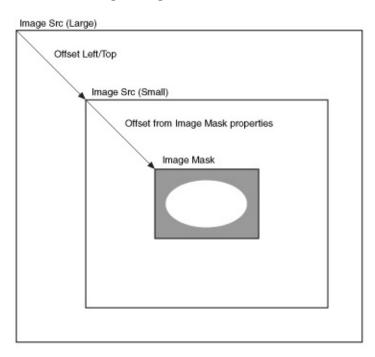


- Offset Left/Top is an array specifying the Image Dst pixel coordinates that receive the image copied from Image Src.
- Image Src (Large) is the reference to the large source image.
- Image Src (Small) is a reference to the small source image, which will be copied into the larger image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src (Large).
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The following image illustrates how this VI functions.



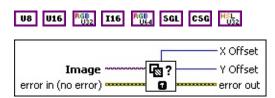
## **IMAQ GetOffset VI**

Owning Palette: <a href="mage-Management-Installed With">Image Management</a> Installed With: NI Vision

**Development Module** 

Returns the position of an image mask in relation to the origin of the coordinate system (0, 0). The default offset value [0, 0] is established when the image is initially created by the <u>IMAQ Create</u> VI. The offset is used only for mask images. By setting different offset values with the <u>IMAQ SetOffset</u> VI, you can move the mask to any location in the image without having to create a new image for each mask.

#### **Details**

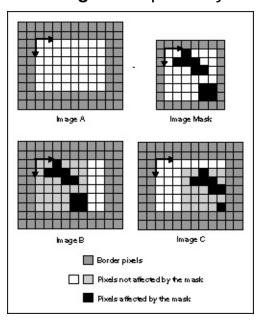


- Image is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- X Offset specifies the horizontal offset of the image mask.
- Y Offset specifies the vertical offset of the image mask.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The following figure illustrates the use of a mask with two different offsets [0, 0] and [3, 1]. A VI processing **Image A** and using the **Image Mask** with an offset of [0, 0] and [3, 1] gives the results as shown in **Image B** and **Image C** respectively. Notice the location of the pixels.



## **IMAQ SetOffset VI**

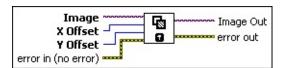
Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Defines the position of an image mask in relation to the origin of the coordinate system (0, 0).

#### **Details Examples**





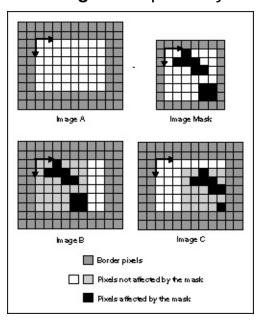
- **Image** is a reference to the source image.
- X Offset specifies the horizontal offset of the image mask.
- Y Offset specifies the vertical offset of the image mask.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

**Image Out** is a reference to **Image**.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The following figure illustrates the use of a mask with two different offsets [0, 0] and [3, 1]. A VI processing **Image A** and using the **Image Mask** with an offset of [0, 0] and [3, 1] gives the results as shown in **Image B** and **Image C** respectively. Notice the location of the pixels.



## **Examples**

Refer to the following for an example of how to use this VI.

• LabVIEW\examples\Vision\2. Functions\Image Management\Mask Example.vi

## **IMAQ Cast Image VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

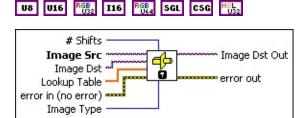
Converts the current image type to the image type specified by **Image Type**. If you specify a lookup table, the IMAQ Cast Image VI converts the image using a lookup table. If converting from a 16-bit image to an 8-bit image, the VI executes this conversion by shifting the 16-bit pixel values to the right by the specified number of shift operations and then truncating to get an 8-bit value.

Refer to the *NI Vision Concepts Manual* for more information about converting image types.



**Note** Refer to <u>Casting Images</u> to learn how the VI casts pixel values when the **Image Src** type and the **Image Type** parameter are not the same.

#### **Examples**



- # Shifts specifies the number of right shifts by which each pixel value in the input image is shifted. This is valid only when converting from a 16-bit image to an 8-bit image. The VI executes this conversion by shifting the 16-bit pixel values to the right by the specified number of shift operations, up to a maximum of 8 shift operations, and then truncating to get an 8-bit value. Enter a value of -1 to ignore the bit depth and shift 0. Enter a value of 0 to use the bit depth to cast the image. Refer to the IMAQ Image Bit Depth VI for information about bit depth.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Lookup Table is an array containing a maximum of 256 elements if Image Src is an 8-bit image or a maximum of 65,536 elements if

**Image Src** is a 16-bit image. The array contains values equal to the index if there are fewer than the amount determined by the image type in **Image Src**. You can use **Lookup Table** to calculate a function that gives a relation between a gray-level value and a user value. This input is valid only when converting from an 8-bit image to a 16-bit image, from a 16-bit image to an 8-bit image, or from an 8-bit or 16-bit image to a 32-bit floating-point image.

After converting **Image Src** into a floating-point type image, you can quantify the image or regions in the image using VIs capable of analyzing floating-point type images.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Type specifies the image type into which the input image is converted. The following values are valid:

	8 bits per pixel (unsigned, standard monochrome)
<b>Grayscale (16)</b> (1)	16 bits per pixel (signed)

Grayscale (SGL) (2)	32 bits per pixel (floating point)
Complex (CSG) (3)	2 × 32 bits per pixel (floating point)
RGB (U32) (4)	32 bits per pixel (red, green, blue, alpha)
<b>HSL (U32)</b> (5)	32 bits per pixel (hue, saturation, luminance, alpha)
<b>RGB (U64)</b> (6)	64 bits per pixel (red, green, blue, alpha)
<b>Grayscale (U16)</b> (7)	16 bits per pixel (unsigned)

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example of how to use this VI.

• LabVIEW\examples\Vision\2. Functions\Operators\Image Averaging Example.vi

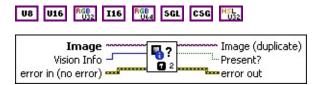
## **IMAQ Is Vision Info Present 2 VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Checks for the presence of extra vision information associated with the image. Extra vision information can include overlay information, color and grayscale pattern matching templates, calibration information, and custom data.

#### **Details**



- Image is the reference to the input image.
- Vision Info specifies the type of vision information you are checking for. Choose from the following values:

<b>Any</b> (0)	Checks if any extra vision information is associated with the image.
Calibration (1)	Checks if any calibration information is associated with the image.
Overlay (2)	Checks if any overlay information is associated with the image.
Grayscale Template (3)	Checks if any grayscale pattern matching template information is associated with the image.
Color Template (4)	Checks if any color pattern matching template information is associated with the image.
Geometric Template (5)	Checks if any geometric matching template information is associated with the image.
Custom Data (6)	Checks if any custom data is associated with the image.
Golden Template (7)	Checks if any golden template information is associated with the image.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image (duplicate) is a reference to the source image.
- **Present?** returns TRUE if the extra vision information specified by **Vision Info** is present in the image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The following VIs associate information with an image:

- The <u>IMAQ Learn Pattern 2</u> VI and the <u>IMAQ Learn Color Pattern</u> VI add pattern matching template information to an image.
- The <u>IMAQ Learn Calibration Template</u> VI, the <u>IMAQ Set</u>
  <u>Calibration Info</u> VI, and the <u>IMAQ Set Simple Calibration</u> VI add calibration information to an image.
- The Overlay VIs add overlay information to an image.
- The <u>IMAQ Write Custom Data</u> VI writes custom data values to an image.

Use the <u>IMAQ Write Image And Vision Info</u> VI to write an image, along with extra vision information associated with the image, to a PNG file. Use the <u>IMAQ Read Image And Vision Info</u> VI to read an image and the extra vision information.

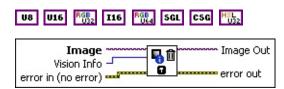
## **IMAQ Remove Vision Info 2 VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Removes specified extra vision information associated with the image. Extra vision information can include overlay information, color and grayscale pattern matching templates, calibration information, and custom data.

#### **Details**



- **Image** is the reference to the input image.
- **Vision Info** specifies the type of vision information to remove. Choose from the following values:

<b>Any</b> (0)	Removes any extra vision information that is associated with the image.
Calibration (1)	Removes any calibration information that is associated with the image.
Overlay (2)	Removes any overlay information that is associated with the image.
Grayscale Template (3)	Removes any grayscale pattern matching template information that is associated with the image.
Color Template (4)	Removes any color pattern matching template information that is associated with the image.
Geometric Template (5)	Removes any geometric matching template information that is associated with the image.
Custom Data (6)	Removes any custom data that is associated with the image.
Golden Template (7)	Removes any golden template information that is associated with the image.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to Image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The following VIs associate information with an image:

- The <u>IMAQ Learn Pattern 2</u> VI and the <u>IMAQ Learn Color Pattern</u> VI add pattern matching template information to an image.
- The <u>IMAQ Learn Calibration Template</u> VI, the <u>IMAQ Set</u>
  <u>Calibration Info</u> VI, and the <u>IMAQ Set Simple Calibration</u> VI add calibration information to an image.
- The Overlay VIs add overlay information to an image.
- The <u>IMAQ Write Custom Data</u> VI writes custom data values to an image.

Use the <u>IMAQ Write Image And Vision Info</u> VI to write an image, along with extra vision information associated with the image, to a PNG file. Use the <u>IMAQ Read Image And Vision Info</u> VI to read an image and the extra vision information.

## **IMAQ Read Custom Data VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Reads the data associated with a key from an image.





- **Image** is the reference to the input image.
- **Key** is the key with which the data is associated.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image (duplicate) is a reference to the source image.
- **Custom Data Out** is the data associated with the key.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

## **IMAQ Write Custom Data VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Associates data with a key in an image. If the key does not exist, it is created. If it does exist, its current value is overwritten.



- **Image** is the reference to the input image.
- **Key** is the key with which the data is associated.
- **Custom Data In** is the data to write to the image.



error in (no error) ----

**Tip** Use the <u>IMAQ Flatten Image to String</u> VI to convert any data type to a string.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Out is a reference to Image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

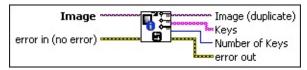
## **IMAQ Get Custom Keys VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Retrieves all the keys from an image that have data associated with them.





- **Image** is the reference to the input image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image (duplicate) is a reference to the source image.
- **Keys** is the array of data keys available in the image.
- Number of Keys is the total number of data keys available in the image.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ Remove Custom Data VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Removes a key and its data from an image.





- Image is the reference to the input image.
- **Key** is the key referencing the data to remove.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to Image.
- **Custom Data Out** is the data associated with the key.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

# **External Library Support**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Use the External Library Support VIs for additional external library functionality that is not provided by LabVIEW.



**Caution** These functions are intended for advanced users. Improper use may result in hanging your application.

Palette Object	Description
IMAQ GetImagePixelPtr	Obtains a pointer on the pixels of an image. This VI also returns information on the organization of the image pixels in memory. Example of IMAQ GetImagePixelPtr.
IMAQ CharPtrToString	Copies a C character string to a LabVIEW string. This VI can accept a char* pointer to get the associated string.
IMAQ MemPeek	Copies memory into a string.
IMAQ Get Window Handle	Returns the operating system handle to an NI Vision external display window.
IMAQ ImageBorderOperation	Fills the border of an image.
IMAQ ImageBorderSize	Sets the border size of the image and determines the current border size of the image.

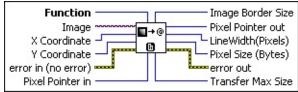
#### IMAQ GetImagePixelPtr VI

Owning Palette: External Library SupportInstalled With: NI Vision Development Module

Obtains a pointer on the pixels of an image. This VI also returns information on the organization of the image pixels in memory. Example of IMAQ GetImagePixelPtr.

#### **Details**





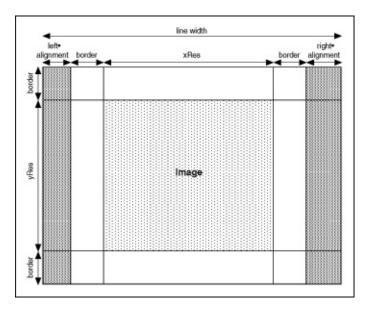
**Function** specifies the method the VI uses to get pixel pointer information. The following options are available:

•	Obtains the pointer on a pixel of an image and obtains information related to the organization of the pixels of this image in memory.
Unmap Pixel Pointer (1)	Frees the pointer and related information previously obtained using <b>Map Pixel Pointer</b> .
Get Pixels Infos (2)	Obtains information related to the organization of the pixels of an image in memory without mapping a pointer.

- Image is a reference to the source image.
- X Coordinate allows you to select the X coordinate of the pixel in the image on which the pointer is required. This parameter is not used in the Unmap Pixel Pointer mode. The default is 0.
- Y Coordinate allows you to select the Y coordinate of the pixel in the image on which the pointer is required. This parameter is not used in the Unmap Pixel Pointer mode. The default is 0.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to

**error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixel Pointer in is used only when Function is set to Unmap Pixel Pointer mode. When the VI is executed to obtain a pointer using the Map Pixel Pointer function, some information regarding the pointer that is required to unmap the pixel pointer is recorded.
  - Note You need to give this pointer to the VI to retrieve this information when executing the Unmap Pixel Pointer function.
- Image Border Size is the border size of the image.
- Pixel Pointer out is the pointer on the pixels of the image. This pointer is obtained only in the Map Pixel Pointer mode.
- LineWidth(Pixels) returns the total number of pixels in a horizontal line in the image. This is the sum of the xRes of the image, the borders of the image, and the left and right alignments of the image, as shown in the following figure. This number may not match the horizontal size of the image.



- Pixel Size (Bytes) returns the size, in bytes, of each pixel in the image. This value multiplied with the **LineWidth** gives the number of bytes occupied by a line of the image in memory.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Transfer Max Size returns the number of bytes from the pixel pointer to the end of the image. This size represents the maximum size of bytes that can be transferred. For example, for an 8-bit image of size 256 × 256 and border 1, the line width is 288, and the maximum transfer size from pixel (0, 0) is 73,728 bytes. Images are 32-byte aligned.



**Note** Pixel (0, 0) is located at the top, left corner of the image.

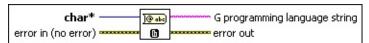
#### **Details**

Refer to <u>IMAQ GetImagePixelPtr Example</u> for a typical implementation scheme of this VI.

#### IMAQ CharPtrToString VI

Owning Palette: External Library SupportInstalled With: NI Vision Development Module

Copies a C character string to a LabVIEW string. This VI can accept a char\* pointer to get the associated string.



- char\* is the C character string pointer. The end of the character string is marked with a 0 (\00) value. The copied string size is limited to 65,536 bytes.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- G programming language string is a LabVIEW string containing all characters before \00 (end of string mark in C).
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

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

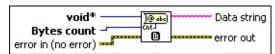
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ MemPeek VI**

Owning Palette: External Library SupportInstalled With: NI Vision

**Development Module** 

Copies memory into a string.



- void\* is the pointer of the memory to be copied.
- Bytes count is the number of bytes to be copied into the string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Data string** is the LabVIEW string containing the bytes of the specified memory zone.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ Get Window Handle VI**

Owning Palette: External Library Support Installed With: NI Vision Development Module

Returns the operating system handle to an NI Vision external display window.



**Note** This VI is not supported on ETS targets or VxWorks targets.



- Window Number specifies the window in memory.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Handle returns the requested external display window handle.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

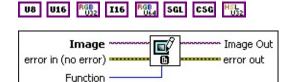
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ ImageBorderOperation VI**

Owning Palette: External Library SupportInstalled With: NI Vision

**Development Module** 

Fills the border of an image.



- Image is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Function** specifies the method used to fill the border of the image. The following options are available:

	Repeats the pixel values of the image near the border into the border by symmetry.
Border	Sets the value of the border pixels to the value of the

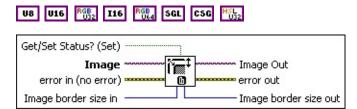
<b>Copy</b> (1)	image pixel near the border.
Border Clear (2)	Sets all border pixels to 0.

- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### IMAQ ImageBorderSize VI

Owning Palette: External Library Support Installed With: NI Vision Development Module

Sets the border size of the image and determines the current border size of the image.



- Get/Set Status? (Set) determines whether the image border size is changed to the image border size value (Set) or the current image border size value is retrieved (Get).
- **Image** is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image border size in determines the new border size of the image.

- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image border size out is the border size of the image.

#### **IMAQ Remote Display Options VI**

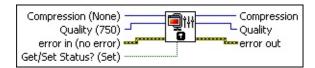
Owning Palette: Vision RTInstalled With: NI Vision Development

Module

Sets or gets the image compression options for remote display. These compression options affect remotely acquired images that are transferred and displayed on a host machine.



**Note** IMAQ Remote Display Options will not affect the remote display options for the Image Display control in versions of LabVIEW prior to LabVIEW 8.0. Use the <u>IMAQ Flatten Image</u> <u>Options</u> VI instead.



Compression (None) specifies the compression method to use when flattening the image to a string. The following values are valid:

None (0)	Performs no image compression on the image.
JPEG (1)	Performs lossy JPEG image compression on the image. JPEG compression may cause data degradation in the flattened image.
	Performs lossless binary packing on the image. This setting is ideal for preserving data integrity when you are compressing binary images. Do not use this setting for nonbinary images.

Quality (750) specifies the quality of the images that you transfer to your host machine for display. The default value is 750. Quality is a measure of the compression method, such that a lower compression factor results in higher image quality. The allowable range for Quality is 0 to 1000. As you lower the Quality value, you increase the amount of loss that occurs during compression.



**Note** The **Quality** parameter is only used when you select **JPEG** compression.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether you want to get or set the remote image display options. The default value is TRUE (Set).
- **Compression** returns the compression method used for remote display.
- Quality returns the quality of the compression method used for remote display.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Flatten Image Options VI**

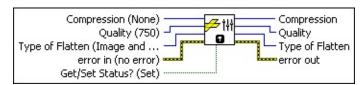
Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Sets or gets the image compression options for flattening an image to string. These compression options affect images flattened using the LabVIEW Flatten To String primitive.



**Note** Values specified for **Compression** and **Quality** also affect the remote display options for the Image Display control in versions of LabVIEW prior to LabVIEW 8.0. In LabVIEW 8.0 and later, the options for flattening and displaying images using the Image Display control can be specified separately. Use the <u>IMAQ</u> Remote <u>Display Options</u> VI to set the remote display options.



**Compression (None)** specifies the compression method to use when flattening the image to a string. The following values are valid:

None (0)	Performs no image compression on the image.
JPEG (1)	Performs lossy JPEG image compression on the image. JPEG compression may cause data degradation in the flattened image.
	Performs lossless binary packing on the image. This setting is ideal for preserving data integrity when you are compressing binary images. Do not use this setting for nonbinary images.

Quality (750) specifies the quality of the images that will be stored when flattening an image to a string. The default value is 750.

Quality is a measure of the compression method, such that a lower compression factor results in higher image quality. The allowable range for Quality is 0 to 1000. As you lower the Quality value, you increase the amount of loss that occurs during compression.



**Note** The **Quality** parameter is only used when you select **JPEG** compression.

Type of Flatten (Image and Vision Info) specifies what type of data to store in the string. The following values are valid:

Image (0)	Stores the image pixel data and overlay information in the string.
Image and Vision Info (1)	(Default) Stores the image pixel data and Vision Extra Information, such as calibration, overlay, template, and custom data information, in the image string.
to	Stores only a reference to the image in the string. Unflattening the string will return another reference to the same image. You will not be able to unflatten the string if the original image has been disposed.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether you want to get or set the flattening options. The default value is TRUE (Set).
- **Compression** returns the compression method used to flatten

- images to string.
- Quality returns the quality of the compression method used to flatten images to string.
- Type of Flatten returns the type of data to store when flattening an image to string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

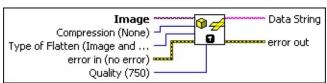
## **IMAQ Flatten Image to String VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Returns the string representation of an image using the provided options.





- **Image** is a reference to the source image.
- **Compression (None)** specifies the compression method to use when flattening the image to a string. The following values are valid:

None (0)	Performs no image compression on the image.
JPEG (1)	Performs lossy JPEG image compression on the image. JPEG compression may cause data degradation in the flattened image.
	Performs lossless binary packing on the image. This setting is ideal for preserving data integrity when you are compressing binary images. Do not use this setting for nonbinary images.

Type of Flatten (Image and Vision Info) specifies what type of data to store in the string. The following values are valid:

Image (0)	Stores the image pixel data and overlay information in the string.
Image and Vision Info (1)	(Default) Stores the image pixel data and Vision Extra Information, such as calibration, overlay, template, and custom data information, in the image string.
to	Stores only a reference to the image in the string. Unflattening the string will return another reference to the same image. You will not be able to unflatten the string if the original image has been disposed.

error in (no error) describes the error status before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Quality (750) specifies the quality of the image stored in the string. The default value is 750. Quality is a measure of the compression method, such that a lower compression factor results in higher image quality. The allowable range for Quality is 0 to 1000. As you lower the Quality value, you increase the amount of loss that occurs during compression.
  - Note The Quality parameter is only used when you select JPEG compression.
- **Data String** is the LabVIEW string containing the string representation of the image. Use the Unflatten from String LabVIEW primitive to convert this string back to an image. If an image has the same name as the flattened image in memory, when the string is unflattened, the contents of the flattened image string are copied to the image. Otherwise, a new image with a unique name is created that contains the contents of the flattened image string.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

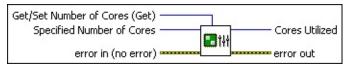
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning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

## **IMAQ Multi-Core Options VI**

Owning Palette: Image ManagementInstalled With: NI Vision

**Development Module** 

Sets the number of available processor cores to use for NI Vision applications.



Get/Set Number of Cores (Get) specifies whether the VI gets or sets the number of processor cores available to NI Vision.

<b>Get</b> (0)	(Default) Returns the number of available processor cores
Set (1)	Specifies the number of processor cores available to NI Vision
Set Max Available (2)	Specifies that NI Vision should have access to all available processor cores

- Specified Number of Cores specifies the number of processor cores available to NI Vision.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Cores Utilized returns the number of processor cores used by NI Vision.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Files**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Files VIs to read, write, and retrieve image file information. The file formats NI Vision supports are BMP, TIFF, JPEG, JPEG2000, PNG, AIPD (internal file format), and AVI (for multiple images).

Palette Object	Description
IMAQ ReadFile	Reads an image file. The file format can be a standard format (BMP, TIFF, JPEG, JPEG2000, PNG, and AIPD) or a nonstandard format known to the user.
IMAQ Write File 2	Writes the image to a file in the selected format.
IMAQ Read Image And Vision Info	Reads an image file, including any extra vision information saved with the image. This includes overlay information, pattern matching template information, and calibration information, as written by the <a href="IMAQ Write Image And Vision Info">IMAQ Write Image And Vision Info</a> VI.
IMAQ Write String	Returns a string that contains the entire contents of the image encoded as an image file. The string can be saved to file, or, for example, transferred to a remote computer using FTP.
IMAQ GetFileInfo	Obtains information regarding the contents of the file. This information is supplied for standard file formats only: BMP, TIFF, JPEG, JPEG2000, PNG, or AIPD.
IMAQ Load Image Dialog	Displays a dialog box with which you can specify a path or paths to images. This dialog box is similar to the LabVIEW File Dialog primitive, but it provides a preview of the image files selected.

Subpalette	Description
<u>AVI</u>	Use the AVI VIs to read and write multiple images to an AVI
	file. You can write compressed AVIs and additional data,

such as time-stamp data, with your images.

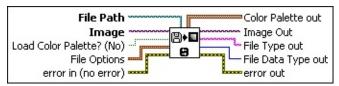
#### **IMAQ ReadFile VI**

Owning Palette: Files Installed With: NI Vision Development Module

Reads an image file. The file format can be a standard format (BMP, TIFF, JPEG, JPEG2000, PNG, and AIPD) or a nonstandard format known to the user. In all cases, the read pixels are converted automatically into the image type passed by **Image**.

#### **Examples**





- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- Image is a reference to the image to which data from the image file is applied.
- **Load Color Palette? (No)** determines whether to load the color table present in the file, if the color table exists. If loaded, this table is read and returned by the output **Color Palette**. The default is FALSE.
- **File Options** is a cluster of user-optional values that you can use to read nonstandard file formats. The structure must be known to the user. This cluster consists of the following elements:
  - Read Raw File indicates whether the file to be read has a nonstandard file format. If so, the remaining options in this cluster describe how to read the data.
    - Note If you are reading a nonstandard file format, you must set the size of **Image** to match the contents of the file before using IMAQ ReadFile.
    - Note If you specify an unsigned 16-bit integer for File Data Type and Image is a signed 16-bit integer, all values greater than 32,767 are truncated to 32,767.

Don't Read Raw	Specifies that the file to be read has a
<b>File</b> (0)	standard file format.

Read Raw	Specifies that the file to be read has a
File (1)	nonstandard file format.

- File Data Type indicates how the image file is encoded.
- Offset to Data specifies the size, in bytes, of the file header. This part of the file is not taken into account when read. The pixel values are read from the byte immediately after the offset size. The default is 0.
- **Use Min Max** determines if the user is using a predetermined minimum and maximum. The technique to determine this minimum and maximum depends on the following input values:

Don't use min max (0)	Minimum and maximum are dependent on the type of image. For an 8-bit image, min = 0 and max = 255.
Use file values (1)	Pixel values from the file are scanned one time to determine the minimum and maximum, and a linear interpolation is performed before loading the image.
Use optional values (2)	Uses the optional values <b>Optional Min Value</b> and <b>Optional Max Value</b> .

- Optional Min Value is the minimum value of the pixels if Use Min Max is set to Use optional values. In this case, pixels with a smaller value are altered to match the chosen minimum. The default is 0.
- Optional Max Value is the maximum value of the pixels if Use Min Max is set to Use optional values. In this case, pixels with a greater value are truncated to match the chosen maximum. The default is 255.
- Byte Order determines if the byte weight is to be swapped (Intel or Motorola). The default is FALSE, which specifies big endian (Motorola). TRUE specifies little endian (Intel). This input is useful only if the pixels are encoded with more than 8 bits.

**error in (no error)** describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the **Simple Error Handler** or **General Error Handler** VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette out contains the RGB color table (if the file has one) read from the file when the user passes the value TRUE for the input Load Color Palette? (No).
  - **Red** is the output red value.
  - **Green** is the output green value.
  - **Blue** is the output blue value.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- **File Type out** indicates the file type that is read. This string returns an identifier of the file format, which can be BMP, TIFF, JPEG, PNG, or AIPD (internal file format). **File Type** returns xxx if the file format is unknown.
- File Data Type out indicates the pixel size defined in the header for standard image file types. File Options are not necessary for

reading standard image files. For other types of image files, the returned values are passed from **File Options/File Data Type**.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Analysis\Histogram Example.vi

# **IMAQ Write File 2 VI**

Owning Palette: Files Installed With: NI Vision Development Module

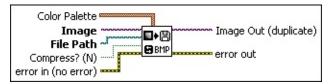
Writes the image to a file in the selected format.

Use the pull-down menu to select an instance of this VI.

Select an instance	•

## **IMAQ Write BMP File 2**

Writes an image to a file in BMP format. Color Palette is used to apply a color palette to an image.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - **Blue** is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- **File Path** is the complete pathname, including drive, directory, and filename, of the file to write.
- Compress? (N) designates whether to compress the BMP file. By default, the image file is not compressed (FALSE). If it is compressed, it uses the **Run Length Encoded** compression type.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

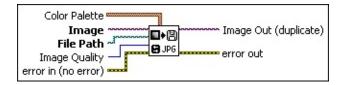
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Write JPEG File 2**

Writes an image to a file in JPEG format.



**Note** JPEG format is not recommended if you want to make precise gauging or inspection measurements on the image.



- **Color Palette** is an unused input. The JPEG file format does not support color palettes. The input exists to address backward-compatibility issues.
- Image is a reference to the image structure to write as an image file.
- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- Image Quality specifies the amount of compression applied to the image. Image Quality can range from 0 to 1000. The default value is 750. The higher the value, the less compression the VI applies.
  - D
- **Note** JPEG compression is lossy, which means that the lower the quality specified, the more loss takes place during the compression. Also, a small amount of loss can occur even when the quality is 1000.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

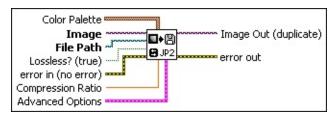
- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write JPEG2000 File 2**

Writes an image to a file in JPEG2000 format.



**Note** This VI is not supported on VxWorks targets.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - **Green** is the value of the green color plane.
  - Blue is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- Lossless? (true) specifies the type of compression used to save the image. Set this parameter to TRUE to save the image with no loss of data. Set this parameter to FALSE to save the image with a loss of data.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

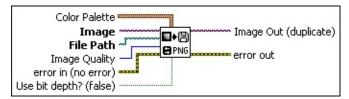
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Compression Ratio is the degree to which to compress the JPEG2000 file. For example, if the Compression Ratio is 50, the resulting file will be 50 times smaller than the size of the image in memory. This parameter is ignored if Lossless? is TRUE.
- Advanced Options is a cluster of optional values that you can use to set advanced JPEG2000 options. This cluster consists of the following elements:
  - Wavelet transform mode determines the wavelet transform to use when writing the file.
  - **Use multi-component transform?** specifies whether or not to use a multi-component transform. This parameter has no effect when encoding grayscale images.
  - Max wavelet transform level specifies the maximum allowed level of wavelet transform. Increasing this value will result in a more accurate image, but will increase the time to write the image. Valid values are from 0 to 255.
  - Quantization step size specifies the absolute base quantization step size for derived quantization mode. This parameter has no effect when Wavelet transform mode is Integer.
- Image Out (duplicate) is a reference to Image. This VI does not

modify the image connected to the **Image** input.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write PNG File 2**

Writes an image to a file in PNG format.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - Blue is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- Image Quality specifies the amount of compression applied to the image. Image Quality can range from 0 to 1000. The default value is 750. The higher the value, the less compression the VI applies.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Use bit depth? (false) When saving a signed 16-bit image to a PNG file, NI Vision must convert the data to an unsigned format and shift the data so that most significant bit is always the leftmost bit. Set this parameter to TRUE to use the bit depth information attached to image to perform these conversions. Set this parameter to FALSE to bias the image by adding a constant value to all the pixels in the image such that the lowest negative pixel value in the image maps to zero, and then shifting the image data based on the highest pixel value in the image. The default is FALSE.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

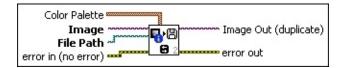
error or warning. The default is an empty string.

### **IMAQ Write Image And Vision Info File 2**

Writes an image, along with extra vision information associated with the image, to a PNG file. This extra vision information includes overlay information, pattern matching template information, and calibration information. Note that this vision information can only be saved with the PNG file format.



**Note** The extra vision information can be saved only with the PNG file format.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - **Blue** is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

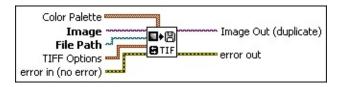
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Write TIFF File 2**

Writes an image to a file in TIFF format.



**Note** 16-bit monochrome images and 64-bit RGB images are nonstandard extensions of the TIFF standard. Most third-party applications cannot read 16-bit monochrome or 64-bit RGB TIFF files. For compatibility with most applications, write 16-bit monochrome images or 64-bit RGB images into PNG files.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - **Blue** is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- File Path is the complete pathname, including drive, directory, and filename, of the file to read.
- **TIFF Options** is a cluster of optional values that you can use to set TIFF options. This cluster consists of the following elements:
  - Rows Per Strip designates the number of rows you want to designate per strip of data. The default value of 0 specifies to write the entire data as one strip.
  - Photometric designates whether to interpret white or black as a zero value.

WhiteIsZero (0)	Designates white as the zero value
BlackIsZero (1)	Designates black as the zero value

Compression Type determines the compression type. By default, no compression is used. Choose from the following options:



**Note** The **JPEG** compression type is not valid for 16-bit monochrome or 64-bit RGB images.



**Note** If you select the **JPEG** compression type, the VI ignores the **Color Palette**.

None (0)	No compression is used on the image
JPEG (1)	JPEG compression is used on the image
Run Length Encoded (2)	Run Length Encoding is used on the image
<b>Zip</b> (3)	Zip compression is used on the image

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

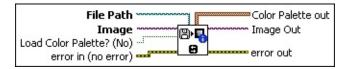
## **IMAQ Read Image And Vision Info VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Reads an image file, including any extra vision information saved with the image. This includes overlay information, pattern matching template information, and calibration information, as written by the <a href="IMAQ Write">IMAQ Write</a> <a href="Image And Vision Info">Image And Vision Info</a> VI.

#### **Examples**





- File Path is the complete pathname, including drive, directory, and filename, of the file to read.
- Image is a reference to the image structure to which the data from the image file is applied.
- **Load Color Palette? (No)** determines whether to load the color table present in the file, if the color table exists. If loaded, this table is read and returned by the output **Color Palette**. The default is FALSE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette out indicates an array of clusters composed of 256 elements for each of the three color planes. A specific color is the result of applying a value between 0 and 255 for each of the three color planes (red, green, and blue). If the three planes have an identical value, a gray level is obtained (0 specifies black and 255 specifies white).
  - **Red** is the output red value.
  - **Green** is the output green value.
  - **Blue** is the output blue value.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for examples that use this VI.

- LabVIEW\examples\Vision\3. Applications\Battery Clamp Inspection Example.vi
- LabVIEW\examples\Vision\2. Functions\Calibration\Simple Calibration Example.llb

# **IMAQ Write String VI**

Owning Palette: Files Installed With: NI Vision Development Module

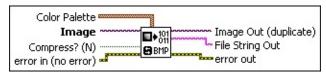
Returns a string that contains the entire contents of the image encoded as an image file. The string can be saved to file, or, for example, transferred to a remote computer using FTP.

Use the pull-down menu to select an instance of this VI.

Select an instance	-
--------------------	---

## **IMAQ Write BMP String**

Returns a string that contains the entire contents of the image encoded as a BMP image file.

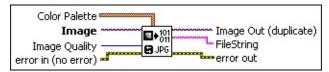


- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - Blue is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- Compress? (N) designates whether to compress the BMP file. By default, the image file is not compressed (FALSE). If it is compressed, it uses the **Run Length Encoded** compression type.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- File String Out is a string that contains the entire contents of the encoded image file. The string can then be saved to disk using binary file writing VIs, or transmitted to another computer.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write JPEG String**

Returns a string that contains the entire contents of the image encoded as a JPEG image file.



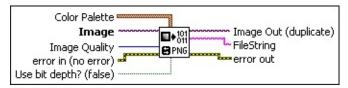
- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - Blue is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- Image Quality specifies the amount of compression applied to the image. Image Quality can range from 0 to 1000. The default value is 750. The higher the value, the less compression the VI applies.
  - Note JPEG compression is lossy, which means that the lower the quality specified, the more loss takes place during the compression. Also, a small amount of loss can occur even when the quality is 1000.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- FileString is a string that contains the entire contents of the encoded image file. The string can then be saved to disk using binary file writing VIs, or transmitted to another computer.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write PNG String**

Returns a string that contains the entire contents of the image encoded as a PNG image file.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - Blue is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- Image Quality specifies the amount of compression applied to the image. Image Quality can range from 0 to 1000. The default value is 750. The higher the value, the less compression the VI applies.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Use bit depth? (false) When saving a signed 16-bit image to a PNG file, NI Vision must convert the data to an unsigned format and shift the data so that most significant bit is always the leftmost bit. Set this parameter to TRUE to use the bit depth information attached to image to perform these conversions. Set this parameter to FALSE to bias the image by adding a constant value to all the pixels in the image such that the lowest negative pixel value in the image maps to zero, and then shifting the image data based on the highest pixel value in the image. The default is FALSE.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- **FileString** is a string that contains the entire contents of the encoded image file. The string can then be saved to disk using binary file writing VIs, or transmitted to another computer.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

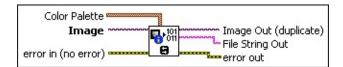
most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write Image And Vision Info String**

Writes an image, along with extra vision information associated with the image, to a PNG file. This extra vision information includes overlay information, pattern matching template information, and calibration information.



**Note** The extra vision information can only be saved with the PNG file format.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - **Blue** is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

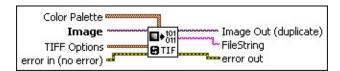
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- **File String Out** is a string that contains the entire contents of the encoded image file. The string can then be saved to disk using binary file writing VIs, or transmitted to another computer.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write TIFF String**

Returns a string that contains the entire contents of the image encoded as a TIFF image file.



**Note** 16-bit monochrome images and 64-bit RGB images are nonstandard extensions of the TIFF standard. Most third-party applications cannot read 16-bit monochrome or 64-bit RGB TIFF files. For compatibility with most applications, write 16-bit monochrome images or 64-bit RGB images into PNG files.



- Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - Red is the value of the red color plane.
  - **Green** is the value of the green color plane.
  - **Blue** is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- **TIFF Options** is a cluster of optional values that you can use to set TIFF options. This cluster consists of the following elements:
  - Rows Per Strip designates the number of rows you want to designate per strip of data. The default value of 0 specifies to write the entire data as one strip.
  - Photometric designates whether to interpret white or black as a zero value.

WhiteIsZero (0) Designates white as the zero value

	1
BlackIsZero (1)	Designates black as the zero value

Compression Type determines the compression type. By default, no compression is used. Choose from the following options:



**Note** The **JPEG** compression type is not valid for 16-bit monochrome or 64-bit RGB images.



**Note** If you select the **JPEG** compression type, the VI ignores the **Color Palette**.

None (0)	No compression is used on the image
JPEG (1)	JPEG compression is used on the image
Run Length Encoded (2)	Run Length Encoding is used on the image
<b>Zip</b> (3)	Zip compression is used on the image

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

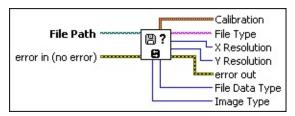
most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- **FileString** is a string that contains the entire contents of the encoded image file. The string can then be saved to disk using binary file writing VIs, or transmitted to another computer.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ GetFileInfo VI**

Owning Palette: Files Installed With: NI Vision Development Module

Obtains information regarding the contents of the file. This information is supplied for standard file formats only: BMP, TIFF, JPEG, JPEG2000, PNG, or AIPD.



- File Path is the complete pathname—including drive, directory, and filename—for the file to be examined. This path can be supplied by either the user or the File Dialog VI from LabVIEW.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Calibration** is a cluster containing the following elements:
  - X Step is the horizontal distance separating two adjacent

- pixels (in units specified by Unit).
- Y Step is the vertical distance separating two adjacent pixels (in units specified by Unit).
- **Unit** is the measuring unit associated with the image. Choose from the following values:

Undefined (0)	Units are undefined.
Angstrom (1)	Units are measured in angstroms.
Micrometer (2)	Units are measured in micrometers.
Millimeter (3)	Units are measured in millimeters.
Centimeter (4)	Units are measured in centimeters.
Meter (5)	Units are measured in meters.
Kilometer (6)	Units are measured in kilometers.
Microinch (7)	Units are measured in microinches.
Inch (8)	Units are measured in inches.
Feet (9)	Units are measured in feet.
<b>Nautical miles</b> (10)	Units are measured in nautical miles.
Standard miles (11)	Units are measured in standard miles.
Steps (12)	Units are measured in steps.

- **File Type** indicates the file type that is read: BMP, TIFF, JPEG, JPEG2000, PNG, or AIPD (internal file format).
- X Resolution specifies the horizontal resolution of the image.
- Y Resolution specifies the vertical resolution of the image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is

- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **File Data Type** indicates the pixel size defined in the header of the file.
- Image Type specifies the type of image that best fits the File Data Type of the file. This output can have the following values:

Grayscale (U8) (0)	8 bits per pixel (unsigned, standard monochrome)
<b>Grayscale (16)</b> (1)	16 bits per pixel (signed)
Grayscale (SGL) (2)	32 bits per pixel (floating point)
Complex (CSG) (3)	2 × 32 bits per pixel (floating point)
<b>RGB (U32)</b> (4)	32 bits per pixel (red, green, blue, alpha)
<b>HSL (U32)</b> (5)	32 bits per pixel (hue, saturation, luminance, alpha)
<b>RGB (U64)</b> (6)	64 bits per pixel (red, green, blue, alpha)
<b>Grayscale (U16)</b> (7)	16 bits per pixel (unsigned, standard monochrome)

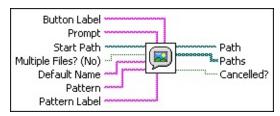
# **IMAQ Load Image Dialog VI**

Owning Palette: Files Installed With: NI Vision Development Module

Displays a dialog box with which you can specify a path or paths to images. This dialog box is similar to the LabVIEW File Dialog primitive, but it provides a preview of the image files selected.



**Note** This VI is not supported on ETS targets or VxWorks targets.



**Button Label** the label to display on the **OK** button in the file dialog box.



**Note** If the value of this parameter is longer than the width of the button, the file dialog box does not display the entire label. In an English version of Windows, the button is approximately 11 characters wide.

- **Prompt** is a custom message that appears in the title bar of the dialog box. The default is Open.
- Start Path is the path of the directory whose contents LabVIEW initially displays in the dialog box. If this parameter is invalid or unwired, the last directory viewed in a file dialog box appears in the dialog box.
- Multiple Files? (No) determines whether multiple files can be selected in the dialog box. The default is FALSE.
- **Default Name** is the name you want to appear as the initial file or directory name in the dialog box. The default is an empty string.
- Pattern restricts the files displayed in the dialog box to those whose name matches pattern. This parameter does not restrict the directories displayed. The pattern matching in this VI is similar to the matching used when matching wildcards in Windows filenames. If you specify characters other than the question mark character (?) or the asterisk character (\*), the VI displays only files or directories that contain those characters. You can use the question mark

character (?) to match any single character. You can use the asterisk character (\*) to match any sequence of one or more characters.

For example, a pattern of \*.png;test\*.bmp returns matches for any file with a .png extension and any file whose filename begins with test and has an .bmp extension.

To match multiple patterns, use a semicolon (;) to separate the patterns. White space—such as blanks, tabs, and carriage returns—are taken literally. Avoid using white spaces unless they are part of the extension pattern. For example, if you use \*.jpg;\*.tif, the dialog box displays all files that end with .jpg and .tif. If you use \*.jpg; \*.tif, the dialog box displays only files that end with .jpg.

- Pattern Label is the label to display in the file dialog box next to the custom pattern. If you do not wire a string to this input, the default label next to any custom pattern is Custom Filter. If you do not wire a string to pattern, LabVIEW ignores this input.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Path is the full path to the file selected using this dialog box, if multiSelect is set to FALSE.
- Paths is the full path to the files selected using this dialog box, if multiSelect is set to TRUE.
- **Cancelled?** is TRUE if you cancel the dialog box or if an error occurs during the execution of the dialog box.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **AVI**

Owning Palette: Files Installed With: NI Vision Development Module Use the AVI VIs to read and write multiple images to an AVI file. You can write compressed AVIs and additional data, such as time-stamp data, with your images.

Palette Object	Description
IMAQ AVI Create	Creates a new AVI file or rewrites an old AVI file.
IMAQ AVI Write Frame	Writes an image to the AVI file specified by <b>Avi Refnum</b> .
IMAQ AVI Close	Closes the AVI file associated with the AVI Refnum.
IMAQ AVI Open	Opens an existing AVI file.
IMAQ AVI Read Frame	Reads an image from the AVI file specified by Avi Refnum.
IMAQ AVI Get Info	Obtains information about the AVI file associated with Avi Refnum.
IMAQ AVI Get Filter Names	Finds the compatible AVI compression filters currently installed on the computer. Each filter has a different compression quality, compression size, and compression time.

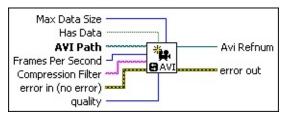
### **IMAQ AVI Create VI**

Owning Palette: AVIInstalled With: NI Vision Development Module Creates a new AVI file or rewrites an old AVI file.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**



- Max Data Size is the maximum number of bytes you can add per frame to an AVI file. The default is 10,000.
- Has Data specifies whether to write additional data with your images.
  - Note This option is available only with NI Vision.
- AVI Path is the complete pathname, including drive, directory, and filename, for the AVI file. This path can be supplied by either the user or the File Dialog VI from LabVIEW.
  - Note The file extension must be .avi.
- Frames Per Second is the number of frames per second at which to play the AVI.
  - Note Frame Per Second indicates the desired playback rate of the AVI you create. The AVI may play at a slower rate depending on the performance of the system on which it plays.
- Compression Filter is the name of the compression filter used to write the frames to the AVI file. Use the <a href="IMAQ AVI Get Filter Names">IMAQ AVI Get Filter Names</a> VI to get the names of the valid compression filters currently installed on your computer.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- quality is the relative quality, from 0–1000, to use with the compression filter. Pass –1 for this parameter to use the default quality of the filter. Not all filters support using different qualities.
- Avi Refnum is a reference to the AVI file created.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\AVI\AVI Read Write with Data.llb

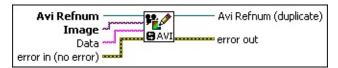
# **IMAQ AVI Write Frame VI**

Owning Palette: AVI Installed With: NI Vision Development Module Writes an image to the AVI file specified by Avi Refnum.



**Note** This VI is not supported on ETS targets or VxWorks targets.





- Avi Refnum is a reference to the AVI file output from the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.

  Create VI or the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.
- Image is a reference to the image that you want to write to an AVI file. This image must be either an 8-bit or RGB image.
- **Data** is a string associated with the image to write. To use this control, the **Has Data** control of the <u>IMAQ AVI Create</u> VI must be TRUE.
  - Note The Data option is available only with NI Vision. Refer to LabVIEW\examples\Vision\2. Functions\AVI\AVI Read Write with Data.llb for information about how to use this option.
  - **Tip** Use the Flatten to String VI to convert any data type to a string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Avi Refnum (duplicate) is the reference to the AVI file.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ AVI Close VI**

Owning Palette: AVIInstalled With: NI Vision Development Module Closes the AVI file associated with the AVI Refnum.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**



- Avi Refnum is a reference to the AVI file output from the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.

  Create VI or the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for

more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\AVI\Read AVI Example.vi

### **IMAQ AVI Open VI**

Owning Palette: AVIInstalled With: NI Vision Development Module Opens an existing AVI file.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**



AVI Path is the complete pathname, including drive, directory, and filename, for the AVI file. This path can be supplied by either the user or the File Dialog VI from LabVIEW.



**Note** The file extension must be .avi.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Avi Refnum is a reference to the AVI file opened.

**error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

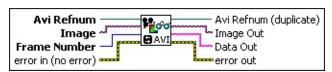
• LabVIEW\examples\Vision\2. Functions\AVI\Read AVI Example.vi

### **IMAQ AVI Read Frame VI**

Owning Palette: AVI Installed With: NI Vision Development Module Reads an image from the AVI file specified by Avi Refnum.



**Note** This VI is not supported on ETS targets or VxWorks targets.



- Avi Refnum is a reference to the AVI file output from the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.

  Create VI or the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.
- Image is a reference to the image that you want to contain the AVI image after you call this VI.
- Frame Number identifies which frame of the AVI file to read.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Avi Refnum (duplicate) is the reference to the AVI file.

- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- **Data Out** is the data associated with the image being read.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

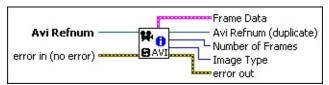
# **IMAQ AVI Get Info VI**

**Owning Palette:** AVIInstalled With: NI Vision Development Module Obtains information about the AVI file associated with Avi Refnum.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**



- Avi Refnum is a reference to the AVI file output from the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.

  Create VI or the <a href="IMAQ AVI Open VI">IMAQ AVI Open VI</a>.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Frame Data is a cluster containing the width and height of the images, frames per second of the AVI, compression filter used to create the AVI file, and whether the AVI file has data associated

with each image.

- width
- height
- **FISSE** framesPerSecond
- **compressionFilter**
- **hasData**
- Avi Refnum (duplicate) is the reference to the AVI file.
- Image Type specifies the type of image used in the AVI file.
- Number of Frames is the number of frames contained in the AVI file.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\AVI\Read AVI Example.vi

### **IMAQ AVI Get Filter Names VI**

Owning Palette: AVIInstalled With: NI Vision Development Module

Finds the compatible AVI compression filters currently installed on the computer. Each filter has a different compression quality, compression size, and compression time.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Filter Names** is an array of the compatible AVI compression filters currently installed on the machine.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

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

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\AVI\AVI Compressor Comparison.llb

# **External Display**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the External Display VIs to do the following in an external display window: display an image, get and set ROIs, manipulate the floating ROI tools window, configure an ROI constructor window, and set up and use an image browser.

Palette Object	Description
IMAQ WindDraw	Displays an image in an image window. The image window appears automatically when the VI is executed. The default image window does not have scrollbars. You can add scrollbars by using the IMAQ WindSetup VI.
IMAQ WindClose	Closes an image window.
IMAQ WindShow	Shows or hides an image window.
IMAQ WindMove	Indicates and sets the position of an image window.
IMAQ WindSize	Indicates and sets the size of an image window. You also can use this VI to add or remove scrollbars for image windows and test for the presence of scrollbars in an image window.
IMAQ WindZoom 2	Obtains or modifies the status of the zoom factor. An image can be displayed using either a positive zoom factor, which represents the image larger than actual size, or a negative zoom factor, which represents the image smaller than actual size. When the size of the image window is smaller than the image displayed, use this VI to specify the position of the image that is centered in the image window.
IMAQ WindSetup	Configures the appearance and attributes of a window.
IMAQ WindDisplayMapping	Sets/gets the pixel mapping policy for displaying 16-bit images with an unspecified bit depth. Because 16-bit grayscale images cannot be

	displayed with their full resolution on 32-bit color displays using common video adapters limited to 8-bit resolution/per pixel/color, 16-bit images need to be mapped to the 8-bit range (0 to 255).
IMAQ GetPalette	Selects a display palette. Five predefined palettes are available. To activate a color palette, choose a code for <b>Palette Number</b> and connect the <b>Color Palette</b> output to the input <b>Color Palette</b> of the <a href="IMAQ WindDraw">IMAQ WindDraw</a> VI.
IMAQ ConstructROI	Allows the user to specify a region of interest (ROI) on the image. IMAQ ConstructROI displays the image in the specified window and provides the set of tools defined to select an ROI. IMAQ ConstructROI also returns the ROI descriptor that corresponds to the ROI drawn on the image when you click <b>OK</b> in the constructor window.
IMAQ WindGetROI	Returns the descriptor for an ROI.
IMAQ WindSetROI	Associates an ROI with an image window.
IMAQ WindEraseROI	Erases the active region of interest associated with an image window.
IMAQ WindROIColor	Selects the color to use to depict ROI contours in the image window.
IMAQ WindLastEvent	Returns the events generated through the image windows as well as the data associated with them.
IMAQ WindGrid	Obtains or modifies the status of the grid. If enabled, the grid is invisible and drawn ROIs will be constrained to it. The grid can be used to help trace a region of interest accurately.
IMAQ WindBackground	Allows you to change the way the background of an image window is displayed.
IMAQ WindNonTearing	Allows you to configure the non-tearing functionality of a display window. Tearing images can occur when the image display rate is not synchronized with the refresh rate of the monitor. The difference between the display rate and the monitor refresh rate can cause parts of two

	different images to be displayed at the same time, which results in a split in the displayed image. By enabling non-tearing, the image display is synchronized to the refresh of the monitor and tearing is eliminated.
IMAQ WindToolsSetup	Configures the appearance and availability of the region tools found in the NI Vision Tools palette. With no input connections, the palette displays all nine region tools. The NI Vision Tools palette is a floating palette and is always visible.
IMAQ WindToolsSelect	Obtains or modifies the status of the region tools.
IMAQ WindToolsShow	Shows or hides the WindTools palette. This VI functions in the same way as the IMAQ WindShow VI, which is used for displaying image windows.
IMAQ WindToolsMove	Obtains or sets the position of the NI Vision Tools palette. This VI functions in the same way as the <a href="MAQ WindMove">IMAQ WindMove</a> VI, which is used for moving image windows.
IMAQ WindToolsClose	Closes the NI Vision Tools palette. This VI functions in the same way as the IMAQ WindClose VI, which is used for closing image windows.

Subpalette	Description	
<u>Browser</u>	Use the Browser VIs to create, configure, and use an image	
	browser.	

### **IMAQ WindDraw VI**

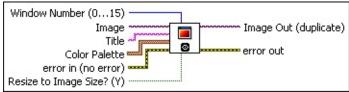
Owning Palette: External DisplayInstalled With: NI Vision Development Module

Displays an image in an image window. The image window appears automatically when the VI is executed. The default image window does not have scrollbars. You can add scrollbars by using the <a href="MAQ">IMAQ</a> WindSetup VI.

Use the <u>IMAQ Image Bit Depth</u> VI to set the bit depth of 16-bit monochrome and 64-bit RGB images. IMAQ WindDraw uses this specified bit depth to display the image.

Use the <u>IMAQ WindDisplayMapping</u> VI to set the pixel mapping policy for displaying 16-bit images of an unspecified bit depth.





Window Number (0...15) specifies the window in which to display the image. You can display up to 16 windows simultaneously. Each window is identified by a number ranging from 0 to 15. Only the specified image window is affected; all other image windows remain the same. The default value is 0.



**Note** When using LabVIEW Real-Time, set this control to 15 to display to RT Video Out. You also can use IMAQ RT Video Out to display to RT Video Out.

Image is a reference to the displayed image.



**Note** Floating point images are displayed by scaling the pixel values to 8 bits. 16-bit images that have a specified bit depth are displayed by scaling the pixel values to 8 bits based on the specified bit depth. 16-bit images that do not have a specified bit depth are displayed according to the IMAO WindDisplayMapping VI.

Title is an image window name. When a string is connected to this

input, the image window automatically takes that name.

Color Palette is used to apply a color palette to the image window. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.



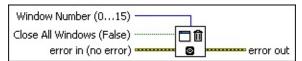
- **Tip** For best results, set your video adapter to high color or true color.
- Red is the value of the red color plane.
- Green is the value of the green color plane.
- **Blue** is the value of the blue color plane.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Resize to Image Size? (Y) specifies whether the user wants to resize the image window automatically to fit the image size. The default is TRUE (yes), in which case the user does not have to know the size of a source image before displaying it.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image In input.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ WindClose VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Closes an image window.



- Window Number (0...15) specifies the image window to close. The window is identified by a number from 0 to 15. The default value is 0.
- Close All Windows (False) specifies whether all the image windows are to be closed. The default value FALSE (No) closes only the specified window. Setting this value to TRUE closes all windows simultaneously.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an

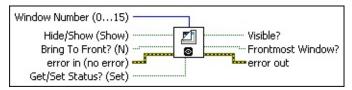
error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ WindShow VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Shows or hides an image window.



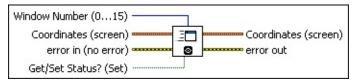
- Window Number (0...15) specifies the image window to show or hide. The window is identified by a number from 0 to 15. The default value is 0.
- Hide/Show (Show) specifies whether an image window is visible. The VI considers this input only when **Get/Set Status?** is TRUE (Set).
- Bring To Front? (N) determines whether to bring a window to the foreground. This input is used only when Get/Set Status? is TRUE (Set) and Hide/Show is also TRUE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

- most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the visibility status of an image window. The default is TRUE (Set).
- **Visible?** returns the current visibility status of the image window. A visible image window returns TRUE.
- **Frontmost Window?** returns TRUE if an image window is in the front.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ WindMove VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Indicates and sets the position of an image window.



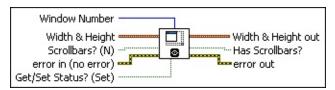
- Window Number (0...15) specifies the image window. The window is identified by a number from 0 to 15. The default value is 0.
- **Coordinates (screen)** specifies the coordinates (x and y) of an image window.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the coordinates of an image window. The default is TRUE (Set).

- **Coordinates (screen)** returns the current coordinates (x and y) of an image window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ WindSize VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Indicates and sets the size of an image window. You also can use this VI to add or remove scrollbars for image windows and test for the presence of scrollbars in an image window.



- Window Number is a number from 0 to 15 that specifies the image window. The default value is 0.
- Width & Height is a cluster containing two elements. When the input Get/Set Status? is TRUE (Set), you can specify the width and height of an image window. If the input is not connected, or if the value is (0, 0), the image window is resized automatically to the image associated with it.
  - S

**Note** This value includes the size of the scrollbars and title bar.

- Scrollbars? (N) controls the presence of scrollbars in an image window. By default, scrollbars are not used. You can resize and move an image window whether scrollbars are present or absent.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

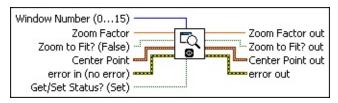
default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the size of an image window. The default value is TRUE (Set).
- **Width & Height out** returns the current width and height of an image window.
- Has Scrollbars? returns the current scrollbar status for an image window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ WindZoom 2 VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Obtains or modifies the status of the zoom factor. An image can be displayed using either a positive zoom factor, which represents the image larger than actual size, or a negative zoom factor, which represents the image smaller than actual size. When the size of the image window is smaller than the image displayed, use this VI to specify the position of the image that is centered in the image window.



- Window Number (0...15) specifies the image window. The window is identified by a number from 0 to 15. The default value is 0.
- **Zoom Factor** can have the following values: 1 to 16 and –1 to –16. The default value is 1 (image is displayed at the original size).
- **Zoom to Fit? (False)** specifies whether the window will automatically zoom to fit the image in the image window.
- **Center Point** is a cluster containing two elements that describe the (x, y) coordinates used to center the image in the image window.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

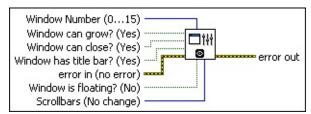
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the present values of **Zoom Factor** and **Center Point**.
- Zoom Factor out returns the current zoom factor.
- **Zoom to Fit? out** returns whether the window will automatically zoom to fit the image.
- Center Point out returns the current coordinates of the Center Point.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ WindSetup VI**

Owning Palette: External DisplayInstalled With: NI Vision Development

Module

Configures the appearance and attributes of a window.



- Window Number (0...15) specifies the image window to configure. The window is identified by a number from 0 to 15. The default value is 0. Specify a value of 15 to configure the video out port on a LabVIEW Real-Time system.
- Window can grow? (Yes) enables or disables the capability for the user to resize the image window. The default is TRUE, which indicates the user can resize the image window.
- Window can close? (Yes) shows or does not show the close box of the window. The default is TRUE, which shows the close box.
- Window has title bar? (Yes) shows or does not show the title bar. The default is TRUE, which shows the title bar.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

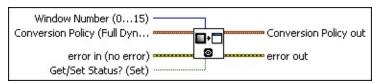
- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Window is floating? (No) produces either a normal or a floating (always on top) window. The default is FALSE, which produces a standard, nonfloating window.
- Scrollbars (No change) controls the presence of scrollbars in a window. You can either force scrollbars to be hidden, force them to be shown, or use the current window scrollbar settings.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ WindDisplayMapping VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Sets/gets the pixel mapping policy for displaying 16-bit images with an unspecified bit depth. Because 16-bit grayscale images cannot be displayed with their full resolution on 32-bit color displays using common video adapters limited to 8-bit resolution/per pixel/color, 16-bit images need to be mapped to the 8-bit range (0 to 255).

#### **Details**



- Window Number (0...15) specifies the image window to configure. The window is identified by a number from 0 to 15. The default value is 0. Specify a value of 15 to configure the video out port on a LabVIEW Real-Time system.
- **Conversion Policy (Full Dynamic)** is the display conversion to use if Get/Set Status? is set to TRUE.
  - Conversion Method has the following values:

Full Dynamic (0)	The full dynamic range of the 16-bit image is mapped to an 8-bit (256 grayscale values) scale. 16-bit images are displayed by scaling the data to 8 bits, calculated as a function of the dynamic range from the image source. The minimum value (min) and the maximum value (max) are calculated automatically. Then the following formula is applied to each pixel:
	Display(x, y) = (Src(x, y) - min) $\times$ 255/(max - min)
Downshift (1)	The 16-bit image pixels are shifted to the right the number of times specified by the # <b>Shifts</b> cluster member.
Given	The pixel values in the range specified by

Range (2)	Minimum Value and Maximum Value are mapped to an 8-bit scale.
90 % Dynamic (3)	The dynamic range containing the middle 90% of the cumulated histogram of the image is mapped to an 8-bit (256 grayscale values) scale.
Given Percent Range (4)	The pixel values in the relative percentage range (0 to 100) of the cumulated histogram specified by <b>Minimum Value</b> and <b>Maximum Value</b> are mapped to an 8-bit scale.

- Minimum Value is the minimum value used for Given Range and Given Percent Range. When Conversion Method is set to Given Range, Minimum Value represents the value that is mapped to 0. When Conversion Method is set to Given Percent Range, Minimum Value represents the percentage of the range used to compute the pixel value mapped to 0.
- Maximum Value is the maximum value used for Given Range and Given Percent Range. When Conversion Method is set to Given Range, Maximum Value represents the value that is mapped to 255. When Conversion Method is set to Given Percent Range, Maximum Value represents the percentage of the range used to compute the pixel value mapped to 255.
- # **Shifts** is the number of bits to right-shift the pixel values for the **Downshift** conversion method.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the conversion policy for displaying a 16-bit image in an image window The default value is TRUE (Set).
- **Conversion Policy out** returns the current display conversion policy for the Window Number specified.

**Conversion Method** has the following values:

Full Dynamic (0)	The full dynamic range of the 16-bit image is mapped to an 8-bit (256 grayscale values) scale. 16-bit images are displayed by scaling the data to 8 bits, calculated as a function of the dynamic range from the image source. The minimum value (min) and the maximum value (max) are calculated automatically. Then the following formula is applied to each pixel:
	Display(x, y) = $(Src(x, y) - min) \times 255/(max - min)$
Downshift (1)	The 16-bit image pixels are shifted to the right the number of times specified by the # <b>Shifts</b> cluster member.
Given Range (2)	The pixel values in the range specified by <b>Minimum Value</b> and <b>Maximum Value</b> are mapped to an 8-bit scale.
90 % Dynamic (3)	The dynamic range containing the middle 90% of the cumulated histogram of the image is mapped to an 8-bit (256 grayscale values)

scale.
The pixel values in the relative percentage range (0 to 100) of the cumulated histogram specified by <b>Minimum Value</b> and <b>Maximum Value</b> are mapped to an 8-bit scale.

- Minimum Value is the minimum value used for Given Range and Given Percent Range. When Conversion Method is set to Given Range, Minimum Value represents the value that is mapped to 0. When Conversion Method is set to Given Percent Range, Minimum Value represents the percentage of the range used to compute the pixel value mapped to 0.
- Maximum Value is the maximum value used for Given Range and Given Percent Range. When Conversion Method is set to Given Range, Maximum Value represents the value that is mapped to 255. When Conversion Method is set to Given Percent Range, Maximum Value represents the percentage of the range used to compute the pixel value mapped to 255.
- # **Shifts** is the number of bits to right-shift the pixel values for the Downshift conversion method.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

IMAQ WindDisplayMapping enables the mapping technique used when displaying a 16-bit grayscale image. According to the **Conversion Method** selected, the image full dynamic is mapped to 8 bits (**Full Dynamic**), a given number of right shift is applied to the 16-bit pixels (**Downshift**), the range corresponding to the center 90% of the cumulative histogram is mapped to 8-bits (**90% Dynamic**). The default mapping technique is the **Full Dynamic**, where 16-bit images are displayed by scaling the data to 8 bits, calculated as a function of the dynamic range from the image source. The minimum value (min) and the maximum value (max) are calculated automatically. Then the following formula is applied to each pixel:

 $Display(x, y) = (Src(x, y) - min) \times 255/(max - min)$ 

## **IMAQ GetPalette VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Selects a display palette. Five predefined palettes are available. To activate a color palette, choose a code for **Palette Number** and connect the **Color Palette** output to the input **Color Palette** of the <u>IMAQ</u> WindDraw VI.



Palette Number (gray) gives you a choice of five predefined palettes. You can choose from the following values:

<b>Gray</b> (0)	Grayscale is the default palette. Gradual grayscale variation from black to white. Each value is assigned an equal amount of the RGB intensities.
Binary (1)	Designed especially for binary images. 16 cycles of 16 different colors, where g is the grayscale value and $g=0$ corresponds to $R=0$ , $G=0$ , $B=0$ (black); $g=1$ corresponds to $R=1$ , $G=0$ , $B=0$ (red); $g=2$ corresponds to $R=0$ , $G=1$ , $G=0$ (green); and so on.
Gradient (2)	Gradation from red to white with a prominent range of light blue in the upper range. 0 is black and 255 is white.
Rainbow (3)	Gradation from blue to red with a prominent range of greens in the middle value range. 0 is black and 255 is white.
Temperature (4)	Gradation from light brown to dark brown. 0 is black and 255 is white.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette indicates an array of clusters composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 for each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white).
  - Note This output is to be directly connected to the input Color Palette of the IMAQ WindDraw VI.
  - Red is the output red value.
  - Green is the output green value.
  - **Blue** is the output blue value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is

a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

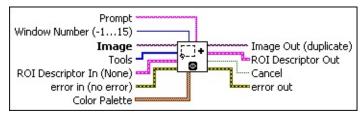
**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ ConstructROI VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Allows the user to specify a region of interest (ROI) on the image. IMAQ ConstructROI displays the image in the specified window and provides the set of tools defined to select an ROI. IMAQ ConstructROI also returns the ROI descriptor that corresponds to the ROI drawn on the image when you click **OK** in the constructor window.





- **Prompt** specifies a message string to display in the title bar of the window. Use this control to provide the user with instructions about selecting the object.
- Window Number (-1...15) specifies the window number to display the image The default value is -1. When you select -1, the IMAQ ConstructROI VI displays a modal dialog window centered on the screen. When you select a regular NI Vision window number (0-15), the VI displays the Image in the specified window and temporarily sets the NI Vision window to modal mode. When the user clicks OK or Cancel, the attributes of the window are reset to their initial values.
- Image is a reference to the source image to use in the constructor window.
- **Tools** is an array of tools specifying which tools are available. The following tools are available:

Selection Tool (0)	Select an ROI in the image.	
Point Tool (1)	Select a pixel in the image.	
Line Tool (2)	Draw a line in the image.	
Rectangle Tool (3)	Draw a rectangle or square in the image.	

Oval Tool (4)	Draw an oval or circle in the image.	
Polygon Tool (5)	Draw a polygon in the image.	
<b>Freehand Region Tool</b> (6)	Draw a freehand region in the image.	
Annulus Tool (7)	Draw an annulus in the image.	
Zoom Tool (8)	Zoom in or out in an image.	
Panning Tool (9)	Shift the view of the image.	
<b>Broken Line Tool</b> (10)	Draw a broken line in the image.	
Freehand Line Tool (11)	Draw a freehand line in the image.	
Rotated Rectangle Tool (12)	Draw a rotated rectangle in the image.	

- ROI Descriptor In (None) is a descriptor of the ROI that appears initially on the ROI Constructor window. Use this input to specify an ROI that the user can modify by resizing or moving it.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette is used to apply a color palette to an image window. Color Palette is an array of clusters constructed by the user or supplied by the <a href="IMAQ GetPalette">IMAQ GetPalette</a> VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue).

A specific color is the result of applying a value between 0 and 255 to each of the three color planes (red, green, and blue). If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). The color palette can be used only for 8-bit images.

- Red is the value of the red color plane.
- Green is the value of the green color plane.
- Blue is the value of the blue color plane.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- **ROI Descriptor Out** returns the ROI descriptor created with the constructor window.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the

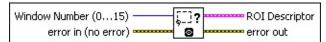
#### contour.

- **Cancel** returns TRUE if the user clicks **Cancel** in the constructor window and returns FALSE if the user clicks **OK**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ WindGetROI VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Returns the descriptor for an ROI.



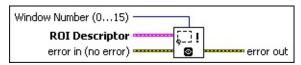
- Window Number (0...15) specifies the image window. The window is identified by a number from 0 to 15. The default value is 0.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- ROI Descriptor is a descriptor that defines the region of interest.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or

- internal edge of an ROI.
- **Type** is the shape type of the contour.
- **Coordinates** indicates the relative position of the contour.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ WindSetROI VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Associates an ROI with an image window.



- Window Number (0...15) specifies the image window. The window is identified by a number from 0 to 15. The default value is 0.
- **ROI Descriptor** is a descriptor that defines the region of interest to associate with an image window.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

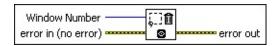
## IMAQ WindEraseROI VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Erases the active region of interest associated with an image window.



**Note** You can erase an ROI in an image window by selecting it and pressing <Delete>.



- Window Number is a number from 0 to 15 that specifies the image window. The default value is 0.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

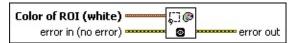
the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

## **IMAQ WindROIColor VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Selects the color to use to depict ROI contours in the image window.



- Color of ROI (white) is a cluster that specifies the color of the ROI.
  - Red gives the red plane intensity. The default is 255.
  - Green gives the green plane intensity. The default is 255.
  - Blue gives the blue plane intensity. The default is 255.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

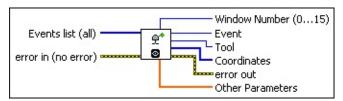
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning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

## IMAQ WindLastEvent VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Returns the events generated through the image windows as well as the data associated with them.

#### **Details**



Events list (all) specifies which events to obtain. The default case returns all events generated through the image windows as well as the data associated with them. This VI enables you to specify the image window events that interest you.

No event (0)	No event.	
Click event (1)	A user has clicked in an image window.	
Draw event (2)	A user has drawn in an image window.	
Move event (3)	A user has moved an image window.	
Size event (4)	A user has resized an image window.	
Scroll event (5)	A user has moved the scrollbars in an image window.	
Activate event (6)	A user has chosen (clicked once to activate) an image window.	
Close event (7)	A user has closed an image window.	
Double- click (8)	A user has double-clicked in an image window.	

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to

**error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Window Number (0...15) indicates the image window that generated the event.
- **Event** indicates the type of event.
- **Tool** returns the tool used to generate the event.
- **Coordinates** indicates the relative position of the event.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

Other Parameters supplies information associated with an event, such as positioning and region distances.

**Details** 

The following table describes the possible values for the **Event**, **Tool**, **Coordinates**, and **Other Parameters** indicators.

Event	Tool	Coordinates	Other Parameters
None	NA	empty	empty
Click	Selection, Point	[0, 1] position (x, y) of click	[0, 1, 2] pixel value (See Note below.)
	Zoom	[0, 1] position of click	[0] zoom factor
		[2, 3] position of image center	
Draw	Line	[0, 1] position of starting point	[0, 1] width and height
		[2, 3] position of ending point	[2] vertical segment angle [3] segment length
	Rectangle, Oval, Annulus, Rotated Rectangle	[03] bounding rectangle	[0, 1] width and height
	Polygon, Freehand Region, Broken Line, Freehand Line	[03] bounding rectangle, [4(n x 2) + 3] the location (x, y) of n vertices	[0, 1] width and height
Move	NA	[0, 1] position of image window	empty
Size	NA	[0, 1] width and height of image window	empty
Scroll	None, Pan	[0, 1] center position of image	empty
Activate	NA	empty	empty
Close	NA	empty	empty

Double- click		[0, 1, 2] pixel value (See Note
CIICK		below.)



**Note** Pixel values are stored in the first element of the array for 8-bit, 16-bit, and floating-point images. The RGB, RGB U64, and HSL values of color images are stored in the order [0, 1, 2]. The real and imaginary values of a complex image are stored in the order [0, 1].

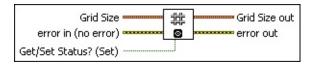
## **IMAQ WindGrid VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Obtains or modifies the status of the grid. If enabled, the grid is invisible and drawn ROIs will be constrained to it. The grid can be used to help trace a region of interest accurately.



**Note** This VI is not supported on ETS targets or VxWorks targets.



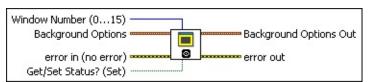
- Grid Size is the new size of the grid if Get/Set Status (Set) is TRUE. Otherwise, the VI ignores this input.
  - X Offset specifies the horizontal offset of the grid.
  - Y Offset specifies the vertical offset of the grid.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the step values for the grid. The default is TRUE (Set).
- Grid Size out returns the present grid size.
  - X Offset indicates the horizontal offset of the grid.
  - Y Offset Indicates the vertical offset of the grid.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ WindBackground VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Allows you to change the way the background of an image window is displayed.



Window Number (0...15) selects the window to configure. The default is 0.

Background Options are the options used to configure the window if Get/Set Status? is TRUE. This parameter is ignored if Get/Set Status? is False. The following table describes which parameters are valid for each Fill Style option:

Fill Style	Hatch Style	Fill Color	<b>Background Color</b>
Default	_	_	_
Solid	_	×	_
Hatch	×	×	×

Fill Style specifies the color used to fill the image window. The following are valid values:

Default (0)	uses the default NI Vision background.	
Solid (1)	specifies a solid background for the window.	
<b>Hatch</b> (2) specifies a hatch pattern background for the window.		

Hatch Style specifies the background color of the image window. The following values are valid:

Horizontal (0)	specifies that the background of the display window will be horizontal bars.
Vertical (1)	specifies that the background of the display window will be vertical bars.
Forward	specifies that the background of the display

Diagonal (2)	window will be diagonal bars. The bars start in the lower-left corner and end in the upper-right corner of the window.
Backward Diagonal (3)	specifies that the background of the display window will be diagonal bars. The bars start in the upper-left corner and end in the lower-right corner of the window.
Cross (4)	specifies that the background of the display window will be intersecting horizontal and vertical bars.
Cross Hatch (5)	specifies that the background of the display window will be intersecting forward and backward diagonal bars.

- Fill Color specifies the color used to fill the image window.
- **Background Color** specifies the background color of the image window.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

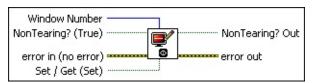
error or warning. The default is an empty string.

- Get/Set Status? (Set) specifies whether you want to get or set the flattening options. The default value is TRUE (Set).
- **Background Options Out** returns the current background options for the specified window.
  - **Fill Style** returns the type of fill selected for the specified window.
  - Hatch Style is the pattern of the background.
  - Fill Color is the color used to fill the image window.
  - **Background Color** is the background color of the image window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## IMAQ WindNonTearing VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Allows you to configure the non-tearing functionality of a display window. Tearing images can occur when the image display rate is not synchronized with the refresh rate of the monitor. The difference between the display rate and the monitor refresh rate can cause parts of two different images to be displayed at the same time, which results in a split in the displayed image. By enabling non-tearing, the image display is synchronized to the refresh of the monitor and tearing is eliminated.



- Window Number selects the window to configure. The default is 0.
- NonTearing? (True) enables non-tearing on a display window when TRUE and disables non-tearing when FALSE. The default is TRUE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Set / Get (Set) specifies whether you want to retrieve or configure the non-tearing functionality of the window. The default value is TRUE (Set).
- NonTearing? Out is the status of the non-tearing functionality for the specified display window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Browser**

Owning Palette: External DisplayInstalled With: NI Vision Development

Module

Use the Browser VIs to create, configure, and use an image browser.

Palette Object	Description
IMAQ Browser Setup	Configures the browser. This VI sets all the required parameters that determine the aspect and the thumbnail organization on the browser.
IMAQ Browser Insert	Adds or inserts an image into the browser. The image is resampled to fit the size of the thumbnail image. The resampling value applied to the image is the same on both axis to maintain the image aspect. Adds the thumbnail at either the last position or the first vacant position, depending on the selected mode. If all positions are occupied, an additional line of thumbnail images is added to the browser.
	Replaces an image in the browser. The resampling ratio is computed to be as small as possible and the same on both axes. Set only one of the following controls: Index, Matrix Indexes, or Pixel Position.
IMAQ Browser Delete	Deletes an image from the browser. Set only one of the following controls: <b>Index</b> , <b>Matrix Indexes</b> , or <b>Pixel Position</b> . After deleting an image, the browser reorganizes the remaining images by filling the free space or leaving it vacant.
IMAQ Browser Focus Setup	Configures the appearance of a thumbnail that is selected on the browser. A selected image is framed using the <b>Focus Color</b> .
IMAQ Browser Focus	Gets or sets the status of images on the browser. You can choose either of the possible status values: selected or not selected. Selected images are framed using the Focus Color (refer to the <a href="IMAQ Browser Focus Setup">IMAQ Browser Focus Setup</a> VI description).

## **IMAQ Browser Setup VI**

Owning Palette: BrowserInstalled With: NI Vision Development Module

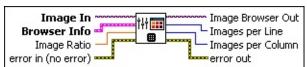
Configures the browser. This VI sets all the required parameters that determine the aspect and the thumbnail organization on the browser. You create and configure a browser using this VI. These tasks are mandatory because all the functions in this library act on browsers and not on standard images.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**





- Image In is the RGB image used by the browser. This image forms the base of the browser. All thumbnails placed in the browser are pasted onto this image.
- Browser Info is the cluster configuring the initial aspect of the browser. This cluster contains the following parameters:
  - Browser Size is the size of the browser in pixels, where X is the width and Y is the height.
    - X is the width to make the browser.
    - Y is the height to make the browser.
  - Images per Line is the number of thumbnail images per line on the browser.
  - **Background Color** is the color filling the background of the browser.
  - Frame Size is the width of the frame that contains each thumbnail.
  - Frame Style defines the frame style of the browser. You can choose from the following frame styles:
    - Raised
    - Beveled

- Outline
- Hidden
- Step
- Raised Outline
- Image Ratio is the width/height ratio of the thumbnail images of the browser. This value is 3/2 for PAL images and 4/3 for NTSC images.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Browser Out is the configured browser image.
- Images per Line is the number of thumbnail images per line on the browser.
- Images per Column is the number of thumbnail images per column on the browser.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

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

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

 $\bullet \quad LabVIEW \land examples \land Vision \land 2. \ Functions \land Display \land Browser \ Example. vi$ 

### **IMAQ Browser Insert VI**

Owning Palette: BrowserInstalled With: NI Vision Development Module

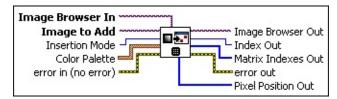
Adds or inserts an image into the browser. The image is resampled to fit the size of the thumbnail image. The resampling value applied to the image is the same on both axis to maintain the image aspect. Adds the thumbnail at either the last position or the first vacant position, depending on the selected mode. If all positions are occupied, an additional line of thumbnail images is added to the browser.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**





- Image Browser In is the RGB image that the browser uses.
- Image to Add is the image to add or insert in the browser. This image can be an 8-bit, 16-bit, floating-point, RGB, or HSL image.
- Insertion Mode is the mode used for inserting the thumbnail. Two modes are available:

Adds the thumbnail at the first vacant position in the browser.
Adds the thumbnail at the last position in the browser.

Color Palette is used to apply a color palette to the thumbnail image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). The default is a grayscale palette. The color palette can be used only for 8-bit images.

- **Tip** For best results, set your video adapter to high color or true color.
- Red is the value of the red color plane.
- Green is the value of the green color plane.
- Blue is the value of the blue color plane.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Browser Out is the resulting browser image.
- Index Out is the index of the browser position of the replaced thumbnail.
- Matrix Indexes Out is the (Column, Line) position of the added thumbnail.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixel Position Out returns the x- and y-coordinates of the topmost-leftmost pixel and bottommost-rightmost pixel of the browser area in which the image is inserted.

# **Examples**

Refer to the following for an example that uses this VI.

 $\bullet \quad LabVIEW \land examples \land Vision \land 2. \ Functions \land Display \land Browser \ Example. vi$ 

## **IMAQ Browser Replace VI**

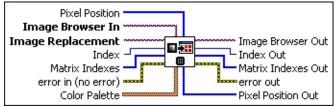
Owning Palette: BrowserInstalled With: NI Vision Development Module

Replaces an image in the browser. The resampling ratio is computed to be as small as possible and the same on both axes. Set only one of the following controls: **Index**, Matrix **Indexes**, or **Pixel Position**.



**Note** This VI is not supported on ETS targets or VxWorks targets.





- **Pixel Position** is the (x, y) pixel position of the image to replace.
- Image Browser In is the RGB image that the browser uses.
- Image Replacement is the image to place in the browser.
- **Index** is the position of the image to replace.
- Matrix Indexes is the (Column, Line) position of the thumbnail to replace.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette is used to apply a color palette to the thumbnail image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). The default is a grayscale palette. The color palette can be used only for 8-bit images.
  - (P)
- **Tip** For best results, set your video adapter to high color or true color.
- Red is the value of the red color plane.
- Green is the value of the green color plane.
- **Blue** is the value of the blue color plane.
- Image Browser Out is the resulting browser image.
- Index Out is the index of the browser position of the replaced thumbnail.
- Matrix Indexes Out is the (Column, Line) position of the replaced thumbnail.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a

warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Pixel Position Out returns the x- and y-coordinates of the topmost-leftmost pixel and bottommost-rightmost pixel of the position of the replaced thumbnail.

## **IMAQ Browser Delete VI**

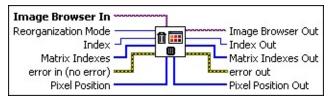
Owning Palette: BrowserInstalled With: NI Vision Development Module

Deletes an image from the browser. Set only one of the following controls: **Index**, **Matrix Indexes**, or **Pixel Position**. After deleting an image, the browser reorganizes the remaining images by filling the free space or leaving it vacant.



**Note** This VI is not supported on ETS targets or VxWorks targets.





- Image Browser In is the RGB image that the browser uses.
- **Reorganization Mode** is the remapping mode of the browser.
- **Index** is the position of the image to delete.
- Matrix Indexes is the (Column, Line) position of the image to delete.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixel Position defines the pixel coordinates of the area or image to delete.
- Image Browser Out is the resulting browser image.
- Index Out is the index of the browser position of the deleted thumbnail.
- Matrix Indexes Out is the (Column, Line) position of the removed thumbnail.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixel Position Out returns the x- and y-coordinates of the topmost-leftmost pixel and bottommost-rightmost pixel of the position of the deleted thumbnail.

## **IMAQ Browser Focus Setup VI**

Owning Palette: BrowserInstalled With: NI Vision Development Module

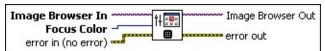
Configures the appearance of a thumbnail that is selected on the browser. A selected image is framed using the **Focus Color**.



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**





- Image Browser In is the RGB image that the browser uses.
- Focus Color is the color used to draw the frame around a selected thumbnail image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Browser Out is the resulting browser image.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

 $\bullet \quad LabVIEW \land examples \land Vision \land 2. \ Functions \land Display \land Browser \ Example. vi$ 

## **IMAQ Browser Focus VI**

Owning Palette: BrowserInstalled With: NI Vision Development Module

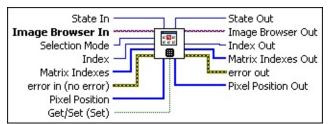
Gets or sets the status of images on the browser. You can choose either of the possible status values: selected or not selected. Selected images are framed using the Focus Color (refer to the <a href="IMAQ Browser Focus">IMAQ Browser Focus</a> Setup VI description).



**Note** This VI is not supported on ETS targets or VxWorks targets.

#### **Examples**





- **State In** is the image status value.
- Image Browser In is the RGB image that the browser uses.
- Selection Mode has two possible modes: multiple selection or single selection.
- Index is the position of the image to select or deselect.
- Matrix Indexes is the (Column, Line) position of the image to select or deselect.
- Pixel Position defines the pixel coordinates of the area of the image to select or deselect.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set (Set) is the VI action.
- **State Out** is the image status.
- Image Browser Out is the resulting browser image.
- Index Out is the index of the browser position of the selected thumbnail.
- Matrix Indexes Out is the (Column, Line) position of the selected thumbnail.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixel Position Out returns the x- and y-coordinates of the topmost-leftmost pixel and bottommost-rightmost pixel of the position of the selected thumbnail.

# **Examples**

Refer to the following for an example that uses this VI.

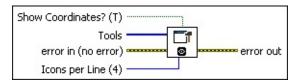
 $\bullet \quad LabVIEW \land examples \land Vision \land 2. \ Functions \land Display \land Browser \ Example. vi$ 

# IMAQ WindToolsSetup VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Configures the appearance and availability of the region tools found in the NI Vision Tools palette. With no input connections, the palette displays all nine region tools. The NI Vision Tools palette is a floating palette and is always visible.

#### **Details**



Show Coordinates? (T) specifies whether to show the active pixel coordinates. Coordinates are shown (TRUE) by default.



**Note** The IMAQ Tools window is not visible unless activated by calling the <a href="IMAQ WindToolsShow">IMAQ WindToolsShow</a> VI.

**Tools** is an array of tools specifying which tools are available. The following tools are available:

Selection Tool (0)	Select an ROI in the image.
Point Tool (1)	Select a pixel in the image.
Line Tool (2)	Draw a line in the image.
Rectangle Tool (3)	Draw a rectangle or square in the image.
Oval Tool (4)	Draw an oval or circle in the image.
Polygon Tool (5)	Draw a polygon in the image.
<b>Freehand Region Tool</b> (6)	Draw a freehand region in the image.
<b>Annulus Tool</b> (7)	Draw an annulus in the image.
Zoom Tool (8)	Zoom in or out in an image.
<b>Panning Tool</b> (9)	Shift the view of the image.
<b>Broken Line Tool</b> (10)	Draw a broken line in the image.
Freehand Line Tool (11)	Draw a freehand line in the image.
Rotated Rectangle	Draw a rotated rectangle in the image.

<b>Tool</b> (12)	
------------------	--

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- lcons per Line (4) determines the number of icons per line. The subsequent lines are set as a function of the number of remaining available icons.
  - Note The IMAQ Tools palette automatically displays cursor information if the input **Icons per Line** is set to 4 (or higher).
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

Use the <u>IMAQ WindLastEvent</u> VI to retrieve the draw events on the window and find the coordinates of a selected region.

You can alter the functionality of region tools by using a tool while pressing certain keyboard keys, as follows:

- <Shift> while drawing constrains x and y dimensions of an ROI to be the same. This forces rectangles into squares, ellipses into circles, and line segments into horizontal or vertical segments. You can also use it to constrain the rotation of the rotated rectangles and annuluses to multiples of 45 degrees.
- <Control> <Click> adds an ROI without erasing the previous ROI elements. The previous elements are erased if you do not use <Control> when starting a new element.
- Double-click while drawing produces the last point of a polygon or broken line.



**Note** You can erase an ROI in an image window by selecting it and pressing <Delete>.

Use the selection tool to select any currently existing ROI by clicking its border. You can then manipulate the ROI in the following ways:

- To resize a rectangle or ellipse, click in a grab handle and drag it to a new location.
- To reposition a vertex in a broken line, polygon, or line, click in a grab handle and move it to a new location.
- To reposition a rectangle or ellipse, click in the interior and drag it to a new location.
- To reposition a point, click on it and drag it to a new location.
- To reposition lines, broken lines, and polygons, click on any segment and drag it to a new location.
- To reposition freehand lines and freehand regions, click anywhere on the line and drag it to a new location.

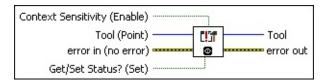
# **IMAQ WindToolsSelect VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Obtains or modifies the status of the region tools.



**Note** You can use this VI even if the WindTools palette is not displayed.



- **Context Sensitivity (Enable)** can enable or disable context sensitive tool selection when modifying the status of the region tools.
- **Tool (Point)** specifies the current tool. The following tools are available:

Selection Tool (0)	Select an ROI in the image.
Point Tool (1)	Select a pixel in the image.
Line Tool (2)	Draw a line in the image.
Rectangle Tool (3)	Draw a rectangle or square in the image.
Oval Tool (4)	Draw an oval or circle in the image.
Polygon Tool (5)	Draw a polygon in the image.
Freehand Region Tool (6)	Draw a freehand region in the image.
Annulus Tool (7)	Draw an annulus in the image.
Zoom Tool (8)	Zoom in or out in an image.
Panning Tool (9)	Shift the view of the image.
<b>Broken Line Tool</b> (10)	Draw a broken line in the image.
Freehand Line Tool (11)	Draw a freehand line in the image.
Rotated Rectangle Tool (12)	Draw a rotated rectangle in the image.

error in (no error) describes the error status before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

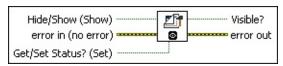
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the present status of the available region tools. The default is TRUE (Set).
- Tool returns the currently selected tool.
- **Context Sensitivity** indicates if **Context Sensitivity** was enabled.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## IMAQ WindToolsShow VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Shows or hides the WindTools palette. This VI functions in the same way as the <u>IMAQ WindShow</u> VI, which is used for displaying image windows.



- Hide/Show (Show) specifies whether the tools palette is visible. This VI considers this input only when **Get/Set Status?** is TRUE (Set).
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the present status of the available region tools. The default is TRUE (Set).
- Visible? returns the current visibility status of the tools palette. A

visible tools palette returns TRUE.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

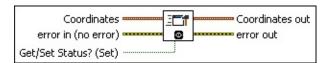
## IMAQ WindToolsMove VI

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Obtains or sets the position of the NI Vision Tools palette. This VI functions in the same way as the <u>IMAQ WindMove</u> VI, which is used for moving image windows.



**Note** You can use this VI even if the WindTools palette is not displayed.



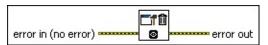
- **Coordinates** is a cluster that contains the screen coordinates that specify where to place the tools palette. This input is necessary only when **Get/Set Status?** is TRUE (Set).
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **Get/Set Status? (Set)** specifies whether to retrieve (Get) or modify (Set) the present status of the available region tools. The default is TRUE (Set).
- **Coordinates out** is a cluster returning the screen coordinates (the x and y positions) where the tools palette is located.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ WindToolsClose VI**

Owning Palette: External DisplayInstalled With: NI Vision Development Module

Closes the NI Vision Tools palette. This VI functions in the same way as the <u>IMAQ WindClose</u> VI, which is used for closing image windows.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Region of Interest**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Region of Interest VIs to focus processing and analysis on part of an image. The Region of Interest Conversion subpalette contains VIs convert machine vision point, line, rectangle, and annulus datatypes to or from a Region of Interest descriptor.

Palette Object	Description
IMAQ ROIToMask	Transforms a region of interest into a mask.
IMAQ MaskToROI	Transforms an image mask into a region of interest descriptor.
IMAQ Group ROIs	Builds a single ROI descriptor from an of array ROI descriptors.
IMAQ Ungroup ROIs	Separates an ROI descriptor describing many contours into an array of ROI descriptors. Each of the ROI descriptors returned contains a single contour.
IMAQ TransformROI	Rotates and translates an ROI in an image to transform the ROI from one coordinate system (Base Reference) to another (New Reference).

Subpalette	Description
Interest	Use the Region of Interest Conversion VIs to convert machine vision point, line, rectangle, and annulus datatypes to or from a Region of Interest descriptor. The Region of Interest Conversion palette is a subpalette of the Region of Interest palette.

## **IMAQ ROIToMask VI**

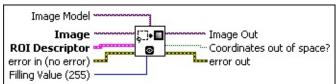
Owning Palette: Region of InterestInstalled With: NI Vision

**Development Module** 

Transforms a region of interest into a mask.

#### **Details** Examples





- Image Model serves as a template for the destination image where the mask is placed. This can be any image type that NI Vision supports. Image takes the characteristics of Image Model (size and location of ROI) when Image Model is connected. However, the connection of Image Model is optional. When no Image Model is connected, the size of the image mask generated is limited to the bounding rectangle of the ROI Descriptor, which reduces the amount of memory used. The offset of the image mask generated
- Image is a reference to the destination image in which the mask is created.
- **ROI Descriptor** is the descriptor that defines the region of interest.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - **Contours** are each of the individual shapes that define the ROI.
    - ID refers to whether the contour is the external or internal edge of an ROI. If the contour is external, all of the area inside it is considered part of the ROI. This VI draws the contours in the order that they are listed in the ROI Descriptor.
      - Q

**Tip** Internal contours do not become part of the mask. You can use internal contours to create non-mask regions inside external contours.

**Type** is the shape type of the contour.

- Coordinates are the coordinates that define the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Filling Value (255) is the pixel value of the mask. All pixels inside the region of interest take this value. The default value is 255.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Coordinates out of space? returns TRUE if any ROI data is found outside the space associated with the Image Model.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

#### **Details**

You can use this VI in two ways. The simplest technique is to connect the input **Image Model**. In this case you can use the source image, in which the image ROI was drawn, as a template for the final destination image by connecting it to **Image Model**. The output image (**Image Out**) automatically acquires the size of the image and location of the ROI as found in the original source image.

However, you do not have to connect an **Image Model**. In this case, the ROI requires an offset that is determined automatically from the upper-left corner of the bounding rectangle described by the ROI. These offset values are automatically set to the image mask.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Analysis\Histogram Example.vi

## IMAQ MaskToROI VI

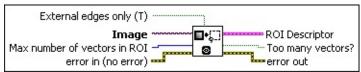
Owning Palette: Region of InterestInstalled With: NI Vision

**Development Module** 

Transforms an image mask into a region of interest descriptor.

### **Examples**





- **External edges only (T)** specifies whether only the external edges are transformed. The default is TRUE.
- **Image** is a reference to the image containing the image mask that is transformed into a region of interest.
- Max number of vectors in ROI is the limit of points that define the contour of a region of interest. This value is 2500 by default.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

- **ROI Descriptor** is a descriptor that defines the region of interest.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Too many vectors? returns TRUE if the number of vectors needed to represent the ROI exceeds the value specified by Max number of vectors in ROI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Grayscale Processing\MagicWand Example.vi

## **IMAQ Group ROIs VI**

Owning Palette: Region of InterestInstalled With: NI Vision

**Development Module** 

Builds a single ROI descriptor from an of array ROI descriptors.

### **Examples**



- **ROI Descriptors** is an array of ROI descriptors that are to be merged.
  - **Global Rectangle** contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

- warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **ROI Descriptor out** is an ROI descriptor containing all the contours in the array of ROIs.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

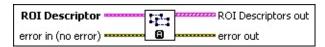
 $\bullet \quad LabVIEW \land amples \land Vision \land 3. \ Applications \land Meter\ Example. vi$ 

# **IMAQ Ungroup ROIs VI**

Owning Palette: Region of InterestInstalled With: NI Vision

**Development Module** 

Separates an ROI descriptor describing many contours into an array of ROI descriptors. Each of the ROI descriptors returned contains a single contour.



- ROI Descriptor is a descriptor that contains multiple contours.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

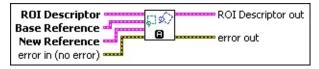
- warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- ROI Descriptors out is the returned array of ROI descriptors. Each ROI descriptor contains a single contour.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ TransformROI VI**

Owning Palette: Region of InterestInstalled With: NI Vision

**Development Module** 

Rotates and translates an ROI in an image to transform the ROI from one coordinate system (Base Reference) to another (New Reference).



- ROI Descriptor is a descriptor that defines the region of interest.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- **Base Reference** is the base coordinate system. This cluster contains the following elements:
  - Origin is a point cluster that specifies the origin of the base reference coordinate system.
  - Angle (deg) is the angle the base reference coordinate system makes with the image coordinate system.
  - Axis Reference is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.
- **New Reference** is the new coordinate system.

This cluster contains the following elements:

- Origin is a point cluster that specifies the origin of the base reference coordinate system.
- Angle (deg) is the angle the base reference coordinate system makes with the image coordinate system.

- Axis Reference is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **ROI Descriptor out** is a new ROI descriptor for the transformed ROI.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

# **Region of Interest Conversion**

Owning Palette: Region of InterestInstalled With: NI Vision

**Development Module** 

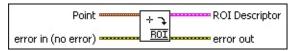
Use the Region of Interest Conversion VIs to convert machine vision point, line, rectangle, and annulus datatypes to or from a Region of Interest descriptor. The Region of Interest Conversion palette is a subpalette of the Region of Interest palette.

Palette Object	Description
IMAQ Convert Point to ROI	Converts a point element to an ROI Descriptor.
IMAQ Convert Line to ROI	Converts a line to an ROI Descriptor.
IMAQ Convert Rectangle to ROI	Converts a rectangle or rotated rectangle to an ROI Descriptor. The contour of the ROI Descriptor returned is always of the type rotated rectangle.
IMAQ Convert Rectangle to ROI (Polygon)	Converts a rectangle or rotated rectangle to an ROI Descriptor. The contour of the ROI Descriptor returned is always of the type rotated rectangle polygon.
IMAQ Convert Annulus to ROI	Converts an annulus to an ROI Descriptor.
IMAQ Convert ROI to Point	Converts an ROI Descriptor to a point element.
IMAQ Convert ROI to Line	Converts an ROI Descriptor to a line.
IMAQ Convert ROI to Rectangle	Converts an ROI Descriptor to a rectangle or rotated rectangle. The contour of the ROI Descriptor is always of the type rotated rectangle.
IMAQ Convert ROI to Annulus	Converts an ROI Descriptor to an annulus.

# **IMAQ Convert Point to ROI VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts a point element to an ROI Descriptor.



- **Point** is a point-coordinate cluster.
  - **X** is the x-coordinate of the point.
  - Y is the y-coordinate of the point.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- ROI Descriptor is a descriptor that defines the point.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an

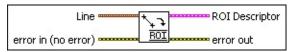
ROI.

- **ID** refers to whether the contour is the external or internal edge of an ROI.
- **Type** is the shape type of the contour.
- **Coordinates** indicates the relative position of the contour.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Convert Line to ROI VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts a line to an ROI Descriptor.



- Line specifies the coordinates of the line.
  - Start X specifies the x-coordinate of the beginning point of the line.
  - Start Y specifies the y-coordinate of the beginning point of the line.
  - **End X** specifies the x-coordinate of the end point of the line.
  - **End Y** specifies the y-coordinate of the end point of the line.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **ROI Descriptor** is a descriptor that defines the line.
- Global Rectangle contains the coordinates of the bounding rectangle.
- **Contours** are each of the individual shapes that define an ROI.
  - **ID** refers to whether the contour is the external or internal edge of an ROI.
  - **Type** is the shape type of the contour.
  - **Coordinates** indicates the relative position of the contour.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Convert Rectangle to ROI VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts a rectangle or rotated rectangle to an ROI Descriptor. The contour of the ROI Descriptor returned is always of the type rotated rectangle.



- Rectangle specifies a rectangular region. The Left, Top, Right, and Bottom coordinates describe a non-rotated rectangle. The Rotation parameter specifies the rotation angle of the rectangle around its center.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **ROI Descriptor** is a descriptor that defines the rectangle.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Rectangle Points returns the actual coordinates of the summits of the rectangle.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a

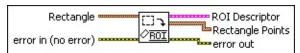
warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ Convert Rectangle to ROI (Polygon) VI

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts a rectangle or rotated rectangle to an ROI Descriptor. The contour of the ROI Descriptor returned is always of the type rotated rectangle polygon.



- Rectangle specifies the coordinates of a rectangular region of interest.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

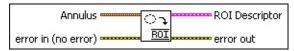
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **ROI Descriptor** is a descriptor that defines the rectangle.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Rectangle Points returns the actual coordinates of the summits of the rectangle.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Convert Annulus to ROI VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts an annulus to an ROI Descriptor.



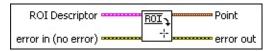
- Annulus specifies the coordinates of an annular region of interest in an image.
  - **Center X** specifies the x-coordinate of the center of the circle or annulus.
  - **Center Y** specifies the y-coordinate of the center of the circle or annulus.
  - Inner Radius is the radius (in pixels) of the inner circle defining one edge of the circular strip.
  - Outer Radius is the radius (in pixels) of the outer circle defining the second edge of the circular strip.
  - Start Angle is the first angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
  - **End Angle** is the second angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- ROI Descriptor is a descriptor that defines an annulus.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Convert ROI to Point VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts an ROI Descriptor to a point element.



- ROI Descriptor is the descriptor that defines the point.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

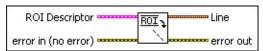
**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Point is a point-coordinate cluster.
  - X is the x-coordinate of the point.
  - **Y** is the y-coordinate of the point.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Convert ROI to Line VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts an ROI Descriptor to a line.



- ROI Descriptor is a descriptor that defines the line.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

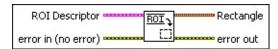
**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **Line** specifies the coordinates of the line.
  - Start X specifies the x-coordinate of the beginning point of the line.
  - Start Y specifies the y-coordinate of the beginning point of the line.
  - **End X** specifies the x-coordinate of the end point of the line.
  - **End Y** specifies the y-coordinate of the end point of the line.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Convert ROI to Rectangle VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts an ROI Descriptor to a rectangle or rotated rectangle. The contour of the ROI Descriptor is always of the type rotated rectangle.



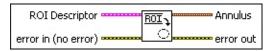
- **ROI Descriptor** is the descriptor that defines the rectangle.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Rectangle specifies a rectangular region. The Left, Top, Right, and Bottom coordinates describe a non-rotated rectangle. The Rotation parameter specifies the rotation angle of the rectangle around its center.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Convert ROI to Annulus VI**

Owning Palette: Region of Interest ConversionInstalled With: NI Vision Development Module

Converts an ROI Descriptor to an annulus.



- ROI Descriptor is the descriptor that defines an annulus.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Annulus specifies the coordinates of an annular region of interest in an image.
  - Center X specifies the x-coordinate of the center of the circle or annulus.
  - Center Y specifies the y-coordinate of the center of the circle or annulus.
  - Inner Radius is the radius (in pixels) of the inner circle defining one edge of the circular strip.
  - Outer Radius is the radius (in pixels) of the outer circle defining the second edge of the circular strip.
  - Start Angle is the first angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
  - **End Angle** is the second angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Image Manipulation**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Image Manipulation VIs to modify an image. Use these VIs to resample an image; extract parts of an image; and rotate, shift, and unwrap images.

Palette Object	Description
IMAQ Resample	Resamples an image to a user-defined size. You can use this VI to display a reduced or enlarged image.
IMAQ Expand	Expands (duplicates) an image or part of an image by adjusting the horizontal and vertical resolution.
IMAQ Extract	Extracts (reduces) an image or part of an image with adjustment of the horizontal and vertical resolution.
IMAQ Interlace	Extracts odd and even fields from an interlaced image or builds an image using two field images.
IMAQ Symmetry	Transforms an image through its symmetry.
<b>IMAQ Rotate</b>	Rotates an image.
IMAQ Shift	Translates an image based on a horizontal and vertical offset.
IMAQ Unwrap	Unwraps a circular strip from an image into a rectangular strip.
IMAQ ClipboardToImage	Copies the clipboard data into an image.
IMAQ ImageToClipboard	Copies the image to the operating system clipboard.
IMAQ 3DView	Displays an image using an isometric view. Each pixel from the image source is represented as a column of pixels in the 3D view. The pixel value corresponds to the altitude.

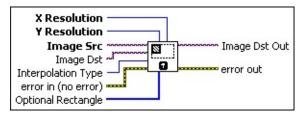
## **IMAQ Resample VI**

Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Resamples an image to a user-defined size. You can use this VI to display a reduced or enlarged image.





- X Resolution specifies the horizontal resolution of the image.
- Y Resolution specifies the vertical resolution of the image.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Interpolation Type specifies the interpolation method used to resample the image. The default is **Zero Order**. The following options are valid:

Zero Order (0)	Rounds to the nearest integral pixel location
Bi-Linear (1)	Uses linear interpolation in both the x- and y-directions to compute the pixel location
Quadratic (2)	Uses the quadratic formula to compute the pixel location
Cubic Spline (3)	Uses cubic splines to compute the pixel location

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

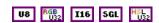
## **IMAQ Expand VI**

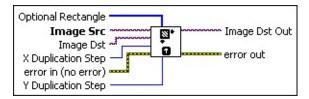
Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Expands (duplicates) an image or part of an image by adjusting the horizontal and vertical resolution.

#### **Details**





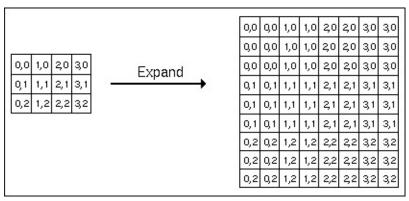
- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- X Duplication Step specifies the number of pixel duplications per column. The column is recopied if the default value (1) is used.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Y Duplication Step specifies the number of pixel duplications per line. The row is recopied if the default value (1) is used.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

For example, if a 256  $\times$  256 image is connected and the **X Duplication Step** and **Y Duplication Step** are both equal to 2, the resulting image has a resolution of 512  $\times$  512. Each pixel in the original image now is represented by four pixels (2  $\times$  2) in the new image.

The following figure illustrates an expansion of an image where **X Duplication Step** equals 2 and **Y Duplication Step** equals 3.



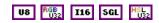
#### **IMAQ Extract VI**

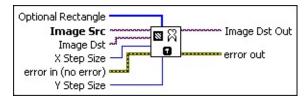
Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Extracts (reduces) an image or part of an image with adjustment of the horizontal and vertical resolution.

#### **Details Examples**





- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- X Step Size is the vertical sampling step, which defines the columns to be extracted (the horizontal reduction ratio). For example, with an X Step Size equal to 3, one out of every three columns is extracted from the Image Src into the Image Dst. Each column is extracted if the default value (1) is used.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

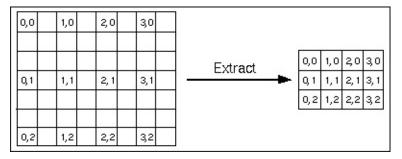
- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Y Step Size is the horizontal sampling step, which defines the lines to be extracted (the vertical reduction ratio). Each row is extracted if the default value (1) is used.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

For example, if a  $512 \times 512$  image is connected and the **X Step Size** and **Y Step Size** are both equal to 2, the resulting image has a resolution of  $256 \times 256$ . The resulting image contains the lines from the **Image Src** 0, 2, 4, ..., 510 and the columns 0, 2, 4, ..., 510 from the **Image Src**.

**Image Src** and **Image Dst** must be of the same image type.

The following figure illustrates an extraction of an image where **X Step Size** equals 2 and **Y Step Size** equals 3. Example of Image Extraction



## **Examples**

Refer to the following for an example that uses this VI.

- LabVIEW\examples\Vision\2. Functions\Image Management\Extract Example.vi
- LabVIEW\examples\Vision\3. Applications\Mechanical Assembly Example.vi

#### **IMAQ Interlace VI**

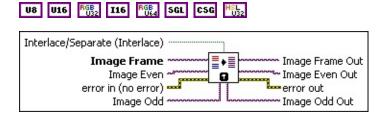
Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Extracts odd and even fields from an interlaced image or builds an image using two field images.



**Note** When two fields are interlaced, the first line in the resulting frame comes from the even field, and the second comes from the odd field.



- Interlace/Separate (Interlace) specifies if the image is built using two field images or one interlaced image. The default is the interlace mode, which specifies that an interlaced image is built using two field images (Image even and Image odd). In the separate mode, the odd and even fields from an interlaced image (Image frame) are extracted.
- Image Frame is the reference to the image from which odd and even fields have to be extracted.
- Image Even is the reference to the image that forms the even lines of the interlaced image. To extract the even field, you must connect an image to this control for the even lines to be written to memory.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

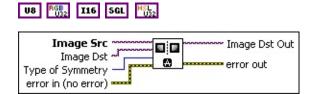
- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Odd is the reference to the image that forms the odd lines of the interlaced image. To extract the odd field, you must connect an image to this control for the odd lines to be written to memory.
- Image Frame Out contains the interlaced image.
- Image Even Out contains the even lines of the input image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Odd Out contains the odd lines of the input image.

## **IMAQ Symmetry VI**

Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Transforms an image through its symmetry.



- **Image Src** is a reference to the source image.
- Image Dst is a reference to the destination image.
- Type of Symmetry specifies the symmetry used. The default is Horizontal. You can choose from the following values:

Horizontal (0)	Based on the horizontal axis of the image
<b>Vertical</b> (1)	Based on the vertical axis of the image
Central (2)	Based on the center of the image
1st Diagonal (3)	Based on the first diagonal of the image (the image must be square)
2nd Diagonal (4)	Based on the second diagonal of the image (the image must be square)

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

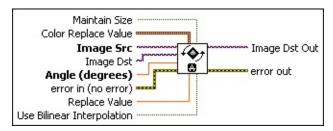
## **IMAQ Rotate VI**

Owning Palette: <a href="mage-Manipulation">Image Manipulation</a> Installed With: NI Vision

**Development Module** 

Rotates an image.





- Maintain Size specifies whether you want the resulting image to be the same size as the source image. Set this value to TRUE (default) if you want the resulting image to be the same size as the source image. Set this value to FALSE if you want the resulting image size to change to accommodate all the image data from the source image.
- **Color Replace Value** is a cluster containing the alpha, red, green, and blue channel values used for filling a color image. The default is 0 for all channels, resulting in black.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Angle (degrees) defines the angle to rotate. The default is 0 degrees.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Replace Value is the value that the VI uses to fill pixels in the rotated image that do not have a corresponding pixel in the Image Src. The default is 0.
- **Use Bilinear Interpolation** specifies the type of interpolation to use to compute the values of the rotated pixels. If set to TRUE (default), IMAQ Rotate uses bilinear interpolation. If set to FALSE, IMAQ Rotate uses zero-order interpolation.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

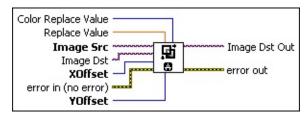
## **IMAQ Shift VI**

Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Translates an image based on a horizontal and vertical offset.





- Replace Value is the value that the VI uses to fill pixels in the shifted image that do not have a corresponding pixel in the Image Src. The default value is 0.
- **Image Src** is a reference to the source image.
- Image Dst is a reference to the destination image.
- **XOffset** is the horizontal offset added to the image. The default is 0.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **YOffset** is the vertical offset added to an image. The default is 0.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Unwrap VI**

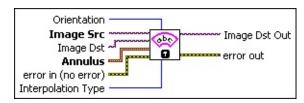
Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Unwraps a circular strip from an image into a rectangular strip.

#### **Examples**





Orientation specifies the orientation of the image data within the circular strip.

(Default) Specifies that the base of the image data is located along the inside edge of the circular strip.
Specifies that the base of the image data is located along the outside edge of the circular strip.

- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Annulus is a cluster of parameters that define a circular strip.
  - Center X specifies the x-coordinate of the center of the circle or annulus.
  - Center Y specifies the y-coordinate of the center of the circle or annulus.
  - Inner Radius is the radius (in pixels) of the inner circle defining one edge of the circular strip.
  - Outer Radius is the radius (in pixels) of the outer circle defining the second edge of the circular strip.
  - Start Angle is the first angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
  - **End Angle** is the second angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate

system centered on the center of the circle or annulus.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Interpolation Type specifies the interpolation method used in the unwrapping process. The following options are valid:

Zero Order (0)	Rounds to the nearest integral pixel location
Bi-Linear (1)	Uses linear interpolation in both the x- and y-directions to compute the pixel location
Quadratic (2)	Uses the quadratic formula to compute the pixel location
Cubic Spline (3)	Uses cubic splines to compute the pixel location

Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
Otherwise, Image Dst Out refers to the image referenced by Image Src.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Image Management\Unwrap Barcode Example.vi

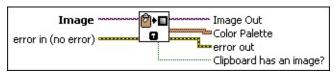
## **IMAQ ClipboardToImage VI**

Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Copies the clipboard data into an image.





- Image is a reference to the source image.
- Color Palette is the color palette that is stored on the clipboard.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - Blue is the value of the blue color plane.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Color Palette is the color palette that is stored on the clipboard. A grayscale palette is returned if no color palette is found on the clipboard.
  - **Red** is the output red value.
  - **Green** is the output green value.
  - **Blue** is the output blue value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Clipboard has an image? returns a TRUE value if the clipboard contains an image.

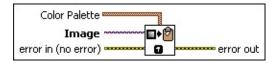
## IMAQ ImageToClipboard VI

Owning Palette: Image ManipulationInstalled With: NI Vision

**Development Module** 

Copies the image to the operating system clipboard.





Color Palette is used to apply a color palette to the image. Color Palette is an array of clusters constructed by the user or supplied by the <a href="IMAQ GetPalette">IMAQ GetPalette</a> VI. This palette can be composed of up to 256 elements for each of the three color planes (red, green, and blue). If the array has less than 256 elements, the VI maps all pixel values past the last element in the array to the associated grayscale value.

A specific color is the result of applying a value between 0 and 255 for each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black, and 255 specifies white). The default is a grayscale palette. The color palette can be used only for 8-bit images.



**Tip** For best results, set your video adapter to high color or true color.

- Red is the value of the red color plane.
- Green is the value of the green color plane.
- **Blue** is the value of the blue color plane.
- **Image** is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ 3DView VI**

Owning Palette: Image ManipulationInstalled With: NI Vision

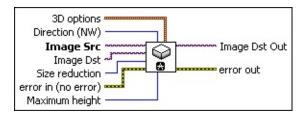
**Development Module** 

Displays an image using an isometric view. Each pixel from the image source is represented as a column of pixels in the 3D view. The pixel value corresponds to the altitude.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.





- **3D options** is a cluster containing the following elements:
  - **alpha** defines the angle between the horizontal and the baseline. The value can be between 0° and 45°. The default value is 30°.
  - beta defines the angle between the horizontal and the second baseline. The value can be between 0° and 45°. The default value is 30°.
  - **border** defines the border size in the 3D view. The default value is 20.
  - background defines the background color for the 3D view.
  - plane specifies the view to display if the image is complex. The following four possible planes can be visualized from a complex image. For complex images, the default is the magnitude.
    - real (0)
    - imaginary (1)
    - magnitude (2)
    - phase (3)

Direction (NW) defines the viewing orientation shown for the 3D view. The following viewing angles are possible. The default is North West.

North West (0)	Sets the viewing orientation to northwest
South West (1)	Sets the viewing orientation to southwest.
South East (2)	Sets the viewing orientation to southeast.
North East (3)	Sets the viewing orientation to northeast.

- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Size reduction is a factor applied to the source image to calculate the final dimensions of the 3D view image. This factor is a divisor that is applied to the source image when determining the final height and width of the 3D view image. A factor of 1 uses all of the pixels of the source image when determining the 3D view image. A factor of 2 uses every other line and every other column of the pixels of the source image to determine the 3D view image. The default is 2.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Maximum height defines the maximum height of a pixel from the image source that is drawn in 3D. This value is mapped from a maximum of 255 (from the source image) in relation to the baseline in the 3D view. A value of 255, therefore, gives a one-to-one correspondence between the intensity value in the source image and the display in 3D view. The default value of 64 results in a four-fold reduction between the original intensity value of the pixel in the source image and the final displayed 3D image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Pixel Manipulation**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Pixel Manipulation VIs to modify specific areas of an image. Use these VIs to read and set pixel values in an image, read and set values along a row or column in an image, and fill the pixels in an image with a particular value.

Palette Object	Description
IMAQ GetPixelValue	Reads a pixel value from an image.
IMAQ GetRowCol	Extracts a range of pixel values, either a row or column, from an image.
IMAQ GetPixelLine	Extracts the intensity values of a line of pixels.
IMAQ ImageToArray	Extracts (copies) the pixels from an image, or part of an image, into a LabVIEW 2D array. This array is encoded in 8 bits, 16 bits, or floating point, as determined by the type of input image.
IMAQ SetPixelValue	Changes a pixel value in an image.
IMAQ SetRowCol	Changes the intensity values in either a row or a column of pixels in an image.
IMAQ SetPixelLine	Changes the intensity values in a line of pixels of an image.
IMAQ ArrayTolmage	Creates an image from a 2D array.
IMAQ FillImage	Fills an image and its border with a specified value.
IMAQ Draw	Draws geometric objects in an image.
IMAQ Draw Text	Inserts text in an image.

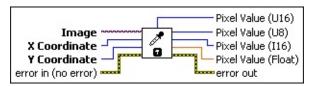
## **IMAQ GetPixelValue VI**

Owning Palette: Pixel Manipulation Installed With: NI Vision

**Development Module** 

Reads a pixel value from an image.





- **Image** is a reference to the source image.
- X Coordinate is the horizontal coordinate of the pixel.
- Y Coordinate is the vertical coordinate of the pixel.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Pixel Value (U16)** returns the specified pixel value. This output is used only for an 8-bit or unsigned 16-bit image.

- **Pixel Value (U8)** returns the specified pixel value. This output is used only for an 8-bit image.
- Pixel Value (I16) returns the specified pixel value. This output is used only for an 8-bit or signed 16-bit image.
- Pixel Value (Float) returns the specified pixel value. The float format can accept values from all supported image types (8-bit, 16-bit, or 32-bit floating point).
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ GetRowCol VI**

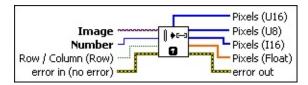
Owning Palette: Pixel Manipulation Installed With: NI Vision

**Development Module** 

Extracts a range of pixel values, either a row or column, from an image.

#### **Examples**

U8 U16 I16 SGL



- Image is a reference to the source image.
- Number is the row or column number to be extracted from the image.
- Row / Column (Row) defines Number as a row number when FALSE and as a column number when TRUE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

- Pixels (U16) returns the intensity values for the specified row or column of pixels. Use this output only when **Image** is an unsigned 16-bit image.
- Pixels (U8) returns the intensity values for the specified row or column of pixels. Use this output only when **Image** is an 8-bit image.
- Pixels (I16) returns the intensity values for the specified row or column of pixels. Use this output only when **Image** is a signed 16-bit image.
- Pixels (Float) returns the intensity values for the specified row or column of pixels. Use this output only when Image is a 32-bit floating-point image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Gauging Example.llb

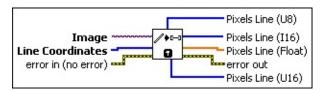
## IMAQ GetPixelLine VI

Owning Palette: Pixel Manipulation Installed With: NI Vision

**Development Module** 

Extracts the intensity values of a line of pixels.

U8 U16 I16 SGL



- Image is a reference to the source image.
- **Line Coordinates** is an array specifying the pixel coordinates that form the end points of the line.

**Note** A line with the coordinates (0, 0, 0, 255) is formed from 256 pixels. Any pixels designated by **Line Coordinates** that lie outside the actual image are set to 0 in **Pixels Line**.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

- Pixels Line (U8) returns the intensity values for the specified line of pixels. Use this output only when **Image** is an 8-bit image.
- Pixels Line (I16) returns the intensity values for the specified line of pixels. Use this output only when **Image** is a signed 16-bit image.
- Pixels Line (Float) returns the intensity values for the specified line of pixels. Use this output only when **Image** is a 32-bit floating-point image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixels Line (U16) returns the intensity values for the specified line of pixels. Use this output only when **Image** is an unsigned 16-bit image.

# IMAQ ImageToArray VI

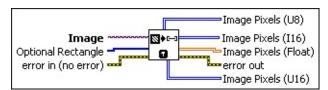
Owning Palette: Pixel Manipulation Installed With: NI Vision

**Development Module** 

Extracts (copies) the pixels from an image, or part of an image, into a LabVIEW 2D array. This array is encoded in 8 bits, 16 bits, or floating point, as determined by the type of input image.

#### **Examples**





- **Image** is a reference to the source image.
- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to extract. The operation applies to the entire image if the input is empty or not connected.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Pixels (U8) returns the extracted pixel values into a 2D array. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes Image Pixels to be the same size as Image or the size of Optional Rectangle. Use this output only when Image is an unsigned 8-bit image.
- Image Pixels (I16) returns the extracted pixel values into a 2D array. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes Image Pixels to be the same size as Image or the size of Optional Rectangle. Use this output only when Image is a signed 16-bit image.
- Image Pixels (Float) returns the extracted pixel values into a 2D array. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes Image Pixels to be the same size as Image or the size of Optional Rectangle. Use this output only when Image is a 32-bit floating-point image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- [U16] Image Pixels (U16) returns the extracted pixel values into a 2D

array. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes **Image Pixels** to be the same size as **Image** or the size of **Optional Rectangle**. Use this output only when **Image** is an unsigned 16-bit image.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Image Management\ImageToArray Example.vi

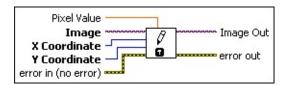
# **IMAQ SetPixelValue VI**

Owning Palette: Pixel ManipulationInstalled With: NI Vision

**Development Module** 

Changes a pixel value in an image.





- **Pixel Value** is the replacement pixel value.
- Image is a reference to the source image.
- X Coordinate is the horizontal coordinate of the pixel.
- Y Coordinate is the vertical coordinate of the pixel.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to the destination image. If Image Dst is

connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ SetRowCol VI

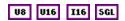
Owning Palette: Pixel ManipulationInstalled With: NI Vision

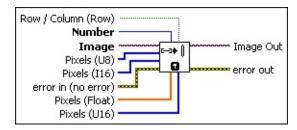
**Development Module** 

Changes the intensity values in either a row or a column of pixels in an image.



**Note** The data type of the **Pixels** input you select must correspond to the data type of the input image.





- Row / Column (Row) defines Number as a row number when FALSE and as a column number when TRUE.
- Number is the row or column number to replace in the image.
- Image is a reference to the source image.
- Pixels (U8) is an array of unsigned 8-bit integers containing the new values for the specified row or column of the image. This input is required if **Image** is an unsigned 8-bit image.
- Pixels (I16) is an array of signed 16-bit integers containing the new values for the specified row or column of the image. This input is required if Image is a signed 16-bit image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Pixels (Float)** is an array of 32-bit floating-point values containing the new values for the specified row or column of the image. This input is required if **Image** is a 32-bit floating-point image.
- Pixels (U16) is an array of unsigned 16-bit integers containing the new values for the specified row or column of the image. This input is required if **Image** is an unsigned 16-bit image.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ SetPixelLine VI

Owning Palette: Pixel ManipulationInstalled With: NI Vision

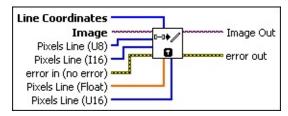
**Development Module** 

Changes the intensity values in a line of pixels of an image.



**Note** Each **Pixels Line** input is specific for a particular type of image.





Line Coordinates is a four-element array specifying the pixel coordinates that form the end points of the line to modify. The first two elements (left, top) in the array correspond to the coordinates for the first endpoint of the line. The last two elements (right, bottom) correspond to the second endpoint of the line.

If the size of the **Pixels Line** is greater than the number of pixels on the straight line between the **Line Coordinates**, the remaining pixel values in the **Pixels Line** will not be used. If the size of the **Pixels Line** is smaller than the number of pixels on the straight line between the **Line Coordinates**, The pixel values are changed starting from the (left, top) pixel and continuing until all of the elements in the **Pixel Line** are used. Any remaining pixels on the straight line between the **Line Coordinates** will not be modified.

- Image is a reference to the source image.
- Pixels Line (U8) is an array of unsigned 8-bit integers containing the new values for the pixel line. This input is required if **Image** is an unsigned 8-bit image.
- Pixels Line (I16) is an array of signed 16-bit integers containing the new values for the pixel line. This input is required if **Image** is a signed 16-bit image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixels Line (Float) is an array of 32-bit floating point values containing the new values for the pixel line. This input is required if **Image** is a 32-bit floating-point image.
- Pixels Line (U16) is an array of unsigned 16-bit integers containing the new values for the pixel line. This input is required if Image is an unsigned 16-bit image.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ ArrayTolmage VI**

Owning Palette: Pixel ManipulationInstalled With: NI Vision

**Development Module** 

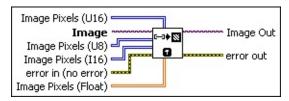
Creates an image from a 2D array.



**Note** Each **Image Pixels** input is specific for a particular type of image.

#### **Details**





- Image Pixels (U16) is a 2D array of unsigned 16-bit integers containing the pixel values that comprise the image. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes Image to be the same size as Image Pixels. Use this input only when Image is an unsigned 16-bit image.
- **Image** is a reference to the source image.
- Image Pixels (U8) is a 2D array of unsigned 8-bit integers containing the pixel values that comprise the image. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes Image to be the same size as Image Pixels. Use this input only when Image is an unsigned 8-bit image.
- Image Pixels (I16) is a 2D array of signed 16-bit integers containing the pixel values that comprise the image. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes Image to be the same size as Image Pixels. Use this input only when Image is a signed 16-bit image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Pixels (Float) is a 2D array of 32-bit floating-point values containing the pixel values that comprise the image. The first index corresponds to the vertical axis and the second index corresponds to the horizontal axis. The VI resizes Image to be the same size as Image Pixels. Use this input only when Image is a 32-bit floating-point image.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is

a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

Refer to the additional VIs in the <u>Frequency Domain</u> VIs group for performing array-to-image transformations with complex images. Refer to the <u>Color Utilities</u> VIs for performing array-to-image transformations with color images.

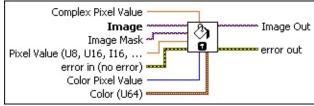
# **IMAQ FillImage VI**

Owning Palette: Pixel ManipulationInstalled With: NI Vision

**Development Module** 

Fills an image and its border with a specified value.





- **Complex Pixel Value** specifies the value used for filling a complex image.
- Image is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region in the image to modify. Only those pixels in the original image that correspond to an equivalent non-zero pixel in the mask image are processed. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Pixel Value (U8, U16, I16, Float) specifies the value with which the image is to be filled. This value is used for 8-bit, 16-bit, and 32-bit floating-point images.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Color Pixel Value** specifies the value used for filling a 32-bit RGB or 32-bit HSL image.
- Color (U64) specifies the value used for filling a 64-bit RGB.
  - Plane 1 (Red) is the value of the red plane.
  - Plane 2 (Green) is the value of the green plane.
  - Plane 3 (Blue) is the value of the blue plane.
  - Plane 4 (alpha) is the value of the alpha plane.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Draw VI**

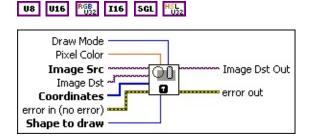
Owning Palette: Pixel Manipulation Installed With: NI Vision

**Development Module** 

Draws geometric objects in an image.



**Note** This VI is not supported on ETS targets or VxWorks targets.



**Draw Mode** defines how to draw the object. The following values are valid:

Frame (0)	(Default) Specifies to use <b>Pixel Color</b> when tracing the contour
Paint (1)	Specifies to use <b>Pixel Color</b> when tracing the contour and the interior of the shape
Invert Frame (2)	Specifies to use the inverse of the pixel values when drawing the contour
Invert Paint (3)	Specifies to use the inverse of the pixel values when drawing the contour and the interior of the shape

- Pixel Color is the pixel value used for tracing the design. This value is not used when in the mode **Invert Frame** or **Invert Paint**. The default is 0.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Coordinates is an array of four elements. A line is specified by the two points forming it. Rectangles and ovals are specified by their bounding rectangle, with the format (Left/Top/Right/Bottom). In these cases, the tracing of a rectangle or oval stops at the column (Right 1) and at the row (Bottom 1). The values by default are (0, 0, SizeX, SizeY) where (SizeX, SizeY) is the resolution of the image. The default is used if the input is 0 or is not connected.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Shape to draw is the form to draw. The following shapes are available:

` ,	(Default) Defined by the two points specified in the array <b>Coordinates</b>
• ( )	Defined by the bounding rectangle specified in the array <b>Coordinates</b>
` ,	Defined by the bounding rectangle specified in the array <b>Coordinates</b>

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

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

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Draw Text VI**

Owning Palette: Pixel ManipulationInstalled With: NI Vision

**Development Module** 

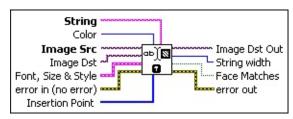
Inserts text in an image.



**Note** This VI is not supported on ETS targets or VxWorks targets.

### **Examples**





- String is the text to write in an image. The string can be composed of multiple lines separated by a hard return. This control is empty by default.
- **Color** is the mode for writing the text. The default is 0, which specifies white. The following values are possible:

White (0)	(Default) Writes white text on the image background
Black (1)	Writes black text on the image background
Inverted (2)	Inverts the text color against the image background color
Black on White (3)	Writes black text on a white background on the image background
White on Black (4)	Writes white text on a black background on the image background

- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Font, Size & Style is a cluster that enables the user to choose the font, size, style, and alignment of the text. This cluster contains the following elements:
  - **Desired Font (Application)** specifies the character type of

the text. The following values are possible:

User- specified Font (0)	Specify the attributes of a font to use. If no font is specified, the default font is used.
Application Font (1)	(Default) Uses the LabVIEW application font, which is the font used on the Controls and Functions palettes and on text in new objects on the front panel or block diagram.
System Font (2)	Uses the LabVIEW system font, which is the font used in menus.
Dialog Font (3)	Uses the LabVIEW dialog font, which is the font used in dialog boxes.

[User-specified Font] is a cluster containing the specific font characteristics for the text to draw. The specification is ignored unless **Desired Font** is set to **User-specified Font**.



Note The Outline?, Shadow?, and Rotation Angle characteristics are not currently supported on RT targets.

- Font Name is the font to use for the string.
- **Size** is the size of the font in points.
- **Strikeout?** causes the text to appear in strikeout if TRUE.
- **Italic?** causes the text to appear in italic if TRUE.
- **Underline?** causes the text to appear underlined if TRUE.
- **Outline?** causes the text to appear in outline if TRUE.
- **Shadow?** causes the text to appear in shadow if TRUE.
- Bold? causes the text to appear in bold if TRUE
- **Rotation Angle** is the orientation of the text in the image. NI Vision supports rotated text for TrueType fonts only.
- Horizontal Alignment specifies the horizontal alignment of

the text. The following values are possible:

Left (0)	(Default) Left-aligns the text.
Center (1)	Center-aligns the text.
Right (2)	Right-aligns the text.

Vertical Alignment specifies the vertical alignment of the text. The following values are possible:

Bottom (0)	(Default) Aligns the text along the bottom.
<b>Top</b> (1)	Aligns the text along the top.
Baseline (2)	Aligns the text along the baseline.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Insertion Point is an array (x and y) specifying the location in which the text is inserted. The text position depends on the alignment mode chosen. The default is (0, 0).
- Image Dst Out is a reference to the destination image. If Image

**Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

- **String width** returns the string length of the text.
- Face Matches returns TRUE if the user-specified font name was found. Returns FALSE if the font name was not found. Note that if the requested font name is not found, the operating system chooses the best match.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Image Management\DrawText Example.vi

# **Overlay**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Overlay VIs to overlay figures, text, and bitmaps onto an image without destroying the image data. Use these VIs to overlay the results of your inspection application onto the images you inspected.

Palette Object	Description
IMAQ Overlay Points	Overlays a point or an array of points on an image.
IMAQ Overlay Line	Overlays a line on an image.
IMAQ Overlay Multiple Lines	Overlays multiple lines, as either a polyline or a polygon, on an image.
IMAQ Overlay Rectangle	Overlays a rectangle on an image.
IMAQ Overlay Oval	Overlays an oval on an image.
IMAQ Overlay Arc	Overlays an arc on an image.
IMAQ Overlay Bitmap	Overlays a bitmap on an image.
IMAQ Overlay Text	Overlays text on an image.
IMAQ	Clears the image overlay.

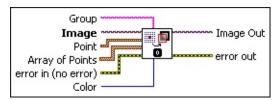
<u>Clear</u> <u>Overlay</u>	
IMAQ Copy Overlay	Copies an image overlay from the source image to the destination image. If the destination image already has an overlay, this VI adds the source overlay to the existing overlay.
IMAQ Overlay ROI	Overlays a region of interest (ROI) on an image.
IMAQ Merge Overlay	Makes a nondestructive overlay part of the image content. This process creates a destructive overlay. The VI then removes the nondestructive overlay. The resulting image is an RGB image.
IMAQ Read Image And Vision Info	Reads an image file, including any extra vision information saved with the image. This includes overlay information, pattern matching template information, and calibration information, as written by the <a href="MAQ Write Image And Vision Info">IMAQ Write Image And Vision Info</a> VI.
IMAQ Write Image And Vision Info	Writes an image, along with extra vision information associated with the image, to a PNG file. This extra vision information includes overlay information, pattern matching template information, and calibration information.
IMAQ Get Overlay Properties	Get the transformation behavior properties for a group of overlays.
IMAQ Set Overlay Properties	overlays.

# **IMAQ Overlay Points VI**

Owning Palette: OverlayInstalled With: NI Vision Development Module Overlays a point or an array of points on an image.

#### **Details**





- Group specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image is a reference to the source image.
- **Point** is a cluster that specifies the coordinates of the point to overlay on the image. This input is ignored if the **Array of Points** input is connected.
  - **X** is the x-coordinate of the point.
  - Y is the y-coordinate of the point.
- Array of Points is an array of point-coordinate clusters to overlay on the image.
  - $\mathbf{X}$  is the x-coordinate of a point in the array.
  - $\mathbf{Y}$  is the y-coordinate of a point in the array.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

**status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Color** is the color of the overlay.
  - Note The alpha color channel is not supported. Setting the Color to transparent has the same effect as selecting black.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

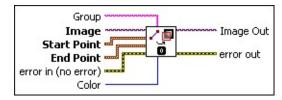
Overlays are nondestructive, which means that they do not overwrite the underlying pixel values. You can save overlay information with an image by using the <a href="MAQ Read Image And Vision Info">IMAQ Read Image And Vision Info</a> VI. If you want to replace the underlying pixel values with the overlay values, use the <a href="IMAQ Merge Overlay">IMAQ Merge Overlay</a> VI.

# **IMAQ Overlay Line VI**

Owning Palette: OverlayInstalled With: NI Vision Development Module Overlays a line on an image.

#### **Examples**





- **Group** specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image is a reference to the source image.
- Start Point is a point-coordinate cluster that defines the starting point of the line.
  - **X** is the x-coordinate of the starting point.
  - Y is the y-coordinate of the starting point.
- **End Point** is a point-coordinate cluster that defines the end point of the line.
  - $\mathbf{X}$  is the x-coordinate of the ending point.
  - Y is the y-coordinate of the ending point.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Color** is the color of the overlay.
  - Note The alpha color channel is not supported. Setting the Color to transparent has the same effect as selecting black.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for examples that use this VI.

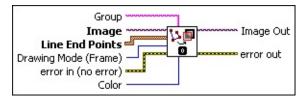
- LabVIEW\examples\Vision\3. Applications\Circle Distance Example.vi
- LabVIEW\examples\Vision\2. Functions\Pattern Matching\Pattern Matching Example.llb

# **IMAQ Overlay Multiple Lines VI**

**Owning Palette:** Overlay Installed With: NI Vision Development Module Overlays multiple lines, as either a polyline or a polygon, on an image.

#### **Details**





- **Group** specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image is a reference to the source image.
- Line End Points is an array of the point-coordinate clusters defining the ends of the lines to overlay.
  - $\mathbf{X}$  is the x-coordinate of a point in the array.
  - Y is the y-coordinate of a point in the array.
- **Drawing Mode (Frame)** specifies whether to draw only the frame of the polygon or fill in the polygon. The default value is **Frame**.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

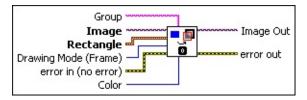
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Color** is the color of the overlay.
  - Note The alpha color channel is not supported. Setting the Color to transparent has the same effect as selecting black.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Overlay Rectangle VI**

**Owning Palette:** Overlay Installed With: NI Vision Development Module Overlays a rectangle on an image.

#### **Details** Examples





- **Group** specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image is a reference to the source image.
- Rectangle specifies the coordinates of the rectangle to overlay on the image.
  - **Left** is the value for the left side of the rectangle.
  - **Top** is the value for the top of the rectangle.
  - **Right** is the value for the right side of the rectangle.
  - **Bottom** is the value for the bottom of the rectangle.
- **Drawing Mode (Frame)** specifies whether to draw only the frame of the rectangle, fill in the rectangle, or highlight the rectangle.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Color** is the color of the overlay.
  - Note The alpha color channel is not supported. Setting the Color to transparent has the same effect as selecting black.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for examples that use this VI.

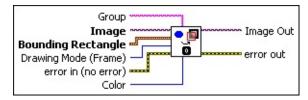
- LabVIEW\examples\Vision\3. Applications\Circle Distance Example.vi
- LabVIEW\examples\Vision\3. Applications\Gauging Example.llb

# **IMAQ Overlay Oval VI**

Owning Palette: OverlayInstalled With: NI Vision Development Module Overlays an oval on an image.

#### **Details** Examples





- **Group** specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image is a reference to the source image.
- **Bounding Rectangle** is the coordinates of the bounding rectangle that define the oval along which the arc lies.
  - **Left** is the value for the left side of the rectangle.
  - **Top** is the value for the top of the rectangle.
  - **Right** is the value for the right side of the rectangle.
  - **Bottom** is the value for the bottom of the rectangle.
- **Drawing Mode (Frame)** specifies whether to draw only the frame of the oval or fill in the oval.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Color** is the color of the overlay.
  - Note The alpha color channel is not supported. Setting the Color to transparent has the same effect as selecting black.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for examples that use this VI.

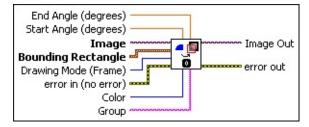
- LabVIEW\examples\Vision\2. Functions\Caliper\Edge Detection Example.llb
- LabVIEW\examples\Vision\2. Functions\Pattern Matching\Pattern Matching Example.llb

# **IMAQ Overlay Arc VI**

Owning Palette: OverlayInstalled With: NI Vision Development Module Overlays an arc on an image.

#### **Details**



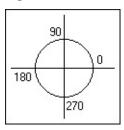


- **End Angle (degrees)** is the end angle for the arc.
- **Start Angle (degrees)** is the starting angle for the arc.
- **Image** is a reference to the source image.
- Bounding Rectangle is the coordinates of the bounding rectangle that define the oval along which the arc lies.
  - **Left** is the value for the left side of the rectangle.
  - **Top** is the value for the top of the rectangle.
  - **Right** is the value for the right side of the rectangle.
  - **Bottom** is the value for the bottom of the rectangle.
- **Drawing Mode (Frame)** specifies whether to draw only the frame of the arc or fill in the arc. The default value is **Frame**.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Color** is the color of the overlay.
  - Note The alpha color channel is not supported. Setting the Color to transparent has the same effect as selecting black.
- **Group** specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Overlays are nondestructive, which means that they do not overwrite the underlying pixel values. You can save overlay information with an image by using the <a href="MAQ Read Image And Vision Info">IMAQ Read Image And Vision Info</a> VI. If you want to replace the underlying pixel values with the overlay values, use the <a href="IMAQ Merge Overlay">IMAQ Merge Overlay</a> VI.

The arc is always drawn counterclockwise, as shown in the following figure:

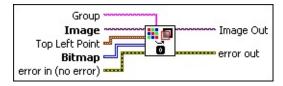


# **IMAQ Overlay Bitmap VI**

Owning Palette: OverlayInstalled With: NI Vision Development Module Overlays a bitmap on an image.

#### **Details**





- **Group** specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image is a reference to the source image.
- **Top Left Point** specifies the location in the image where the top, left point of the bitmap is to be placed.
  - **X** is the x-coordinate of the point.
  - Y is the y-coordinate of the point.
- Bitmap is the bitmap pixels to overlay.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

warning code.

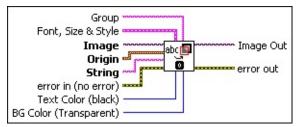
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Overlay Text VI**

**Owning Palette:** Overlay Installed With: NI Vision Development Module Overlays text on an image.

#### **Details Examples**





- **Group** specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- **Font, Size & Style** is a cluster that enables the user to choose the font, size, style, and alignment of the text. This cluster contains the following elements:
  - **Desired Font (Application)** specifies the character type of the text. The following values are possible:

User- specified Font (0)	Specify the attributes of a font to use. If no font is specified, the default font is used.
Application Font (1)	(Default) Uses the LabVIEW application font, which is the font used on the Controls and Functions palettes and on text in new objects on the front panel or block diagram.
System Font (2)	Uses the LabVIEW system font, which is the font used in menus.
Dialog Font (3)	Uses the LabVIEW dialog font, which is the font used in dialog boxes.

[User-specified Font] is a cluster containing the specific font characteristics for the text to draw. The specification is ignored unless **Desired Font** is set to **User-specified Font**.

- Note The Outline?, Shadow?, and Rotation Angle characteristics are not currently supported on RT targets.
- Font Name is the font to use for the string.
- **Size** is the size of the font in points.
- **Strikeout?** causes the text to appear in strikeout if TRUE.
- **Italic?** causes the text to appear in italic if TRUE.
- **Underline?** causes the text to appear underlined if TRUE.
- Outline? causes the text to appear in outline if TRUE.
- **Shadow?** causes the text to appear in shadow if TRUE.
- Bold? causes the text to appear in bold if TRUE
- Rotation Angle is the orientation of the text in the image. NI Vision supports rotated text for TrueType fonts only.
- Horizontal Alignment specifies the horizontal alignment of the text. The following values are possible:

Left (0)	(Default) Left-aligns the text.
Center (1)	Center-aligns the text.
Right (2)	Right-aligns the text.

Vertical Alignment specifies the vertical alignment of the text. The following values are possible:

Bottom (0)	(Default) Aligns the text along the bottom.
<b>Top</b> (1)	Aligns the text along the top.
Baseline (2)	Aligns the text along the baseline.

- **Image** is a reference to the source image.
- Origin is the location on the image to overlay the text. This parameter is dependent on the values of Horizontal Alignment and Vertical Alignment.
  - $\mathbf{X}$  is the x-coordinate.

- $\mathbf{Y}$  is the y-coordinate.
- **String** is the text to overlay.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Text Color (black)** is the color of the text.
- **BG Color (Transparent)** is the background color of the text overlay.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE

(checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

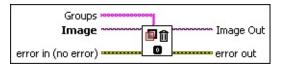
• LabVIEW\examples\Vision\3. Applications\Fuse Inspection Example.llb

# **IMAQ Clear Overlay VI**

**Owning Palette:** Overlay Installed With: NI Vision Development Module Clears the image overlay.

### **Examples**





- **Groups** is an array of strings consisting of the overlay group names to clear. If no names are specified, all available groups are cleared.
- Image is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

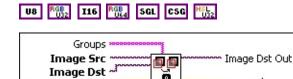
Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Fuse Inspection Example.llb

# **IMAQ Copy Overlay VI**

Owning Palette: Overlay Installed With: NI Vision Development Module

Copies an image overlay from the source image to the destination image. If the destination image already has an overlay, this VI adds the source overlay to the existing overlay.



Ō

- Groups is an array of strings consisting of the overlay group names to copy. If no names are specified, all available groups are copied.
- Image Src is a reference to the source image.

error out

- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image

**Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

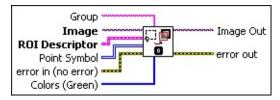
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Overlay ROI VI**

Owning Palette: OverlayInstalled With: NI Vision Development Module Overlays a region of interest (ROI) on an image.

#### **Details** Examples





- Group specifies the group to which you want to add the overlay information. If no name is specified, the overlay information is added to the default group.
- Image is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the region of interest to overlay on the image.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- Point Symbol is a bitmap that describes the symbol to use for a point ROI. If not specified, the VI overlays a single point for a point ROI.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or

function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Colors (Green)** is an array of colors associated with each of the ROIs described in the ROI descriptor.
  - **Note** The alpha color channel is not supported. Setting the color to transparent has the same effect as selecting black.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

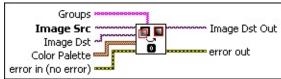
• LabVIEW\examples\Vision\2. Functions\Image Management\Unwrap Barcode Example.vi

# **IMAQ Merge Overlay VI**

Owning Palette: Overlay Installed With: NI Vision Development Module

Makes a nondestructive overlay part of the image content. This process creates a destructive overlay. The VI then removes the nondestructive overlay. The resulting image is an RGB image.





- **Groups** is an array of strings consisting of the overlay group names to merge to the image. If no names are specified, all available groups are merged.
- Image Src is a reference to the source image.
- Image Dst is the reference to the destination image. If connected, Image Dst is converted to an RGB image. If not connected, Image Src is converted to an RGB image.
- Color Palette is used to apply a color palette to the image. Color Palette is an array of clusters constructed by the user or supplied by the <a href="IMAQ GetPalette">IMAQ GetPalette</a> VI. This palette can be composed of up to 256 elements for each of the three color planes (red, green, and blue). If the array has less than 256 elements, the VI maps all pixel values past the last element in the array to the associated grayscale value.

A specific color is the result of applying a value between 0 and 255 for each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black, and 255 specifies white). The default is a grayscale palette. The color palette can be used only for 8-bit images.



**Tip** For best results, set your video adapter to high color or true color.

- Red is the value of the red color plane.
- Green is the value of the green color plane.
- Blue is the value of the blue color plane.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

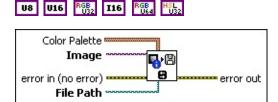
# **IMAQ Write Image And Vision Info VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Writes an image, along with extra vision information associated with the image, to a PNG file. This extra vision information includes overlay information, pattern matching template information, and calibration information.



**Note** The extra vision information can be saved only with the PNG file format.



Color Palette is used to apply a color palette to an image. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.

- Red is the value of the red color plane.
- Green is the value of the green color plane.
- Blue is the value of the blue color plane.
- Image is a reference to the image structure to write as an image file.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **File Path** is the complete pathname, including drive, directory, and filename, of the file to write.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Get Overlay Properties VI**

**Owning Palette:** Overlay Installed With: NI Vision Development Module Get the transformation behavior properties for a group of overlays.



error in (no error) 🕶

- Image is a reference to the source image.
- **Group** is the overlay group name for which you want to get the properties.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to Image.
- Properties returns the overlay properties for the specified overlay group
  - **Property** is the name of the overlay property. The following

#### values are valid:

Scale (0)	Specifies the behavior of an overlay group when a scale operation is applied to an image.
Shift (1)	Specifies the behavior of an overlay group when a shift operation is applied to an image.
Rotate (2)	Specifies the behavior of an overlay group when a rotate operation is applied to an image.
Symmetry (3)	Specifies the behavior of an overlay group when a symmetry operation is applied to an image.

**Behavior** is the behavior of the overlay property. The following values are valid:

Clear (0)	Sets the behavior of the overlay group to clear the current settings when an image is transformed.
Keep (1)	Sets the behavior of the overlay group to keep the current settings when an image is transformed.
Transform (2)	Sets the behavior of the overlay group to transform with the image.

error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

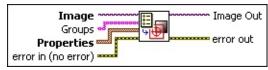
source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Set Overlay Properties VI**

**Owning Palette:** OverlayInstalled With: NI Vision Development Module Sets the transformation behavior properties for groups of overlays.





- **Image** is a reference to the source image.
- Groups is an array of strings consisting of the overlay group names for which you want to set the properties.
- Properties specifies the overlay properties for an overlay group.
  - **Property** is the name of the overlay property. The following values are valid:

Scale (0)	Specifies the behavior of an overlay group when a scale operation is applied to an image.
Shift (1)	Specifies the behavior of an overlay group when a shift operation is applied to an image.
Rotate (2)	Specifies the behavior of an overlay group when a rotate operation is applied to an image.
Symmetry (3)	Specifies the behavior of an overlay group when a symmetry operation is applied to an image.

**Behavior** is the behavior of the overlay property. The following values are valid:

Clear (0)	Sets the behavior of the overlay group to clear the current settings when an image is transformed.
Keep (1)	Sets the behavior of the overlay group to keep the current settings when an image is transformed.

<b>Transform</b> (2)	Sets the behavior of the overlay group to
	transform with the image.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to Image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Calibration**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Calibration VIs to spatially calibrate an image. Spatial calibration converts pixel coordinates to real-world coordinates while compensating for potential perspective errors or nonlinear distortions in your imaging system.

Palette Object	Description
IMAQ Learn Calibration Template	Learns a calibration from a template. The template can be an image of a grid of circles, or a set of pixel coordinates and corresponding real-world coordinates.
IMAQ Set Simple Calibration	Sets a simple calibration for an image or resets the real- world coordinate system in a calibrated image.
IMAQ Set Calibration Info	Sets calibration information from a calibrated image to an uncalibrated image. Both images must be the same size.
IMAQ Get Calibration Info	Returns calibration information associated with an image.
IMAQ Convert Real World to Pixel	Transforms real-world coordinates to pixel coordinates, according to the calibration information acquired from the <a href="MAQ Learn Calibration Template">IMAQ Learn Calibration Template</a> VI, the <a href="IMAQ Set Simple Calibration">IMAQ Set Simple Calibration</a> VI, or copied from a calibrated image using the <a href="IMAQ Set Calibration Info">IMAQ Set Calibration Info</a> VI.
IMAQ Convert Pixel to Real World	Transforms pixel coordinates to real-world coordinates, according to the calibration information contained in the image. Calibration information is attached to this image by the IMAQ Learn Calibration Template VI, the IMAQ Set Simple Calibration VI, or the IMAQ Set Calibration Info VI.
IMAQ Correct Calibrated	Corrects a calibrated image by applying a calibration to create a spatially correct image.

IMAQ	Reads an image file, including any extra vision information
	saved with the image. This includes overlay information,
	pattern matching template information, and calibration
And Vision	information, as written by the <u>IMAQ Write Image And Vision</u>
<u>Info</u>	<u>Info</u> VI.
IMAQ	Writes an image, along with extra vision information
<u>Write</u>	associated with the image, to a PNG file. This extra vision
<u>lmage</u>	information includes overlay information, pattern matching
And Vision	template information, and calibration information.
<u>Info</u>	

### **IMAQ Learn Calibration Template VI**

Owning Palette: <u>Calibration</u>Installed With: NI Vision Development Module

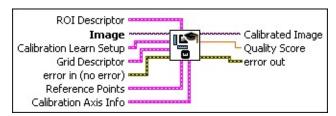
Learns a calibration from a template. The template can be an image of a grid of circles, or a set of pixel coordinates and corresponding real-world coordinates.



**Note** When learning a calibration template from an image of a grid of circles, the image type must be either U8 or I16.

#### **Examples**





- **ROI Descriptor** defines the region of interest to use in the learning process. All circles in the grid that are outside the defined region are not used when estimating the calibration transformation.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Image is a reference to the template used for calibrating your system. It should be an image of a grid of dots. The contents of this image are not used if the **Reference Points** array is provided to the VI.
- **Calibration Learn Setup** contains the following values:



**Distortion** indicates the type of distortion you can correct when correcting an image:

Perspective (0)	Image contains perspective calibration information.
Nonlinear (1)	Image contains nonlinear calibration information.
Simple Calibration (2)	Image contains simple calibration information (no perspective or nonlinear information).
Corrected (3)	Image has been corrected and contains calibration unit and scaling factor.

Corrected Image Scaling defines the aspect scaling to use when correcting an image.

Scale to	The corrected image is scaled such that the	
	features in the image have the same size as the	
Area (0)	input image.	
	<b>3</b>	
Fit (1)	size as the input image.	

Corrected Image Learn ROI specifies the ROI to use when correcting an image.

Full Image (0)	The whole image is always corrected, regardless of the user-defined or calibration-defined ROIs.
Calibration ROI (1)	The area defined by the Calibration ROI is corrected. The calibration ROI is computed by the algorithm and corresponds to the area of the calibration template containing dots.
<b>ROI</b> (2)	The area defined by <b>ROI Descriptor</b> is corrected.
ROI AND Calibration ROI (3)	The area defined by the intersection of the ROI and Calibration ROI is corrected.
ROI OR Calibration	The area defined by the union of the ROI and Calibration ROI is corrected.

<b>ROI</b> (4)		
----------------	--	--

- Learn Error Map? is a Boolean input value. Set this input to TRUE if you want the software to calculate and store an error map during the learning process. The image error map reflects error bounds on the calibration transformation.
- Learn Correction Table? is a Boolean input value. Set this input to TRUE if you want the correction table to be determined and stored. The correction table accelerates the process of correcting an image. It is useful if several images are going to be corrected.
- Grid Descriptor contains information about a grid image that is used to learn the calibration. If the image has been calibrated previously, using the IMAQ Learn Calibration Template VI, this input is ignored and the previously defined grid descriptor is used. This is an optional input. This input contains the following information:
  - **X Step** is the real-world distance between circle centers in the x direction.
  - Y Step is the real-world distance between circle centers in the y direction.
  - Unit is the units of X Step and Y Step.
  - Threshold Range is the range of grayscale values in a grid image that represent the circles. This is the threshold range for detecting the centers of the circles. Lower value and Upper value must be between 0–255.
    - **Lower value** is the lowest pixel value used during a manual threshold. The default is 1.
    - **Upper value** is the highest pixel value used during a manual threshold. The default is 255.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Reference Points is the set of reference points to use in learning a calibration transformation. The minimum number of points is 4. If you connect Reference Points, a grid image is not required.
  - **Pixel Coordinates** is the coordinates of the pixel reference points.
  - Real World Coordinates is the measuring unit associated with the image.
  - **Unit** is the measuring unit associated with the image. Choose from the following values:

undefined (0)	Units are undefined.
angstrom (1)	Units are measured in angstroms.
micrometer (2)	Units are measured in micrometers.
millimeter (3)	Units are measured in millimeters.
centimeter (4)	Units are measured in centimeters.
meter (5)	Units are measured in meters.
kilometer (6)	Units are measured in kilometers.
microinch (7)	Units are measured in microinches.
inch (8)	Units are measured in inches.
feet (9)	Units are measured in feet.

nautical miles (10)	Units are measured in nautical miles.
standard miles (11)	Units are measured in standard miles.
steps (12)	Units are measured in steps.

- Calibration Axis Info defines a Reference Coordinate System for the real-world coordinates.
  - Origin Pixel is the origin of the reference coordinate system, expressed in pixel units.
  - Angle Relative to Horizontal (deg) is the angle of the x-axis of the real-world coordinate system, in relation to the horizontal. The horizontal is expressed by a row of circle centers.
  - Axis Reference is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.
- Calibrated Image is the reference to the output image. This image contains the learned calibration information. Use this image to propagate the calibration information through your application.

  The calibration information is associated with the image until the image is resized or disposed of. Use the <a href="IMAQ Write Image And Vision Info">IMAQ Write Image And Vision Info</a> VI to save the calibration information to disk with the image. You can use the <a href="IMAQ Read Image And Vision InfoVI">IMAQ Read Image And Vision InfoVI</a> to
- Quality Score is the quality score of the learning process between 0–1000. A quality of 1000 means that the chosen algorithm learned the feature points well. It does not necessarily reflect the absolute accuracy of the estimated calibration mapping.

retrieve the calibration information.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is

a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Examples**

Refer to the following for examples that use this VI.

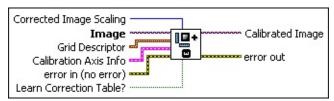
- LabVIEW\examples\Vision\2. Functions\Calibration\Perspective Calibration Example.llb
- $\bullet \quad LabVIEW \land 2. Functions \land Calibration \land Simple \ Calibration \\ Example.llb$

# **IMAQ Set Simple Calibration VI**

Owning Palette: <u>Calibration</u>Installed With: NI Vision Development Module

Sets a simple calibration for an image or resets the real-world coordinate system in a calibrated image. When used to reset a coordinate system, if the correction table is still required, the table must be relearned. When **Angle Relative to Horizontal** is set to 0, you do not need to make a correction.





Corrected Image Scaling defines the scaling to be used in the corrected image. If the image has been calibrated previously, using the <a href="IMAQ Learn Calibration Template">IMAQ Learn Calibration Template</a> VI, this input is ignored and the previously defined scaling is used. This is an optional input.

The corrected image is scaled such that the features in the image have the same size as the input image.
The corrected image is scaled to be the same size as the input image.

- Image is a reference to the uncalibrated image.
- Grid Descriptor contains information about a grid image that is used to learn the calibration. If the image has been calibrated previously, using the <a href="IMAQ Learn Calibration Template">IMAQ Learn Calibration Template</a> VI, this input is ignored and the previously defined grid descriptor is used. This is an optional input. This input contains the following information:
  - **Step** is the width of a pixel in real-world units.
  - Y Step is the height of a pixel in real-world units.
  - **Unit** is the unit for **X Step** and **Y Step**. The following options are valid:

Undefined (0)	Units are undefined.
	l l

į	1
Angstrom (1)	Units are measured in angstroms.
Micrometer (2)	Units are measured in micrometers.
Millimeter (3)	Units are measured in millimeters.
Centimeter (4)	Units are measured in centimeters.
Meter (5)	Units are measured in meters.
Kilometer (6)	Units are measured in kilometers.
Microinch (7)	Units are measured in microinches.
Inch (8)	Units are measured in inches.
Feet (9)	Units are measured in feet.
Nautical miles (10)	Units are measured in nautical miles.
Standard miles (11)	Units are measured in standard miles.
Steps (12)	Units are measured in steps.

- Calibration Axis Info defines a Reference Coordinate System for the real-world coordinates. For an image that has already been calibrated, this axis information overrides the old axis information.
  - Origin Pixel is the origin of the reference coordinate system, expressed in pixel units.
  - Angle Relative to Horizontal (deg) is the angle of the x-axis of the real-world coordinate system, in relation to the horizontal. The horizontal is expressed by a row of circle centers.
  - Axis Reference is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Learn Correction Table? is a Boolean input. If TRUE, the correction table is processed and stored. The correction table accelerates the process of correcting an image. It is useful if several images are going to be corrected. If the current operation is resetting the real-world coordinate system, or if the correction table is still required, the correction table will be relearned.
- Calibrated Image is the reference to the output image. This image contains the learned calibration information. Use this image to propagate the calibration information through your application.

  The calibration information is associated with the image until the image is resized or disposed of. Use the <a href="IMAQ Write Image And Vision Info">IMAQ Write Image And Vision Info</a> VI to save the calibration information to disk with the image. You can use the <a href="IMAQ Read Image And Vision Info">IMAQ Read Image And Vision Info</a> VI to retrieve the calibration information.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

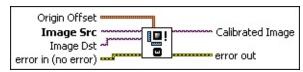
### **IMAQ Set Calibration Info VI**

Owning Palette: <u>Calibration</u>Installed With: NI Vision Development Module

Sets calibration information from a calibrated image to an uncalibrated image. Both images must be the same size.

#### **Examples**





- Origin Offset specifies the (x,y) location of the origin (top-left corner) of the destination image (Image Dst) with respect to the origin of template image (Image Src). The default is (-1, -1).
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

**Calibrated Image** is the reference to the output image. This image contains the learned calibration information. Use this image to propagate the calibration information through your application.

The calibration information is associated with the image until the image is resized or disposed of. Use the <u>IMAQ Write Image And Vision Info</u> VI to save the calibration information to disk with the image. You can use the <u>IMAQ Read Image And Vision Info</u>VI to retrieve the calibration information.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Examples**

Refer to the following for examples that use this VI.

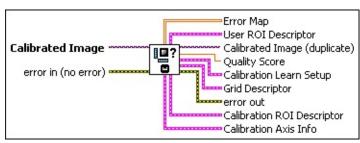
- LabVIEW\examples\Vision\2. Functions\Calibration\Perspective Calibration Example.llb
- LabVIEW\examples\Vision\2. Functions\Calibration\Perspective Calibration Example.llb

### **IMAQ Get Calibration Info VI**

Owning Palette: <u>Calibration</u>Installed With: NI Vision Development Module

Returns calibration information associated with an image. Use the <u>IMAQ</u> <u>Learn Calibration Template</u> VI or the <u>IMAQ Set Simple Calibration</u> VI to set this calibration information.





- **Calibrated Image** is the reference to the input image that determines the output calibration information.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **Error Map** returns the error map if it was determined by the <u>IMAQ</u> Learn Calibration Template VI.
- **User ROI Descriptor** defines the region you specified at the time of calibration.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Calibrated Image (duplicate) is a reference to the output image. This image is identical to the input Calibrated Image.
- Quality Score is the quality score of the learning process between 0–1000. A quality of 1000 means that the chosen algorithm learned the feature points well. It does not necessarily reflect the absolute accuracy of the estimated calibration mapping.
- **Calibration Learn Setup** contains the following values:
  - **Distortion** indicates the type of distortion you can correct when correcting an image. Choose from the following options:

Perspective (0)	Image contains perspective calibration information.
Nonlinear (1)	Image contains nonlinear calibration information.
Simple Calibration (2)	Image contains simple calibration information (no perspective or nonlinear information).
Corrected (3)	Image has been corrected and contains calibration unit and scaling factor.

Corrected Image Scaling defines the aspect scaling to use when correcting an image.

Scale to	The corrected image is scaled such that the
Preserve	features in the image have the same size as the
<b>Area</b> (0)	input image.
	The corrected image is scaled to be the same
Fit (1)	size as the input image.

Corrected Image Learn ROI specifies the ROI to use when correcting an image.

Full Image (0)	The whole image is always corrected, regardless of the user-defined or calibration-defined ROIs.
Calibration ROI (1)	The area defined by the Calibration ROI is corrected. The calibration ROI is computed by the algorithm and corresponds to the area of the calibration template containing dots.
<b>ROI</b> (2)	The area defined by <b>ROI Descriptor</b> is corrected.
ROI AND Calibration ROI (3)	The area defined by the intersection of the ROI and Calibration ROI is corrected.
ROI OR Calibration ROI (4)	The area defined by the union of the ROI and Calibration ROI is corrected.

- Learn Error Map? is a Boolean input value. Set this input to TRUE if you want the software to calculate and store an error map during the learning process. The image error map reflects error bounds on the calibration transformation.
- Learn Correction Table? is a Boolean input value. Set this input to TRUE if you want the correction table to be determined and stored. The correction table accelerates the process of correcting an image. It is useful if several images are going to be corrected.
- Grid Descriptor returns scaling constants used to calibrate the image.
  - X Step is the real-world distance between circle centers in the

- x direction.
- Y Step is the real-world distance between circle centers in the y direction.
- **Unit** is the units of **X Step** and **Y Step**.
- Threshold Range is the range of grayscale values in a grid image that represent the circles. This is the threshold range for detecting the centers of the circles. Lower value and Upper value must be between 0?255.
  - **Lower value** is the lowest pixel value used during a manual threshold. The default is 1.
  - **Upper value** is the highest pixel value used during a manual threshold. The default is 255.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Calibration ROI Descriptor** returns an ROI that corresponds to the region of the image where the calibration information is accurate.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.

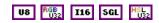
- **Type** is the shape type of the contour.
- **Coordinates** indicates the relative position of the contour.
- **Calibration Axis Info** defines a **Reference Coordinate System** for the real-world coordinates.
  - Origin Pixel is the origin of the reference coordinate system, expressed in pixel units.
  - Angle Relative to Horizontal (deg) is the angle of the x-axis of the real-world coordinate system, in relation to the horizontal. The horizontal is expressed by a row of circle centers.
  - Axis Reference is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.

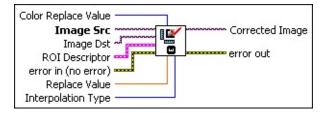
### **IMAQ Correct Calibrated Image VI**

Owning Palette: <u>Calibration</u>Installed With: NI Vision Development Module

Corrects a calibrated image by applying a calibration to create a spatially correct image.

### **Examples**





- Color Replace Value is the value used to replace pixels in a corrected color image that were not part of the original image.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- **ROI Descriptor** is the ROI to be corrected.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Replace Value is the value that the VI uses to fill pixels in the corrected image that the VI could not correct in Image Src. The default is 0.
- Interpolation Type specifies the interpolation method used to obtain the corrected image. The default is **Zero Order**. The following options are valid:

Zero Order (0)	Rounds to the nearest integral pixel location
Bi-Linear (1)	Uses linear interpolation in both the x- and y-directions to compute the pixel location
Quadratic (2)	Uses the quadratic formula to compute the pixel location
Cubic Spline (3)	Uses cubic splines to compute the pixel location

- Corrected Image is the reference to the corrected destination (output) image. If the Image Dst is connected, then Corrected Image is the same as Image Dst. Otherwise, Corrected Image refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Calibration\Simple Calibration Example.vi

### **Color Utilities**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Color Utilities VIs to manipulate the colors and color planes of an image. Use these VIs to extract different color planes from an image, replace the planes of a color image with new data, convert a color image into a 2D array and back, read and set pixel values in a color image, and convert pixel values from one color space to another.

Palette Object	Description
IMAQ ExtractColorPlanes	Extracts the three planes (RGB, HSL, HSV, or HSI) from an image.
IMAQ ExtractSingleColorPlane	Extracts a single plane from a color image.
IMAQ ReplaceColorPlane	Replaces one or more image planes from a color image (RGB, HSL, HSV, or HSI). Only the planes connected at the input are replaced.
IMAQ GetColorPixelValue	Reads the pixel values from a color image.
IMAQ SetColorPixelValue	Changes a pixel value in a color image.
IMAQ GetColorPixelLine	Extracts a line of pixels from a color image.
IMAQ SetColorPixelLine 2	Changes a line of pixels from a color image.
IMAQ ColorImageToArray	Extracts the pixels from a color image or from part of a color image into a 2D array.
IMAQ ArrayToColorImage	Creates a color image from a 2D array.
IMAQ RGBToColor 2	Converts an RGB color value into another format (HSL, HSV, HSI, CIE L*a*b*, or CIE XYZ).
IMAQ ColorToRGB	Converts an HSL, HSV, or HSI color value into an RGB color value.

IMAQ ColorValueToInteger	Converts clusters composed of three colors in mode (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into colors encoded in the form of an unsigned 32-bit integer.
IMAQ IntegerToColorValue	Converts colors in the form of an unsigned 32-bit integer into a cluster composed of the three colors in mode (R, G, B), (H, S, L), (H, S, V), or (H, S, I).

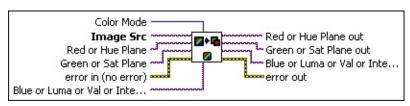
### **IMAQ ExtractColorPlanes VI**

Owning Palette: Color Utilities Installed With: NI Vision Development

Module

Extracts the three planes (RGB, HSL, HSV, or HSI) from an image.





**Color Mode** defines the image color format to use for the operation. Choose from the following values:



**Note** Unsigned 64-bit RGB images support only a **Color Mode** of **RGB**.

<b>RGB</b> (0)	(Default) Specifies the color format RGB (red, green, and blue)
<b>HSL</b> (1)	Specifies the color format HSI (hue, saturation, and luminance)
<b>HSV</b> (2)	Specifies the color format HSV (hue, saturation, and value)
<b>HSI</b> (3)	Specifies the color format HSI (hue, saturation, and intensity)

- Image Src is a reference to the source image.
- Red or Hue Plane is the reference to the destination image for the first color plane. This plane can be either the red plane (Color Mode RGB) or the hue plane (Color Mode HSL, HSV, or HSI). If Red or Hue Plane is connected, it must have as many bits per pixel as the extracted color plane. If Red or Hue Plane is not connected, the VI does not extract the first color plane.
- Green or Sat Plane is the reference to the destination image for the second color plane. This plane can be either the green plane (Color Mode RGB) or the saturation plane (Color Mode HSL, HSV, or HSI). If Green or Saturation Plane is connected, it must have as many bits per pixel as the extracted color plane. If Green or

**Saturation Plane** is not connected, the VI does not extract the second color plane.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Blue or Luma or Val or Inten Plane is the reference to the destination image for the third color plane. This plane can be either the blue plane (Color Mode RGB), the luminance plane (Color Mode HSL), the value plane (Color Mode HSV) or the intensity plane (Color Mode HSI). If Blue or Luma or Val or Inten Plane is connected, it must have as many bits per pixel as the extracted color plane. If Blue or Luma or Val or Inten Plane is not connected, the VI does not extract the third color plane.
- Red or Hue Plane out is the reference to the image containing the red (or hue) plane of the source (input) image.
- Green or Sat Plane out is the reference to the image containing the green (or saturation) plane of the source (input) image.
- Blue or Luma or Val or Inten Plane out is the reference to the image containing the blue (or luminance, value, or intensity) plane

of the source (input) image.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

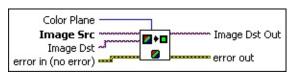
# IMAQ ExtractSingleColorPlane VI

Owning Palette: Color Utilities Installed With: NI Vision Development

Module

Extracts a single plane from a color image.





**Color Plane** defines the color plane to extract. Choose from the following values:



**Note** Unsigned 64-bit RGB images support only **Color Plane** values of **Red**, **Green**, or **Blue**.

<b>Red</b> (0)	(Default) Extracts the red color plane
Green (1)	Extracts the green color plane
Blue (2)	Extracts the blue color plane
<b>Hue</b> (3)	Extracts the hue color plane
<b>Saturation</b> (4)	Extracts the saturation color plane
<b>Luminance</b> (5)	Extracts the luminance color plane
Value (6)	Extracts the value color plane
Intensity (7)	Extracts the intensity color plane

- Image Src is the reference to a color image that has one of its color planes extracted. If Image Dst is not connected, the source image is converted to an image that contains the extracted plane.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

to check errors and to specify execution order by wiring error out from one node to error in of the next node.

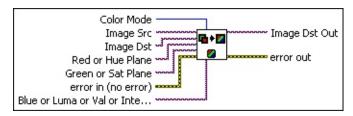
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ ReplaceColorPlane VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Replaces one or more image planes from a color image (RGB, HSL, HSV, or HSI). Only the planes connected at the input are replaced. If all three planes are connected, the input **Image Src** is not necessary, and only the **Image Dst** is used. The image is resized to the dimensions of the planes passed on input. Therefore, their sizes must be identical. If one or two planes are connected, the planes must have the same dimension as the source image.





**Color Mode** defines the image color format to use for the operation. Choose from the following values:



**Note** Unsigned 64-bit RGB images support only a **Color Mode** of **RGB**.

<b>RGB</b> (0)	(Default) Specifies the color format RGB (red, green, and blue)
	Specifies the color format HSI (hue, saturation, and luminance)
<b>HSV</b> (2)	Specifies the color format HSV (hue, saturation, and value)
<b>HSI</b> (3)	Specifies the color format HSI (hue, saturation, and intensity)

- Image Src is a reference to the source image. This image is not necessary if the Image Dst and the three color planes are connected.
- Image Dst is a reference to the destination image.
- Red or Hue Plane is the reference to the source image for the first

color plane. This plane can be either the red plane (**Color Mode** RGB) or the hue plane (**Color Mode** HSL, HSV, or HSI). If **Red or Hue Plane** is connected, it must have as many bits per pixel as the replaced color plane. If **Red or Hue Plane** is not connected, the VI does not replace the first color plane.

- Green or Sat Plane is the reference to the source image for the second color plane. This plane can be either the green plane (Color Mode RGB) or the saturation plane (Color Mode HSL, HSV, or HSI). If Green or Saturation Plane is connected, it must have as many bits per pixel as the replaced color plane. If Green or Saturation Plane is not connected, the VI does not replace the second color plane.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Blue or Luma or Val or Inten Plane is the reference to the third color plane. This plane can be either the blue plane (Color Mode RGB), the luminance plane (Color Mode HSL), the value plane (Color Mode HSV) or the intensity plane (Color Mode HSI). If Blue

or Luma or Val or Inten Plane is connected, it must have as many bits per pixel as the replaced color plane. If Blue or Luma or Val or Inten Plane is not connected, the VI does not replace the third color plane.

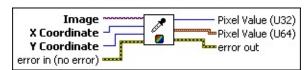
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### IMAQ GetColorPixelValue VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Reads the pixel values from a color image. This VI returns the pixel value as either an unsigned 32-bit integer or a cluster of four unsigned 16-bit integers, depending on the bit depth of **Image**. You can convert the 32-bit integer into a cluster containing three elements possessing either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) using the <u>IMAQ IntegerToColorValue</u> VI.





- **Image** is a reference to the source image.
- X Coordinate is the horizontal coordinate of the pixel.
- Y Coordinate is the vertical coordinate of the pixel.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

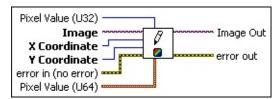
- most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixel Value (U32) returns the pixel value as an unsigned 32-bit integer.
- Pixel Value (U64) returns the pixel value as a cluster of four unsigned 16-bit integers.
  - Plane 1 (Red) is the value of the red plane.
  - Plane 2 (Green) is the value of the green plane.
  - Plane 3 (Blue) is the value of the blue plane.
  - Plane 4 (alpha) is the value of the alpha plane.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### IMAQ SetColorPixelValue VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Changes a pixel value in a color image. This VI receives the pixel value as either an unsigned 32-bit integer or a cluster of four unsigned 16-bit integers, depending on the bit depth of **Image**. You can convert the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into an unsigned 32-bit integer control using the <u>IMAQ ColorValueToInteger</u> VI.





- Pixel Value (U32) is the pixel value as an unsigned 32-bit integer.
- **Image** is a reference to the source image.
- X Coordinate is the horizontal coordinate of the pixel.
- Y Coordinate is the vertical coordinate of the pixel.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

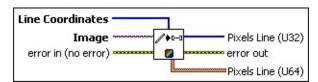
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixel Value (U64) is the pixel value as a cluster of four unsigned 16-bit integers.
  - Plane 1 (Red) is the value of the red plane.
  - Plane 2 (Green) is the value of the green plane.
  - Plane 3 (Blue) is the value of the blue plane.
  - Plane 4 (alpha) is the value of the alpha plane.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### IMAQ GetColorPixelLine VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Extracts a line of pixels from a color image. This VI returns an array of either unsigned 32-bit integers or clusters of four unsigned 16-bit integers, depending on the bit depth of Image. You can convert an array of 32-bit integers into an array of clusters coding the three color values as either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) using the <a href="IMAQ">IMAQ</a> IntegerToColorValue VI.





Line Coordinates is an array specifying the start x, start y, end x, and end y coordinates of the two end points of the line to extract.



**Note** A line designated by the coordinates [0, 0, 0, 255] consists of 256 pixels. The output **Pixels Line** contains the values specified by this line. Any pixel values outside the image automatically are set to 0 in **Pixels Line**.

- **Image** is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

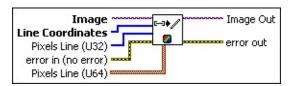
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixels Line (U32) returns the pixel values as a 1D array of unsigned 32-bit integer indicators.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixels Line (U64) returns the pixel values as clusters of four unsigned 16-bit integers.
  - Plane 1 (Red) is the value of the red plane.
  - Plane 2 (Green) is the value of the green plane.
  - Plane 3 (Blue) is the value of the blue plane.
  - Plane 4 (alpha) is the value of the alpha plane.

# **IMAQ SetColorPixelLine 2 VI**

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Changes a line of pixels from a color image. This VI receives an array of either unsigned 32-bit integers or clusters of four unsigned 16-bit integers, depending on the bit depth of Image. You can convert an array of clusters coding the color three values (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into an array of pixels (unsigned 32-bit integers) using the <a href="IMAQ">IMAQ</a> IntegerToColorValue VI.





- **Image** is a reference to the source image.
- Line Coordinates is an array specifying the pixel coordinates that form the endpoints of the line to modify. Any coordinates located outside the actual image are not used.
- Pixels Line (U32) contains the pixel values as a 1D array of unsigned 32-bit integers.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

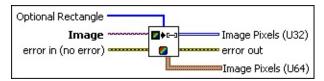
- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Pixels Line (U64) contains the pixel values as clusters of four unsigned 16-bit integers.
  - Plane 1 (Red) is the value of the red plane.
  - Plane 2 (Green) is the value of the green plane.
  - Plane 3 (Blue) is the value of the blue plane.
  - Plane 4 (alpha) is the value of the alpha plane.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ ColorImageToArray VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Extracts the pixels from a color image or from part of a color image into a 2D array. This VI returns the values as a 2D array of either unsigned 32-bit integers or clusters of four unsigned 16-bit integers, depending on the bit depth of **Image**. You can convert a 2D array of unsigned 32-bit integers into a 2D array of clusters coding the three color values as either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) using the <u>IMAQ</u> IntegerToColorValue VI.





- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- **Image** is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

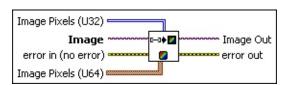
- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Pixels (U32) returns the pixel values as a 2D array of unsigned 32-bit integers.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Pixels (U64) returns the pixel values as a cluster of four unsigned 16-bit integers.
  - Plane 1 (Red) is the value of the red plane.
  - Plane 2 (Green) is the value of the green plane.
  - Plane 3 (Blue) is the value of the blue plane.
  - Plane 4 (alpha) is the value of the alpha plane.

# IMAQ ArrayToColorImage VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Creates a color image from a 2D array. This VI receives the values as a 2D array of either unsigned 32-bit integers or clusters of four unsigned 16-bit integers, depending on the bit depth of **Image**. You can convert a 2D array of clusters coding the three color values as either (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into a 2D array of unsigned 32-bit integers using the <u>IMAQ ColorValueToInteger</u> VI.





- Image Pixels (U32) contains the pixel values as a 2D array of unsigned 32-bit integer controls.
- **Image** is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

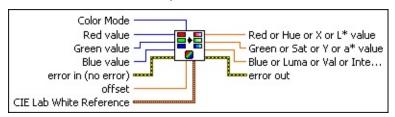
most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Pixels (U64) contains the pixel values as clusters of four unsigned 16-bit integers.
  - Plane 1 (Red) is the value of the red plane.
  - Plane 2 (Green) is the value of the green plane.
  - Plane 3 (Blue) is the value of the blue plane.
  - Plane 4 (alpha) is the value of the alpha plane.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ RGBToColor 2 VI**

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Converts an RGB color value into another format (HSL, HSV, HSI, CIE L\*a\*b\*, or CIE XYZ).



Color Mode defines the image color format conversion to perform. Choose from the following values:

<b>RGB</b> (0)	(Default) no change
HSL (1)	Convert to HSL
<b>HSV</b> (2)	Convert to HSV
<b>HSI</b> (3)	Convert to HSI
CIE L*a*b* (4)	Convert to CIE L*a*b*
CIE XYZ (5)	Convert to CIE XYZ

- Red value is the input red value.
- Green value is the input green value.
- Blue value is the input blue value.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- offset adds an offset to the calculated **Hue** value when **Color Mode** is set to HSL. The offset represents the angle by which the hue plane is rotated. **Offset** can range from 0 to 360.
  - The default offset value of 0 results in a hue value of 0 for the color red (R=255, G=0, B=0). By changing the offset value, you can specify the RGB color that maps to a hue value of 0. When you want to analyze red or colors close to red in the HSL space, you can add an offset so that the hue values associated with these colors are not zero.
- CIE Lab White Reference is the CIE X, Y, and Z values associated with white. The default values of this control work with RGB values of (255, 255, 255) as white.
- Red or Hue or X or L\* value is the output value for the first color plane, depending on the Color Mode chosen.
- Green or Sat or Y or a\* value is the output value for the second color plane, depending on the Color Mode chosen.
- Blue or Luma or Val or Inten or Z or b\*value is the output value for the third color plane, depending on the Color Mode chosen.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is

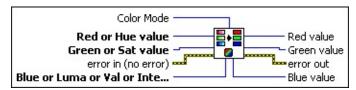
a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ ColorToRGB VI**

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Converts an HSL, HSV, or HSI color value into an RGB color value.



Color Mode defines the image color format to use for the operation. The default is **RGB**, which specifies that the input and output values are the same. Choose from the following values:

<b>RGB</b> (0)	(Default) No change	
<b>HSL</b> (1)	Convert to HSL	
<b>HSV</b> (2)	Convert to HSV	
<b>HSI</b> (3)	Convert to HSI	

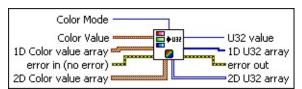
- Red or Hue value is the input value for the first color plane, depending on the Color Mode chosen.
- Green or Sat value is the input value for the second color plane, depending on the Color Mode chosen.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Blue or Luma or Val or Inten value is the input value for the third color plane, depending on the Color Mode chosen.
- **Red value** is the output red value.
- Green value is the output green value.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Blue value is the output blue value.

### IMAQ ColorValueToInteger VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Converts clusters composed of three colors in mode (R, G, B), (H, S, L), (H, S, V), or (H, S, I) into colors encoded in the form of an unsigned 32-bit integer. The elements of these clusters can contain single values, 1D arrays, 2D arrays, or a combination of the above.



Color Mode defines the image color format to use for the operation. The default is **RGB**, which specifies that the input and output values are the same. Choose from the following values:

<b>RGB</b> (0)	RGB (0) (Default) No change	
<b>HSL</b> (1)	Convert to HSL	
<b>HSV</b> (2)	Convert to HSV	
<b>HSI</b> (3)	Convert to HSI	

- Color Value is a cluster containing the value the VI converts. The VI interprets the cluster as (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the Color Mode. The cluster is composed of the following elements:
  - Red or Hue value is the first color plane value, depending on the Color Mode.
  - Green or Sat value is the second color plane value, depending on the Color Mode.
  - Blue or Luma or Val or Inten value is the third color plane value, depending on the Color Mode.
- **1D Color value array** is a 1D array of clusters containing the color values.
  - Red or Hue value is the first color plane value, depending on the Color Mode.
  - Green or Sat value is the second color plane value,

depending on the Color Mode.

- Blue or Luma or Val or Inten value is the third color plane value, depending on the Color Mode.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **2D Color value array** is a 2D array of clusters containing the color values. The VI interprets the array of clusters as (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the value **Color Mode**. These clusters are the same type as **Color Value**.
  - Red or Hue value is the first color plane value, depending on the Color Mode.
  - Green or Sat value is the second color plane value, depending on the Color Mode.
  - Blue or Luma or Val or Inten value is the third color plane value, depending on the Color Mode.
- **U32 value** receives the color value resulting from the input **Color Value**. This array is encoded as an unsigned 32-bit integer.

- 1D U32 array receives the color value resulting from the input 1D Color value array. This array is encoded as a 1D array of unsigned 32-bit integers.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **2D U32 array** receives the color value resulting from the input **2D Color value array**. This array is encoded as a 2D array of unsigned 32-bit integers.

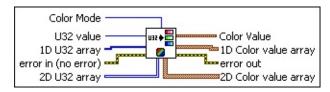
# IMAQ IntegerToColorValue VI

Owning Palette: Color Utilities Installed With: NI Vision Development Module

Converts colors in the form of an unsigned 32-bit integer into a cluster composed of the three colors in mode (R, G, B), (H, S, L), (H, S, V), or (H, S, I). You can enter these colors as a single value, a 1D array, a 2D array, or a combination of the above.



**Note** This VI ignores any information contained in the alpha channel of a color value.



Color Mode defines the image color format to use for the operation. The default is **RGB**, which specifies that the input and output values are the same. Choose from the following values:

<b>RGB</b> (0)	<b>GB</b> (0) (Default) No change	
<b>HSL</b> (1)	Convert to HSL	
<b>HSV</b> (2)	Convert to HSV	
<b>HSI</b> (3)	Convert to HSI	

- U32 value is a color value encoded as an unsigned 32-bit integer.
- **1D U32 array** is a set of color values encoded as a 1D array of unsigned 32-bit integers.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **2D U32 array** is a set of color values encoded as a 2D array of unsigned 32-bit integers.
- Color Value is a cluster containing the color value resulting from the input U32 value. This cluster can contain the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the value of Color Mode. The cluster is composed of the following elements:
  - Red (or Hue) value is the first color plane value, depending on the Color Mode.
  - Green (or Sat) value is the second color plane value, depending on the Color Mode.
  - Blue (or Light or Val) value is the third color plane value, depending on the Color Mode.
- **1D Color value array** is a 1D array containing the color value resulting from the input **1D U32 Array**. This array can contain the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the status of the set **Color Mode**.
  - Red (or Hue) value is the first color plane value, depending on the Color Mode.
  - Green (or Sat) value is the second color plane value, depending on the Color Mode.
  - Blue (or Light or Val) value is the third color plane value, depending on the Color Mode.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **2D Color value array** is a 2D array of clusters containing the color value resulting from the input **2D U32 Array**. This array can contain the values (R, G, B), (H, S, L), (H, S, V), or (H, S, I), depending on the value of **Color Mode**.
  - Red (or Hue) value is the first color plane value, depending on the Color Mode.
  - Green (or Sat) value is the second color plane value, depending on the Color Mode.
  - Blue (or Light or Val) value is the third color plane value, depending on the Color Mode.

#### **Vision RT**

Owning Palette: <u>Vision Utilities</u>Installed With: NI Vision Development Module

Use the Vision RT VIs to configure and control display settings for your RT system and to setup time-bounded operations within your application.

	Ter bystem and to setup time bounded operations within your application.		
Palette Object	Description		
IMAQ RT Video Out	Displays an image to the video out port on a LabVIEW Real- Time Module system.		
IMAQ Video Out Display Mode	Sets/gets the Video Out Display Mode settings for RT Video Out on a LabVIEW RT system.		
IMAQ Remote Display Options	Sets or gets the image compression options for remote display. These compression options affect remotely acquired images that are transferred and displayed on a host machine.		
IMAQ Initialize Timed Execution	Reserves memory for the algorithms used during time- bounded execution and enables timing capabilities.		
IMAQ Start Timed Execution	Sets the time limit bounding the execution of Vision VIs. All Vision VIs executing while the time limit is set are subject to the time bound. Only one time limit is active at any time.		
IMAQ Stop Timed Execution	Turns off the time limit established by the <u>IMAQ Start Timed</u> <u>Execution</u> VI and returns the time elapsed since the start of the time limit.		
IMAQ Uninitialize Timed Execution	Frees the resources reserved by the IMAQ Initialize Timed Execution VI and disables timing capabilities.		
IMAQ Check Timed Execution	Returns a timeout error if the time limit is reached or exceeded.		

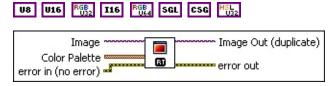
### **IMAQ RT Video Out VI**

Owning Palette: <u>Vision RT</u>Installed With: NI Vision Development Module

Displays an image to the video out port on a LabVIEW Real-Time Module system.



**Note** This VI is only supported on ETS targets.



**Image** is a reference to the displayed image.



**Note** Floating point images are displayed by scaling the pixel values to 8 bits. 16-bit images that have a specified bit depth are displayed by scaling the pixel values to 8 bits based on the specified bit depth. 16-bit images that do not have a specified bit depth are displayed according to the <a href="IMAQ WindDisplayMapping">IMAQ WindDisplayMapping</a> VI.

Color Palette is used to apply a color palette to the image window. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.



**Tip** For best results, set your video adapter to high color or true color.

- Red is the value of the red color plane.
- Green is the value of the green color plane.
- **Blue** is the value of the blue color plane.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out (duplicate) is a reference to Image. This VI does not modify the image connected to the Image input.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

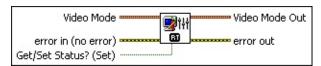
### **IMAQ Video Out Display Mode VI**

Owning Palette: <u>Vision RT</u>Installed With: NI Vision Development Module

Sets/gets the Video Out Display Mode settings for RT Video Out on a LabVIEW RT system.



**Note** This VI is only supported on ETS targets.



- **Video Mode** settings control how your images are displayed on your remote system monitor.
  - Screen Area specifies the screen resolution settings for the monitor connected to your LabVIEW RT target. Refer to your monitor documentation for information about the maximum supported screen area.
  - Color Depth specifies the number of bits used to represent a screen pixel.
  - Refresh Frequency allows you to choose a refresh frequency for your monitor. Refer to your monitor documentation for information about the maximum refresh frequencies. If the video adapter is unable to display the specified refresh frequency, then the default refresh frequency will be used. Refer to your monitor documentation for information about maximum refresh frequencies.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Status? (Set) specifies whether to retrieve (Get) or modify (Set) the video mode for RT Video Out. The default value is **True** (Set).
- Video Mode Out returns the current video settings for the LabVIEW RT target.
  - Screen Area is the screen resolution settings for the monitor connected to your LabVIEW RT target.
  - **Color Depth** is the number of bits used to represent a screen pixel.
  - Refresh Frequency is the refresh frequency for your monitor. Refer to your monitor documentation for information about maximum refresh frequencies.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

### **IMAQ Initialize Timed Execution VI**

Owning Palette: <u>Vision RT</u>Installed With: NI Vision Development Module

Reserves memory for the algorithms used during time-bounded execution and enables timing capabilities.



**Note** This VI is available only with LabVIEW 7.0 or later.



**Note** This VI is only supported on ETS targets.

#### **Examples**



- Reserved Memory Size (in KB) is the size of the reserved memory block used by algorithms.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

 $\bullet$  LabVIEW\examples\Vision\4. Real-Time\Time-bounded Particle Analysis.llb

## **IMAQ Start Timed Execution VI**

Owning Palette: <u>Vision RT</u>Installed With: NI Vision Development Module

Sets the time limit bounding the execution of Vision VIs. All Vision VIs executing while the time limit is set are subject to the time bound. Only one time limit is active at any time.



**Note** This VI is available only with LabVIEW 7.0 or later.



**Note** This VI is only supported on ETS targets.

### **Examples**



- Time Limit (ms) is the time limit in milliseconds.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

 $\bullet$  LabVIEW\examples\Vision\4. Real-Time\Time-bounded Particle Analysis.llb

### **IMAQ Stop Timed Execution VI**

Owning Palette: Vision RTInstalled With: NI Vision Development

Module

Turns off the time limit established by the <u>IMAQ Start Timed Execution</u> VI and returns the time elapsed since the start of the time limit.



**Note** This VI is available only with LabVIEW 7.0 or later.



**Note** This VI is only supported on ETS targets.

### **Examples**



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Execution Time (ms)** is the execution time that has elapsed since the setting of the time limit.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

 $\bullet$  LabVIEW\examples\Vision\4. Real-Time\Time-bounded Particle Analysis.llb

## **IMAQ Uninitialize Timed Execution VI**

Owning Palette: Vision RTInstalled With: NI Vision Development

Module

Frees the resources reserved by the <u>IMAQ Initialize Timed Execution</u> VI and disables timing capabilities.



**Note** This VI is available only with LabVIEW 7.0 or later.



**Note** This VI is only supported on ETS targets.

### **Examples**



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

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

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

 $\bullet$  LabVIEW\examples\Vision\4. Real-Time\Time-bounded Particle Analysis.llb

## **IMAQ Check Timed Execution VI**

Owning Palette: Vision RTInstalled With: NI Vision Development

Module

Returns a timeout error if the time limit is reached or exceeded.



Note This VI is available only with LabVIEW 7.0 or later.



**Note** This VI is only supported on ETS targets.

### **Examples**



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

 $\bullet$  LabVIEW\examples\Vision\4. Real-Time\Time-bounded Particle Analysis.llb

# **Image Processing**

Owning Palette: Vision and MotionInstalled With: NI Vision

Development Module

Use the Image Processing VIs to analyze, filter, and process images in NI Vision.

Subpalette	Description
Processing	Use the Processing VIs to process grayscale and binary images. Use these VIs to transform images using predefined or custom lookup tables, change the contrast information in an image, invert the values in an image, and segment the image.
<u>Filters</u>	Use the Filter VIs to filter your images to enhance the information in the image. Use these VIs to smooth your image, remove noise, and find edges in the image. You can use a predefined filter kernel or create custom filter kernels.
Morphology	Use the Morphology VIs to perform morphological operations on an image. Some of these VIs perform basic morphological operations, such as dilation and erosion, on grayscale and binary images. Other VIs improve the quality of binary images by filling holes in particles, removing particles that touch the border of an image, removing noisy particles, and removing unwanted particles based on different characteristics of the particle.
Analysis	Use the Analysis VIs to return information about the contents of grayscale and binary images as well as particles in the image. Use these VIs to compute the histogram information and grayscale statistics of an image, retrieve pixel information and statistics along any one-dimensional profile in an image, and detect and measure particles in binary images.
Color Processing	Use the Color Processing VIs to perform basic processing on color images. Use these VIs to compute the histogram of a color image; apply lookup tables to color images; change the brightness, contrast, and gamma information associated with a color image; and threshold a color image.

<u>Operators</u>	Use the Operators VIs to perform arithmetic and bit-wise operations in NI Vision. Use some of these VIs to add, subtract, multiply, and divide an image with other images or constants. Use other VIs in this palette to apply logical operations and make pixel comparisons between an image and other images or a constant.
Frequency Domain	Use the Frequency Domain VIs to perform frequency processing and other tasks on images. Use these VIs to convert an image from the spatial domain to the frequency domain using a two-dimensional Fast Fourier Transform (FFT) and convert an image from the frequency domain to the spatial domain using the inverse FFT. These VIs also extract the magnitude, phase, real, and imaginary planes of the complex image.

## **Processing**

Owning Palette: Image ProcessingInstalled With: NI Vision

**Development Module** 

Use the Processing VIs to process grayscale and binary images. Use these VIs to transform images using predefined or custom lookup tables, change the contrast information in an image, invert the values in an image, and segment the image.

Palette Object	Description
IMAQ UserLookup	Performs a user-specified lookup-table transformation by remapping the pixel values in an image.
IMAQ MathLookup	Converts the pixel values of an image by replacing them with values from a defined lookup table. This VI modifies the dynamic range of either part of an image or the complete image, depending on the type of curve chosen.
IMAQ Equalize	Produces a histogram equalization of an image. This VI redistributes the pixel values of an image to linearize the accumulated histogram. The precision of the VI is dependent on the histogram precision, which in turn is dependent on the number of classes used in the histogram.
IMAQ BCGLookup	Applies a brightness, contrast, and gamma correction to an image. The correction is performed by computing and applying a lookup table. Brightness, Contrast, and Gamma control the changes made to the transfer function represented by the lookup table.
<b>IMAQ Threshold</b>	Applies a threshold to an image.
IMAQ MultiThreshold	Performs thresholds of multiple intensity ranges to an image.
IMAQ Label	Labels the particles in a binary image.
IMAQ Watershed Transform	Computes the watershed transform on an image.
IMAQ Inverse	Inverts the pixel intensities of an image to compute the

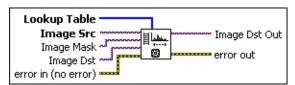
	negative of an image.
_	Computes the optimal values to threshold an image into a given number of classes.
IMAQ Local Threshold	Thresholds an image into a binary image based on the specified local adaptive thresholding method.
IMAQ MagicWand	Creates an image mask by extracting a region surrounding a reference pixel, called the origin, and using a tolerance of intensity variations based on this reference pixel.

## IMAQ UserLookup VI

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Performs a user-specified lookup-table transformation by remapping the pixel values in an image.





- Lookup Table is a grayscale replacement table. This input is an array containing a maximum of 256 elements if Image Src is an 8-bit image or a maximum of 65,536 elements if Image Src is a 16-bit image. Individual pixels within the image are not modified when the lookup table is missing a value that corresponds to those pixels.
- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

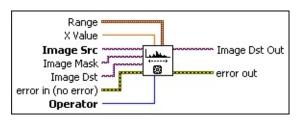
## IMAQ MathLookup VI

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Converts the pixel values of an image by replacing them with values from a defined lookup table. This VI modifies the dynamic range of either part of an image or the complete image, depending on the type of curve chosen.

#### **Examples**





Range is a cluster containing the minimum and maximum pixel values of the range to modify. If you do not connect this cluster or you keep the default values of (0,0), the VI modifies the dynamic range of the entire image. The dynamic range of the destination image is dependent on the type of input image. The dynamic range for an 8-bit image is between 0 and 255. The dynamic range for 16-bit and floating-point images is the smallest and largest pixel value contained in the original image before processing.



**Note** The VI does not modify the dynamic range for 16-bit and floating-point images. The VI changes only the distribution of the values.

- Minimum is the smallest value used for processing. After processing, all pixel values that are less than or equal to the Minimum in the original image are set to 0 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the smallest pixel value found in the original image.
- Maximum is the largest value used for processing. After processing, all pixel values that are greater than or equal to the Maximum in the original image are set to 255 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the largest pixel value found in the original image.

- X Value is a value used only for the operators Power X and Power 1/X.
- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Operator specifies the remapping procedure to use. The horizontal axis represents the pixel values before processing (between Minimum and Maximum), and the vertical axis represents the pixel values after processing. The default is Linear. You can choose from the following valid operators:



**Note** For an 8-bit image, the minimum is always 0 and the maximum is always 255. For 16-bit and floating-point images, the minimum and maximum are the endpoint values found in the image before processing.

Linear (0)	(Default) Linear remapping.
Log (1)	A logarithmic remapping operation that gives extended contrast for small pixel values and less contrast for large pixel values.
<b>Exp</b> (2)	An exponential remapping operation that gives extended contrast for large pixel values and less contrast for small pixel values.
Square (3)	Similar to exponential but with a more gradual effect.
Square Root (4)	Similar to logarithmic but with a more gradual effect.
Power X (5)	Gives variable effects depending on the value of $X$ . The default value of $X$ is 1.5.
Power 1/X (6)	Gives variable effects depending on the value of $X$ . The default value of $X$ is 1.5.

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Grayscale Processing\MathLookup Example.vi

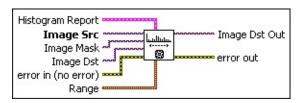
## **IMAQ Equalize VI**

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Produces a histogram equalization of an image. This VI redistributes the pixel values of an image to linearize the accumulated histogram. The precision of the VI is dependent on the histogram precision, which in turn is dependent on the number of classes used in the histogram.

#### **Details**





- Histogram Report is the histogram of an image that is used to compute the equalization. If this input is not connected, the VI computes the histogram from the source (input) image to equalize the image. The cluster contains the following elements:
  - histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The *n*th class contains all pixel values belonging to the interval [(Starting Value + (n ? 1) × Interval Width), (Starting Value + n × (Interval Width ? 1))].
  - Minimal Value returns the smallest pixel value used in calculating the histogram.
  - Maximal Value returns the largest pixel value used in calculating the histogram.
  - Starting Value returns the smallest pixel value from the first class calculated in the histogram. It can be equal to the Minimum value from the Interval Range or the smallest value found for the image type connected.
  - Interval Width returns the length of each class.
  - Mean Value returns the mean value of the pixels used in calculating the histogram.

**Standard Variation** returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.

- Area (pixels) returns the number of pixels used in the histogram calculation. This is influenced by the values specified in Interval Range and the contents of Image Mask.
- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Range is a cluster containing the minimum and maximum values

for the range of pixel values to equalize. If you do not connect this cluster or you keep the default values of (0,0), the VI considers **Minimal Value** and **Maximal Value** of **Histogram Report** to be the minimum and maximum values to equalize, and the VI equalizes the entire image.

- Minimum is the smallest value used for processing. After processing, all pixel values that are less than or equal to the Minimum in the original image are set to 0 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the smallest pixel value found in the original image.
- Maximum is the largest value used for processing. After processing, all pixel values that are greater than or equal to the Maximum in the original image are set to 255 for an 8-bit image. In 16-bit and floating-point images, these pixel values are set to the largest pixel value found in the original image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

Use the optional **Histogram Report** input to equalize an image based on its histogram. To equalize a source image based on the histogram of another image, run the <u>IMAQ Histogram</u> VI before the IMAQ Equalize VI to supply an input for **Histogram Report**.

The modification to the pixel value depends on the histogram contents, regardless of the image type used. All pixels entering into the same histogram class have an identical value after equalization.

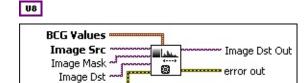
## IMAQ BCGLookup VI

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Applies a brightness, contrast, and gamma correction to an image. The correction is performed by computing and applying a lookup table. Brightness, Contrast, and Gamma control the changes made to the transfer function represented by the lookup table.

### **Examples**

error in (no error) 🕶



- **BCG Values** is a cluster of the brightness, contrast, and gamma values.
  - Brightness (default = 128) sets the brightness of the image. This value is used as the x intercept of the transfer function in the lookup table. The neutral value is 128 (no change in the image). A higher value returns a brighter image. A value less than 128 decreases the overall brightness of the image.
  - Contrast (default = 45.0) sets the contrast of the image. This control is used as the slope of the transfer function in the lookup table and is expressed in degrees. A slope of 45 degrees is neutral. A higher value returns a more contrasted image. A value smaller than 45 decreases the contrast of the image.
  - Gamma (default = 1.0) sets the gamma correction applied to the image. The neutral value is 1. A value greater than 1 gives extended contrast for small pixel values and less contrast for large pixel values. A value smaller than 1 returns an image with less contrast for small pixel values and extended contrast for large pixel values.
- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src

**(Small)** image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if **Image Mask** is not connected.

- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - **status** is TRUE (X) if an error occurred or FALSE

(checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Grayscale Processing\BCG Lookup Example.vi

## **IMAQ Threshold VI**

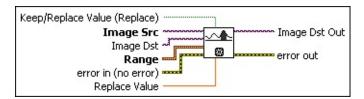
Owning Palette: ProcessingInstalled With: NI Vision Development

Module

Applies a threshold to an image.

### **Examples**

U8 I16 SGL



- Keep/Replace Value (Replace) determines whether to replace the value of the pixels existing in the range between Lower value and Upper value. The default status, TRUE, replaces these pixel values, and the status FALSE keeps the original values.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Range is a cluster specifying the threshold range.
  - **Lower value** is the lowest pixel value used during a threshold. The default is 128.
  - **Upper value** is the highest pixel value used during a threshold. The default is 255.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Replace Value is the value used to replace pixels between the Lower value and Upper value. This operation requires that Keep/Replace Value (Replace) is TRUE.
  - Note Use a binary palette when you plan to visualize an image to which a threshold has been applied in Replace mode. However, which palette to use for visualization depends on the value of **Replace Value**. For example, the visualization of a threshold image could be performed with a gray palette. However, in this case it is advised that you use a replacement value of 255 (white) to see the threshold image better.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Examples**

Refer to the following for an example that uses this VI.

- LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Analysis Report Example.vi
- LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Orientation Example.vi

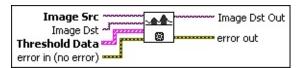
## IMAQ MultiThreshold VI

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Performs thresholds of multiple intensity ranges to an image.

#### **Details** Examples





- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Threshold Data is an array of clusters specifying the mode and threshold range. This operation is analogous to the process in the <a href="IMAQ Threshold">IMAQ Threshold</a> VI. Each cluster is composed of the following elements:
  - **Lower value** is the lowest pixel value to be taken into account during a threshold. The default is 128.
  - **Upper value** is the highest pixel value to be taken into account during a threshold. The default is 255.
  - Replace Value is the value used to replace pixels between the Lower value and Upper value. This operation requires that Keep/Replace Value (Replace) is TRUE.
  - Keep/Replace Value (Replace) determines whether to replace the value of the pixels existing in the range between Lower value and Upper value. The default status, TRUE, replaces these pixel values, and the status FALSE keeps the original values.

All pixels outside the range between **Lower value** and **Upper value** are set to 0. All values found between this range are replaced by the value entered in **Replace Value** if **Keep/Replace Value (Replace)** is TRUE.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The threshold operations are performed in reverse order that the data is received from **Threshold Data**. A pixel can be taken into account only once, even if the pixel is included in the threshold range of more than one threshold by **Threshold Data**.

For example, a VI contains two clusters on input:

Cluster Lower value = 80, Upper value = 150, Keep/Replace Value = 1 TRUE, Replace Value = 255.

Cluster Lower value = 120, Upper value = 200, Keep/Replace Value 2 = FALSE.

This example shows two threshold ranges with an overlap between 120 and 150. Therefore, the pixels between 120 and 150 are treated only by the second threshold. The following results occur after execution of this VI.

- Pixel values between 0 and 79 are replaced by 0.
- Pixel values between 80 and 119 are replaced by 255.
- Pixel values between 120 and 200 keep their original values.
- Pixel values greater than 200 are replaced by 0.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Circle Distance Example.vi

### **IMAQ Label VI**

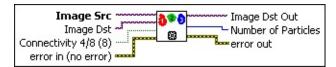
Owning Palette: ProcessingInstalled With: NI Vision Development

Module

Labels the particles in a binary image.

#### **Details** Examples

#### U8 I16 SGL



- Image Src is a reference to the source image.
- **Image Dst** is a reference to the destination image.
- Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Number of Particles indicates the number of particles detected in the image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

This operation applies a unique grayscale value to all pixels composing the same group of pixels (a particle). This value is encoded in 8 or 16 bits, depending on the image type. Therefore, 255 particles can be labeled in an 8-bit image and 65,535 particles in a 16-bit image.

If you want to label more than 255 particles in an 8-bit image, you can cast the image into a 16-bit image using the <u>IMAQ Cast Image</u> VI and then label the particles.

### **Examples**

Refer to the following for an example that uses this VI.

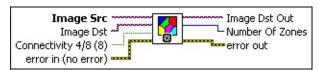
- LabVIEW\examples\Vision\2. Functions\Grayscale Processing\Label Example.vi
- LabVIEW\examples\Vision\2. Functions\Calibration\Nonlinear Calibration Example.llb

### **IMAQ Watershed Transform VI**

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Computes the watershed transform on an image. Refer to the *NI Vision Concepts Manual* for more information about watershed transform.





- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image. If Image Dst is connected, it must be the same size as the Image Src.
  - Note If Image Dst is an 8-bit image, the VI can store up to 255 unique labels not including the watershed line value of 0. If Image Dst is a 16-bit image, the VI can store up to 32,767 unique labels not including the watershed line value of 0.
- Connectivity 4/8 (8) specifies how the watershed transform algorithm determines whether an adjacent pixel belongs to the same, or different, catchment basin or watershed line. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Number Of Zones indicates the number of zones detected in the image. A zone is a region of the image in which all of the pixels belong to the same catchment basin.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

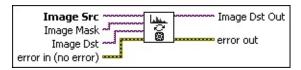
### IMAQ Inverse VI

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Inverts the pixel intensities of an image to compute the negative of an image.

#### <u>Details</u> <u>Examples</u>





- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

For 8-bit images, this VI performs an 8-bit inversion of the pixel intensities in the [0, 255] range. For 16-bit and floating-point image types, the inversion is performed within the minimum and maximum intensity values found in the image.

# **Examples**

Refer to the following for an example that uses this VI.

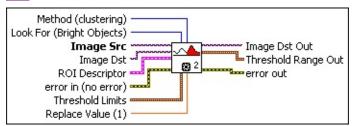
• LabVIEW\examples\Vision\2. Functions\Image Management\Unwrap Barcode Example.vi

## **IMAQ AutoBThreshold 2 VI**

Installed With: NI Vision Development Module

Computes the optimal threshold value for an image or a region of interest and applies the computed threshold.

U8



Look For (Bright Objects) indicates the type of objects for which you want to look. The following values are valid:

	Looks for objects in the image represented by pixels with values greater than the value computed by the threshold method.
• ,	Looks for objects in the image represented by pixels with values less than the value computed by the threshold method.

- ROI Descriptor is a descriptor that defines the Region of Interest (ROI) within which the threshold is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Method (clustering) specifies the threshold method to use.

#### Choose from the following options:



**Note** Refer to the *NI Vision Concepts Manual* for more information about these methods.

clustering (0)	Use the clustering method as a general-purpose threshold method.
entropy (1)	Use the entropy method to detect particles that are present in minuscule proportions on the image.
metric (2)	The metric method calculates an optimal threshold value using the surfaces representing the initial gray scale.
moments (3)	Use the moments method for images that have poor contrast.
inter-class variance (4)	The interclass variance method determines an optimal threshold by maximizing the between-class variation with respect to the threshold.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Replace Value (1) specifies the replacement value the VI uses for the pixels of the kept objects in the destination image.
- Threshold Limits specifies the upper and lower bounds of the threshold values used to calculate the threshold.
  - **Lower value** specifies the lower bound of the threshold range.
  - **Upper value** specifies the upper bound of the threshold range.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Threshold Range Out specifies the range used to threshold the image or ROI.
  - **Lower value** is the lowest pixel value to be taken into account during a threshold.
  - **Upper value** is the highest pixel value to be taken into account during a threshold.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

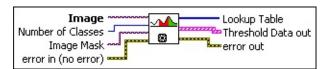
## IMAQ AutoMThreshold VI

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Computes the optimal values to threshold an image into a given number of classes. This algorithm uses a variant of the classification by clustering method. Starting from a random sort, a clustering algorithm is iterated until a stable and reliable result is found. Use the <a href="IMAQ MultiThreshold">IMAQ MultiThreshold</a> VI or the <a href="IMAQ UserLookup">IMAQ UserLookup</a> VI to threshold an image based on the optimal threshold range computed.

#### **Details**





- **Image** is a reference to the source image.
- Number of Classes is the number of desired phases. This algorithm uses a clustering method and can use any value between 2 and 256. The default value is 2.
- Image Mask is an 8-bit image specifying the region in the image to use for the calculation. Only those pixels in the original image that correspond to an equivalent non-zero pixel in the mask image are used for the calculation. The entire image is used in the calculation if Image Mask is not connected.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Lookup Table** is an array containing the values of the 256 transformed elements encoded between 0 and (n 1), where n is the **Number of Classes**. You can connect this array to the <u>IMAQ</u> <u>UserLookup</u> VI to perform a threshold operation directly.
- Threshold Data out outputs an array containing the Number of Classes compatible with the <u>IMAQ MultiThreshold</u> VI. The results range from 0 to (n-1), where n is the Number of Classes.
  - **Lower value** is the lowest pixel value to be taken into account during a threshold.
  - **Upper value** is the highest pixel value to be taken into account during a threshold.
  - Replace Value is the value used to replace pixels between the Lower value and Upper value. This operation requires that Keep/Replace Value (Replace) is TRUE.
  - Keep/Replace Value (Replace) determines whether to replace the value of the pixels existing in the range between Lower value and Upper value. The default TRUE replaces these pixel values and the status FALSE keeps the original values.

All pixels outside the range between **Lower value** and **Upper value** are set to 0. All values found between this range are replaced by the value entered in **Replace Value** if **Keep/Replace Value (Replace)** is TRUE.

error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

This method is based on an iterated measurement of an histogram. After finding the best result, the histogram is segmented into n groups. These groups are based on the fact that each point in a group is closer to the barycenter of its own group than the other group. The VI outputs the threshold data in two forms:

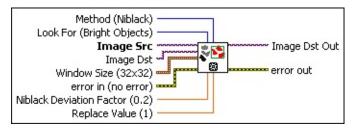
- A LUT directly usable by the **IMAQ UserLookup** VI
- An array directly usable by the <u>IMAQ MultiThreshold</u> VI (Threshold Data)

## **IMAQ Local Threshold VI**

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Thresholds an image into a binary image based on the specified local adaptive thresholding method.





Method (Niblack) indicates the local thresholding algorithm the VI uses. The following values are valid:

Niblack (0)	Computes thresholds for each pixel based on its local statistics using the Niblack local thresholding algorithm.
Correction (1)	Performs background correction to eliminate non- uniform lighting effects and then performs thresholding using the interclass variance thresholding algorithm.

Look For (Bright Objects) indicates the type of objects for which you want to look. The following values are valid:

	Looks for objects in the image represented by pixels with values greater than the value computed by the threshold method.
• • • • • • • • • • • • • • • • • • • •	Looks for objects in the image represented by pixels with values less than the value computed by the threshold method.

- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Window Size (32x32) is a cluster specifying the size of the window the VI uses when calculating a local threshold. The window should

be sized as large as possible but small enough that each window contains pixels with consistent levels of contrast. The VI will produce inconsistent results for windows that contain uniform pixel values. A typical window should be about the size of the object you want to segment in the image.

- SizeX is the size of the window in the x dimension. The size must be at least 3 and cannot be larger than the width of Image Src.
- SizeY is the size of the window in the *y* dimension. The size must be at least 3 and cannot be larger than the height of **Image Src**.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Niblack Deviation Factor (0.2) specifies the *k* constant used in the Niblack local thresholding algorithm, which determines the weight applied to the variance calculation. Valid *k* constants range from 0 to 1. The lower the Deviation Factor, the closer the pixel value must be to the mean value to be selected as part of a particle. Setting the

- **Niblack Deviation Factor** to 0 will increase the performance of the VI because the VI will not calculate the variance for any of the pixels. The VI ignores this value if **Method** is not set to Niblack.
- Replace Value (1) specifies the replacement value the VI uses for the pixels of the kept objects in the destination image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

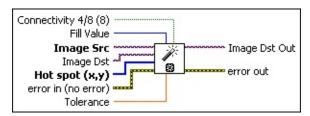
### **IMAQ MagicWand VI**

Owning Palette: ProcessingInstalled With: NI Vision Development Module

Creates an image mask by extracting a region surrounding a reference pixel, called the origin, and using a tolerance of intensity variations based on this reference pixel. Using this origin, the VI searches for its neighbors with an intensity that equals or falls within the tolerance value of the point of reference. The resulting image is binary. The image passed as input for **Image Dst** must be an 8-bit image. If the same image is entered for **Image Src** and **Image Dst**, the image must be an 8-bit image.

#### **Examples**





**Connectivity 4/8 (8)** specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

- Fill Value is the value used for the highlighted pixels in the destination image. The default is 1.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Hot spot (x,y) is an array containing the (x, y) coordinates of the origin pixel chosen from the image source.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to

**error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Tolerance** is the maximum allowed deviation from the origin. All pixels satisfying the tolerance criteria (between origin pixel tolerance and origin pixel +tolerance) and connectivity criteria (as specified in **Connectivity 4/8 (8)**), are highlighted. All other pixels are turned off. The default is 20.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a

warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Grayscale Processing\MagicWand Example.vi

### **Filters**

Owning Palette: <a href="mage-ProcessingInstalled-With">Image ProcessingInstalled With</a>: NI Vision

**Development Module** 

Use the Filter VIs to filter your images to enhance the information in the image. Use these VIs to smooth your image, remove noise, and find edges in the image. You can use a predefined filter kernel or create custom filter kernels.

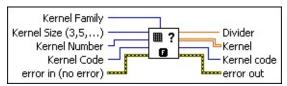
Palette Object	Description
IMAQ GetKernel	Reads a predefined kernel.
IMAQ BuildKernel	Constructs a convolution matrix by converting a string. This string can represent either integers or floating-point values.
IMAQ Convolute	Filters an image using a linear filter. The calculations are performed with either integers or floating points, depending on the image type and the contents of the kernel.
IMAQ Correlate	Computes the normalized cross correlation between the source image and the template image.
IMAQ LowPass	Calculates the inter-pixel variation between the pixel being processed and those pixels surrounding it. If the pixel being processed has a variation greater than a specified percentage, it is set to the average pixel value as calculated from the neighboring pixels.
IMAQ NthOrder	Orders, or classifies, the pixel values surrounding the pixel being processed. The data is placed into an array and the pixel being processed is set to the nth pixel value, the nth pixel being the ordered number.
IMAQ EdgeDetection	Extracts the contours (detects edges) in gray-level values.
IMAQ CannyEdgeDetection	Uses a specialized edge detection method to accurately estimate the location of edges even under conditions of poor signal-to-noise ratios.

### IMAQ GetKernel VI

Owning Palette: Filters Installed With: NI Vision Development Module

Reads a predefined kernel. This code consists of three separate units: **Kernel Family**, **Kernel Size**, and **Kernel Number**. If you already know the code, you can enter it directly with **Kernel Code**.

#### **Details** Examples



Kernel Family determines the type of matrix. This value corresponds to the thousandth unit in the researched code. The matrix types are as follows:

<b>Gradient</b> (1)	Specifies the kernel family as gradient
Laplacian (2)	Specifies the kernel family as Laplacian
Smoothing (3)	Specifies the kernel family as smoothing
Gaussian (4)	Specifies the kernel family as Gaussian

- Kernel Size (3,5,...) determines the horizontal and vertical matrix size. The values are 3, 5, and 7, corresponding to the convolutions  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$  supplied in the matrix catalog. This value corresponds to the hundredth unit in the researched code.
- Kernel Number is the matrix family number. It is a two-digit number, between 0 and n, belonging to a family and a size. A number of predefined matrices are available for each type and size.
- Kernel Code is a code you can use to directly access a convolution matrix. Each code specifies a specific convolution matrix. You can use this input if it is connected and is not 0. The kernel located in the file then is transcribed into a 2D array that is available from the output Kernel. You can use the codes to specify a predefined kernel.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Divider** is the normalization factor associated with the retrieved kernel.
- Kernel is the resulting matrix. It corresponds to a kernel encoded by a code specified from the inputs Kernel Family, Kernel Size, and Kernel Number or a from a code directly passed through the input Kernel Code. You can connect this output directly to the input Kernel in the IMAO Convolute VI.
- **Kernel code** indicates the code that was used to retrieve the kernel.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

For the kernel code 1300, the kernel family is gradient, the kernel size is  $3 \times 3$ , and the kernel number (nn) is 00. The matrix is as follows:

- -1 0 1
- -1 0 1
- -1 0 1

Refer to Kernels for a list of predefined kernels used in NI Vision.

# **Examples**

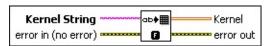
Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Analysis Example.vi

# IMAQ BuildKernel VI

**Owning Palette:** Filters Installed With: NI Vision Development Module Constructs a convolution matrix by converting a string. This string can represent either integers or floating-point values.

#### **Details**



- **Kernel String** is a string listing the coefficients that form the matrix.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Kernel** is the resulting matrix converted from the input string. You can connect this output directly to the input **Kernel** in the <u>IMAQ</u> Convolute VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

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

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

The column separator can be either a comma, a semicolon, or a blank space. The line separator is a linefeed character. For example, the string  $1\,1\,1\,1\,1\,1\,1\,1$  produces a  $3\times 3$  matrix with all coefficients set to 1 as follows:

- 1 1 1
- 1 1 1
- 1 1 1

### **IMAQ Convolute VI**

Owning Palette: Filters Installed With: NI Vision Development Module

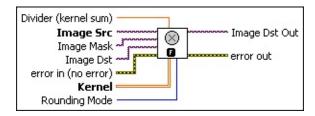
Filters an image using a linear filter. The calculations are performed with either integers or floating points, depending on the image type and the contents of the kernel.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> before using this VI.

### **Details Examples**





- Divider (kernel sum) is a normalization factor that can be applied to the sum of the obtained products. Under normal conditions the divider should not be connected. If connected and not equal to 0, the elements internal to the matrix are summed and then divided by this normalization factor.
- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Kernel is a 2D array that contains the convolution matrix to apply to the image. The size of the convolution is fixed by the size of this array. The array can be generated by standard LabVIEW programming techniques or the <a href="MAQ GetKernel">IMAQ GetKernel</a> VI or the <a href="IMAQ BuildKernel">IMAQ BuildKernel</a> VI. If the kernel contains fewer than three rows or three columns, no convolution is performed.
- Rounding Mode specifies the type of rounding to use when dividing image pixels. The following options are available.

• ,	Rounds the result of a division using the best available method.
Truncate (1)	Truncates the result of a division.

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

#### **Details**

The connected source image must have been created with a border capable of supporting the size of the convolution matrix. A  $3 \times 3$  matrix must have a minimum border of 1, a  $5 \times 5$  matrix must have a minimum border of 2, and so on. The border size of the destination image is not important.

A convolution matrix must have odd-sized dimensions so that it contains a central pixel. If one of the **Kernel** dimensions is even, the function does not take into account the odd boundary, farthest out on the matrix. For example, if the input **Kernel** is  $6 \times 4$  (X = 6 and Y = 4), the actual convolution is  $5 \times 3$ . Both the sixth line and the fourth line are ignored.

Calculations using an 8-bit or 16-bit **Image Src** input are made in integer mode provided that the kernel contains only integers. Calculations using a 32-bit floating-point **Image Src** input are made in floating-point mode.

The processing speed corresponds to the size of the kernel. A  $3 \times 3$  convolution processes 9 pixels and a  $5 \times 5$  convolution processes 25 pixels.

# **Examples**

Refer to the following for an example that uses this VI.

• Examples\Vision\2. Functions\Binary Analysis\Particle Analysis Example.vi

### **IMAQ Correlate VI**

Owning Palette: <u>Filters</u>Installed With: NI Vision Development Module Computes the normalized cross correlation between the source image and the template image.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.



- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Image Src is a reference to the source image.
- Image Template is a reference to the template. This image must be an 8-bit image. The center of the template is used as the origin for the correlation.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is the reference to the destination. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src. IMAQ Correlate is an 8-bit image that contains the cross-correlation values normalized to lie in the range [0, 255]. A value of 255 indicates a very high correlation and a value of 0 indicates no correlation.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ LowPass VI**

Owning Palette: Filters Installed With: NI Vision Development Module

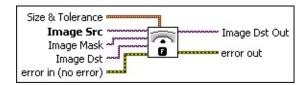
Calculates the inter-pixel variation between the pixel being processed and those pixels surrounding it. If the pixel being processed has a variation greater than a specified percentage, it is set to the average pixel value as calculated from the neighboring pixels.



**Note** Refer to the *NI Vision Concepts Manual* for more information about this filter.

#### **Details**





- Size & Tolerance is a cluster that specifies the X Size, Y Size, and % Tolerance variables.
  - X Size is the size of the horizontal matrix axis. The default is 3.
  - Y Size is the size of the vertical matrix axis. The default is 3.
  - **Tolerance** is the maximum variation allowed. The default is 40%.
- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

The connected source image must have been created with a border capable of supporting the size of the convolution matrix. A  $3 \times 3$  matrix must have a minimum border of 1, a  $5 \times 5$  matrix must have a minimum border of 2, and so on. The border size of the destination image is not important.

### **IMAQ NthOrder VI**

Owning Palette: Filters Installed With: NI Vision Development Module

Orders, or classifies, the pixel values surrounding the pixel being processed. The data is placed into an array and the pixel being processed is set to the nth pixel value, the nth pixel being the ordered number.



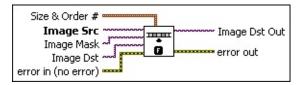
**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.



**Note** Refer to the *NI Vision Concepts Manual* for more information about this filter.

#### **Details**





- Size & Order # is a cluster that specifies the X Size, Y Size, and Order # variables. The default values correspond to a 3 × 3 median filter.
  - X Size is the size of the horizontal matrix axis. The default is 3.
  - Y Size is the size of the vertical matrix axis. The default is 3.
  - Order # is the order number chosen after classing the values. The default is 4.
- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The connected source image must have been created with a border capable of supporting the size of the convolution matrix. A  $3 \times 3$  matrix must have a minimum border of 1, a  $5 \times 5$  matrix must have a minimum border of 2, and so on. The border size of the destination image is not important.

The default for this VI is a  $3 \times 3$  median operation with X = 3, Y = 3, and **Order** = 4. To change to a  $5 \times 5$  median operation, the cluster must take the values X = 5, Y = 5, and **Order** = 12. In this last example, the order number is determined by calculating the central pixel number in the array. For a  $5 \times 5$  convolution, **Order** = 12 (the 13th pixel) because that pixel is the center pixel number for a 2D array of 25 pixels.

A lighter image results when using a higher-order number (such as 7 in a  $3 \times 3$  matrix). Darker images result when using a lower-order number (such as 1 in a  $3 \times 3$  matrix).

A median (center-pixel) operation is advantageous because it standardizes the gray-level values without significantly modifying the form of the objects or the overall brightness in the image. You can use this VI to apply a median filter by selecting the correct order,  $(f^2-1)/2$ , where f is the size of the convolution matrix.

If the order value that is entered is 0, the image obtained is representative of the local minimum from the source image. This operation is equivalent to a gray morphology erosion.

If the order value that is passed is equal to the following equation, the obtained image is representative of the local maximum from the source image. This operation is equivalent to a gray morphology dilation.

# **IMAQ EdgeDetection VI**

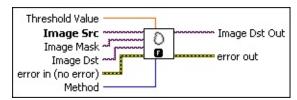
Owning Palette: Filters Installed With: NI Vision Development Module

Extracts the contours (detects edges) in gray-level values. Any image connected to the input **Image Dst** must be the same image type connected to **Image Src**. The image type connected to the input **Image Mask** must be an 8-bit image. The connected source image must have been created with a border capable of supporting the size of the processing matrix. For example, a  $3 \times 3$  matrix has a minimum border size of 1. The border size of the destination image is not important.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.





- Threshold Value is the minimum pixel value to appear in the resulting image. It is rare to use a value greater than 0 for this type of processing because the results from this processing are usually very dark and are not very dynamic. The default is 0.
- **Image Src** is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or

function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Method specifies the type of edge-detection filter to use. The default is **Differentiation**. The following filters are valid:
  - D

**Note** Refer to the *NI Vision Concepts Manual* for more information about these filters.

<b>Differentiation</b> (0)	Processing with a $2 \times 2$ matrix
<b>Gradient</b> (1)	Processing with a $2 \times 2$ matrix
Prewitt (2)	Processing with a $3 \times 3$ matrix
Roberts (3)	Processing with a $2 \times 2$ matrix
Sigma (4)	Processing with a $3 \times 3$ matrix
Sobel (5)	Processing with a 3 × 3 matrix

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for

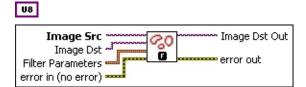
more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ CannyEdgeDetection VI

Owning Palette: Filters Installed With: NI Vision Development Module

Uses a specialized edge detection method to accurately estimate the location of edges even under conditions of poor signal-to-noise ratios.



- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- **Filter Parameters** is a cluster containing the following filter parameters:
  - Sigma is the sigma of the Gaussian smoothing filter that the VI applies to the image before performing the edge detection.
  - HThresh (High Threshold) defines the upper percentage of pixel values in the image from which the edge detection algorithm chooses the seed or starting point of an edge segment. You can choose values from 0 to 1.
  - LThres (Low Threshold) is multiplied by the HThresh value to define a lower threshold for all the pixels in an edge segment.
  - WindowSize defines the size of the Gaussian filter that the VI applies to the image. The size must be an odd number.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# Morphology

Owning Palette: Image ProcessingInstalled With: NI Vision

**Development Module** 

Use the Morphology VIs to perform morphological operations on an image. Some of these VIs perform basic morphological operations, such as dilation and erosion, on grayscale and binary images. Other VIs improve the quality of binary images by filling holes in particles, removing particles that touch the border of an image, removing noisy particles, and removing unwanted particles based on different characteristics of the particle.

Palette Object	Description
IMAQ Distance	Encodes a pixel value of a particle as a function of the location of that pixel in relation to the distance to the border of the particle. The source image must have a border size of at least 1 pixel and must be an 8-bit binary image. All distances greater than 255 are rounded down to 255.
IMAQ FillHole	Fills the holes found in a particle. The holes are filled with a pixel value of 1. The source image must be an 8-bit binary image. The holes found in contact with the image border are never filled because it is impossible to determine whether these holes are part of a particle.
IMAQ RejectBorder	Eliminates particles that touch the border of an image. The source image must be an 8-bit binary image.
IMAQ RemoveParticle	Eliminates or keeps particles resistant to a specified number of 3 x 3 erosions.
IMAQ Morphology	Performs primary morphological transformations.
IMAQ GrayMorphology	Performs grayscale morphological transformations.
IMAQ Segmentation	Starting from a labeled image, calculates the zones of influence between particles.
IMAQ Danielsson	Returns a distance map based on the algorithms of Danielsson.

IMAQ Find Circles	Separates overlapping circular particles and classifies them based on their radius, surface area, and perimeter.
IMAQ Convex Hull	Draws the convex hull for each particle in the image.
IMAQ Skeleton	Starting from a binary image, calculates the skeletons of the particles within an image or the lines delineating the zones of influence of the objects (skeleton of an inverse image). The source image must have a border greater than or equal to 1.
IMAQ Separation	Separates touching particles, particularly small isthmuses found between particles.
IMAQ Particle Filter 3	Filters (keeps or removes) each particle in an image according to its measurements.

## **IMAQ Distance VI**

Owning Palette: MorphologyInstalled With: NI Vision Development Module

Encodes a pixel value of a particle as a function of the location of that pixel in relation to the distance to the border of the particle. The source image must have a border size of at least 1 pixel and must be an 8-bit binary image. All distances greater than 255 are rounded down to 255.



**Note** In most cases National Instruments recommends using the <u>IMAQ Danielsson</u> VI instead of IMAQ Distance because it produces more accurate results.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.





- Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

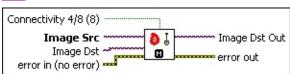
## **IMAQ FillHole VI**

Owning Palette: MorphologyInstalled With: NI Vision Development Module

Fills the holes found in a particle. The holes are filled with a pixel value of 1. The source image must be an 8-bit binary image. The holes found in contact with the image border are never filled because it is impossible to determine whether these holes are part of a particle.

#### **Examples**





Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The

default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Analysis Example.vi

# IMAQ RejectBorder VI

Owning Palette: MorphologyInstalled With: NI Vision Development Module

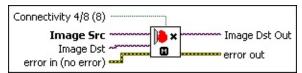
Eliminates particles that touch the border of an image. The source image must be an 8-bit binary image.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.

#### **Examples**





- Connectivity 4/8 (8) specifies how the algorithm determines whether an adjacent pixel belongs to the same or to a different particle. The default is 8.
- **Image Src** is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a

warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Analysis Example.vi

## **IMAQ RemoveParticle VI**

Owning Palette: MorphologyInstalled With: NI Vision Development Module

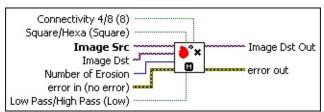
Eliminates or keeps particles resistant to a specified number of  $3 \times 3$  erosions. The particles that are kept are exactly the same shape as those found in the original source image. The source image must be an 8-bit binary image.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.

#### **Examples**

#### U8



Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

- Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Number of Erosion specifies the number of 3 x 3 erosions to apply to the image. The default is 2.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Low Pass/High Pass (Low) specifies whether the objects resistant to n erosions are discarded or kept (default).
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

• LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Analysis Example.vi

# **IMAQ Morphology VI**

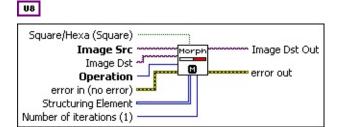
Owning Palette: MorphologyInstalled With: NI Vision Development Module

Performs primary morphological transformations. All source and destination images must be 8-bit binary images. The connected source image for a morphological transformation must have been created with a border capable of supporting the size of the structuring element. A  $3\times 3$  structuring element requires a minimal border of 1, a  $5\times 5$  structuring element requires a minimal border of 2, and so on. The border size of the destination image is not important.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.

#### **Details** Examples



- Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Operation specifies the type of morphological transformation procedure to use. The default is **AutoM**. You can choose from the following values:

AutoM (0)	Auto median
Close (1)	Dilation followed by an erosion
Dilate (2)	Dilation (the opposite of an erosion)
Erode (3)	Erosion that eliminates isolated background pixels
Gradient (4)	Extraction of internal and external contours of a

	particle
Gradient out (5)	Extraction of exterior contours of a particle
Gradient in (6)	Extraction of interior contours of a particle
Hit Miss (7)	Elimination of all pixels that do not have the same pattern as found in the structuring element
<b>Open</b> (8)	Erosion followed by a dilation
PClose (9)	A succession of seven closings and openings
<b>POpen</b> (10)	A succession of seven openings and closings
Thick (11)	Activation of all pixels matching the pattern in the structuring element
Thin (12)	Activation of all pixels matching the pattern in the structuring element

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **Structuring Element** is a 2D array that contains the structuring element to apply to the image. The size of the structuring element (the size of this array) determines the processing size. A structuring element of  $3 \times 3$  is used if this input is not connected.
- Number of iterations (1) is the number of times the VI performs a dilate or erode operation.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

A structuring element must have odd-sized dimensions so that it contains a central pixel. If one of the dimensions for the structuring element is even, the function does not take into account the odd boundary, farthest out on the matrix. For example, if the input structuring element is  $6 \times 4$  (X = 6 and Y = 4), the actual processing is performed at  $5 \times 3$ . Both the sixth column and the fourth row are ignored. The processing speed is correlated with the size of the structuring element. For example, a  $3 \times 3$  structuring element processes nine pixels, and a  $5 \times 5$  structuring element processes 25 pixels.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Grayscale Processing\Label Example.vi

# IMAQ GrayMorphology VI

Owning Palette: MorphologyInstalled With: NI Vision Development Module

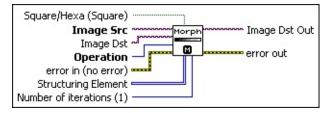
Performs grayscale morphological transformations. All source and destination image types must be the same. The connected source image for a morphological transformation must have been created with a border capable of supporting the size of the structuring element. A  $3\times 3$  structuring element requires a minimal border of 1, a  $5\times 5$  structuring element requires a minimal border of 2, and so on. The border size of the destination image is not important.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.

#### **Details**





- Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Operation specifies the type of morphological transformation procedure to use. The default is **AutoM**. You can choose from the following values:

AutoM (0)	Auto median
Close (1)	Dilation followed by an erosion
Dilate (2)	Dilation (the opposite of an erosion)
Erode (3)	Erosion that eliminates isolated background pixels

Open (4)	Erosion followed by a dilation
PClose (5)	A succession of seven closings and openings
POpen (6)	A succession of seven openings and closings

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Structuring Element is a 2D array that contains the structuring element to apply to the image. The size of the structuring element (the size of this array) determines the processing size. A structuring element of 3 × 3 is used if this input is not connected.
- Number of iterations (1) is the number of times the VI performs a dilate or erode operation.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

A structuring element must have odd-sized dimensions so that it contains a central pixel. If one of the dimensions for the structuring element is even, the function does not take into account the odd boundary, farthest out on the matrix. For example, if the input structuring element is  $6 \times 4$  (X = 6 and Y = 4), the actual processing is performed at  $5 \times 3$ . Both the sixth line and the fourth row are ignored. The processing speed is correlated with the size of the structuring element. For example, a  $3 \times 3$  structuring element processes nine pixels, and a  $5 \times 5$  structuring element processes 25 pixels.

# **IMAQ Segmentation VI**

Owning Palette: MorphologyInstalled With: NI Vision Development Module

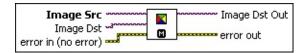
Starting from a labeled image, calculates the zones of influence between particles. Each labeled particle is dilated (grows) until the particles reach their neighbors, at which time this growth is stopped. The source image must have a border greater than or equal to 1.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.



**Note** Use the <u>IMAQ Label</u> VI to label the image prior to using IMAQ Segmentation.



- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Danielsson VI**

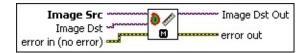
Owning Palette: MorphologyInstalled With: NI Vision Development Module

Returns a distance map based on the algorithms of Danielsson. The Danielsson distance map produces images and data that are similar to the <u>IMAQ Distance</u> VI but are more accurate. In most cases, National Instruments recommends using this function instead of the IMAQ Distance VI. All distances greater than 255 are rounded down to 255.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.





- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

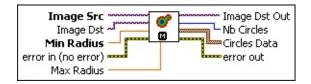
### **IMAQ Find Circles VI**

Owning Palette: MorphologyInstalled With: NI Vision Development Module

Separates overlapping circular particles and classifies them based on their radius, surface area, and perimeter. Starting from a binary image, it finds the radius and center of the circular particles even when multiple circular particles overlap. In addition, this VI can trace the circles in the destination image. It constructs and uses a Danielsson distance map to determine the radius of each particle.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.



- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Min Radius specifies the smallest radius, in pixels, to detect. The default is 1.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Max Radius specifies the largest radius, in pixels, to detect. The default is 10.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Nb Circles returns the number of detected circles in the image.
- Circles Data returns an array of measurements for all detected circles. Each element in the array has a structure containing the following elements.
  - **Pos. X** is the horizontal position, in pixels, of the center of the circle.
  - **Pos. Y** is the vertical position, in pixels, of the center of the circle.
  - **Radius** is the radius of the circle in pixels.
  - **Core Area** is the surface area, in pixels, of the nucleus of the circle as defined by the Danielsson distance map.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Convex Hull VI**

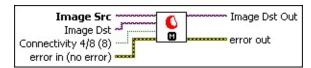
Owning Palette: MorphologyInstalled With: NI Vision Development Module

Draws the convex hull for each particle in the image.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.





- **Image Src** is a reference to the source image.
- **Image Dst** is a reference to the destination image.



**Note** Image Dst may contain a new, nonconvex particle if the image source contains a particle that, when made convex, overlaps another particle.

Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

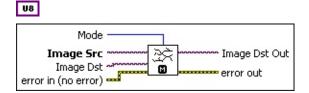
### **IMAQ Skeleton VI**

Owning Palette: MorphologyInstalled With: NI Vision Development Module

Starting from a binary image, calculates the skeletons of the particles within an image or the lines delineating the zones of influence of the objects (skeleton of an inverse image). The source image must have a border greater than or equal to 1.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.



Mode specifies the type of skeleton to perform. The default is **Skeleton L**.

**Skeleton L** uses this type structuring element:

0	<b>?</b> ·	1
0	1	1
0	?	1

**Skeleton M** uses this type structuring element:

?	?	1
0	1	1
?	?	1

**Skiz** is an inverse skeleton (**Skeleton L** on an inverse image).

- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use <u>error in</u> and <u>error out</u> to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

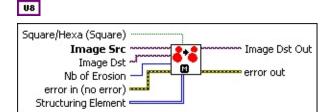
# **IMAQ Separation VI**

Owning Palette: MorphologyInstalled With: NI Vision Development Module

Separates touching particles, particularly small isthmuses found between particles. This VI performs *n* erosions where *n* equals **Nb of Erosion** and then reconstructs the final image based on the results of the erosion. If an existing isthmus is broken or removed during the erosion process, the particles are reconstructed without the isthmus. The reconstructed particles, however, have the same size as the initial particles except that they are separated. If no isthmus is broken during the erosion process, the particles are reconstructed as they were initially found. No change is made. The source image must be an 8-bit binary image and have a border greater than or equal to 1pixel.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.



- Square/Hexa (Square) specifies whether to treat the pixel frame as square or hexagonal during the transformation. The default is square.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Nb of Erosion specifies the number of erosions that are used to separate the particles. The default is 1.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Structuring Element is a 2D array that contains the structuring element to apply to the image. The size of the structuring element (the size of this array) determines the processing size. A structuring element of 3 × 3 is used if this input is not connected.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Particle Filter 3 VI**

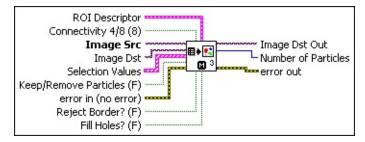
Owning Palette: MorphologyInstalled With: NI Vision Development Module

Filters (keeps or removes) each particle in an image according to its measurements.



**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.





- ROI Descriptor is a descriptor that defines the Region of Interest (ROI) within which the particle filter is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8

- 4 (FALSE) Particle detection is performed in connectivity mode 4
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Selection Values controls the criteria used to filter the particle in the image. This control is made of an array of clusters composed of the following values:
  - **Measurement Parameter** is the measurement on which you want to filter.
  - Range Lower Value specifies the lower value of the range for the chosen parameter.
  - Range Upper Value specifies the upper value of the range for the chosen parameter.
  - Range specifies whether to include or exclude the values given in Range Lower Value and Range Upper Value.

When Range is Include (FALSE), the particle meets the criteria if **Range Lower Value** ≤ particle measurement < **Range Upper Value**.

When Range is Exclude (TRUE), the particle meets the criteria if **Range Upper Value** ≤ particle measurement or particle measurement < **Range Lower Value**.

- Measurement Type specifies whether the measurement is in uncalibrated pixels or calibrated real-world values.
- Keep/Remove Particles (F) controls whether particles that meet any of the criteria specified in Selection Values are removed. When Keep/Remove Particles is TRUE, particles meeting any of the criteria are removed. If FALSE, only particles meeting any of the criteria remain.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Reject Border? (F)**, when TRUE, rejects border particles. The default value is FALSE.
- **Fill Holes? (F)**, when TRUE, fills holes in the binary object. The default value is FALSE. If **Fill Holes?** is TRUE and **Reject Border?** is TRUE, then holes in the binary object are filled after border particles are rejected.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Number of Particles indicates the number of particles detected in the image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Analysis**

Owning Palette: <a href="mage-ProcessingInstalled-With">Image ProcessingInstalled With</a>: NI Vision

**Development Module** 

Use the Analysis VIs to return information about the contents of grayscale and binary images as well as particles in the image. Use these VIs to compute the histogram information and grayscale statistics of an image, retrieve pixel information and statistics along any one-dimensional profile in an image, and detect and measure particles in binary images.

Palette Object	Description
IMAQ Histograph	Calculates the histogram from an image. This VI returns a data type (cluster) compatible with a LabVIEW graph.
IMAQ Histogram	Calculates the histogram of an image.
IMAQ Quantify	Quantifies the contents of an image or the regions within an image. The region definition is performed with a labeled image mask. Each region of the mask has a single unique value.
IMAQ Centroid	Computes the center of the energy of an image or of a portion of an image.
IMAQ LineProfile	Calculates the profile of a line of pixels. This VI returns a data type (cluster) compatible with a LabVIEW graph. The relevant pixel information is taken from the specified vector (line).
IMAQ ROIProfile	Calculates the profile of the pixels along the boundary of an ROI descriptor. This VI returns a data type (cluster) that is compatible with a LabVIEW graph. This VI also returns other information such as pixel statistics and the coordinates of the pixels along the ROI boundary.
IMAQ LinearAverages	Computes the average pixel intensity (mean line profile) on the whole or part of the image.
IMAQ Extract Curves	Finds curves in an image.
IMAQ Particle	Returns the number of particles detected in a binary

	image and an array of reports containing the most commonly used particle measurements.
<u>Analysis</u>	Returns the number of particles detected in a binary image and a 2D array of requested measurements about the particle.

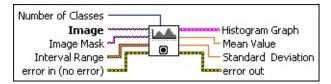
# **IMAQ Histograph VI**

Owning Palette: Analysis Installed With: NI Vision Development Module

Calculates the histogram from an image. This VI returns a data type (cluster) compatible with a LabVIEW graph.

#### **Examples**

U8 I16 SGL



- Number of Classes specifies the number of classes used to classify the pixels. The number of obtained classes differs from the specified amount in a case in which the minimum and maximum boundaries are overshot in the Interval Range. It is advised to specify a number of classes that is a power of two (for example, 2, 4, or 8) for 8-bit or 16-bit images. The default value is 256, which is designed for 8-bit images. This value gives a uniform class distribution or one class for each grayscale intensity in an 8-bit image.
- Image is a reference to the source image.
- Image Mask is an 8-bit image specifying the region in the image to use for the calculation. Only those pixels in the original image that correspond to an equivalent non-zero pixel in the mask image are used for the calculation. The entire image is used in the calculation if Image Mask is not connected.
- Interval Range is a cluster specifying the minimum and maximum boundaries for the histogram calculation. Only those pixels having a value that falls in this range are taken into account by the histogram calculation. This cluster is composed of the following elements:
  - Minimum is the minimum interval value. The default value of (0, 0) ensures that the real minimum value is determined by the source image, as described in the following table.

Image Type	Minimum Value Used
U8	0

116	Minimum pixel value found in the image
SGL	Minimum pixel value found in the image

Maximum is the maximum interval value. The default value of (0, 0) ensures that the real maximum value is determined by the source image, as described in the following table.

Image Type	Maximum Value Used
U8	0
116	Maximum pixel value found in the image
SGL	Maximum pixel value found in the image

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Histogram Graph is a cluster that returns the histogram values. This cluster contains the following elements:
  - Starting Value returns the smallest pixel value from the first class calculated in the histogram. It can be equal to the Minimum value from the Interval Range or the smallest

value found for the image type connected.

- Incremental Value returns the incrementing value that specifies how much to add to **Starting Value** in calculating the median value of each class from the histogram. The median value  $x_n$  from the nth class is expressed as follows:  $x_n = Starting Value + n \times Incremental Value$ .
- Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [(Starting Value +  $(n-1) \times Interval \ Width$ ), (Starting Value +  $n \times (Interval \ Width 1)$ )].
- Mean Value returns the mean value of the pixels used in calculating the histogram.
- Standard Deviation returns the standard deviation from the histogram. The higher this value, the better the distribution of the values in the histogram and the image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

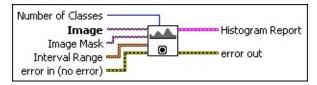
Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Analysis\Histogram Example.vi

# **IMAQ Histogram VI**

Owning Palette: <u>Analysis</u>Installed With: NI Vision Development Module Calculates the histogram of an image.

U8 I16 SGL



- Number of Classes specifies the number of classes used to classify the pixels. The number of obtained classes differs from the specified amount in a case in which the minimum and maximum boundaries are overshot in the Interval Range. It is advised to specify a number of classes that is a power of two (for example, 2, 4, or 8) for 8-bit or 16-bit images. The default value is 256, which is designed for 8-bit images. This value gives a uniform class distribution or one class for each grayscale intensity in an 8-bit image.
- Image is a reference to the source image.
- Image Mask is an 8-bit image specifying the region in the image to use for the calculation. Only those pixels in the original image that correspond to an equivalent non-zero pixel in the mask image are used for the calculation. The entire image is used in the calculation if Image Mask is not connected.
- Interval Range is a cluster specifying the minimum and maximum boundaries for the histogram calculation. Only those pixels having a value that falls in this range are taken into account by the histogram calculation. This cluster is composed of the following elements:
  - Minimum is the minimum interval value. The default value of (0, 0) ensures that the real minimum value is determined by the source image, as described in the following table.

Image Type	Minimum Value Used
US	0
116	Minimum pixel value found in the image
SGL	Minimum pixel value found in the image

Maximum is the maximum interval value. The default value of (0, 0) ensures that the real maximum value is determined by the source image, as described in the following table.

Image Type	Maximum Value Used
U8	0
116	Maximum pixel value found in the image
SGL	Maximum pixel value found in the image

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Histogram Report is a cluster that returns the histogram values. This cluster contains the following elements:
  - histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [(Starting Value +  $(n-1) \times Interval \ Width$ ), (Starting Value +  $n \times (Interval \ Width 1)$ )].

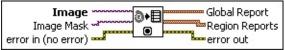
- Minimal Value returns the smallest pixel value used in calculating the histogram.
- Maximal Value returns the largest pixel value used in calculating the histogram.
- Starting Value returns the smallest pixel value from the first class calculated in the histogram. It can be equal to the Minimum value from the Interval Range or the smallest value found for the image type connected.
- Interval Width returns the length of each class.
- Mean Value returns the mean value of the pixels used in calculating the histogram.
- Standard Variation returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.
- Area (pixels) returns the number of pixels used in the histogram calculation. This is influenced by the values specified in Interval Range and the contents of Image Mask.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Quantify VI**

Owning Palette: Analysis Installed With: NI Vision Development Module

Quantifies the contents of an image or the regions within an image. The region definition is performed with a labeled image mask. Each region of the mask has a single unique value.





- **Image** is a reference to the source image.
- Image Mask is a labeled version of the source image that specifies the regions to quantify. This image must be an 8-bit image or 16-bit image. Only the pixels in the original image that correspond to an equivalent pixel in the mask different from 0 are used for the quantification. Each pixel in this image (mask) indicates, by its value, which region belongs to the corresponding pixel in Image. Up to 255 different regions for an 8-bit image, or 65,535 regions for a 16-bit image, can be quantified directly from the Image. A quantification is performed on the complete image if the Image Mask is not connected.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero **error code**. If **status** is FALSE, **code** is 0 or a

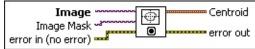
- warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Global Report is a cluster containing the quantification data relative to all the regions within an image, or to the entire image if the Image Mask is not connected. This cluster contains the following elements:
  - Mean Value of the pixels.
  - Standard Variation of the pixel values indicates the distribution of the values in relation to the average. The higher this value, the better the distribution of the pixel values.
  - Minimal Value returns the smallest pixel value.
  - Maximal Value returns the largest pixel value.
  - Area (calibrated) returns the analyzed surface area in userdefined units.
  - **Area (pixels)** returns the analyzed surface area in pixels.
  - % returns the percentage of the analyzed surface in relation to the complete image.
- Region Reports is a cluster containing the quantification data relative to each region within an image, or the entire image if the Image Mask is not connected. The nth element in this array contains the data regarding the *n*th region. The size of this array is equal to the largest pixel value in Image Mask. The returned data is identical to the data in Global Report.
  - Mean Value of the pixels.
  - Standard Variation of the pixel values indicates the distribution of the values in relation to the average. The higher this value, the better the distribution of the pixel values.
  - Minimal Value returns the smallest pixel value.
  - Maximal Value returns the largest pixel value.
  - Area (calibrated) returns the analyzed surface area in userdefined units.

- Area (pixels) returns the analyzed surface area in pixels.
- % returns the percentage of the analyzed surface in relation to the complete image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Centroid VI**

**Owning Palette:** Analysis Installed With: NI Vision Development Module Computes the center of the energy of an image or of a portion of an image.





- **Image** is a reference to the source image.
- Image Mask is an 8-bit image specifying the region in the image to use for the calculation. Only those pixels in the original image that correspond to an equivalent non-zero pixel in the mask image are used for the calculation. The entire image is used in the calculation if Image Mask is not connected.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Em Centroid is a cluster containing the x- and y-coordinates of the

centroid of the image.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

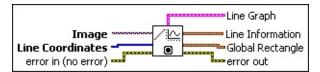
### IMAQ LineProfile VI

Owning Palette: Analysis Installed With: NI Vision Development Module

Calculates the profile of a line of pixels. This VI returns a data type (cluster) compatible with a LabVIEW graph. The relevant pixel information is taken from the specified vector (line).

#### **Examples**





- **Image** is a reference to the source image.
- Line Coordinates is an array specifying the pixel coordinates that form the end points of the line.
  - D

**Note** A line with the coordinates (0, 0, 0, 255) is formed from 256 pixels. Any pixels designated by **Line Coordinates** that lie outside the actual image are set to 0 in **Pixels Line**.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **Line Graph** is a cluster that contains the line profile with an x-origin at 0 and an increment of 1. The cluster contains the following elements:
  - **x0** always returns 0.
  - **dx** always returns 1.
  - Pixels Line returns the line profile calculated in an array in which elements represent the pixel values belonging to the specified vector.
- **Line Information** is a cluster containing relevant information about the pixels found in the specified vector. This cluster contains the following elements:
  - Min returns the smallest pixel value found in the line profile.
  - **Max** returns the largest pixel value found in the line profile.
  - Mean returns the mean value of the pixels found in the line profile.
  - **Std Dev** returns the standard deviation from the line profile.
  - **Count** returns the number of pixels found in the line profile.
- Global Rectangle is a cluster that contains the coordinates of a bounding rectangle for the line in the image. This cluster includes the following parameters:
  - **x1Left** indicates the x-coordinate of the top-left corner of the rectangle.
  - y1Top indicates the y-coordinate of the top-left corner of the rectangle.
  - **x2Right** indicates the x-coordinate of the bottom-right corner of the rectangle.
  - y2Bottom indicates the y-coordinate of the bottom-right corner of the rectangle.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Analysis\LineProfile Example.vi

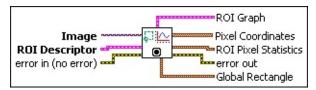
### **IMAQ ROIProfile VI**

Owning Palette: Analysis Installed With: NI Vision Development Module

Calculates the profile of the pixels along the boundary of an ROI descriptor. This VI returns a data type (cluster) that is compatible with a LabVIEW graph. This VI also returns other information such as pixel statistics and the coordinates of the pixels along the ROI boundary.

#### **Examples**





- Image is a reference to the source image.
- ROI Descriptor is a descriptor that defines the region of interest.
  - Global Rectangle is the minimum rectangle required to contain all of the contours in the ROI.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- **status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **ROI Graph** is a cluster that contains the ROI profile with an x-origin at 0 and an increment of 1. The cluster contains the following elements:
  - **x0** always returns 0.
  - **dx** always returns 1.
  - **Pixels Line** returns the ROI profile calculated in an array in which elements represent the pixel values belonging to the specified vector.
- Pixel Coordinates is an array containing the spatial coordinates of each pixel along the ROI boundary.
- ROI Pixel Statistics is a cluster containing relevant information about the pixels found along the ROI boundary. The cluster contains the following elements:
  - Min returns the smallest pixel value found in the ROI profile.
  - **Max** returns the largest pixel value found in the ROI profile.
  - Mean returns the mean value of the pixels found in the ROI profile.
  - **Std Dev** returns the standard deviation from the ROI profile.
  - **Count** returns the count of pixels found in the ROI profile.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Global Rectangle is a cluster that contains the coordinates of a bounding rectangle for the ROI in the image. This cluster includes the following parameters:
  - **x1Left** indicates the x-coordinate of the top-left corner of the rectangle.
  - y1Top indicates the y-coordinate of the top-left corner of the rectangle.
  - **x2Right** indicates the x-coordinate of the bottom-right corner of the rectangle.
  - y2Bottom indicates the y-coordinate of the bottom-right corner of the rectangle.

### **Examples**

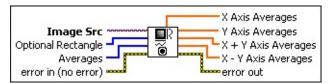
Refer to the following for examples that use this VI.

- LabVIEW\examples\Vision\2. Functions\Caliper\Edge Detection Example.llb
- LabVIEW\examples\Vision\3. Applications\Mechanical Assembly Example.vi

## **IMAQ LinearAverages VI**

**Owning Palette:** Analysis Installed With: NI Vision Development Module Computes the average pixel intensity (mean line profile) on the whole or part of the image.





- Image Src is a reference to the source image.
- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Averages specifies which linear averages the VI calculates. Specify one or more of the following values in the array:
  - **>** 1

**Note** If **Averages** is empty, the VI will calculate all four types of linear averages.

X Axis Averages (0)	Calculates the linear average along each column in the image
Y Axis Averages (1)	Calculates the linear average along each row in the image
X + Y Axis Averages (2)	Calculates the linear average along each diagonal running from bottom-left to top-right.
X – Y Axis Averages (3)	Calculates the linear average along each diagonal running from top-left to bottom-right.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

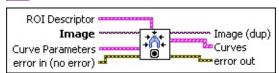
to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- X Axis Averages is the linear average along each column in the image.
- Y Axis Averages is the linear average along each row in the image.
- X + Y Axis Averages is the linear average along each diagonal running from bottom-left to top-right.
- X Y Axis Averages is the linear average along each diagonal running from top-left to bottom-right.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Extract Curves VI**

**Owning Palette:** Analysis Installed With: NI Vision Development Module Finds curves in an image.





- ROI Descriptor specifies the region of the image in which to extract curves. This region must be a single rectangle or rotated rectangle contour. By default, the VI returns the curves for the entire image.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Image is a reference to the source image.
- **Curve Parameters** is a cluster containing information about how curves are extracted from the image.
  - **Extraction Mode** specifies how the VI identifies curves in the image. Choose from the following options:

Normal (0)	(Default) Select <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background.
Uniform Regions (1)	Select <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.

**Edge Threshold** specifies the minimum contrast a seed point must have for it to be begin a curve. Valid values range from 1 to 360. The default value is 75.

Edge Filter Size specifies the width of the edge filter that the VI uses to identify curves in the image. The following options are available:

<b>Fine</b> (0)	Sets the <b>Edge Filter Size</b> to fine.
Normal (1)	(Default) Sets the <b>Edge Filter Size</b> to normal.

- Min Length specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value. The default value is 25.
- Row Search Step Size specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Col Search Step Size specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Max Endpoint Gap specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
- Closed specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
- **Subpixel** is ignored by the VI.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Image (dup)** is a reference to the source image.
- **Curves** is an array describing the curves found in the image.
  - **Points** specifies the location of every point detected on the curve.
  - Closed? is TRUE (Closed) when the distance between the endpoints of the curve are less than or equal to **Max**Endpoint Gap or FALSE (Open) when the distance between the endpoints of the curve are greater than **Max Endpoint**Gap.
  - Curve Length returns the length of the curve.
  - Min Edge Strength returns the lowest edge strength detected on the curve.
  - Max Edge Strength returns the highest edge strength detected on the curve.
  - Average Edge Strength returns the average of all edge strengths detected on the curve.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for

more information about the error.

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

### **IMAQ Particle Analysis Report VI**

Owning Palette: Analysis Installed With: NI Vision Development Module

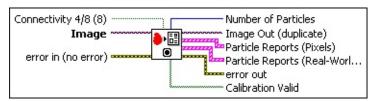
Returns the number of particles detected in a binary image and an array of reports containing the most commonly used particle measurements.



**Note** This VI modifies the source image. If you need the original source image, Create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.

### **Examples**





**Connectivity 4/8 (8)** specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

- **Image** is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number of Particles indicates the number of particles detected in the image.
- Image Out (duplicate) is a reference to the Image input.
- Particle Reports (Pixels) is an array that returns a set of uncalibrated pixel measurements from the detected particles. This cluster contains the following elements:
  - **Area** is the area of the particle.
  - Number of Holes is the number of holes in the particle. NI Vision can detect holes inside a particle as small as one pixel.
  - **Bounding Rect** is the smallest rectangle with sides parallel to the x-axis and y-axis that completely encloses the particle.
  - Center of Mass is the point representing the average position of the total mass of the particle assuming every point in the particle has a constant density. Center of Mass may be located outside the particle if the particle is not convex.
  - Orientation is the angle of the line passing through the particle center of mass with the lowest moment of inertia.
  - **Dimensions** indicates the width and height of **Bounding Rect**.
- Particle Reports (Real-World) is an array that returns a set of calibrated real-world measurements from the detected particles. If the image has no calibration information, uncalibrated pixel measurements are returned. This cluster contains the following elements:
  - Area is the area of the particle.
  - Number of Holes is the number of holes in the particle. NI

Vision can detect holes inside a particle as small as one pixel.

- **Bounding Rect** is the smallest rectangle with sides parallel to the x-axis and y-axis that completely encloses the particle.
- Center of Mass is the point representing the average position of the total mass of the particle assuming every point in the particle has a constant density. Center of Mass may be located outside the particle if the particle is not convex.
- Orientation is the angle of the line passing through the particle center of mass with the lowest moment of inertia.
- **Dimensions** indicates the width and height of **Bounding Rect**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Calibration Valid indicates whether the calibration information for a particle is valid. Calibration Valid has one Boolean for each report in Particle Reports (Real-World). If the calibration information is invalid for any pixel in the particle, the corresponding Calibration Valid Boolean is FALSE.
  - Note A pixel has invalid calibration information when the pixel lies outside the area of the image that was calibrated.
  - Note Particle Report (Real-World) contains measurements for every particle regardless of whether the

particle has valid or invalid calibration information.

### **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Analysis Report Example.vi

### **IMAQ Particle Analysis VI**

Owning Palette: Analysis Installed With: NI Vision Development Module

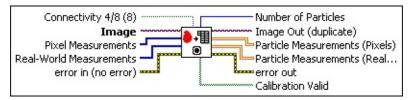
Returns the number of particles detected in a binary image and a 2D array of requested measurements about the particle.



**Note** This VI modifies the source image. If you need the original source image, Create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.

### **Examples**





Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:

8 (TRUE)	Particle detection is performed in connectivity mode 8
4 (FALSE)	Particle detection is performed in connectivity mode 4

- **Image** is a reference to the source image.
- Pixel Measurements is an array of measurement parameters you can request for each particle. The parameters are returned as uncalibrated pixel measurements.
- Real-World Measurements is an array of measurement parameters you can request for each particle. The parameters are returned as calibrated real-world measurements. If **Image** does not have any attached calibration information, the VI returns pixel measurements.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to

**error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number of Particles indicates the number of particles detected in the image.
- Image Out (duplicate) is a reference to the Image input.
- Particle Measurements (Pixels) is a 2D array that returns the requested pixel measurements from the detected particles. The array has one column for each measurement requested in Pixel Measurements and one row for each particle detected.
- Particle Measurements (Real-World) is a 2D array that returns the requested calibrated real-world measurements from the detected particles. The array has one column for each measurement requested in Real-World Measurements and one row for each particle detected. If Image does not have any attached calibration information, the VI returns pixel measurements.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Calibration Valid indicates whether the calibration information for a particle is valid. Calibration Valid has one Boolean for each report in Particle Reports (Real-World). If the calibration information is invalid for any pixel in the particle, the corresponding Calibration Valid Boolean is FALSE.
  - Note A pixel has invalid calibration information when the pixel lies outside the area of the image that was calibrated.
  - Note Particle Report (Real-World) contains measurements for every particle regardless of whether the particle has valid or invalid calibration information.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Binary Analysis\Particle Analysis Example.vi

# **Color Processing**

Owning Palette: Image ProcessingInstalled With: NI Vision

**Development Module** 

Use the Color Processing VIs to perform basic processing on color images. Use these VIs to compute the histogram of a color image; apply lookup tables to color images; change the brightness, contrast, and gamma information associated with a color image; and threshold a color image.

Palette Object	Description
IMAQ ColorUserLookup	Applies a lookup table (LUT) to each color plane.
IMAQ ColorBCGLookup	Applies a brightness, contrast, and gamma correction to each color plane separately.
IMAQ ColorEqualize	Equalizes a color image. This VI equalizes either the luminance plane (default) or all three planes.
IMAQ ColorThreshold	Applies a threshold to the three planes of an RGB or HSL image and places the result into an 8-bit image.
IMAQ ColorHistogram	Calculates the histograms extracted from the three planes of an image. This VI can function in one of six modes corresponding to the six color models (RGB, HSL, HSV, HSI, CIE L*a*b*, or CIE XYZ). The IMAQ ColorHistograph VI, a variant of the IMAQ ColorHistogram VI, has the advantage that its output data is directly compatible with a LabVIEW graph.
IMAQ ColorHistograph	Calculates the histograms extracted from the three planes of an image. This VI can function in one of four modes corresponding to the color models RGB, HSL, HSV, and HSI. The output from this VI is directly compatible with a LabVIEW graph.
IMAQ ColorLearn	Extracts the color features of an image, which can be used for color matching or other applications related with color information, such as color identification and color image segmentation.
IMAQ ColorMatch	Finds the match between the color content of multiple regions in an image and those defined by the input

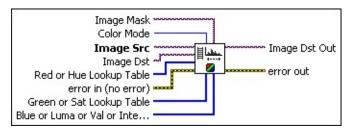
# IMAQ ColorUserLookup VI

Owning Palette: Color ProcessingInstalled With: NI Vision

**Development Module** 

Applies a lookup table (LUT) to each color plane.





- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- **Color Mode** defines the image color format to use for the operation. Choose from the following values:

**Note** Unsigned 64-bit RGB images support only a **Color Mode** of **RGB**.

<b>RGB</b> (0)	(Default) Specifies the color format RGB (red, green, and blue)
HSL (1)	Specifies the color format HSI (hue, saturation, and luminance)
<b>HSV</b> (2)	Specifies the color format HSV (hue, saturation, and value)
<b>HSI</b> (3)	Specifies the color format HSI (hue, saturation, and intensity)

- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Red or Hue Lookup Table is the LUT applied to the first color plane, depending on the Color Mode. This array can contain a

maximum of 256 elements. The array is filled automatically when you specify fewer than 256 elements. This procedure does not change pixel values that are not explicitly specified from the values of the LUT you input. By default this array is empty, and no replacement occurs on this plane.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Green or Sat Lookup Table is the LUT applied to the second color plane, depending on the Color Mode. This array can contain a maximum of 256 elements. The array is filled automatically when you specify fewer than 256 elements. This procedure does not change pixel values that are not explicitly specified from the values of the LUT you input. By default this array is empty, and no replacement occurs on this plane.
- Blue or Luma or Val or Inten Lookup Table is the LUT applied to the third color plane, depending on the Color Mode. This array can contain a maximum of 256 elements. The array is filled automatically when you specify fewer than 256 elements. This

- procedure does not change pixel values that are not explicitly specified from the values of the LUT you input. By default this array is empty, and no replacement occurs on this plane.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

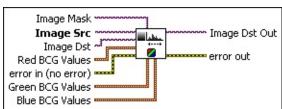
## IMAQ ColorBCGLookup VI

Owning Palette: Color Processing Installed With: NI Vision

**Development Module** 

Applies a brightness, contrast, and gamma correction to each color plane separately.





- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Red BCG Values is a cluster of the following three values to adjust for the Red plane:
  - Brightness sets the brightness of the red plane of the image. This value is used as the x-intercept of the transfer function in the lookup table. The neutral value is 128 (no change in the plane). A higher value returns a brighter plane. A value less than 128 decreases the overall brightness of the plane. The default is 128.
  - Contrast sets the contrast of the red plane of the image. This control is used as the slope of the transfer function in the lookup table and is expressed in degrees. A slope of 45 is neutral. A higher value returns a more contrasted red plane. A value smaller than 45 decreases the contrast of the red plane. The default is 45.0.
  - Gamma sets the gamma correction applied to the red plane of the image. The neutral value is 1. A value greater than 1

gives extended contrast for small pixel values and less contrast for large pixel values. A value smaller than 1 returns a plane with less contrast for small pixel values and extended contrast for large pixel values. The default is 1.0.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Green BCG Values is a cluster of the following three values to adjust for the Green plane:
  - Brightness sets the brightness of the green plane of the image. The default is 128.
  - **Contrast** sets the contrast of the green plane of the image. The default is 45.0.
  - Gamma sets the gamma correction applied to the green plane of the image. The default is 1.0.
- Blue BCG Values is a cluster of the following three values to adjust for the Blue plane:
  - Brightness sets the brightness of the blue plane of the

- image. The default is 128.
- **Contrast** sets the contrast of the blue plane of the image. The default is 45.0.
- **Gamma** sets the gamma correction applied to the blue plane of the image. The default is 1.0.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

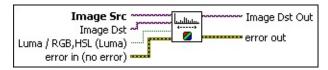
### **IMAQ ColorEqualize VI**

Owning Palette: Color ProcessingInstalled With: NI Vision

**Development Module** 

Equalizes a color image. This VI equalizes either the luminance plane (default) or all three planes.





- **Image Src** is a reference to the source image.
- Image Dst is a reference to the destination image.
- Luma / RGB,HSL (Luma) specifies whether to perform the operation on the luminance plane or on all three planes. An equalization on the luminance plane conserves the hue and saturation from the color image. An equalization of the three planes gives a stronger contrast but changes the hue and saturation of the color image. The default is Luma, which is FALSE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ ColorThreshold VI**

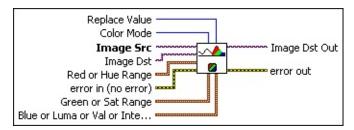
Owning Palette: Color ProcessingInstalled With: NI Vision

**Development Module** 

Applies a threshold to the three planes of an RGB or HSL image and places the result into an 8-bit image.

#### **Details Examples**





- Replace Value is the value that the VI uses to fill pixels in the destination image when the corresponding pixels in the Image Src are found in all three ranges. The default is 1.
- Color Mode defines the image color format to use for the operation. Choose from the following values:

<b>RGB</b> (0)	(Default) Specifies the color format RGB (red, green, and blue)
HSL (1)	Specifies the color format HSI (hue, saturation, and luminance)
<b>HSV</b> (2)	Specifies the color format HSV (hue, saturation, and value)
<b>HSI</b> (3)	Specifies the color format HSI (hue, saturation, and intensity)

- Image Src is the reference to the image to threshold. The source image is converted to 8 bit and contains the result of the threshold if Image Dst is not connected.
- Image Dst is a reference to the destination image. If Image Dst is connected, it must be an unsigned 8-bit image.
- Red or Hue Range is a cluster used to determine the thresholding range for the red or hue plane, depending on the Color Mode. Any

pixel values not included in this range are reset to zero in the destination image. The pixel values included in this range are altered depending on the status of the **Replace** input. By default, all pixel values are included (0, 255). **Red or Hue Range** includes the following elements:

- **Lower value** is the minimum pixel value in the red or hue plane that is used for the threshold. The default is 0.
- **Upper value** is the maximum pixel value in the red or hue plane that is used for the threshold. The default is 255.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Green or Sat Range is a cluster used to determine the thresholding range for the green or saturation plane, depending on the Color Mode. Any pixel values not included in this range are reset to zero in the destination image. The pixel values included in this range are altered depending on the status of the Replace input. By default, all pixel values are included (0, 255). Green or Sat Range includes the following elements:

- **Lower value** is the minimum pixel value in the green or saturation plane that is used for the threshold. The default is 0.
- Upper value is the maximum pixel value in the green or saturation plane that is used for the threshold. The default is 255.
- Blue or Luma or Val or Inten Range is a cluster used to determine the thresholding range for the blue, luminance, value, or intensity plane, depending on the Color Mode. Any pixel values not included in this range are reset to zero in the destination image. The pixel values included in this range are altered depending on the status of the Replace input. By default, all pixel values are included (0, 255). Blue or Luma or Val or Inten Range includes the following elements:
  - Lower value the minimum pixel value in the blue, luminance, value, or intensity plane that is used for the threshold. The default is 0.
  - **Upper value** the maximum pixel value in the blue, luminance, value, or intensity plane that is used for the threshold. The default is 255.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

A test is performed with each range (Red or Hue Range, Green or Sat Range, and Blue or Luma or Val or Inten Range) to determine whether the corresponding pixel from the **Image Src** is set to the value specified in **Replace Value**. If a pixel from **Image Src** does not have corresponding pixel values specified in all three ranges, the corresponding pixel in **Image Dst Out** is set to 0. If the image destination **Image Dst** is not connected, IMAQ ColorThreshold converts the image source **Image Src** to an 8-bit image and saves the resulting image in the image source.



**Note** If a range is not specified for a given plane, the value of the pixel for this color plane is ignored (i.e. the algorithm always considers the pixel value for this plane to be in range). Therefore you easily can apply a threshold to one of the three ranges without having to set the values of the other two ranges.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Color\Color Threshold Example.vi

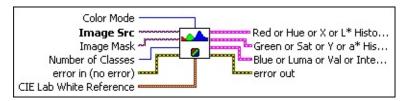
# **IMAQ ColorHistogram VI**

Owning Palette: Color ProcessingInstalled With: NI Vision

**Development Module** 

Calculates the histograms extracted from the three planes of an image. This VI can function in one of six modes corresponding to the six color models (RGB, HSL, HSV, HSI, CIE L\*a\*b\*, or CIE XYZ). The <a href="IMAQ">IMAQ</a> ColorHistograph VI, a variant of the IMAQ ColorHistogram VI, has the advantage that its output data is directly compatible with a LabVIEW graph.





**Color Mode** defines the image color format to use for the operation. Choose from the following values:

<b>RGB</b> (0)	(Default) Specifies the color format RGB (red, green, and blue)
HSL (1)	Specifies the color format HSI (hue, saturation, and luminance)
<b>HSV</b> (2)	Specifies the color format HSV (hue, saturation, and value)
<b>HSI</b> (3)	Specifies the color format HSI (hue, saturation, and intensity)

- Image Src is a reference to the source image.
- Image Mask is an 8-bit image specifying the region in the image to use for the calculation. Only those pixels in the original image that correspond to an equivalent non-zero pixel in the mask image are used for the calculation. The entire image is used in the calculation if Image Mask is not connected.
- Number of Classes specifies the number of classes used to classify the pixels. The default value is 256.
- error in (no error) describes the error status before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- CIE Lab White Reference is the CIE X, Y, and Z values associated with white. The default values of this control work with RGB values of (255, 255, 255) as white.
- Red or Hue or X or L\* Histogram Report is a cluster that returns the detailed results from a histogram calculated on a red or hue plane, depending on the Color Mode. This cluster is the same as the cluster used by the <a href="IMAQ Histogram">IMAQ Histogram</a> VI. It contains the following elements:
  - histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [(Starting Value +  $(n 1) \times Interval \ Width$ ), (Starting Value +  $n \times (Interval \ Width 1)$ )].
  - Minimal Value returns the smallest pixel value used in calculating the histogram.
  - Maximal Value returns the largest pixel value used in calculating the histogram.

- **Starting Value** is always equal to 0. It returns the smallest pixel value from the first class calculated in the histogram.
- Interval Width returns the length of each class.
- Mean Value returns the mean value of the pixels used in calculating the histogram.
- Standard Variation returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.
- Area (pixels) returns the number of pixels used in the histogram calculation. This is influenced by the values specified in Interval Range and the contents of Image Mask.
- Green or Sat or Y or a\* Histogram Report is a cluster that returns the detailed results from a histogram calculated on the green or saturation plane, depending on the Color Mode. This cluster is the same as the cluster used by the <a href="IMAQ Histogram">IMAQ Histogram</a> VI. It contains the following elements:
  - histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [(Starting Value +  $(n 1) \times Interval \ Width$ ), (Starting Value +  $n \times (Interval \ Width 1)$ )].
  - Minimal Value returns the smallest pixel value used in calculating the histogram.
  - Maximal Value returns the largest pixel value used in calculating the histogram.
  - Starting Value is always equal to 0. It returns the smallest pixel value from the first class calculated in the histogram.
  - Interval Width returns the length of each class.
  - Mean Value returns the mean value of the pixels used in calculating the histogram.
  - Standard Variation returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.

**Area (pixels)** returns the number of pixels used in the histogram calculation. This is influenced by the values specified in **Interval Range** and the contents of **Image Mask**.

- Blue or Luma or Val or Inten or Z or b\* Histogram Report is a cluster that returns the detailed results from a histogram calculated on the blue, luminance, value, or intensity planes, depending on the Color Mode. This cluster is the same as the cluster used by the IMAO Histogram VI. It contains the following elements:
  - histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [( $Starting\ Value + (n-1) \times Interval\ Width$ ), ( $Starting\ Value + n \times (Interval\ Width 1)$ )].
  - Minimal Value returns the smallest pixel value used in calculating the histogram.
  - Maximal Value returns the largest pixel value used in calculating the histogram.
  - **Starting Value** is always equal to 0. It returns the smallest pixel value from the first class calculated in the histogram.
  - Interval Width returns the length of each class.
  - Mean Value returns the mean value of the pixels used in calculating the histogram.
  - Standard Variation returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.
  - Area (pixels) returns the number of pixels used in the histogram calculation. This is influenced by the values specified in Interval Range and the contents of Image Mask.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE

(checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

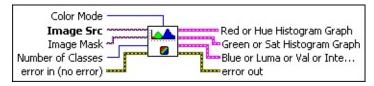
# **IMAQ ColorHistograph VI**

Owning Palette: Color ProcessingInstalled With: NI Vision

**Development Module** 

Calculates the histograms extracted from the three planes of an image. This VI can function in one of four modes corresponding to the color models RGB, HSL, HSV, and HSI. The output from this VI is directly compatible with a LabVIEW graph.





**Color Mode** defines the image color format to use for the operation. Choose from the following values:

<b>RGB</b> (0)	(Default) Specifies the color format RGB (red, green, and blue)
HSL (1)	Specifies the color format HSI (hue, saturation, and luminance)
<b>HSV</b> (2)	Specifies the color format HSV (hue, saturation, and value)
<b>HSI</b> (3)	Specifies the color format HSI (hue, saturation, and intensity)

- Image Src is a reference to the source image.
- Image Mask is an 8-bit image specifying the region in the image to use for the calculation. Only those pixels in the original image that correspond to an equivalent non-zero pixel in the mask image are used for the calculation. The entire image is used in the calculation if Image Mask is not connected.
- Number of Classes specifies the number of classes used to classify the pixels. The default value is 256.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Red or Hue Histogram Graph is a cluster that returns the detailed results from a histogram calculated on a red or hue plane, depending on the Color Mode. This cluster is the same as the cluster used by the <a href="IMAQ Histograph">IMAQ Histograph</a> VI. It contains the following elements:
  - Starting Value is always equal to 0. This parameter is returned in the type Histogram Report, as in the <a href="MAQHistograph">IMAQHISTOGRAPH</a> VI.
  - Incremental Value returns the incrementing value that specifies how much to add to **Starting Value** in calculating the median value of each class from the histogram. The median value  $x_n$  from the nth class is expressed as follows:  $x_n = Starting Value + n \times Incremental Value$ .
  - Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [(Starting Value +  $(n-1) \times Interval \ Width$ ), (Starting Value +  $n \times (Interval \ Width 1)$ )].
- Green or Sat Histogram Graph is a cluster that returns the

detailed results from a histogram calculated on the green or saturation plane, depending on the **Color Mode**. This cluster is the same as the cluster used by the <u>IMAQ Histograph</u> VI. It contains the following elements:

- Starting Value is always equal to 0. This parameter is returned in the type Histogram Report, as in the <a href="IMAQ">IMAQ</a> Histograph VI.
- Incremental Value returns the incrementing value that specifies how much to add to **Starting Value** in calculating the median value of each class from the histogram. The median value  $x_n$  from the nth class is expressed as follows:  $x_n = Starting Value + n \times Incremental Value$ .
- Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [(Starting Value +  $(n-1) \times Interval \ Width$ ), (Starting Value +  $n \times (Interval \ Width 1)$ )].
- Blue or Luma or Val or Inten Histogram Graph is a cluster that returns the detailed results from a histogram calculated on the blue, luminance, value, or intensity planes, depending on the Color Mode. This cluster is the same as the cluster used by the IMAQ Histograph VI. It contains the following elements:
  - Starting Value is always equal to 0. This parameter is returned in the type Histogram Report, as in the <a href="MAQ">IMAQ</a> Histograph VI.
  - Incremental Value returns the incrementing value that specifies how much to add to **Starting Value** in calculating the median value of each class from the histogram. The median value  $x_n$  from the nth class is expressed as follows:  $x_n = Starting Value + n \times Incremental Value$ .
  - Histogram returns the histogram values in an array. The elements found in this array are the number of pixels per class. The nth class contains all pixel values belonging to the interval [(Starting Value +  $(n 1) \times Interval \ Width$ ), (Starting Value +  $n \times (Interval \ Width 1)$ )].

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ ColorLearn VI**

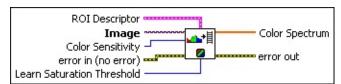
Owning Palette: Color ProcessingInstalled With: NI Vision

**Development Module** 

Extracts the color features of an image, which can be used for color matching or other applications related with color information, such as color identification and color image segmentation.

#### **Examples**





- ROI Descriptor is a descriptor indicating the regions in the image that contain the colors you want to learn. If ROI Descriptor contains multiple regions, the VI accumulates color information in each region before learning. If ROI Descriptor is empty or not connected, the VI considers the entire image as a single region.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Image is a reference to the color image from which you want to learn color information.
- Color Sensitivity specifies the sensitivity of the color information in the image. The default is **Low**. Set this option to **High** when you need to distinguish colors with close hue values.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Learn Saturation Threshold specifies the threshold value to distinguish two colors with the same hue value. The default is 80.
- **Color Spectrum** returns the color features found in the image region. These features represent the color information in the image region in a compact form.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Color\ColorLearn Example.vi

# **IMAQ ColorMatch VI**

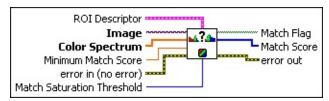
Owning Palette: Color ProcessingInstalled With: NI Vision

**Development Module** 

Finds the match between the color content of multiple regions in an image and those defined by the input **Color Spectrum**. **Color Spectrum** is output from the **IMAQ ColorLearn** VI.

### **Examples**





- ROI Descriptor is a descriptor indicating the regions in the image whose color information you want to compare with the information in Color Spectrum. All region contours are considered to be external. If ROI Descriptor contains multiple regions, the color information in each region is compared with Color Spectrum individually and the match results are reported for each region. If ROI Descriptor is not connected, the VI considers the entire image as a single region.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Image is a reference to the color image from which you want to extract the color feature to compare with the color defined by the input **Color Spectrum** color feature array.
- Color Spectrum contains information about the color to match.

During the matching phase, the color information in each specified image region is compared with the information contained in this array.



**Tip** You can ignore certain color components by replacing the corresponding component in **Color Spectrum** to −1. For example, by replacing the last component in **Color Spectrum** with −1, the white color is ignored during the color matching process. By replacing the second to last component in **Color Spectrum**, the black color is ignored during the color matching process.

- Minimum Match Score is the smallest score a match can have to be considered valid. The data range for a match is between 0 and 1000.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Match Saturation Threshold specifies the threshold value to distinguish two colors with the same hue value. The default is 80. The value used here should be the same as that used in the IMAQ

#### ColorLearn VI.

- Match Flag is an array containing Boolean information that indicates whether there is a match between the input color information and the color in each specified region in the image. For each specified region in the image, the match flag is set to TRUE if its color information matches the input color information (for example, if Match Score is higher or equal to Minimum Match Score).
- Match Score is an array containing the match scores for each region in the image. The match scores are represented on a scale of 0 to 1000 where 0 indicates no match and 1000 indicates a perfect match.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Color\ColorMatching Example.vi

# **Operators**

Owning Palette: Image ProcessingInstalled With: NI Vision

**Development Module** 

Use the Operators VIs to perform arithmetic and bit-wise operations in NI Vision. Use some of these VIs to add, subtract, multiply, and divide an image with other images or constants. Use other VIs in this palette to apply logical operations and make pixel comparisons between an image and other images or a constant.

<b>D</b> 1	
Palette Object	Description
IMAQ Add	Adds two images or an image and a constant.
IMAQ Subtract	Subtracts one image from another or a constant from an image.
IMAQ Absolute Difference	Subtracts one image from another or a constant from an image and returns the absolute value of the difference.
IMAQ And	Performs an AND or NAND operation on two images or an image and a constant.
IMAQ Or	Performs an OR or NOR operation on two images or an image and a constant.
IMAQ Multiply	Multiplies two images or an image and a constant.
IMAQ Divide	Divides one image by another image or an image by a constant.
IMAQ LogDiff	Keeps bits found in Image Src A that are absent from Image Src B.
IMAQ Xor	Performs an XOR or XNOR operation on two images or an image and a constant.
IMAQ MulDiv	Computes a ratio between two images. Each pixel in Image Src A is multiplied by the integer value specified in the input Constant before being divided by the equivalent pixel found in Image Src B. To avoid losing information, a temporary variable giving higher definition is used to perform the

	operation. If the background is lighter than the image, this function can correct the background. In a background correction image, Image Src A is the acquired image, and Image Src B is the light background.
IMAQ Modulo	Executes modulo division (remainder) of one image by another or an image by a constant.
IMAQ Mask	Recopies the Image Src into the Image Dst. If a pixel value is 0 in the Image Mask, the corresponding pixel in Image Dst is set to 0.
IMAQ Compare	Performs comparison operations between two images or an image and a constant. An operation between an image and a constant occurs when Image Src B is not connected.

# **IMAQ Add VI**

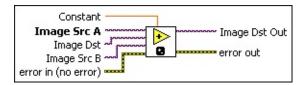
Owning Palette: Operators Installed With: NI Vision Development

Module

Adds two images or an image and a constant.

### **Details** Examples





- **Constant** is the value added to **Image Src A** for image-constant operations. **Constant** is rounded down when the image is encoded as an integer. The default is 0.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

An operation between an image and a constant occurs when Image **Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) + SrcB(x, y)$$
  
or  
 $Dst(x, y) = SrcA(x, y) + Constant$ 

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to **Image Dst**.

To add a constant to an image, **Image Dst** must be connected to the same image type as **Image Src A**.

When an 8-bit image or constant is added to an RGB image, it is added to each plane of the RGB image. When IMAQ Add is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** is added to the corresponding color plane of **Image Src B**. Each of these additions is similar to an IMAQ Add operation on two 8-bit images.

# **Examples**

Refer to the following for examples that use this VI.

- LabVIEW\examples\Vision\2. Functions\Operators\Add Example.vi
- LabVIEW\examples\Vision\2. Functions\Operators\Image Averaging Example.vi

# **IMAQ Subtract VI**

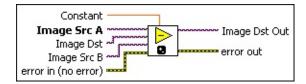
Owning Palette: Operators Installed With: NI Vision Development

Module

Subtracts one image from another or a constant from an image.

#### **Details**





- **Constant** is the value subtracted from **Image Src A** for image-constant operations. **Constant** is rounded down when the image is encoded as an integer. The default is 0.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

An operation between an image and a constant occurs when **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = SrcA(x, y) - SrcB(x, y)$$
  
or  
 $Dst(x, y) = SrcA(x, y) - Constant$ 

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to **Image Dst**.

To subtract a constant from an image, **Image Dst** must be connected to the same image type as the input **Image Src A**.

If one of the two source images is empty, the result is a copy of the other.

When an 8-bit image or constant is subtracted from an RGB image, it is subtracted from each plane of the RGB image. When IMAQ Subtract is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** subtracts the corresponding color plane of **Image Src B**. Each of these subtractions is similar to an IMAQ Subtract operation on two 8-bit images.

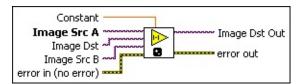
# **IMAQ Absolute Difference VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Subtracts one image from another or a constant from an image and returns the absolute value of the difference.

#### **Details**





- **Constant** is the value subtracted from **Image Src A** for image-constant operations. **Constant** is rounded down when the image is encoded as an integer. The default is 0.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image. If Image Dst is connected, it must have as many bits per pixel as the deepest source image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is

- a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

An operation between an image and a constant occurs when **Image Src B** is not connected. The two possibilities are distinguished in the following equations:

$$Dst(x, y) = |SrcA(x, y) - SrcB(x, y)|$$
  
or  
 $Dst(x, y) = |SrcA(x, y) - Constant|$ 

The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to **Image Dst**.

To set the absolute difference between a constant and an image, **Image Dst** must be connected to the same image type as **Image Src A**.

When setting the absolute difference between an 8-bit image or constant and an RGB image, the absolute difference is calculated for each plane of the RGB image. When IMAQ Absolute Difference is performed on two RGB images, the absolute difference is calculated for each color plane (red, green, and blue) of **Image Src A** and the corresponding color plane of **Image Src B**. Each of these absolute difference calculations is similar

to an IMAQ Absolute Difference operation on two 8-bit images.

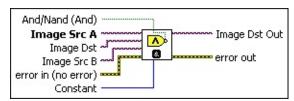
## **IMAQ And VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Performs an AND or NAND operation on two images or an image and a constant.

#### **Details**





- And/Nand (And) specifies which logic operation the VI performs. If set to TRUE, the result of the AND is inverted, producing a NAND. The default is FALSE, which specifies an AND operation.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image. If Image Dst is connected, it must be the same size as Image Src A.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Constant** is the value used for image-constant operations. The default is 0.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

All connected images must be the same image type. An operation between an image and a constant occurs when **Image Src B** is not connected.

This VI is performed for each pixel (x, y) in the following manner: If two images are connected on input, then Dst(x, y) = SrcA(x, y) AND SrcB(x, y).

If **Image Src B** is not connected, then Dst(x, y) = SrcA(x, y) AND *Constant*.

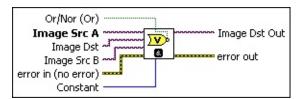
### **IMAQ Or VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Performs an OR or NOR operation on two images or an image and a constant.

#### **Details**





- Or/Nor (Or) specifies which logic operation the VI performs. If set to TRUE, the result of the OR is inverted, producing a NOR. The default is FALSE, which specifies an OR operation.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image. If Image Dst is connected, it must be the same size as Image Src A.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Constant** is the value used for image-constant operations. The default is 0.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

All connected images must be the same image type. An operation between an image and a constant occurs when **Image Src B** is not connected.

This VI is performed for each pixel (x, y) in the following manner: If two images are connected on input, then Dst(x, y) = SrcA(x, y) OR SrcB(x, y).

If **Image Src B** is not connected, then Dst(x, y) = SrcA(x, y) OR *Constant*.

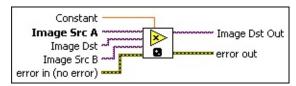
## **IMAQ Multiply VI**

Owning Palette: OperatorsInstalled With: NI Vision Development

Module

Multiplies two images or an image and a constant.





- **Constant** is the value multiplied by Image Src A for image-constant operations.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

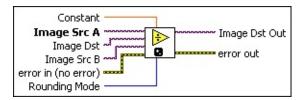
## **IMAQ Divide VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Divides one image by another image or an image by a constant.

#### **Details**





- **Constant** is the value that divides Image Src A for image-constant operations.
- Image Src A is a reference to the first source image.
- Image Dst is the reference to the destination image. If connected, Image Dst is converted to an RGB image. If not connected, Image Src is converted to an RGB image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Rounding Mode specifies the type of rounding to use when dividing image pixels. The following options are available.

• ,	Rounds the result of a division using the best available method.
	available metrod.
Truncate (1)	Truncates the result of a division.

- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The image type of **Image Src B** depends on the image type of **Image Src A**.

- If Image Src A is a signed 16-bit image, Image Src B must be a signed 16-bit image or an unsigned 8-bit image.
- If Image Src A is a 32-bit floating-point image, Image Src B must be a 32-bit floating-point image, signed 16-bit image, or unsigned 8-bit image.
- If **Image Src A** is a complex image, **Image Src B** must be a complex image, 32-bit floating-point image, signed 16-bit image, or unsigned 8-bit image.
- If Image Src A is a 32-bit RGB image, Image Src B must be a 32-bit RGB image, complex image, 32-bit floating-point image, signed 16-bit image, or unsigned 8-bit image.

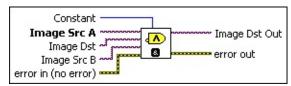
Otherwise, **Image Src B** must be the same type as **Image Src A**.

When the value of a pixel in **Image Src B** is 0, the result of the divide operation for the pixel is either the maximum or minimum pixel value for the image type. If the corresponding pixel value in **Image Src A** is greater than 0, then the result is the maximum possible pixel value for the image type. Otherwise, the result is the minimum possible value for the image type.

## **IMAQ LogDiff VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Keeps bits found in Image Src A that are absent from Image Src B.



- **Constant** is the value used for image-constant operations. The default is 0.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is the reference to the destination (output) image

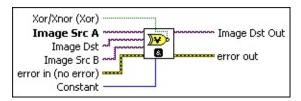
that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## IMAQ Xor VI

Owning Palette: Operators Installed With: NI Vision Development Module

Performs an XOR or XNOR operation on two images or an image and a constant.



- Xor/Xnor (Xor) specifies which logic operation the VI performs.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Constant is the value used for image-constant operations. The

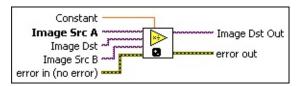
default is 0.

- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## IMAQ MulDiv VI

Owning Palette: Operators Installed With: NI Vision Development Module

Computes a ratio between two images. Each pixel in Image Src A is multiplied by the integer value specified in the input Constant before being divided by the equivalent pixel found in Image Src B. To avoid losing information, a temporary variable giving higher definition is used to perform the operation. If the background is lighter than the image, this function can correct the background. In a background correction image, Image Src A is the acquired image, and Image Src B is the light background.



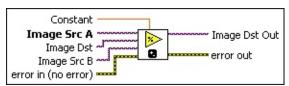
- **Constant** is the value multiplied by each pixel in Image Src A before each pixel in Image Src A is divided by the equivalent pixel in Image Src B.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Modulo VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Executes modulo division (remainder) of one image by another or an image by a constant.



- **Constant** is the value that divides Image Src A for image-constant operations.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

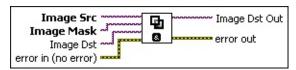
Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Mask VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Recopies the Image Src into the Image Dst. If a pixel value is 0 in the Image Mask, the corresponding pixel in Image Dst is set to 0.



- Image Src is a reference to the source image.
- Image Mask is an 8-bit image that specifies the region of the small image that will be copied. Only those pixels in the Image Src (Small) image that correspond to an equivalent non-zero pixel in the mask image are copied. All other pixels keep their original values. The entire image is processed if Image Mask is not connected.
- Image Dst is a reference to the destination image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

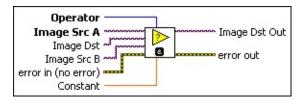
error or warning. The default is an empty string.

- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Compare VI**

Owning Palette: Operators Installed With: NI Vision Development Module

Performs comparison operations between two images or an image and a constant. An operation between an image and a constant occurs when Image Src B is not connected.



- Operator specifies the comparison operator to use.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Src B is the reference to the second source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **Constant** is the value used in comparison with Image Src A for image-constant operations.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src A.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Frequency Domain**

Owning Palette: Image ProcessingInstalled With: NI Vision

**Development Module** 

Use the Frequency Domain VIs to perform frequency processing and other tasks on images. Use these VIs to convert an image from the spatial domain to the frequency domain using a two-dimensional Fast Fourier Transform (FFT) and convert an image from the frequency domain to the spatial domain using the inverse FFT. These VIs also extract the magnitude, phase, real, and imaginary planes of the complex image.

Palette Object	Description
IMAQ FFT	Computes the FFT of an image. The FFT creates a complex image in which high frequencies are grouped at the center, while low frequencies are located at the edges.
IMAQ InverseFFT	Computes the inverse FFT of a complex image.
IMAQ ArrayToComplexImage	Creates a complex image from a 2D array of complex numbers.
IMAQ ArrayToComplexPlane	Replaces the real part or the imaginary part of a complex image, starting from a 2D array of floating-point values.
IMAQ ImageToComplexPlane	Extracts the pixels from an 8-bit, 16-bit, 32-bit floating-point, or Complex image into the real part or imaginary part of a complex image.
IMAQ ComplexImageToArray	Extracts the pixels from a complex image into a complex 2D array.
IMAQ ComplexPlaneToArray	Extracts the pixels from the real part, imaginary part, magnitude, or phase of a complex image into a 2D array of floating-point numbers.
IMAQ ComplexPlaneToImage	Extracts the pixels from the real part, imaginary part, magnitude, or phase of a complex image into an 8-bit, 16-bit, or 32-bit floating-point image.
IMAQ	Computes the conjugate of a complex image.

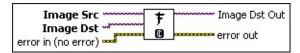
<u>ComplexConjugate</u>	This VI converts the complex pixel data $z = a + ib$ of a complex image into $z = a \& #150$ ; ib.
IMAQ ComplexAttenuate	Attenuates the frequencies of a complex image.
IMAQ ComplexAdd	Adds two images where the first is a complex image, or adds a complex image and a complex number.
IMAQ ComplexSubtract	Subtracts two images where the first is a complex image, or subtracts a complex number from a complex image.
IMAQ ComplexFlipFrequency	Transposes the frequency components of a complex image. The high and low frequency components of a complex image are inverted to produce a central symmetric representation of the spatial frequencies.
IMAQ ComplexTruncate	Truncates the frequencies of a complex image.
IMAQ ComplexMultiply	Multiplies two images where the first is a complex image, or multiples a complex image and a complex number.
IMAQ ComplexDivide	Divides one image by another where the first is a complex image, or divides a complex image by a complex number.

## **IMAQ FFT VI**

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Computes the FFT of an image. The FFT creates a complex image in which high frequencies are grouped at the center, while low frequencies are located at the edges.



- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

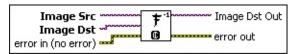
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## IMAQ InverseFFT VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Computes the inverse FFT of a complex image.



- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

to check errors and to specify execution order by wiring error out from one node to error in of the next node.

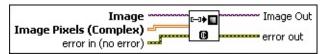
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ ArrayToComplexImage VI**

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Creates a complex image from a 2D array of complex numbers.



- Image Pixels (Complex) is a 2D array of complex numbers containing the pixel values that comprise the image.
- Image is a reference to the complex image to contain the array data.
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

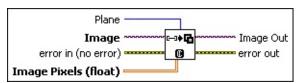
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ ArrayToComplexPlane VI**

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Replaces the real part or the imaginary part of a complex image, starting from a 2D array of floating-point values.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Pixels (float) is a 2D array of 32-bit floating-point values containing the values that comprise one complex plane of the image.
- **Image** is a reference to the source image.
- Plane specifies which component of the complex image is replaced with the values contained in the array of floating points, Image

Pixels.

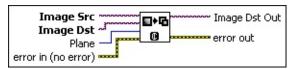
- Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ ImageToComplexPlane VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Extracts the pixels from an 8-bit, 16-bit, 32-bit floating-point, or Complex image into the real part or imaginary part of a complex image.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Plane specifies which component of the complex image to replace.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image

#### Src.

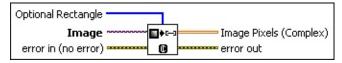
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ ComplexImageToArray VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Extracts the pixels from a complex image into a complex 2D array.



- Image is a reference to the source image.
- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Image Pixels (Complex) returns the extracted pixel values into a 2D array.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

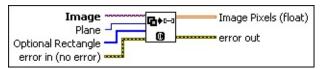
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ ComplexPlaneToArray VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Extracts the pixels from the real part, imaginary part, magnitude, or phase of a complex image into a 2D array of floating-point numbers.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image is a reference to the source image.
- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Plane specifies which component of the complex image is extracted.

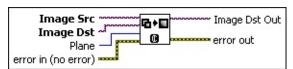
- Image Pixels (float)
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ ComplexPlaneToImage VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Extracts the pixels from the real part, imaginary part, magnitude, or phase of a complex image into an 8-bit, 16-bit, or 32-bit floating-point image.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Plane specifies which component of the complex image is extracted.
- Image Dst Out is a reference to the destination image. If Image

**Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

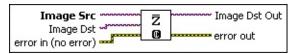
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ ComplexConjugate VI**

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Computes the conjugate of a complex image. This VI converts the complex pixel data z = a + ib of a complex image into z = a & #150; ib.



- **Image Src** is a reference to the source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

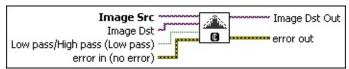
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ ComplexAttenuate VI**

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Attenuates the frequencies of a complex image.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- **Low pass/High pass (Low pass)** determines which frequencies are attenuated.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image

#### Src.

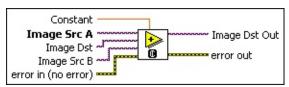
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ ComplexAdd VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Adds two images where the first is a complex image, or adds a complex image and a complex number.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Constant is the value added to Image Src A for image-constant operations.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is

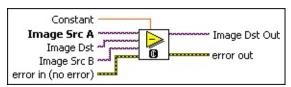
- connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src B is a reference to the second source image.

# **IMAQ ComplexSubtract VI**

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Subtracts two images where the first is a complex image, or subtracts a complex number from a complex image.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Constant** is the value subtracted from Image Src A for image-constant operations.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is

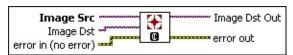
- connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src B is a reference to the second source image.

# IMAQ ComplexFlipFrequency VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Transposes the frequency components of a complex image. The high and low frequency components of a complex image are inverted to produce a central symmetric representation of the spatial frequencies.



- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ ComplexTruncate VI**

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Truncates the frequencies of a complex image.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src is a reference to the source image.
- Image Dst is a reference to the destination image.
- **Truncation Frequency** % is the percentage of the frequencies that are retained within a Fourier-transformed image.
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.

Otherwise, **Image Dst Out** refers to the image referenced by **Image Src**.

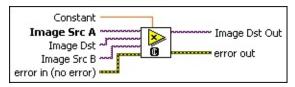
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Low pass/High pass (Low pass)** determines which frequencies are truncated.

# IMAQ ComplexMultiply VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Multiplies two images where the first is a complex image, or multiples a complex image and a complex number.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Constant** is the value multiplied by Image Src A for image-constant operations.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is

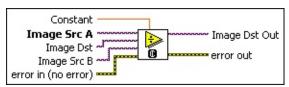
- connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src B is a reference to the second source image.

# IMAQ ComplexDivide VI

Owning Palette: Frequency DomainInstalled With: NI Vision

**Development Module** 

Divides one image by another where the first is a complex image, or divides a complex image by a complex number.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Constant** is the value that divides Image Src A for image-constant operations.
- Image Src A is a reference to the first source image.
- Image Dst is a reference to the destination image.
- Image Dst Out is the reference to the destination (output) image that receives the processing results of the VI. If the Image Dst is

- connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers to the image referenced by **Image Src A**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Src B is a reference to the second source image.

#### **Machine Vision**

Owning Palette: Vision and MotionInstalled With: NI Vision

**Development Module** 

Use the Machine Vision VIs to perform common machine vision inspection tasks, including the presence or absence of parts in an image and measuring the dimensions of parts to see if they meet specifications.

Subpalette	Description
Select Region of Interest	Use the Select Region of Interest VIs to select a specific area of an image from an external display window.
Coordinate System	Use the Coordinate System VIs to find various types of coordinate systems in an image. Use these VIs to find the coordinate system using either edge detection or pattern matching.
Count and Measure Objects	Use the IMAQ Count Objects VI to count and measure objects.
Measure Intensities	Use the Measure Intensities VIs to measure the intensity of a pixel on a point or the intensity statistics of pixels along a line or in a rectangular region of an image.
Measure Distances	Use the Measure Distances VIs to measure distances in an image, such as the minimum and maximum horizontal separation between two vertically oriented edges or the minimum or maximum vertical separation between two horizontally oriented edges.
<u>Locate</u> <u>Edges</u>	Use the Locate Edges VIs to find vertical, horizontal, and circular edges of an image.
Find Patterns	Use the Find Patterns VI to find a pattern in an image.
Searching and Matching	Use the Searching and Matching VIs to locate patterns and sub-images in an image. These VIs allow you to perform color and grayscale pattern matching as well as shape matching.
Caliper	Use the Caliper VIs to detect specific edges, peaks,

	rotational shifts, and other features in an image along regions of interest that you specify.
Analytic Geometry	Use Analytic Geometry VIs to derive results from the coordinates of points returned by image analysis and machine vision algorithms. Use these VIs to fit lines, circles, and ellipses to a set of points in the image; compute the area of a polygon represented by a set of points; measure distances between points; and find angles between lines represented by points.
Inspection	Use Inspection VIs to compare images to a golden template reference image.
Classification	Use Classification VIs to identify unknown objects by comparing significant features to a set of features that conceptually represent classes of known objects.
OCR	Use OCR VIs to develop optical character recognition (OCR) applications. OCR is the process by which the machine vision software reads text and/or characters in an image.
Instrument Readers	Use the Instrument Reader VIs to develop applications that require reading from seven-segment displays, meters or gauges, or one-dimensional barcodes.

# **Select Region of Interest**

Owning Palette: Machine VisionInstalled With: NI Vision Development

Module

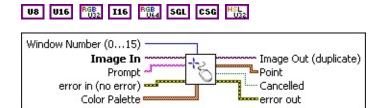
Use the Select Region of Interest VIs to select a specific area of an image from an external display window.

Palette Object	Description
IMAQ Select Point	Allows the user to specify the position of a point in the image. IMAQ Select Point displays the image in the specified window and provides a point tool.
IMAQ Select Line	Allows the user to specify the location of a line in the image. IMAQ Select Line displays the image in the specified window, and provides a line tool.
IMAQ Select Rectangle	Allows the user to specify a rectangular area in the image. IMAQ Select Rectangle displays the image in the specified window and provides the rectangle and rotated rectangle tools.
IMAQ Select Annulus	Allows the user to specify an annulus area in an image. IMAQ Select Annulus displays the image in the specified window, provides the annulus tool, and returns the coordinates of the annulus selected when the user clicks <b>OK</b> in the window.

### **IMAQ Select Point VI**

Owning Palette: <u>Select Region of Interest</u>Installed With: NI Vision Development Module

Allows the user to specify the position of a point in the image. IMAQ Select Point displays the image in the specified window and provides a point tool. IMAQ Select Point returns the coordinates of the point selected when the user clicks **OK** in the window.



- Window Number (0...15) specifies the window in which to display the image. When you select the default value of ?1, this VI uses a modal dialog window, centered in the screen. When you select a regular NI Vision window number (0?15), the VI displays Image In in the specified window and temporarily sets the NI Vision window to modal mode. When the user clicks OK or Cancel in the window, the attributes of the window are set back to their initial values.
- Image In is a reference to the image on which the user selects the **Point**.
- **Prompt** specifies a message string to display in the title bar of the window. Use this control to provide the user with instructions about selecting the object.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette is used to apply a color palette to the image window. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - **Tip** For best results, set your video adapter to high color or true color.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - **Blue** is the value of the blue color plane.
- Image Out (duplicate) is a reference to Image In. This VI does not modify the image connected to the Image In input.
- **Point** is a cluster that specifies the coordinates of the point chosen by the user.
  - **X** is the x-coordinate of the point.
  - Y is the y-coordinate of the point.
- **Cancelled** returns TRUE if the user ends the selection by clicking **Cancel** in the window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

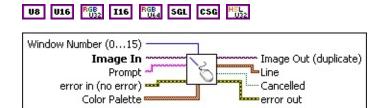
same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Select Line VI**

Owning Palette: <u>Select Region of Interest</u>Installed With: NI Vision Development Module

Allows the user to specify the location of a line in the image. IMAQ Select Line displays the image in the specified window, and provides a line tool. IMAQ Select Line returns the coordinates of the line selected when the user clicks **OK** in the window.



- Window Number (0...15) specifies the window in which to display the image. When you select the default value of ?1, this VI uses a modal dialog window, centered in the screen. When you select a regular NI Vision window number (0?15), the VI displays Image In in the specified window and temporarily sets the NI Vision window to modal mode. When the user clicks OK or Cancel in the window, the attributes of the window are set back to their initial values.
- Image In is a reference to the image on which the user selects the Line.
- **Prompt** specifies a message string to display in the title bar of the window. Use this control to provide the user with instructions about selecting the object.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

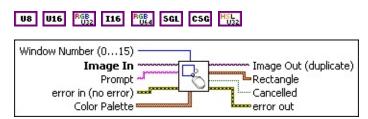
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette is used to apply a color palette to the image window. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - **Tip** For best results, set your video adapter to high color or true color.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - **Blue** is the value of the blue color plane.
- Image Out (duplicate) is a reference to Image In. This VI does not modify the image connected to the Image In input.
- Line specifies the coordinates of the line chosen by the user.
  - Start X specifies the x-coordinate of the beginning point of the line.
  - Start Y specifies the y-coordinate of the beginning point of the line.
  - **End X** specifies the x-coordinate of the end point of the line.
  - **End Y** specifies the y-coordinate of the end point of the line.

- Cancelled returns TRUE if the user ends the selection by clicking Cancel in the window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Select Rectangle VI**

Owning Palette: <u>Select Region of Interest</u>Installed With: NI Vision Development Module

Allows the user to specify a rectangular area in the image. IMAQ Select Rectangle displays the image in the specified window and provides the rectangle and rotated rectangle tools. IMAQ Select Rectangle returns the coordinates of the rectangle selected when the user clicks **OK** in the window.



- Window Number (0...15) specifies the window in which to display the image. When you select the default value of ?1, this VI uses a modal dialog window, centered in the screen. When you select a regular NI Vision window number (0?15), the VI displays Image In in the specified window and temporarily sets the NI Vision window to modal mode. When the user clicks OK or Cancel in the window, the attributes of the window are set back to their initial values.
- Image In is a reference to the image on which the user selects the Rectangle.
- **Prompt** specifies a message string to display in the title bar of the window. Use this control to provide the user with instructions about selecting the object.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

**status** is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

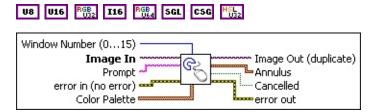
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette is used to apply a color palette to the image window. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - **Tip** For best results, set your video adapter to high color or true color.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - Blue is the value of the blue color plane.
- Image Out (duplicate) is a reference to Image In. This VI does not modify the image connected to the Image In input.
- **Rectangle** specifies the coordinates of the rectangle chosen by the user.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.

- **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
- Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- **Cancelled** returns TRUE if the user ends the selection by clicking **Cancel** in the window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Select Annulus VI**

Owning Palette: <u>Select Region of Interest</u>Installed With: NI Vision Development Module

Allows the user to specify an annulus area in an image. IMAQ Select Annulus displays the image in the specified window, provides the annulus tool, and returns the coordinates of the annulus selected when the user clicks **OK** in the window.



- Window Number (0...15) specifies the window in which to display the image. When you select the default value of ?1, this VI uses a modal dialog window, centered in the screen. When you select a regular NI Vision window number (0?15), the VI displays Image In in the specified window and temporarily sets the NI Vision window to modal mode. When the user clicks OK or Cancel in the window, the attributes of the window are set back to their initial values.
- Image In is a reference to the image on which the user selects the Annulus.
- **Prompt** specifies a message string to display in the title bar of the window. Use this control to provide the user with instructions about selecting the object.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Color Palette is used to apply a color palette to the image window. Color Palette is an array of clusters constructed by the user or supplied by the IMAQ GetPalette VI. This palette is composed of 256 elements for each of the three color planes (red, green, and blue). A specific color is the result of applying a value between 0 and 255 to each of the three color planes. If the three planes have identical values, a gray level is obtained (0 specifies black and 255 specifies white). If the image type requires a color palette and it is not supplied, a grayscale color palette is generated and written.
  - **Tip** For best results, set your video adapter to high color or true color.
  - Red is the value of the red color plane.
  - Green is the value of the green color plane.
  - Blue is the value of the blue color plane.
- Image Out (duplicate) is a reference to Image In. This VI does not modify the image connected to the Image In input.
- Annulus specifies the coordinates of the annulus chosen by the user.
  - **Center X** specifies the x-coordinate of the center of the circle or annulus.
  - Center Y specifies the y-coordinate of the center of the circle or annulus.
  - Inner Radius is the radius (in pixels) of the inner circle defining one edge of the circular strip.

- **Outer Radius** is the radius (in pixels) of the outer circle defining the second edge of the circular strip.
- Start Angle is the first angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
- **End Angle** is the second angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
- **Cancelled** returns TRUE if the user ends the selection by clicking **Cancel** in the window.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Coordinate System**

Owning Palette: Machine VisionInstalled With: NI Vision Development

Module

Use the Coordinate System VIs to find various types of coordinate systems in an image. Use these VIs to find the coordinate system using either edge detection or pattern matching.

Palette Object	Description
IMAQ Find CoordSys (Rect) 2	Computes a coordinate system based on the position of an object in a search area of an image.
IMAQ Find CoordSys (2 Rects) 2	Computes a coordinate system based on the position of an object in two search areas of an image.
IMAQ Find CoordSys (Pattern) 2	Searches for a template image in a rectangular search area of the input Image.

## **IMAQ Find CoordSys (Rect) 2 VI**

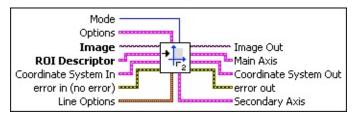
Owning Palette: Coordinate SystemInstalled With: NI Vision

**Development Module** 

Computes a coordinate system based on the position of an object in a search area of an image. The location and orientation of the coordinate system found is used to create the reference position of a coordinate system or to update the current location and orientation of an existing Coordinate System. The function can overlay the following on the returned image: the position of the search area, the search lines, the edges found, and the location and orientation of the coordinate system found.

#### **Details**

U8 I16 SGL



Mode specifies the function that is performed by this VI. You can choose from the following values:

Find Reference (0)	Specifies that the location and orientation of the found edges set the reference system of <b>Coordinate System Out</b> . This is typically the first mode to use when setting up a coordinate system.
Update CoordSys (1)	Specifies that the location and orientation of the found edges set the measurement system of Coordinate System Out. The reference system of Coordinate System In remains unchanged in Coordinate System Out. Use this mode with each new inspection image to update the location and orientation of the edges defining the coordinate system.

Options is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image.

**Direction** specifies the order and direction in which the edges are searched. The arrow in the **Direction** symbol specifies the search direction for the **Main Axis**, while the perpendicular sign determines the position of the perpendicular axis.

Value	Main Axis	<b>Secondary Axis</b>
0	Horizontal – Left to Right	Bottom to Top
1	Horizontal – Left to Right	Top to Bottom
2	Vertical – Top to Bottom	Left to Right
3	Vertical – Top to Bottom	Right to Left
4	Horizontal – Right to Left	Top to Bottom
5	Horizontal – Right to Left	Bottom to Top
6	Vertical – Bottom to Top	Right to Left
7	Vertical – Bottom to Top	Left to Right

**Edge Options** specifies the parameters that are used to compute the edge gradient information and detect the edges.

**Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
Falling Edges (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- Width specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- Minimum Edge Strength specifies the minimum edge strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the following options:

1

Zero Order (0)	Rounds to the nearest integral edge location
Bilinear (1)	Uses bilinear interpolation to compute the edge location
Bilinear Fixed (4)	(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge location

**Data Processing Method** is the method used to process the data extracted for edge detection.

• ,	(Default) Averages the data extracted for edge detection
	Takes the median of the data extracted for edge detection

- Show Search Area determines whether the search area is overlaid on the image.
- Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
- Show Edges Found determines whether the locations of the edges found are overlaid on the result image.
- Show Result determines whether the coordinate system found is overlaid on the result image.
- Search Area Color specifies the color to use to overlay the search area.
- Search Lines Color specifies the color to use to overlay the search lines.
- **Edge Locations Color** specifies the color to use to overlay the edge locations.
- Result Color specifies the color to use to overlay the result.
- Overlay Group Name specifies the overlay group name for the step overlays.
- **Image** is a reference to the image in which the coordinate system is to be located.

- **ROI Descriptor** is a descriptor that defines the rectangle, rotated rectangle, or other four-side rectangular polygon within which the edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Coordinate System In specifies the coordinate system to update. This input is required only when Mode is set to Update CoordSys. When Mode is set to Find Reference, the VI ignores this parameter.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

- Line Options specifies the options that are used to detect the straight edges.
  - Number of Lines specifies the number of straight edges to find. The default is 1.
  - **Type** specifies the method used to find the straight edge. Choose from the following options:

First Edge Rake (0)	Fits a straight edge on the first points detected using a Rake
Best Edge Rake (1)	Fits a straight edge on the best points detected using a Rake
Hough Edge Rake (2)	Finds the strongest straight edge using all points detected on a Rake
First Edge Projection (3)	Uses the location of the first projected edge as the straight edge
Best Edge Projection (4)	(Default) Finds the strongest projected edge location to determine the straight edge

- Min. Score specifies the minimum score of a detected straight edge. The default is 10.
- **Max. Score** specifies the maximum score of a detected straight edge. The default is 1000.
- Orientation specifies the angle at which the straight edge is expected to be found. The default is 0.
- Angle Range specifies the positive and negative range around the **Orientation** within which the straight edge is expected to be found. The default is 10.
- Angle Tolerance specifies the expected angular accuracy of the straight edge. The default is 1.
- Step Size specifies the gap in pixels between the search lines used with the Rake-based methods. The default is 3.
- Min. SNR (db) specifies the minimum signal to noise ratio (SNR) of the edge points used to fit the straight edge. The default is 0.

- Min. Points (%) specifies the minimum number for points as a percentage of the number of search lines that need to be included in the detected straight edge. The default is 25%.
- Hough Iterations specify the number of iterations used in the Hough-based method. The default is 5.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- Main Axis returns a cluster of two point coordinates specifying the coordinates of the edge that corresponds to the Main Axis. The first point is the intersection of the two axes of the coordinate system. The second point is the result of the computation of the intersection of the Main Axis with the ROI Descriptor.
- **Coordinate System Out** is a cluster returning the position of the coordinate system.
  - Reference System is a cluster containing the following elements.
    - **Origin** is a point cluster that specifies the origin of the base-reference coordinate system.
      - **X** is the x-coordinate of the point.
      - Y is the y-coordinate of the point.
    - Angle (deg) is the angle the base-reference coordinate system makes with the image coordinate system.
    - Axis Reference is the direction of the y-axis of the base-reference coordinate system.
  - Measurement System is a cluster containing the following elements.
    - Origin is a point cluster that specifies the origin of the base-reference coordinate system.
      - **X** is the x-coordinate of the point.
      - Y is the y-coordinate of the point.
    - Angle (deg) is the angle the base-reference coordinate system makes with the image coordinate system.

- **Axis Reference** is the direction of the y-axis of the base-reference coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Secondary Axis returns a cluster of two point coordinates specifying the coordinates of the edge that corresponds to the Secondary Axis. The first point is the intersection of the two axis of the coordinate system. The second point is the result of the computation of the intersection of the Secondary Axis with the ROI Descriptor.

#### **Details**

The VI uses the following algorithm. First, the VI determines the position of the Main Axis of the coordinate system. It locates the intersection points between a set of parallel search lines, or rake, and the edge of an object. The intersection points are determined based on their edge strength. The VI calculates a best-fit line using the points found. This line defines the Main Axis of the coordinate system. The VI then locates the intersection points between a set of parallel search lines, or rake, perpendicular to the Main Axis and the edge of the object. It calculates a hit-line to the object from the edge closest to the search area detected and perpendicular to the Main Axis. This line defines the Secondary Axis of the coordinate system. The intersection between the Main Axis and Secondary Axis is the origin of the coordinate system.

The VI sets the reference position and orientation of the coordinate system or updates its position depending on the **Mode** chosen.

### **IMAQ Find CoordSys (2 Rects) 2 VI**

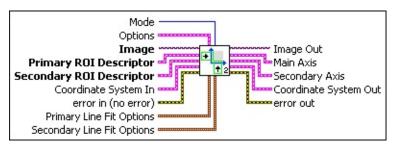
Owning Palette: Coordinate SystemInstalled With: NI Vision

**Development Module** 

Computes a coordinate system based on the position of an object in two search areas of an image. The location and orientation of the coordinate system found is used to create the reference position of a Coordinate System or to update the current location and orientation of an existing Coordinate System. The function has the capability to overlay the following on the returned image: the position of the search area, the search lines, the edges found, and the location and orientation of the Coordinate System found.

### **Details** Examples





Mode specifies the function that is performed by this VI. You can choose from the following values:

Find Reference (0)	Specifies that the location and orientation of the found edges set the reference system of <b>Coordinate System Out</b> . This is typically the first mode to use when setting up a coordinate system.
Update CoordSys (1)	Specifies that the location and orientation of the found edges set the measurement system of Coordinate System Out. The reference system of Coordinate System In remains unchanged in Coordinate System Out. Use this mode with each new inspection image to update the location and orientation of the edges defining the coordinate system.

**Options** is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image.

**Direction** specifies the order and direction in which the edges are searched. The arrow in the **Direction** symbol specifies the search direction for the **Main Axis**, while the perpendicular sign determines the position of the perpendicular axis.

Value	Main Axis	Secondary Axis
0	Horizontal – Left to Right	Bottom to Top
1	Horizontal – Left to Right	Top to Bottom
2	Vertical – Top to Bottom	Left to Right
3	Vertical – Top to Bottom	Right to Left
4	Horizontal – Right to Left	Top to Bottom
5	Horizontal – Right to Left	Bottom to Top
6	Vertical – Bottom to Top	Right to Left
7	Vertical – Bottom to Top	Left to Right

- Primary Edge Options specifies the parameters that are used to compute the edge gradient information and detect the edges along the Primary ROI Descriptor.
  - **Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
<b>Falling Edges</b> (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- **Width** specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- Minimum Edge Strength specifies the minimum edge strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the

### following options:

Zero Order (0)	Rounds to the nearest integral edge location
Bilinear (1)	Uses bilinear interpolation to compute the edge location
Bilinear Fixed (4)	(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge location

**Data Processing Method** is the method used to process the data extracted for edge detection.

• , ,	(Default) Averages the data extracted for edge detection
` '	Takes the median of the data extracted for edge detection

- Secondary Edge Options specifies the parameters that are used to compute the edge gradient information and detect the edges along the Secondary ROI Descriptor.
  - **Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
<b>Falling Edges</b> (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- **Width** specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- Minimum Edge Strength specifies the minimum edge strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the

#### following options:

Zero Order (0)	Rounds to the nearest integral edge location
Bilinear (1)	Uses bilinear interpolation to compute the edge location
Bilinear Fixed (4)	(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge location

**Data Processing Method** is the method used to process the data extracted for edge detection.

• ,	(Default) Averages the data extracted for edge detection
` '	Takes the median of the data extracted for edge detection

- Show Search Area determines whether the search area is overlaid on the image.
- Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
- Show Edges Found determines whether the locations of the edges found are overlaid on the result image.
- Show Result determines whether the coordinate system found is overlaid on the result image.
- Search Area Color specifies the color to use to overlay the search area.
- Search Lines Color specifies the color to use to overlay the search lines.
- Edge Locations Color specifies the color to use to overlay the edge locations.
- Result Color specifies the color to use to overlay the result.
- Overlay Group Name specifies the overlay group name for the step overlays.
- Image is a reference to the image in which the coordinate system is

to be located.

- Primary ROI Descriptor is a descriptor that defines the Region of Interest (ROI) within which the edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Secondary ROI Descriptor is a descriptor that defines the Region of Interest (ROI) within which the edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Coordinate System In specifies the coordinate system to update. This input is required only when **Mode** is set to **Update CoordSys**. When **Mode** is set to **Find Reference**, the VI ignores this parameter.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Primary Line Fit Options specifies the options that are used to detect the straight edges.
  - Number of Lines specifies the number of straight edges to find. The default is 1.
  - Type specifies the method used to find the straight edge. Choose from the following options:

First Edge Rake (0)	Fits a straight edge on the first points detected using a Rake
Best Edge Rake (1)	Fits a straight edge on the best points detected using a Rake
Hough Edge Rake (2)	Finds the strongest straight edge using all points detected on a Rake
First Edge Projection (3)	Uses the location of the first projected edge as the straight edge
Best Edge Projection (4)	(Default) Finds the strongest projected edge location to determine the straight edge

- Min. Score specifies the minimum score of a detected straight edge. The default is 10.
- Max. Score specifies the maximum score of a detected straight edge. The default is 1000.

- **Orientation** specifies the angle at which the straight edge is expected to be found. The default is 0.
- Angle Range specifies the positive and negative range around the **Orientation** within which the straight edge is expected to be found. The default is 10.
- Angle Tolerance specifies the expected angular accuracy of the straight edge. The default is 1.
- Step Size specifies the gap in pixels between the search lines used with the Rake-based methods. The default is 7.
- Min. SNR (db) specifies the minimum signal to noise ratio (SNR) of the edge points used to fit the straight edge. The default is 0.
- Min. Points (%) specifies the minimum number for points as a percentage of the number of search lines that need to be included in the detected straight edge. The default is 25%.
- Hough Iterations specify the number of iterations used in the Hough-based method. The default is 5.
- Secondary Line Fit Options specifies the options that are used to detect the straight edges.
  - Number of Lines specifies the number of straight edges to find. The default is 1.
  - **Type** specifies the method used to find the straight edge. Choose from the following options:

First Edge Rake (0)	Fits a straight edge on the first points detected using a Rake
Best Edge Rake (1)	Fits a straight edge on the best points detected using a Rake
Hough Edge Rake (2)	Finds the strongest straight edge using all points detected on a Rake
First Edge Projection (3)	Uses the location of the first projected edge as the straight edge
Best Edge Projection (4)	(Default) Finds the strongest projected edge location to determine the straight edge

- Min. Score specifies the minimum score of a detected straight edge. The default is 10.
- Max. Score specifies the maximum score of a detected straight edge. The default is 1000.
- Orientation specifies the angle at which the straight edge is expected to be found. The default is 0.
- Angle Range specifies the positive and negative range around the **Orientation** within which the straight edge is expected to be found. The default is 10.
- Angle Tolerance specifies the expected angular accuracy of the straight edge. The default is 1.
- Step Size specifies the gap in pixels between the search lines used with the Rake-based methods. The default is 7.
- Min. SNR (db) specifies the minimum signal to noise ratio (SNR) of the edge points used to fit the straight edge. The default is 0.
- Min. Points (%) specifies the minimum number for points as a percentage of the number of search lines that need to be included in the detected straight edge. The default is 25%.
- Hough Iterations specify the number of iterations used in the Hough-based method. The default is 5.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- Main Axis returns a cluster of two point coordinates specifying the coordinates of the edge that corresponds to the Main Axis. The first point is the intersection of the two axes of the coordinate system. The second point is the result of the computation of the intersection of the Main Axis with the Primary ROI Descriptor.
- Secondary Axis returns a cluster of two point coordinates specifying the coordinates of the edge that corresponds to the Secondary Axis. The first point is the intersection of the two axis of the coordinate system. The second point is the result of the computation of the intersection of the Secondary Axis with the Secondary ROI Descriptor.

- **Coordinate System Out** is a cluster returning the position of the coordinate system.
  - Reference System is a cluster containing the following elements.
    - **Origin** is a point cluster that specifies the origin of the base-reference coordinate system.
      - **X** is the x-coordinate of the point.
      - **Y** is the y-coordinate of the point.
    - Angle (deg) is the angle the base-reference coordinate system makes with the image coordinate system.
    - **Axis Reference** is the direction of the y-axis of the base-reference coordinate system.
  - **Measurement System** is a cluster containing the following elements.
    - **Origin** is a point cluster that specifies the origin of the base-reference coordinate system.
      - **X** is the x-coordinate of the point.
      - **Y** is the y-coordinate of the point.
    - Angle (deg) is the angle the base-reference coordinate system makes with the image coordinate system.
    - **Axis Reference** is the direction of the y-axis of the base-reference coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a

warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The VI uses the following algorithm. First, the VI determines the position of the **Main Axis** of the coordinate system. It locates the intersection points between a set of parallel search lines, in the **Primary ROI Descriptor**, and the edge of an object. The intersection points are determined based on their edge strength. The VI calculates a best-fit line through the points found. This line defines the **Main Axis** of the coordinate system. The process is repeated perpendicularly in the **Secondary ROI Descriptor** in order to locate the **Secondary Axis**. The intersection between the **Main Axis** and **Secondary Axis** is the origin of the coordinate system.

The VI sets the reference position and orientation of the coordinate system or updates its position depending on the **Mode** chosen.

### **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Fuse Inspection Example.llb

## IMAQ Find CoordSys (Pattern) 2 VI

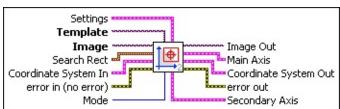
Owning Palette: Coordinate SystemInstalled With: NI Vision

**Development Module** 

Searches for a template image in a rectangular search area of the input Image. The VI uses the location and orientation of the template found to create the reference position of a coordinate system or to update the current location and orientation of an existing coordinate system. The VI has the capability to overlay on the returned image the position of the search area and the location and orientation of the coordinate system found.

#### **Examples**





- Settings is a cluster defining the parameters of the template location algorithm and the information that is overlaid on the result image.
  - Match Mode specifies the technique to use when looking for the template pattern in the image. Choose from the following values:

Invariant (0)	(Default) Searches for the template pattern in the image, assuming it is not rotated more than 4 degrees.
	Searches for the template in the image with no restriction on the rotation of the template.

- Minimum Match Score is the smallest score a match must have to be considered valid.
- SubPixel Accuracy determines whether the match results should be returned with subpixel accuracy.
- Rotation Angle Ranges is an array of angle ranges where each range specifies how much you expect the template

- pattern to rotate in the image. Each range is specified by a lower angle and an upper angle, both of which are expressed in degrees.
- Show Search Area specifies whether the centers and bounding rectangles of the patterns found are overlaid on the result image.
- Show Feature Found specifies whether the position and orientation of the Coordinate System found are overlaid on the result image.
- Show Result specifies whether the position and orientation of the Coordinate System found are overlaid on the result image.
- **Template** is a reference to the image to locate during the match process. The template image is the image obtained from the output of the <a href="IMAQ Learn Pattern 2">IMAQ Learn Pattern 2</a> VI. If the template has not been learned, this VI first goes through the learning process.
- Image is a reference to the source image.
- Search Rect specifies a rectangular search area in an image. The Left, Top, Right, and Bottom coordinates describe a non-rotated rectangle. The Rotation specifies the rotation angle of the rectangle around its center.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.

- Coordinate System In specifies the coordinate system to update. This input is required only when Mode is set to Update CoordSys. When Mode is set to Find Reference, the VI ignores this parameter.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Mode specifies the function that is performed by this VI. You can choose from the following values:

Find Reference (0)	Specifies that the location and orientation of the found edges set the reference system of <b>Coordinate System Out</b> . This is typically the first mode to use when setting up a coordinate system.
Update CoordSys (1)	Specifies that the location and orientation of the found edges set the measurement system of Coordinate System Out. The reference system of Coordinate System In remains unchanged in Coordinate System Out. Use this mode with each new inspection image to update the location and orientation of the edges defining the coordinate system.

Image Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst. Otherwise, Image Dst Out refers to the image referenced by Image Src.

- Main Axis returns a cluster of two point coordinates specifying the coordinates of the edge that corresponds to the Main Axis. The first point is the intersection of the two axes of the coordinate system. The second point is the result of the computation of the intersection of the Main Axis with the Primary Rectangle.
- **Coordinate System Out** is a cluster returning the position of the coordinate system.

When Mode is set to **Find Reference**, the location and orientation of the located coordinate system are returned as the **Reference System** in the **Coordinate System Out** cluster. The **Measurement System** values are set to the same value.

When Mode is set to **Update CoordSys**, the VI sets the **Measurement System** values of the **Coordinate System Out** cluster to the location and orientation of the located coordinate system.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Secondary Axis** returns a cluster of two point coordinates

specifying the coordinates of the edge that corresponds to the **Secondary Axis**. The first point is the intersection of the two axis of the coordinate system. The second point is the result of the computation of the intersection of the **Secondary Axis** with the **Secondary Rectangle**.

### **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Battery Clamp Inspection Example.vi

# **Count and Measure Objects**

Owning Palette: Machine VisionInstalled With: NI Vision Development

Module

Use the IMAQ Count Objects VI to count and measure objects.

Palette Object	Description
Count	Locates, counts, and measures objects in a rectangular search area. This VI uses a threshold on the pixel intensities to segment the objects from their background.

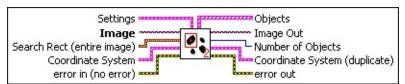
### **IMAQ Count Objects 2 VI**

Owning Palette: <u>Count and Measure Objects</u>Installed With: NI Vision Development Module

Locates, counts, and measures objects in a rectangular search area. This VI uses a threshold on the pixel intensities to segment the objects from their background.

#### **Details**





- Settings is a cluster defining the parameters of the algorithm and the information that is overlaid on the result image.
  - Objects specifies the type of objects to detect.

<b>Bright Objects</b> (0)	Searches for bright objects.
Dark Objects (1)	Searches for dark objects.

- Threshold specifies the grayscale intensity that is used as threshold level. When a **Bright Objects** type is selected, the threshold range used includes **Threshold** to the maximum possible intensity depending of the image type (255 for a 8-bit image). When a **Dark Objects** type is selected, the threshold range used includes the minimum possible intensity depending of the image type (0 for a 8-bit image) up to **Threshold**.
- Ignore Objects touching Image Borders specifies whether the binary objects touching the borders of the search area are ignored.
- Fill Holes within Objects specifies whether the holes in the binary objects are ignored.
- Minimum Object Size specifies whether the detected objects smaller than Min Size are ignored.
- Min Size specifies the size of the objects to ignore. When the

- parameter **Minimum Objects Size** is set to TRUE, the objects which size is smaller than or equal to **Min Size** are ignored.
- Maximum Object Size specifies whether the detected objects larger than Max Size are ignored.
- Max Size specifies the size of the objects to ignore. When the parameter Maximum Object Size is set to TRUE, the objects which size is larger than or equal to Max Size are ignored.
- Show Search Area determines whether the search area is overlaid on the image.
- Show Center determines whether the locations of the objects found are overlaid on the image.
- Show Bounding Box determines whether the bounding rectangles of the objects found are overlaid on the image.
- **Image** is a reference to the source image.
- Search Rect (entire image) specifies the rectangular region of the image where the objects are located. The default value specifies a search on the entire image.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- Coordinate System specifies the coordinate system to which the Search Rect is linked. If the Coordinate System input is used

(connected), the location of the **Search Rect** is shifted and rotated by the difference between the reference position of the coordinate system and its new location.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Objects** is an array of clusters that return the following elements for each object located.
  - Object Center is a point-coordinate cluster specifying the location of the center of mass of the binary object detected.
  - **Bounding Box** is the bounding rectangle of the binary object.
  - Area is the area of the binary object detected.
  - Orientation returns the orientation of the longest segment in the binary object.
  - Aspect Ratio is the ratio between the width and the height of the binary object.
  - Nb Holes returns the number of holes found in the binary object.

- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- Number of Objects is the number of objects located in the search area based on the input **Settings**.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

Optional filters give the capability to ignore the objects smaller or larger than given sizes. Other options allow rejecting the objects touching the borders of the search area and ignoring the holes that the segmentation process may create in the objects. The segmented objects are then located and measured. IMAQ Count Objects 2 can overlay on the image returned the position of the search area, the centers and bounding rectangles of the objects detected.

### **Measure Intensities**

Owning Palette: Machine VisionInstalled With: NI Vision Development

Module

Use the Measure Intensities VIs to measure the intensity of a pixel on a point or the intensity statistics of pixels along a line or in a rectangular region of an image.

Palette Object	Description
IMAQ Light Meter (Point)	Measures the pixel intensities in a 3 x 3 pixel neighborhood centered on a point of an image.
IMAQ Light Meter (Line)	Measures the pixel intensities on a line of an image.
IMAQ Light Meter (Rectangle)	Measures the pixel intensities in a rectangle of an image.

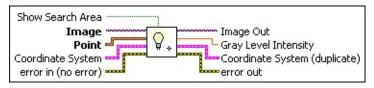
### **IMAQ Light Meter (Point) VI**

Owning Palette: Measure Intensities Installed With: NI Vision

**Development Module** 

Measures the pixel intensities in a  $3 \times 3$  pixel neighborhood centered on a point of an image.





- Show Search Area specifies whether the VI overlays the location of the intensity measurement on the image.
- **Image** is a reference to the source image.
- **Point** is a cluster that specifies the coordinates of the intensity measurement. The intensity measurement is made in a 3 × 3 pixel neighborhood centered on the Point coordinates.
  - **X** is the x-coordinate of the point.
  - Y is the y-coordinate of the point.
- Coordinate System specifies the coordinate system to which the Point is linked. If the Coordinate System input is used (connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image Out is a reference to the result image. The search area may be overlaid on the image according to the setting Show Search Area. The search area may be overlaid on the image according to the setting Show Search Area.
- Gray Level Intensity returns the average intensity of the pixels in a 3 x 3 neighborhood centered on **Point**.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

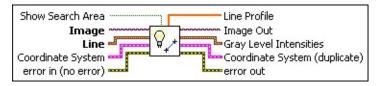
# **IMAQ Light Meter (Line) VI**

Owning Palette: Measure Intensities Installed With: NI Vision

**Development Module** 

Measures the pixel intensities on a line of an image.





- Show Search Area specifies whether the VI overlays the location of the intensity measurement on the image.
- Image is a reference to the source image.
- **Line** is a cluster of two point coordinates that define a line on which the VI measures pixel intensities.
  - Start X specifies the x-coordinate of the beginning point of the line.
  - Start Y specifies the y-coordinate of the beginning point of the line.
  - **End X** specifies the x-coordinate of the end point of the line.
  - **End Y** specifies the y-coordinate of the end point of the line.
- Coordinate System specifies the coordinate system to which the Line is linked. If the Coordinate System input is used (connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Line Profile returns an array of the intensities of the pixels found along the Line.
- Image Out is a reference to the result image. The search area may be overlaid on the image according to the setting Show Search Area. The search area may be overlaid on the image according to the setting Show Search Area.
- Gray Level Intensities is a cluster that contains the intensity measurements.
  - Mean Intensity is the average of the pixel intensities.
  - **Standard Deviation** is the standard deviation of the pixel intensities.
  - Minimum Intensity is the minimum of the pixel intensities.
  - Maximum Intensity is the maximum of the pixel intensities.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Light Meter (Rectangle) VI**

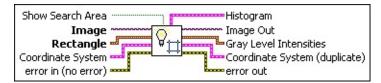
Owning Palette: Measure Intensities Installed With: NI Vision

**Development Module** 

Measures the pixel intensities in a rectangle of an image.

#### **Examples**





- Show Search Area specifies whether the VI overlays the location of the intensity measurement on the image.
- Image is a reference to the source image.
- Rectangle specifies the coordinates of a rectangular area in which the intensity is measured.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- Coordinate System specifies the coordinate system to which the Rectangle is linked. If the Coordinate System input is used (connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the

coordinate system and its new location.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Histogram** contains the histogram of the pixel intensities in the Rectangle area.
- Image Out is a reference to the result image. The search area may be overlaid on the image according to the setting Show Search Area. The search area may be overlaid on the image according to the setting Show Search Area.
- Gray Level Intensities is a cluster that contains the intensity measurements.
  - **Mean Intensity** is the average of the pixel intensities.
  - **Standard Deviation** is the standard deviation of the pixel intensities.
  - Minimum Intensity is the minimum of the pixel intensities.
  - Maximum Intensity is the maximum of the pixel intensities.

**Coordinate System (duplicate)** is a reference to the coordinate system.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Light Meter\Light Meter Example.vi

#### **Measure Distances**

Owning Palette: Machine VisionInstalled With: NI Vision Development Module

Use the Measure Distances VIs to measure distances in an image, such as the minimum and maximum horizontal separation between two vertically oriented edges or the minimum or maximum vertical separation between two horizontally oriented edges.

Palette Object	Description
IMAQ Clamp Horizontal Max	Measures a distance in the horizontal direction, from the vertical sides of the search area towards the center of the search area.
IMAQ Clamp Horizontal Min	Measures a distance in the horizontal direction, from the center of the search area towards the vertical sides of the search area.
IMAQ Clamp Vertical Max	Measures a distance in the vertical direction, from the horizontal sides of the search area towards the center of the search area. T
IMAQ Clamp Vertical Min	Measures a distance in the vertical direction, from the center of the search area towards the horizontal sides of the search area.

### **IMAQ Clamp Horizontal Max VI**

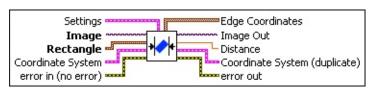
Owning Palette: Measure Distances Installed With: NI Vision

**Development Module** 

Measures a distance in the horizontal direction, from the vertical sides of the search area towards the center of the search area. This VI locates edges along a set of parallel search lines, or rake. The edges are determined based on their contrast and slope.

#### **Details** Examples





- Settings is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image. The first three parameters specify the filter used to detect the edges.
  - Contrast specifies the threshold for the contrast of the edge.
    Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - Filter width specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
  - Subsampling Ratio specifies the number of pixels that separate two consecutive search lines of the rake.
  - Show Search Area determines whether the search area is overlaid on the image.
  - Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
  - Show Edges Found determines whether the locations of the

- edges found are overlaid on the result image.
- Show Result determines whether the hit lines to the object are overlaid on the result image.
- **Image** is a reference to the source image.
- Rectangle specifies the coordinates of a rectangular search area.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- Coordinate System specifies the coordinate system to which the Rectangle is linked. If the Coordinate System input is used (connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Edge Coordinates** is an array of point clusters consisting of the spatial coordinates of the detected edges.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- Distance returns the distance measured between the hit lines.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

A vertical hit-line to the object is calculated through the leftmost edge detected. A second vertical hit-line to the object is calculated through the rightmost edge. The distance between those two lines is returned. IMAQ Clamp Horizontal Max can overlay on the image returned: the position of the search area, the search lines, the edges found, and the result.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Caliper\Clamp Example.vi

### **IMAQ Clamp Horizontal Min VI**

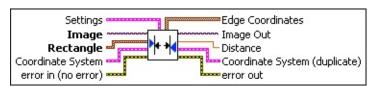
Owning Palette: Measure Distances Installed With: NI Vision

**Development Module** 

Measures a distance in the horizontal direction, from the center of the search area towards the vertical sides of the search area. This VI locates edges along a set of parallel search lines, or rake. The edges are determined based on their contrast and slope.

#### **Details**





- Settings is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image. The first three parameters specify the filter used to detect the edges.
  - Contrast specifies the threshold for the contrast of the edge.
    Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - Filter width specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
  - Subsampling Ratio specifies the number of pixels that separate two consecutive search lines of the rake.
  - Show Search Area determines whether the search area is overlaid on the image.
  - Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
  - Show Edges Found determines whether the locations of the

- edges found are overlaid on the result image.
- Show Result determines whether the hit lines to the object are overlaid on the result image.
- **Image** is a reference to the source image.
- Rectangle specifies the coordinates of a rectangular search area.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- Coordinate System specifies the coordinate system to which the Rectangle is linked. If the Coordinate System input is used (connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Edge Coordinates** is an array of point clusters consisting of the spatial coordinates of the detected edges.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- Distance returns the distance measured between the hit lines.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

A vertical hit-line to the object is calculated through the leftmost edge detected in the right half of the search area. A second vertical hit-line to the object is calculated through the rightmost edge in the left half of the search area. The distance between those two lines is returned. IMAQ Clamp Horizontal Min can overlay on the image returned: the position of the search area, the search lines, the edges found, and the result.

### **IMAQ Clamp Vertical Max VI**

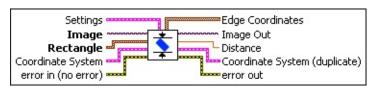
Owning Palette: Measure Distances Installed With: NI Vision

**Development Module** 

Measures a distance in the vertical direction, from the horizontal sides of the search area towards the center of the search area. T his VI locates edges along a set of parallel search lines, or rake. The edges are determined based on their contrast and slope.

#### **Details** Examples





- Settings is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image. The first three parameters specify the filter used to detect the edges.
  - Contrast specifies the threshold for the contrast of the edge.
    Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - Filter width specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
  - Subsampling Ratio specifies the number of pixels that separate two consecutive search lines of the rake.
  - Show Search Area determines whether the search area is overlaid on the image.
  - Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
  - Show Edges Found determines whether the locations of the

- edges found are overlaid on the result image.
- Show Result determines whether the hit lines to the object are overlaid on the result image.
- **Image** is a reference to the source image.
- Rectangle specifies the coordinates of a rectangular search area.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- Coordinate System specifies the coordinate system to which the Rectangle is linked. If the Coordinate System input is used (connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Edge Coordinates** is an array of point clusters consisting of the spatial coordinates of the detected edges.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- Distance returns the distance measured between the hit lines.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

A horizontal hit-line to the object is calculated through the uppermost edge detected. A second horizontal hit-line to the object is calculated through the lowermost edge. The distance between those two lines is returned. IMAQ Clamp Vertical Max can overlay on the image returned: the position of the search area, the search lines, the edges found and the result.

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Battery Clamp Inspection Example.vi

### **IMAQ Clamp Vertical Min VI**

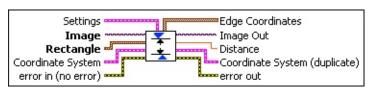
Owning Palette: Measure Distances Installed With: NI Vision

**Development Module** 

Measures a distance in the vertical direction, from the center of the search area towards the horizontal sides of the search area. This VI locates edges along a set of parallel search lines, or rake. The edges are determined based on their contrast and slope.

#### **Details**





- Settings is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image. The first three parameters specify the filter used to detect the edges.
  - Contrast specifies the threshold for the contrast of the edge.
    Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - **Filter width** specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
  - Subsampling Ratio specifies the number of pixels that separate two consecutive search lines of the rake.
  - Show Search Area determines whether the search area is overlaid on the image.
  - Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
  - Show Edges Found determines whether the locations of the

- edges found are overlaid on the result image.
- Show Result determines whether the hit lines to the object are overlaid on the result image.
- **Image** is a reference to the source image.
- Rectangle specifies the coordinates of a rectangular search area.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- Coordinate System specifies the coordinate system to which the Rectangle is linked. If the Coordinate System input is used (connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Edge Coordinates** is an array of point clusters consisting of the spatial coordinates of the detected edges.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- Distance returns the distance measured between the hit lines.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

A horizontal hit-line to the object is calculated through the uppermost edge detected in the lower half of the search area. A second horizontal hit-line to the object is calculated through the lowermost edge in the upper half of the search area. The distance between those two lines is returned. IMAQ Clamp Vertical Min can overlay on the image returned: the position of the search area, the search lines, the edges found, and the result.

# **Locate Edges**

Owning Palette: <u>Machine Vision</u>Installed With: NI Vision Development Module

Use the Locate Edges VIs to find vertical, horizontal, and circular edges of an image.

Palette Object	Description
IMAQ Find Edge	Finds straight edges (lines) within a region in an image.
IMAQ Find Circular Edge	Locates a circular edge in a search area. This VI locates the intersection points between a set of search lines defined by a spoke and the edge of an object. The intersection points are determined based on their contrast and slope.
-	Locates a straight edge in a circular search area. This VI locates the intersection points between a set of concentric search lines and the edge of an object. The intersection points are determined based on their contrast and slope.

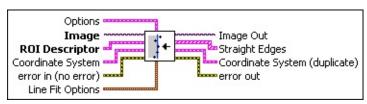
# **IMAQ Find Edge VI**

Owning Palette: Locate Edges Installed With: NI Vision Development

Module

Finds straight edges (lines) within a region in an image.





Options is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image.

Search Direction specifies the direction in which the Region of Interest (ROI) is examined when straight edge detection is performed.

Left to Right (0)	(Default) Searches for edges from left to right
Right to Left (1)	Searches for edges from right to left
Top to Bottom (2)	Searches for edges from top to bottom
Bottom to Top (3)	Searches for edges from bottom to top

**Edge Options** specifies the parameters that are used to compute the edge gradient information and detect the edges.

**Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
Falling Edges (2)	Searches for falling edges

**Kernel Size** specifies the size of the edge detection kernel. The default is 3.

Width specifies the number of pixels averaged perpendicular to the search direction to compute the

- edge profile strength at each point along the search ROI. The default is 3.
- Minimum Edge Strength specifies the minimum edge strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the following options:

Zero Order (0)	Rounds to the nearest integral edge location	
Bilinear (1)	Uses bilinear interpolation to compute the edge location	
Bilinear Fixed (4)	(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge location	

**Data Processing Method** is the method used to process the data extracted for edge detection.

• ,	(Default) Averages the data extracted for edge detection
` '	Takes the median of the data extracted for edge detection

- Show Search Area determines whether the search area is overlaid on the image.
- Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
- Show Edges Found determines whether the locations of the edges found are overlaid on the result image.
- Show Result determines whether the coordinate system found is overlaid on the result image.
- Search Area Color specifies the color to use to overlay the search area.
- Search Lines Color specifies the color to use to overlay the search lines.

- Edge Locations Color specifies the color to use to overlay the edge locations.
- Result Color specifies the color to use to overlay the result.
- Overlay Group Name specifies the overlay group name for the step overlays.
- **Image** is a reference to the image in which you want to search.
- **ROI Descriptor** is a descriptor that defines the rectangle, rotated rectangle, or other four-side rectangular polygon within which the edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- **Coordinate System** specifies the coordinate system to use.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Line Fit Options** specifies the options that are used to detect the straight edges.
  - Number of Lines specifies the number of straight edges to find. The default is 1.
  - **Type** specifies the method used to find the straight edge. Choose from the following options:

First Edge Rake (0)	(Default) Fits a straight edge on the first points detected using a Rake
Best Edge Rake (1)	Fits a straight edge on the best points detected using a Rake
Hough Edge Rake (2)	Finds the strongest straight edge using all points detected on a Rake
First Edge Projection (3)	Uses the location of the first projected edge as the straight edge
Best Edge Projection (4)	Finds the strongest projected edge location to determine the straight edge

- Min. Score specifies the minimum score of a detected straight edge. The default is 10.
- Max. Score specifies the maximum score of a detected straight edge. The default is 1000.
- Orientation specifies the angle at which the straight edge is expected to be found. The default is 0.
- Angle Range specifies the positive and negative range around the **Orientation** within which the straight edge is expected to be found. The default is 45.
- Angle Tolerance specifies the expected angular accuracy of the straight edge. The default is 1.

- Step Size specifies the gap in pixels between the search lines used with the Rake-based methods. The default is 7.
- Min. SNR (db) specifies the minimum signal to noise ratio (SNR) of the edge points used to fit the straight edge. The default is 0.
- Min. Points (%) specifies the minimum number for points as a percentage of the number of search lines that need to be included in the detected straight edge. The default is 25%.
- Hough Iterations specify the number of iterations used in the Hough-based method. The default is 5.
- Image Out is a reference to Image.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

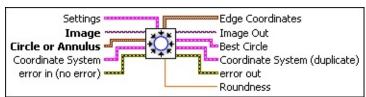
### **IMAQ Find Circular Edge VI**

Owning Palette: Locate Edges Installed With: NI Vision Development Module

Locates a circular edge in a search area. This VI locates the intersection points between a set of search lines defined by a spoke and the edge of an object. The intersection points are determined based on their contrast and slope.

#### **Details** Examples





- Settings is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image. The first three parameters specify the filter used to detect the edges.
  - Contrast specifies the threshold for the contrast of the edge.
    Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - Filter width specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
  - Subsampling Ratio specifies the number of degrees that separates two consecutive search lines of the spoke.
  - Scan Direction specifies the direction in which the search lines are scanned for edges. The default value is **Outside to Inside**, from outside of the annulus to inside the annulus.
  - Show Search Area determines whether the search area is overlaid on the image.

- Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
- Show Edges Found determines whether the locations of the edges found are overlaid on the result image.
- Show Result determines whether the edges used in the bestfit line calculation are overlaid on the result image.
- **Image** is a reference to the source image.
- Circle or Annulus specifies a circular search area.
  - **Center X** specifies the x-coordinate of the center of the circle or annulus.
  - **Center Y** specifies the y-coordinate of the center of the circle or annulus.
  - Inner Radius is the radius (in pixels) of the inner circle defining one edge of the circular strip.
  - Outer Radius is the radius (in pixels) of the outer circle defining the second edge of the circular strip.
  - Start Angle is the first angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
  - **End Angle** is the second angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
- Coordinate System specifies the coordinate system to which the Annulus is linked. If the Coordinate System input is used (connected), the annular search area is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Edge Coordinates** is an array of point clusters consisting of the spatial coordinates of the detected edges.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- **Best Circle** returns in a cluster the center point of the best-fit circle as well as its radius.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

**Roundness** returns the roundness of the circular edge detected. This <u>calculation</u> is based on the location of the edges detected.

### **Details**

A best-fit circle is calculated through the points found. The VI returns the coordinates of the circle found as well as its roundness. IMAQ Find Circular Edge can overlay results on the image returned: the position of the search area, the search lines, the edges found, and the result circle.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Battery Clamp Inspection Example.vi

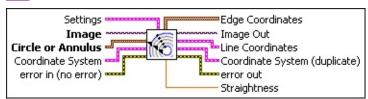
## **IMAQ Find Concentric Edge VI**

Owning Palette: Locate Edges Installed With: NI Vision Development Module

Locates a straight edge in a circular search area. This VI locates the intersection points between a set of concentric search lines and the edge of an object. The intersection points are determined based on their contrast and slope.

#### **Details**





- Settings is a cluster defining the parameters of the edge detection algorithm and the information that is overlaid on the result image. The first three parameters specify the filter used to detect the edges.
  - Contrast specifies the threshold for the contrast of the edge.
    Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - **Filter width** specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
  - Subsampling Ratio specifies the number of pixels that separate two consecutive search lines of the rake.
  - Scan Direction specifies the direction in which the search lines are scanned for edges. The default value is Counter-Clockwise.
  - Show Search Area determines whether the search area is overlaid on the image.

- Show Search Lines determines whether the search lines used to locate the edges are overlaid on the image.
- Show Edges Found determines whether the locations of the edges found are overlaid on the result image.
- Show Result determines whether the edges used in the bestfit line calculation are overlaid on the result image.
- Image is a reference to the source image.
- Circle or Annulus specifies a circular search area.
  - **Center X** specifies the x-coordinate of the center of the circle or annulus.
  - **Center Y** specifies the y-coordinate of the center of the circle or annulus.
  - Inner Radius is the radius (in pixels) of the inner circle defining one edge of the circular strip.
  - Outer Radius is the radius (in pixels) of the outer circle defining the second edge of the circular strip.
  - Start Angle is the first angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
  - **End Angle** is the second angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
- Coordinate System specifies the coordinate system to which the Annulus is linked. If the Coordinate System input is used (connected), the annular search area is shifted and rotated by the difference between the reference position of the coordinate system and its new location.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Edge Coordinates** is an array of point clusters consisting of the spatial coordinates of the detected edges.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.
- **Line Coordinates** returns a cluster of two point coordinates specifying the coordinates of the edge found.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

**Straightness** is the straightness of the edge found. This result is the least-square error of the fitted line to the entire set of points.

### **Details**

A best-fit line is calculated through the points found. The VI returns the coordinates of the edge found as well as its straightness. IMAQ Find Concentric Edge can overlay on the image returned the position of the search area, the search lines, the edges found and the result line.

## **Find Patterns**

Owning Palette: Machine VisionInstalled With: NI Vision Development

Module

Use the Find Patterns VI to find a pattern in an image.

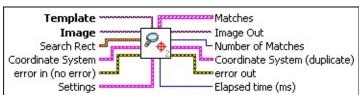
<b>Palette Object</b>	Description
•	Searches for a template image in a rectangular search
Pattern 2	area of the image.

## **IMAQ Find Pattern 2 VI**

Owning Palette: Find Patterns Installed With: NI Vision Development Module

Searches for a template image in a rectangular search area of the image.

U8



- **Template** is a reference to the image to locate during the match process. The template image is the image obtained from the output of the <a href="IMAQ Learn Pattern 2">IMAQ Learn Pattern 2</a> VI. If the template has not been learned, this VI first goes through the learning process.
- **Image** is a reference to the source image.
- Search Rect specifies a rectangular search area in an image. The Left, Top, Right, and Bottom coordinates describe a non-rotated rectangle. The Rotation specifies the rotation angle of the rectangle around its center.
  - **Left** is the x-coordinate of the upper left corner of the rectangle.
  - **Top** is the y-coordinate of the upper left corner of the rectangle.
  - **Right** is the x-coordinate of the bottom right corner of the rectangle.
  - **Bottom** is the y-coordinate of the bottom right corner of the rectangle.
  - Rotation specifies the rotation angle in degrees of the rectangle with its center as point of rotation. If the rotation angle does not equal zero, the Left, Top, Right, and Bottom coordinates are not the actual coordinates of the upper left and bottom right corner of the rectangle, but their position if the rotation angle equals zero.
- Coordinate System specifies the coordinate system to which the Rectangle is linked. If the Coordinate System input is used

(connected), the location of the intensity measurement is shifted and rotated by the difference between the reference position of the coordinate system and its new location.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Settings is a cluster defining the parameters of the template location algorithm and the information that is overlaid on the result image.
  - Match Mode specifies the technique to use when looking for the template pattern in the image. Choose from the following values:

(Default) Searches for the template pattern in the image, assuming it is not rotated more than 4 degrees.
Searches for the template in the image with no restriction on the rotation of the template.

Number of Matches Requested is the number of valid

- matches expected.
- Minimum Match Score is the smallest score a match must have to be considered valid.
- SubPixel Accuracy determines whether the match results should be returned with subpixel accuracy.
- Rotation Angle Ranges is an array of angle ranges where each range specifies how much you expect the template pattern to rotate in the image. Each range is specified by a lower angle and an upper angle, both of which are expressed in degrees.
- Show Search Area specifies whether the centers and bounding rectangles of the patterns found are overlaid on the result image.
- Show Result Show Result specifies whether the centers and bounding boxes of the patterns found are overlaid on the result image.
- Matches is an array of match clusters. Matches returns the following elements:
  - **Position** is a point-coordinate cluster corresponding to the center of the template.
  - Angle is the rotation angle of the template in the source image at the current match location.
  - **Scale** is the change in percentage size of the matched template from the original template. Currently this is always set to 1.
  - Score ranks the match results on a scale of 0 to 1,000, where 0 equals no match and 1,000 equals a perfect match.
  - Bounding Box is an array of five points that define the boundary of the template pattern in the image. You can connect this array directly to the <a href="MAQ Overlay Multiple Lines">IMAQ Overlay Multiple Lines</a> VI to overlay the location of a match on your image.
- Image Out is a reference to the destination image. The search area and/or the result of the measurement may be overlaid on the image according to the settings.

- Number of Matches is the number of template matches found in the inspection image based on the input settings.
- **Coordinate System (duplicate)** is a reference to the coordinate system.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Searching and Matching**

Owning Palette: Machine VisionInstalled With: NI Vision Development Module

Use the Searching and Matching VIs to locate patterns and sub-images in an image. These VIs allow you to perform color and grayscale pattern matching as well as shape matching.

Palette Object	Description
IMAQ Setup Learn Pattern 2	Sets parameters used during the learning phase.
IMAQ Learn Pattern 2	Creates a description of the template image for which you want to search during the matching phase. This description data is appended to the input template image. During the matching phase, the template descriptor is extracted from the template image and used to search for the template in the inspection image.
IMAQ Setup Match Pattern 2	Sets parameters that are used during the matching phase.
IMAQ Match Pattern 2	Searches for a pattern, or template image, in an inspection image.
IMAQ Setup Learn Geometric Pattern	Sets parameters that are used during the learning process. Run this VI before using the IMAQ Learn Geometric Pattern.
IMAQ Learn Geometric Pattern	Creates a description of the template image for which you want to search during the matching phase.
IMAQ	Sets parameters that are used during the matching process.

Setup Match Geometric Pattern	Run this VI before using the <u>IMAQ Match Geometric Pattern</u> VI.
IMAQ Match Geometric Pattern	Searches for a pattern, or template image, in the inspection image.
IMAQ Learn Multiple Geometric Patterns	Combines the descriptions of the patterns you want to search for during the matching phase into a multiple geometric template. Use the multiple geometric template to search for these template images in the target image.
IMAQ Match Multiple Geometric Patterns	Matches the templates specified in the Multiple Template control to the target image.
IMAQ Read Multiple Geometric Template	Creates a new multiple geometric template from the file specified by <b>File Path</b> .
IMAQ Write Multiple Geometric Template	Writes a multiple geometric template to the file specified by <b>File Path</b> . This VI saves the exact state of the multiple geometric template.
IMAQ Dispose Multiple Geometric Template	Destroys a multiple geometric template and frees the space it occupied in memory.
IMAQ Setup Learn Color	Sets parameters that are used during the color learning stage. Run this VI before using the IMAQ Learn Color Pattern VI.

<u>Pattern</u>	
IMAQ Learn Color Pattern	Creates a description of the color template image that you are going to look for during the color pattern matching stage.
IMAQ Setup Match Color Pattern	Sets parameters that are used during the color matching process. Run this VI before using the <a href="IMAQ Match Color Pattern">IMAQ Match Color Pattern</a> VI.
IMAQ Match Color Pattern	Searches for a color pattern, or color template image, in the input Image.
IMAQ Read Image And Vision Info	Reads an image file, including any extra vision information saved with the image. This includes overlay information, pattern matching template information, and calibration information, as written by the <a href="IMAQ Write Image And Vision">IMAQ Write Image And Vision</a> Info VI.
IMAQ Write Image And Vision Info	Writes an image, along with extra vision information associated with the image, to a PNG file. This extra vision information includes overlay information, pattern matching template information, and calibration information.
IMAQ Detect Shapes	Searches for rectangles, lines, ellipses, or circles in an image.
IMAQ Shape Match Tool	Finds objects in an image whose shape matches the shape of the object specified by a template.

Subpalette	Description	
<u>Pattern</u>	Use the Pattern Matching Advanced VIs to optimize and	
<b>Matching</b>	fine-tune advanced parameters used during the learning	

### <u>Advanced</u>

and matching phases of pattern matching. These functions are intended for advanced users. Improper use may cause unexpected results.

### **IMAQ Setup Learn Pattern 2 VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Sets parameters used during the learning phase. Execute this VI before the <u>IMAQ Learn Pattern 2</u> VI.

#### **Examples**



Learn Mode specifies the invariance mode to use when learning the template pattern. Choose from the following values:

<b>AII</b> (0)	(Default) Extracts template information for shift- and rotation-invariant matching.
Shift Information (1)	Extracts information for shift-invariant matching.
Rotation Information (2)	Extracts information for rotation-invariant matching.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Learn Pattern Setup Data is a string that contains information about the setup parameters selected for the learning phase.

  Connect this output to the <a href="IMAQ Learn Pattern 2">IMAQ Learn Pattern 2</a> VI or to the <a href="IMAQ Advanced Setup Learn Pattern 2">IMAQ Advanced Setup Learn Pattern 2</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Pattern Matching\Pattern Matching Example.llb

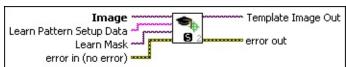
### **IMAQ Learn Pattern 2 VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Creates a description of the template image for which you want to search during the matching phase. This description data is appended to the input template image. During the matching phase, the template descriptor is extracted from the template image and used to search for the template in the inspection image.

#### **Examples**





- Image is a reference to the template image for which you want to search in the inspection image.
- Learn Pattern Setup Data is a string that contains information obtained from the <a href="IMAQ Setup Learn Pattern 2">IMAQ Setup Learn Pattern 2</a> VI or from the <a href="IMAQ Advanced Setup Learn Pattern 2">IMAQ Advanced Setup Learn Pattern 2</a> VI. If this input is not connected, the VI uses default parameters during the learning phase.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a

warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Template Image Out** is a reference to the template. This template image contains the data that defines the template pattern for the matching phase.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

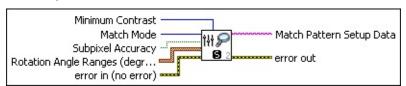
• LabVIEW\examples\Vision\2. Functions\Pattern Matching\Pattern Matching Example.llb

### **IMAQ Setup Match Pattern 2 VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Sets parameters that are used during the matching phase. Execute this VI before the <u>IMAQ Match Pattern 2</u> VI or the <u>IMAQ Refine Matches</u> VI.

#### **Examples**



- Minimum Contrast specifies the minimum expected difference between the maximum pixel value and the minimum pixel value in the inspection image. The default is 0, which allows the algorithm to compute the Minimum Contrast.
- Match Mode specifies the invariance mode to use when looking for the template pattern in the inspection image. The default is **Shift Invariant**. The following options are valid:

Shift Invariant (0)	Searches for the template image, allowing up to $\pm 4^{\circ}$ of rotation.
	Searches for the template in the image regardless of the rotation of the template.

- Subpixel Accuracy determines whether to return the match results with subpixel accuracy. The default is FALSE. If Match Feature Mode is set to color, the VI ignores Subpixel Accuracy.
- Rotation Angle Ranges (degrees) is an array of angle ranges, where each range specifies how much you expect the template to rotate in the inspection image. Specify a lower and upper angle, in degrees, for each range.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Match Pattern Setup Data is a string that contains information about the setup parameters for the matching phase. Connect this output to the <a href="IMAQ Match Pattern 2">IMAQ Match Pattern 2</a> VI or to the <a href="IMAQ Advanced Setup Match Pattern 2">IMAQ Advanced Setup Match Pattern 2</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Pattern Matching\Pattern Matching Example.llb

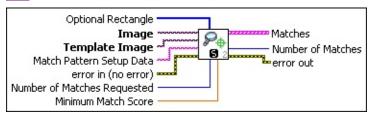
### **IMAQ Match Pattern 2 VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Searches for a pattern, or template image, in an inspection image. Execute the <u>IMAQ Learn Pattern 2</u> VI before this VI to configure the template image for the match phase.

#### **Details** Examples





- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Image is a reference to the inspection image in which you want to search for the template image.
- **Template Image** is a reference to the template for which you want to search during the matching phase. The template image is an output of the <a href="IMAQ Learn Pattern 2">IMAQ Learn Pattern 2</a> VI.
- Match Pattern Setup Data is a string that contains information from the <a href="MAQ Setup Match Pattern 2">IMAQ Setup Match Pattern 2</a> VI or the <a href="IMAQ Advanced Setup Match Pattern 2">IMAQ Advanced Setup Match Pattern 2</a> VI. If this input is not connected, the VI uses default parameters for all unconfigured options.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number of Matches Requested is the number of valid matches expected.
- Minimum Match Score is the smallest score a match can have to be considered valid. The data range for a match is between 0 and 1000.
- Matches is an array of match clusters. Matches returns the following elements:
  - **Position** is a point-coordinate cluster corresponding to the center of the template.
  - Angle is the rotation angle of the template in the source image at the current match location.
  - Scale is the change in percentage size of the matched template from the original template. Currently this is always set to 1.
  - Score ranks the match results on a scale of 0 to 1,000, where 0 equals no match and 1,000 equals a perfect match.
  - Bounding Box is an array of five points that define the boundary of the template pattern in the image. You can connect this array directly to the <a href="MAQ Overlay Multiple Lines">IMAQ Overlay Multiple Lines</a> VI to overlay the location of a match on your image.
- Number of Matches is the number of template matches found in the inspection image based on the input settings.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

### **Details**

If you did not save your template image correctly, this VI returns Error – 1074395247: Invalid Template Descriptor. Refer to the <u>IMAQ Write Image</u> <u>And Vision Info</u> VI for information about saving template data and template images.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Pattern Matching\Pattern Matching Example.llb

## **Pattern Matching Advanced**

Owning Palette: Searching and Matching Installed With: NI Vision

**Development Module** 

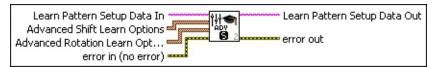
Use the Pattern Matching Advanced VIs to optimize and fine-tune advanced parameters used during the learning and matching phases of pattern matching. These functions are intended for advanced users. Improper use may cause unexpected results.

Palette Object	Description
IMAQ Advanced Setup Learn Pattern 2	Optimizes and fine-tunes advanced parameters used during the learning phase.
IMAQ Advanced Setup Match Pattern 2	Optimizes and fine-tunes advanced parameters used during the matching phase.
IMAQ Refine Matches	Refines matches returned from the <u>IMAQ Match</u> <u>Pattern 2</u> VI using subpixel information learned using the <u>IMAQ Learn Pattern 2</u> VI.
IMAQ Advanced Setup Learn Geometric Pattern	Optimizes and fine-tunes advanced parameters used during the learning phase.
IMAQ Advanced Setup Match Geometric Pattern	Optimizes and fine-tunes advanced parameters used during the matching phase.
IMAQ Advanced Match Geometric Pattern	Searches for a pattern, or template image, in the inspection image.
IMAQ Get Geometric Template Features	Returns the features that describe a geometric template.
IMAQ Get Geometric Features From Curves	Returns the geometric features described by a set of curves.

### **IMAQ Advanced Setup Learn Pattern 2 VI**

Owning Palette: Pattern Matching Advanced Installed With: NI Vision Development Module

Optimizes and fine-tunes advanced parameters used during the learning phase. Execute this VI before the <a href="IMAQ Learn Pattern 2">IMAQ Learn Pattern 2</a> VI. If you use the <a href="IMAQ Setup Learn Pattern 2">IMAQ Setup Learn Pattern 2</a> VI along with this VI, execute the IMAQ Advanced Setup Learn Pattern 2 VI after the IMAQ Setup Learn Pattern 2 VI.



- Learn Pattern Setup Data In is a string that contains information from the <a href="IMAQ Setup Learn Pattern 2">IMAQ Setup Learn Pattern 2</a> VI. If this input is not connected, the VI uses default parameters for all unconfigured options.
- Advanced Shift Learn Options is an array of options to use during the learning phase. The array is composed of the following elements:
  - Setup Data Item determines the option you want to use during the learning phase of shift-invariant matching. The following values are possible:

	·
Initial Step Size (0)	Specifies the largest number of image pixels to shift the sample across the inspection image during the initial phase of shift-invariant matching. The default is 7. The algorithm may reduce the value of the <b>Initial Step Size</b> based on the <b>Initial Sample Size</b> and the template image. If the step size is not an odd integer, the VI coerces it to the next smaller odd integer.
Initial Sample Size (1)	Specifies the number of template pixels that you want to include in a sample for the initial phase of shift-invariant matching. The default is 0, which allows the algorithm to compute the <b>Initial Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple

	of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Initial Sample Size Factor (2)	Specifies the size of the sample for the initial phase of shift-invariant matching as a percent of the template size, in pixels. The default is 0, which causes the VI to use the Initial Sample Size. If you provide values for both the Initial Sample Size Factor and the Initial Sample Size, the VI uses the Initial Sample Size. For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Final Sample Size (3)	Specifies the number of template pixels you want to add to the <b>Initial Sample Size</b> for the final phase of shift-invariant matching. These additional points include edge points. The default is 0, which allows the algorithm to compute the <b>Final Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Final Sample Size Factor (4)	Specifies the size of the sample for the final phase of shift-invariant matching as a percent of the edge points in the template, in pixels. The default is 0, which causes the VI to use the <b>Final Sample Size</b> . If you provide values for both the <b>Final Sample Size Factor</b> and the <b>Final Sample Size</b> , the VI uses the <b>Final Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Subpixel Sample Size (5)	Specifies the number of template pixels that you want to include in a sample for the subpixel phase of shift-invariant matching. The default is 0, which allows the algorithm to compute the <b>Subpixel Sample Size</b> . For optimal speed, the VI coerces

	sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Subpixel Sample Size Factor (6)	Subpixel Sample Size Factor specifies the size of the sample for the subpixel phase of shift-invariant matching as a percent of the template size, in pixels. The default is 0, which causes the VI to use the <b>Subpixel Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.

- Item Value is the value of the option you want to use during the learning phase.
- Advanced Rotation Learn Options is an array of options to use during the learning phase. If this array is connected, the learning phase extracts information for rotation-invariant matching. Each cluster in the array is composed of the following elements:
  - Setup Data Item determines the option you want to use during the learning phase. The following values are possible:

Search Strategy Support (0)	specifies the aggressiveness of the rotation search strategy available during the matching phase. Three types of search strategies are available. The default is 2.
	Conservative—Extracts template information for Conservative matching only.
	<ol> <li>Balanced—Extracts template information for Balanced and Conservative matching.</li> </ol>
	<ol> <li>Aggressive—Extracts template information for Aggressive, Balanced, and Conservative matching.</li> </ol>
Initial Step Size (1)	Specifies the largest number of image pixels to shift the sample across the inspection image during the initial phase of matching. The

	default values are 5 for the Balanced search strategy and 3 for the Conservative search strategy. If the step size is not an odd integer, the VI coerces it to the next smaller odd integer.  Note The Aggressive search strategy does not support the Initial Step Size option.
Initial Sample Size (2)	Specifies the number of template pixels that you want to include in a sample for the initial phase of rotation-invariant matching. The default is 0, which allows the algorithm to compute the <b>Initial Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Initial Sample Size Factor (3)	Specifies the size of the sample for the initial phase of rotation-invariant matching as a percent of the template size, in pixels. The default is 0, which causes the VI to use the Initial Sample Size. If you provide values for both the Initial Sample Size Factor and the Initial Sample Size, the VI uses the Initial Sample Size. For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Initial Angular Accuracy (4)	Initial Angular Accuracy sets the angle accuracy, in degrees, to use during the initial phase of rotation-invariant matching. The default is 6 degrees. The VI coerces the angle to the largest integer smaller than it that evenly divides 360. This option is not used in conjunction with the Aggressive search strategy.
Final	Specifies the number of template pixels you

Sample Size (5)	want to add to the Initial Sample Size for the final phase of rotation-invariant matching. These additional points include edge points. The default is 0, which allows the algorithm to compute the Final Sample Size. For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Final Sample Size Factor (6)	Specifies the size of the sample for the final phase of rotation-invariant matching as a percent of the edge points in the template, in pixels. The default is 0, which causes the VI to use the <b>Final Sample Size</b> . If you provide values for both the <b>Final Sample Size Factor</b> and the <b>Final Sample Size</b> , the VI uses the <b>Final Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.
Final Angular Accuracy (7)	Sets the angle accuracy, in degrees, to use during the final phase of the rotation-invariant matching. The default is 1 degree. Use subpixel accuracy to achieve angle accuracy less than the default. This value must be no greater than the value for <b>Initial Angular Accuracy</b> . The VI coerces the angle to the largest integer smaller than it that evenly divides 360. This option is not used in conjunction with the Aggressive search strategy.
Subpixel Sample Size (8)	Specifies the number of template pixels that you want to include in a sample for the subpixel phase of rotation-invariant matching. The default is 0, which allows the algorithm to compute the <b>Subpixel Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and

	coerces sizes greater than 240 to an integer multiple of 60.
Sample Size Factor (9)	Subpixel Sample Size Factor specifies the size of the sample for the subpixel phase of rotation-invariant matching as a percent of the template size, in pixels. The default is 0, which causes the VI to use the <b>Subpixel Sample Size</b> . For optimal speed, the VI coerces sizes that are less than 240 to an integer multiple of 12 and coerces sizes greater than 240 to an integer multiple of 60.

- **Item Value** is the value of the option you want to use during the learning phase.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Learn Pattern Setup Data Out is a string that contains information about the setup parameters selected for the learning phase.

Connect this output to the **IMAQ Learn Pattern 2** VI.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Advanced Setup Match Pattern 2 VI**

Owning Palette: Pattern Matching Advanced Installed With: NI Vision Development Module

Optimizes and fine-tunes advanced parameters used during the matching phase. Execute this VI before the <u>IMAQ Match Pattern 2</u> VI or the <u>IMAQ Refine Matches</u> VI. If you use the <u>IMAQ Setup Match Pattern 2</u> VI along with this VI, execute this VI after IMAQ Setup Match Pattern 2.



- Match Pattern Setup Data In is a string that contains information from the IMAQ Setup Match Pattern 2 VI. If this input is not connected, the VI uses default parameters for all unconfigured options.
- Advanced Shift Match Options is an array of options to use during the matching phase. If this array is connected, the matching phase searches for the template image allowing up to ±4° of rotation. Each cluster in the array is composed of the following elements:
  - Setup Data Item determines the option you want to use during the matching phase. The following values are possible:

Minimum Contrast (0)	Specifies the minimum expected difference between the maximum pixel value and the minimum pixel value in the inspection image.
Enable Subpixel Accuracy (1)	Determines whether to return the match results with subpixel accuracy. The default is 0, which disables subpixel accuracy.
Subpixel Iterations (2)	Defines the maximum number of incremental improvements used to refine shift-invariant matching using subpixel information. The default is 20.
Subpixel Tolerance (3)	Defines the maximum amount of change, in pixels, between consecutive incremental improvements in the match position that you want to trigger the end of the refinement

	process. The default is 0, which specifies using the <b>Subpixel Iterations</b> value.  If you provide values for both <b>Subpixel Iterations</b> and <b>Subpixel Tolerance</b> , the VI refines the match for at most <b>Subpixel Iterations</b> but may stop early if <b>Subpixel Tolerance</b> is satisfied. If you set <b>Subpixel Tolerance</b> , matches may be invalidated during the subpixel matching process. However, using <b>Subpixel Iterations</b> alone cannot invalidate a match. This behavior is particular important in the <u>IMAQ Refine Matches</u> VI.
Initial Match List Length (4)	Specifies the maximum size of the match list. The match list contains the regions in the inspection image that have the highest probability of containing a match. The default is 200.
Match List Reduction Factor (5)	Specifies the reduction of the match list as matches are refined. The default is 5.
Initial Step Size (6)	Specifies the number of pixels to shift the sample across the inspection image during the initial phase of shift-invariant matching. The default is 0, which uses the <b>Initial Step Size</b> stored in the template. If the step size is not an odd integer, the VI uses the default value.

Item Value is the value for the option you want to use during the matching phase.

Advanced Rotation Match Options is an array of options to use during the matching phase of rotation-invariant matching. If this array is connected, the matching phase searches for the template image regardless of the rotation of the template. Each cluster in the array is composed of the following elements:

Setup Data Item determines the option you want to use

during the matching phase. The following values are possible:

Minimum Contrast (0)	Specifies the minimum expected difference between the maximum pixel value and the minimum pixel value in the inspection image.
Enable Subpixel Accuracy (1)	Determines whether to return the match results with subpixel accuracy. The default is 0, which disables subpixel accuracy.
Search Strategy (2)	Specifies the aggressiveness of the rotation search strategy. The default is 2.
	<ol> <li>Conservative—Extracts template information for Conservative matching only.</li> <li>Balanced—Extracts template information for Balanced and Conservative matching.</li> <li>Aggressive—Extracts template information for Aggressive, Balanced, and Conservative matching.</li> </ol>
Subpixel Iterations (3)	Defines the maximum number of incremental improvements used to refine rotation-invariant matching using subpixel information. The default is 20.
Subpixel Tolerance (4)	improvements in the match position that you want to trigger the end of the refinement process. The default is 0, which specifies using the <b>Subpixel Iterations</b> value.
	If you provide values for both Subpixel Iterations and Subpixel Tolerance, the VI refines the match for at most Subpixel Iterations but may stop early if Subpixel Tolerance is satisfied. If you set Subpixel Tolerance, matches may be invalidated during the subpixel matching process. However, using Subpixel Iterations alone

	cannot invalidate a match. This behavior is particular important in the <u>IMAQ Refine</u> <u>Matches</u> VI.
Initial Match List Length (5)	Specifies the maximum size of the match list. The match list contains the regions in the inspection image that have the highest probability of containing a match. The default is 200.
Match List Reduction Factor (6)	Specifies the reduction of the match list as matches are refined. The default is 5.
Initial Step Size (7)	Specifies the number of pixels to shift the sample across the inspection image during the initial phase of rotation-invariant matching. The default is 0, which uses the <b>Initial Step Size</b> stored in the template. If the step size is not an odd integer, the VI uses the default value.
Intermediate Angular Accuracy (8)	Specifies the accuracy to use during the intermediate phase of rotation-invariant matching. The default is the value of <b>Final Angular Accuracy</b> stored in the template. The VI coerces this value to an integer that evenly divides 360 and lies in the range defined by <b>Initial Angular Accuracy</b> and <b>Final Angular Accuracy</b> .

**Item Value** is the value for the option you want to use during the matching phase.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

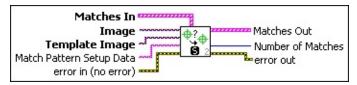
to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Match Pattern Setup Data Out is a string that contains information about the setup parameters for the matching phase. Connect this output to the <a href="IMAQ Match Pattern 2">IMAQ Match Pattern 2</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Refine Matches VI**

Owning Palette: Pattern Matching AdvancedInstalled With: NI Vision Development Module

Refines matches returned from the <u>IMAQ Match Pattern 2</u> VI using subpixel information learned using the <u>IMAQ Learn Pattern 2</u> VI.



- Matches In is a cluster containing the following elements:
  - **Position** is a point-coordinate cluster corresponding to the center of the template.
  - Angle is the rotation angle of the template in the source image at the current match location.
  - Scale is the change in percentage size of the matched template from the original template. Currently, this is always set to 1.
  - Score ranks the match results on a scale of 0 to 1,000, where 0 equals no match and 1,000 equals a perfect match.
  - Bounding Box is an array of five points that define the boundary of the template pattern in the image. You can connect this array directly to the <a href="MAQ Overlay Multiple Lines">IMAQ Overlay Multiple Lines</a> VI to overlay the location of a match on your image.
- Image is a reference to the inspection image in which you originally located the matches you want to refine.
- **Template Image** is a reference to the template for which you want to search during the refinement phase. The template image is an output of the <u>IMAQ Learn Pattern 2</u> VI.
- Match Pattern Setup Data is a string that contains information from the <a href="IMAQ Setup Match Pattern 2">IMAQ Setup Match Pattern 2</a> VI or the <a href="IMAQ Advanced Setup Match Pattern 2">IMAQ Advanced Setup Match Pattern 2</a> VI. If this input is not connected, the VI uses default parameters for all unconfigured options.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Matches Out is an array of match clusters. Matches Out returns the following elements:
  - **Position** is a point-coordinate cluster corresponding to the center of the template.
  - Angle is the rotation angle of the template in the source image at the current match location.
  - Scale is the change in percentage size of the matched template from the original template. Currently this is always set to 1.
  - Score ranks the match results on a scale of 0 to 1,000, where 0 equals no match and 1,000 equals a perfect match.
  - Bounding Box is an array of five points that define the boundary of the template pattern in the image. You can connect this array directly to the <a href="IMAQ Overlay Multiple Lines">IMAQ Overlay Multiple Lines</a> VI to overlay the location of a match on your image.
- Number of Matches is the number of template matches found in the inspection image after refinement.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ Advanced Setup Learn Geometric Pattern VI

Owning Palette: Pattern Matching Advanced Installed With: NI Vision Development Module

Optimizes and fine-tunes advanced parameters used during the learning phase. Execute this VI before the <u>IMAQ Learn Geometric Pattern</u> VI. If you use the <u>IMAQ Setup Learn Geometric Pattern</u> VI along with this VI, execute this VI after the IMAQ Setup Learn Geometric Pattern VI.



- Learn Geometric Pattern Setup Data In is a string that contains information from the <a href="IMAQ Setup Learn Geometric Pattern">IMAQ Setup Learn Geometric Pattern</a> VI. If this input is not connected, the VI uses default parameters for all unconfigured options.
- Advanced Geometric Learn Options is an array of options to use during the learning phase.
  - Setup Data Item determines the option you want to use during the learning phase. The following values are possible:

Curve Extraction Mode (0)	Specifies how the VI identifies curves in the image. Set this option to <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background. Set this option to <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.
Curve Edge Threshold (1)	Specifies the minimum contrast an edge pixel must have for it to be considered part of a curve. Valid values range from 1 to 255.
Curve Edge Filter Size (2)	Specifies the width of the edge filter that the VI uses to identify curves in the image. Set

	this option to 0 to specify a fine filter width, or set the option to 1 to specify the normal filter width.
Min Curve Length (3)	Specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value.
Curve Row Search Step Size (4)	Specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
Curve Col Search Step Size (5)	Specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
Curve Max Endpoint Gap (6)	Specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
Extract Closed Curves (7)	Specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
Enable Subpixel Curve Extraction (8)	The VI ignores this value.
Min Rectangular Dimension (9)	Specifies the minimum length for each side of a rectangular feature. The algorithm ignores rectangular features with a side shorter than this value. The default value is 10.
Min Feature	Specifies the minimum aspect ratio of a

Aspect Ratio (10)	rectangular feature. The algorithm ignores rectangular features with aspect ratios less than this value. Valid values for this element range from .01 to 1.0. The default value is 0.1.
Min Feature Radius (11)	Specifies the minimum radius, in pixels, for a circular feature. The algorithm ignores circular features with radii less than this value. The default value is 5.
Min Feature Length (12)	Specifies the minimum length, in pixels, for a linear feature. The algorithm ignores linear features with lengths shorter than this value. The default value is 15.
Min Feature Strength (13)	Specifies the minimum strength for a feature. The algorithm ignores features with a strength less than this value. Valid values for this element range from 0 to 1. The default value is 0.5.
Max Number of Features (14)	Specifies the maximum number of features the algorithm uses when learning. The default value is 25.
Pixel Tolerance For Lines (15)	Specifies the maximum number of pixels between an edge pixel and a linear feature for the algorithm to consider that edge pixel as part of the linear feature. The default value is 2.

Item Value is the value of the option you want to use during the learning phase.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

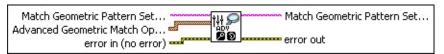
to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Learn Geometric Pattern Setup Data Out is a string that contains information about the setup parameters selected for the learning phase. Connect this output to the <a href="IMAQ Learn Geometric Pattern">IMAQ Learn Geometric Pattern</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# IMAQ Advanced Setup Match Geometric Pattern VI

Owning Palette: Pattern Matching Advanced Installed With: NI Vision Development Module

Optimizes and fine-tunes advanced parameters used during the matching phase. Execute this VI before the <u>IMAQ Match Geometric Pattern</u> VI. If you use the <u>IMAQ Setup Match Geometric Pattern</u> VI along with this VI, execute this VI after IMAQ Setup Match Geometric Pattern.



- Match Geometric Pattern Setup Data In is a string that contains information from the <a href="IMAQ Setup Match Geometric Pattern">IMAQ Setup Match Geometric Pattern</a> VI. If this input is not connected, the VI uses default parameters for all unconfigured options.
- Advanced Geometric Match Options is an array of options to use during the matching phase.
  - Setup Data Item determines the option you want to use during the matching phase. The following options are possible:

Curve Extraction	Specifies how the VI identifies curves in the image. Set this option to <b>Normal</b> if you want
Mode (0)	the VI to make no assumptions about the uniformity of objects in the image or the image background. Set this option to <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This option allows the VI to more accurately calculate the external curves of the objects.
Curve Edge Threshold (1)	Specifies the minimum contrast an edge pixel must have for it to be considered part of a curve. Valid values range from 1 to 255.
Curve Edge	Specifies the width of the edge filter that the

Filter Size (2)	VI uses to identify curves in the image. Set this option to 0 to specify a fine filter width, or set the option to 1 to specify the normal filter width.
Min Curve Length (3)	Specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value. The default value is 25.
Curve Row Search Step Size (4)	Specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
Curve Col Search Step Size (5)	Specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
Curve Max Endpoint Gap (6)	Specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
Extract Closed Curves (7)	Specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
Enable Subpixel Curve Extraction (8)	The VI ignores this option.
Min Features (9)	Specifies the minimum number of features the VI uses when matching. Set this option to 0 if you want the algorithm to use all of the features. The default value is 5.

Max Features (10)	Specifies the maximum number of features the VI uses to finalize a match. The default value is 5.
Enable Correlation Score (11)	Specifies that the VI should calculate the Correlation Score and return it for each match result. The default value is 1, which enables the Correlation Score.
Enable Subpixel Accuracy (12)	Determines whether to return the match results with subpixel accuracy. Set this option to 1 to enable subpixel accuracy. The default value is 0, which disables subpixel accuracy.
Subpixel Iterations (13)	Specifies the maximum number of incremental improvements used to refine matches using subpixel information. The minimum value for this option is 20.
Subpixel Tolerance (14)	Specifies the maximum amount of change, in pixels, between consecutive incremental improvements in the match position before the VI stops refining the match position. The default value is 0, which specifies that the VI should always use a number of refinements equal to <b>Subpixel Iterations</b> . If you provide values for both <b>Subpixel Iterations</b> and <b>Subpixel Tolerance</b> , the VI refines the match for, at most, <b>Subpixel Iterations</b> but may stop early if <b>Subpixel Tolerance</b> is satisfied.
Initial Match List Length (15)	Specifies the maximum size of the match list. The match list contains the regions in the inspection image that have the highest probability of containing a match. The minimum value for this option is 5. The default value is 200.
Enable Target Template Curve Score (16)	Specifies whether the VI should calculate the match curve to template curve score and return it for each match result. Set this option to 1 to enable the target template curve

	score. The default value is 0, which disables the target template curve score.
Min Match Separation Distance (17)	Specifies the minimum separation distance, in pixels, between the origins of two matches that have unique positions. The VI does not return matches that have the same position, scale, and angle. Set this value to -1 if you do not want the VI to use the position of a match to determine whether the match is unique. The default value is 20.
Min Match Separation Angle (18)	Specifies the minimum angular difference, in degrees, between two matches that have unique angles. The VI does not return matches that have the same position, scale, and angle. Set this value to -1 if you do not want the VI to use the angle of a match to determine whether the match is unique. The default value is 10.
Min Match Separation Scale (19)	Specifies the minimum difference in scale, expressed as a percentage, between two matches that have unique scales. The VI does not return matches that have the same position, scale, and angle. Set this value to -1 if you do not want the VI to use the scale of a match to determine whether the match is unique. The default value is 10.
Max Match Overlap (20)	Specifies the maximum amount of overlap, expressed as a percentage, allowed between the bounding rectangles of two matches. Set this value to -1 to ignore this option. The default value is 80.
Enable Coarse Result (21)	Specifies whether you want the algorithm to spend less time accurately estimating the location of a match. Set this value to 1 if you want to quickly determine whether a part is present in the inspection image without an accurate estimate of its position, angle, and

	scale. The default value is 0, which specifies that the VI returns matches with pixel or subpixel accuracy.
Smooth Contours (22)	Specifies whether or not smoothing be done on the contours of the inspection image before feature extraction. The default value is FALSE.
Enable Calibration Support (23)	Specifies whether or not the algorithm treat the inspection image as a calibrated image. Use the IMAQ Set Simple Calibration VI or the IMAQ Set Calibration Info VI to calibrate the inspection image. The default value is TRUE.

- Item Value is the value for the option you want to use during the matching phase.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

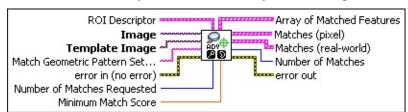
- Match Geometric Pattern Setup Data Out is a string that contains information about the setup parameters for the matching phase.

  Connect this output to the <a href="IMAQ Match Geometric Pattern">IMAQ Match Geometric Pattern</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Advanced Match Geometric Pattern VI**

Owning Palette: Pattern Matching AdvancedInstalled With: NI Vision Development Module

Searches for a pattern, or template image, in the inspection image.



- ROI Descriptor specifies the region of the inspection image in which to perform geometric matching. This region must be a single rectangle or rotated rectangle contour. By default, the VI uses the entire inspection image for matching.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- Image is a reference to the inspection image in which you want to search for the template image
- **Template Image** is a reference to the template for which you want to search during the matching phase. The template image is an output of the <a href="IMAQ Learn Geometric Pattern">IMAQ Learn Geometric Pattern</a> VI.
- Match Geometric Pattern Setup Data is a string that contains information from the IMAQ Setup Match Geometric Pattern VI or the IMAQ Advanced Setup Match Geometric Pattern VI. If this input is not connected, the VI uses default parameters for all unconfigured options.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number of Matches Requested is the number of valid matches expected.
- Minimum Match Score is the smallest score a match can have to be considered valid. The data range for a match is between 0 and 1000.
- Array of Matched Features is an array of a list of geometric features associated with each match returned.
  - Matched Features is a list of geometric features associated with a given match.
    - **Feature Type** is the type of geometric feature.
    - Feature Data is an array that describes the geometric feature. The elements of this array should be interpreted according to the feature type in the following manner:

Circle	<ul> <li>Element 0—X coordinate of the center of circle.</li> </ul>
	<ul> <li>Element 1—Y coordinate of the center of circle.</li> </ul>

	• Element 2—Radius of circle.
Ellipse	<ul> <li>Element 0—X coordinate of the center of ellipse.</li> <li>Element 1—Y coordinate of the center of ellipse.</li> <li>Element 2—Orientation in degrees of the major axis.</li> <li>Element 3—The length of the semi-major axis.</li> <li>Element 4—The length of the semi-minor axis.</li> </ul>
Constant Curve	<ul> <li>Element 0—X coordinate of the center of the constant curve.</li> <li>Element 1—Y coordinate of the center of the constant curve.</li> <li>Element 2—The starting angle in degrees of the constant curve.</li> <li>Element 3—The ending angle in degrees of the constant curve.</li> <li>Element 4—The radius of the circle subtended by the constant curve.</li> </ul>
Rectangle/Leg	<ul> <li>Element 0—X coordinate of the location of the center of the rectangle/leg.</li> <li>Element 1—Y coordinate of the location of the center of the rectangle/leg.</li> <li>Elements 2 and 3—The first corner point (x, y) of the rectangle/leg.</li> <li>Elements 4 and 5—The second corner point (x, y) of the rectangle/leg.</li> </ul>

	<ul> <li>Elements 6 and 7—The third corner point (x, y) of the rectangle/leg.</li> <li>Elements 8 and 9—The fourth corner point (x, y) of the rectangle/leg.</li> <li>Element 10—The orientation of the rectangle/leg in degrees with respect to the horizontal axis.</li> <li>Element 11—The width of the rectangle/leg.</li> <li>Element 12—The height of the rectangle/leg.</li> </ul>
Corner	<ul> <li>Element 0—X coordinate of the corner position.</li> <li>Element 1—Y coordinate of the corner position.</li> <li>Element 2—The orientation of the corner.</li> <li>Element 3—The enclosed angle of the corner.</li> <li>Element 4—Specifies whether or not the corner is a virtual corner. A value of 0 means that the corner is not a virtual corner. A value of 1 means that the corner is a virtual corner.</li> </ul>
Parallel Line Pair	<ul> <li>Elements 0 and 1 —The start point (x, y) of the first line.</li> <li>Elements 2 and 3—The end point (x, y) of the first line.</li> <li>Elements 4 and 5—The start point (x, y) of the second line.</li> <li>Elements 6 and 7—The end</li> </ul>

	<ul> <li>point (x, y) of the second line.</li> <li>Element 8—The orientation in degrees of the parallel line pair with respect to the horizontal axis.</li> <li>Element 9—The distance of the two lines from one another.</li> </ul>
Parallel Line Pairs	<ul> <li>Elements 0–9—The first parallel line pair.</li> <li>Elements 10–19—The second parallel line pair.</li> <li>Element 20—The orientation in degrees of the pair of parallel line pairs with respect to the horizontal axis.</li> <li>Element 21—The distance the two parallel line pairs are from one another as measured by the midline of the first to the midline of the second.</li> </ul>
Line	<ul> <li>Elements 0 and 1—The start point (x,y) of the line.</li> <li>Elements 2 and 3—The end point (x,y) of the line.</li> <li>Element 4—The length of the line.</li> <li>Element 5—The orientation in degrees of the line with respect to the horizontal axis.</li> </ul>
Closed Curve	<ul> <li>Elements 0 and 1—The center point (x,y) of the closed curve.</li> <li>Element 2—The arc length of the closed curve element.</li> <li>Elements 3 and 4—The first of</li> </ul>

- *N* contour points (x,y) of the closed curve.
- Elements 5 and 6—The second of N contour points (x,y) of the closed curve.
- Elements k and k+1—The (k–
  1)/2 contour point (x,y) of the
  closed curve.

.

Elements 2\*N+1 and 2\*N+2—
 The last of N contour points (x,y) of the closed curve.

- Matches (pixel) contains information about a match found in the image.
  - **Position** is a point-coordinate cluster corresponding to the location of the template origin.
  - **Angle** is the rotation angle, in degrees, of the template at the current match location.
  - **Scale** is the ratio of the size of the matched template relative to the original template image, expressed as a percentage.
  - Score ranks the match results on a scale of 0 to 1000, where 0 equals no match and 1000 equals a perfect match.
  - **Bounding Box** is an array of five points that define the boundary of the template.
  - Contrast Reversed specifies whether the match is an inverse of the template image. This parameter is TRUE if the match is an inverse of the template image. For example, the match is a white object on a black background but the template image is a black object on a white background. This parameter is FALSE if the match is not the inverse of the template image.
  - Occlusion % is the percentage of the match that is occluded.

- Template Target Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all template curves have a corresponding curve in the match region of the inspection image.
- Target Template Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all curves in the match region of the inspection image have a corresponding curve in the template image.
  - Note The VI does not compute the Target Template Curve Score by default. Use the <a href="IMAQ Advanced Setup Match Geometric Pattern">IMAQ Advanced</a> Setup Match Geometric Pattern VI to configure the algorithm to compute this value.
- Correlation Score specifies the accuracy of the match obtained by comparing the template image to the match region using a correlation metric that compares the two regions as a function of their pixel values. A score of 1000 indicates a perfect match, and a score of 0 indicates no match.
- Matches (real-world) contains information about a match found in the image.
  - Note Matches (real-world) may differ from Matches (pixel) if Image is a calibrated image. If Image is a calibrated image, Position, Angle, and Bounding Box are returned in real-world units.
  - **Position** is a point-coordinate cluster corresponding to the location of the template origin.
  - **Angle** is the rotation angle, in degrees, of the template at the current match location.
  - **Scale** is the ratio of the size of the matched template relative to the original template image, expressed as a percentage.

- Score ranks the match results on a scale of 0 to 1000, where 0 equals no match and 1000 equals a perfect match.
- **Bounding Box** is an array of five points that define the boundary of the template.
- Contrast Reversed specifies whether the match is an inverse of the template image. This parameter is TRUE if the match is an inverse of the template image. For example, the match is a white object on a black background but the template image is a black object on a white background. This parameter is FALSE if the match is not the inverse of the template image.
- Occlusion % is the percentage of the match that is occluded.
- Template Target Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all template curves have a corresponding curve in the match region of the inspection image.
- Target Template Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all curves in the match region of the inspection image have a corresponding curve in the template image.



Note The VI does not compute the Target Template Curve Score by default. Use the IMAQ Advanced Setup Match Geometric Pattern VI to configure the algorithm to compute this value.

Correlation Score specifies the accuracy of the match obtained by comparing the template image to the match region using a correlation metric that compares the two regions as a function of their pixel values. A score of 1000 indicates a perfect match, and a score of 0 indicates no match.

Number of Matches is the number of template matches found in the inspection image based on the input settings.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Get Geometric Template Features VI**

Owning Palette: Pattern Matching AdvancedInstalled With: NI Vision Development Module

Returns the features that describe a geometric template.

Template Image in	 Learned Features
error in (no error)	 error out

- **Template Image in** is a reference to the template for which you want to search during the matching phase.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Learned Features** is a list of geometric features associated with a match.
  - **Feature Type** is the type of geometric feature.
  - Feature Data is an array that describes the geometric feature. The elements of this array should be interpreted according to the feature type in the following manner:

Circle	<ul> <li>Element 0—X coordinate of the center of circle.</li> <li>Element 1—Y coordinate of the center of circle.</li> <li>Element 2—Radius of circle.</li> </ul>
Ellipse	<ul> <li>Element 0—X coordinate of the center of ellipse.</li> <li>Element 1—Y coordinate of the center of ellipse.</li> <li>Element 2—Orientation in degrees of the major axis.</li> <li>Element 3—The length of the semimajor axis.</li> <li>Element 4—The length of the semiminor axis.</li> </ul>
Constant Curve	<ul> <li>Element 0—X coordinate of the center of the constant curve.</li> <li>Element 1—Y coordinate of the center of the constant curve.</li> <li>Element 2—The starting angle in degrees of the constant curve.</li> <li>Element 3—The ending angle in degrees of the constant curve.</li> <li>Element 4—The radius of the circle subtended by the constant curve.</li> </ul>
Rectangle/Leg	<ul> <li>Element 0—X coordinate of the location of the center of the rectangle/leg.</li> <li>Element 1—Y coordinate of the location of the center of the rectangle/leg.</li> <li>Elements 2 and 3—The first corner point (x, y) of the rectangle/leg.</li> <li>Elements 4 and 5—The second</li> </ul>

	<ul> <li>corner point (x, y) of the rectangle/leg.</li> <li>Elements 6 and 7—The third corner point (x, y) of the rectangle/leg.</li> <li>Elements 8 and 9—The fourth corner point (x, y) of the rectangle/leg.</li> <li>Element 10—The orientation of the rectangle/leg in degrees with respect to the horizontal axis.</li> <li>Element 11—The width of the rectangle/leg.</li> <li>Element 12—The height of the rectangle/leg.</li> </ul>
Corner	<ul> <li>Element 0—X coordinate of the corner position.</li> <li>Element 1—Y coordinate of the corner position.</li> <li>Element 2—The orientation of the corner.</li> <li>Element 3—The enclosed angle of the corner.</li> <li>Element 4—Specifies whether or not the corner is a virtual corner. A value of 0 means that the corner is not a virtual corner. A value of 1 means that the corner is a virtual corner.</li> </ul>
Parallel Line Pair	<ul> <li>Elements 0 and 1 —The start point (x, y) of the first line.</li> <li>Elements 2 and 3—The end point (x, y) of the first line.</li> <li>Elements 4 and 5—The start point (x, y) of the second line.</li> <li>Elements 6 and 7—The end point (x, y) of the second line.</li> <li>Element 8—The orientation in degrees of the parallel line pair with</li> </ul>

	respect to the horizontal axis.  • Element 9—The distance of the two lines from one another.
Parallel Line Pairs	<ul> <li>Elements 0–9—The first parallel line pair.</li> <li>Elements 10–19—The second parallel line pair.</li> <li>Element 20—The orientation in degrees of the pair of parallel line pairs with respect to the horizontal axis.</li> <li>Element 21—The distance the two parallel line pairs are from one another as measured by the midline of the first to the midline of the second.</li> </ul>
Line	<ul> <li>Elements 0 and 1—The start point (x,y) of the line.</li> <li>Elements 2 and 3—The end point (x,y) of the line.</li> <li>Element 4—The length of the line.</li> <li>Element 5—The orientation in degrees of the line with respect to the horizontal axis.</li> </ul>
Closed Curve	<ul> <li>Elements 0 and 1—The center point (x,y) of the closed curve.</li> <li>Element 2—The arc length of the closed curve element.</li> <li>Elements 3 and 4—The first of N contour points (x,y) of the closed curve.</li> <li>Elements 5 and 6—The second of N contour points (x,y) of the closed curve.</li> </ul>

 Elements k and k+1—The (k-1)/2 contour point (x,y) of the closed curve.

.

.

.

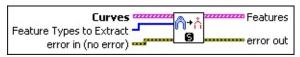
 Elements 2\*N+1 and 2\*N+2—The last of N contour points (x,y) of the closed curve.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Get Geometric Features From Curves VI**

Owning Palette: Pattern Matching Advanced Installed With: NI Vision Development Module

Returns the geometric features described by a set of curves.



- **Curves** is an array describing the curves found in the image.
  - **Points** specifies the location of every point detected on the curve.
  - Closed? is TRUE (Closed) when the distance between the endpoints of the curve are less than or equal to **Max**Endpoint Gap or FALSE (Open) when the distance between the endpoints of the curve are greater than **Max Endpoint**Gap.
  - Curve Length returns the length of the curve.
  - Min Edge Strength returns the lowest edge strength detected on the curve.
  - Max Edge Strength returns the highest edge strength detected on the curve.
  - Average Edge Strength returns the average of all edge strengths detected on the curve.
- Feature Types to Extract are the types of features to extract from the passed curves. Pass an empty array to extract all features. The default value is an empty array.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Features is a list of geometric features associated with a match.
  - **Feature Type** is the type of geometric feature.
  - Feature Data is an array that describes the geometric feature. The elements of this array should be interpreted according to the feature type in the following manner:

Circle	<ul> <li>Element 0—X coordinate of the center of circle.</li> <li>Element 1—Y coordinate of the center of circle.</li> <li>Element 2—Radius of circle.</li> </ul>
Ellipse	<ul> <li>Element 0—X coordinate of the center of ellipse.</li> <li>Element 1—Y coordinate of the center of ellipse.</li> <li>Element 2—Orientation in degrees of the major axis.</li> <li>Element 3—The length of the semimajor axis.</li> <li>Element 4—The length of the semiminor axis.</li> </ul>
Constant Curve	<ul> <li>Element 0—X coordinate of the center of the constant curve.</li> <li>Element 1—Y coordinate of the</li> </ul>

	<ul> <li>center of the constant curve.</li> <li>Element 2—The starting angle in degrees of the constant curve.</li> <li>Element 3—The ending angle in degrees of the constant curve.</li> <li>Element 4—The radius of the circle subtended by the constant curve.</li> </ul>
Rectangle/Leg	<ul> <li>Element 0—X coordinate of the location of the center of the rectangle/leg.</li> <li>Element 1—Y coordinate of the location of the center of the rectangle/leg.</li> <li>Elements 2 and 3—The first corner point (x, y) of the rectangle/leg.</li> <li>Elements 4 and 5—The second corner point (x, y) of the rectangle/leg.</li> <li>Elements 6 and 7—The third corner point (x, y) of the rectangle/leg.</li> <li>Elements 8 and 9—The fourth corner point (x, y) of the rectangle/leg.</li> <li>Element 10—The orientation of the rectangle/leg in degrees with respect to the horizontal axis.</li> <li>Element 11—The width of the rectangle/leg.</li> <li>Element 12—The height of the rectangle/leg.</li> </ul>
Corner	<ul> <li>Element 0—X coordinate of the corner position.</li> <li>Element 1—Y coordinate of the corner position.</li> <li>Element 2—The orientation of the corner.</li> <li>Element 3—The enclosed angle of</li> </ul>

	<ul> <li>the corner.</li> <li>Element 4—Specifies whether or not the corner is a virtual corner. A value of 0 means that the corner is not a virtual corner. A value of 1 means that the corner is a virtual corner.</li> </ul>
Parallel Line Pair	<ul> <li>Elements 0 and 1 —The start point (x, y) of the first line.</li> <li>Elements 2 and 3—The end point (x, y) of the first line.</li> <li>Elements 4 and 5—The start point (x, y) of the second line.</li> <li>Elements 6 and 7—The end point (x, y) of the second line.</li> <li>Element 8—The orientation in degrees of the parallel line pair with respect to the horizontal axis.</li> <li>Element 9—The distance of the two lines from one another.</li> </ul>
Parallel Line Pairs	<ul> <li>Elements 0–9—The first parallel line pair.</li> <li>Elements 10–19—The second parallel line pair.</li> <li>Element 20—The orientation in degrees of the pair of parallel line pairs with respect to the horizontal axis.</li> <li>Element 21—The distance the two parallel line pairs are from one another as measured by the midline of the first to the midline of the second.</li> </ul>
Line	<ul> <li>Elements 0 and 1—The start point (x,y) of the line.</li> <li>Elements 2 and 3—The end point</li> </ul>

	<ul> <li>(x,y) of the line.</li> <li>Element 4—The length of the line.</li> <li>Element 5—The orientation in degrees of the line with respect to the horizontal axis.</li> </ul>
Closed Curve	<ul> <li>Elements 0 and 1—The center point (x,y) of the closed curve.</li> <li>Element 2—The arc length of the closed curve element.</li> <li>Elements 3 and 4—The first of N contour points (x,y) of the closed curve.</li> <li>Elements 5 and 6—The second of N contour points (x,y) of the closed curve.</li> <li>Elements k and k+1—The (k-1)/2 contour point (x,y) of the closed curve.</li> <li>.</li> <li>Elements 2*N+1 and 2*N+2—The last of N contour points (x,y) of the closed curve.</li> </ul>

error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a

warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Setup Learn Geometric Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Sets parameters that are used during the learning process. Run this VI before using the <u>IMAQ Learn Geometric Pattern</u>.



**Curve Parameters** is a cluster containing information about how curves are extracted from the template image during the learn phase.

**Extraction Mode** specifies how the VI identifies curves in the image. Choose from the following options:

Normal (0)	(Default) Select <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background.
Uniform Regions (1)	Select <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.

Edge Threshold specifies the minimum contrast a seed point must have for it to be begin a curve. Valid values range from 1 to 360. The default value is 75.

Edge Filter Size specifies the width of the edge filter that the VI uses to identify curves in the image. The following options are available:

Fine (0)	Sets the <b>Edge Filter Size</b> to fine.
Normal (1)	(Default) Sets the <b>Edge Filter Size</b> to normal.

Min Length specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value. The default value is 25.

Row Search Step Size specifies the distance, in the y

direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.

- Col Search Step Size specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Max Endpoint Gap specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
- Closed specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
- **Subpixel** is ignored by the VI.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

**Learn Geometric Pattern Setup Data** is a string that contains information about to the setup parameters selected for the learning phase. Connect this output to the <a href="IMAQ Learn Geometric Pattern">IMAQ Learn Geometric Pattern</a> VI or the <a href="IMAQ Advanced Setup Learn Geometric Pattern">IMAQ Advanced Setup Learn Geometric Pattern</a> VI.

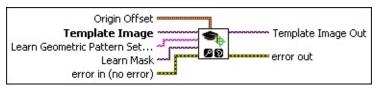
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Learn Geometric Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Creates a description of the template image for which you want to search during the matching phase. This description data is appended to the input template image. During the matching phase, the description data is extracted from the template image and used to search for the template in the inspection image.





- Origin Offset specifies the number of pixels the VI shifts the origin of the template from the center of the template image. The origin offset is used by the <a href="IMAQ Match Geometric Pattern">IMAQ Match Geometric Pattern</a> VI to set the Position element of the resulting Matches cluster for each template match within a target image. The default value is (0, 0), which sets the center of the template image as the origin of the template.
- **Template Image** is a reference to the template image for which you want to search in the inspection image during the matching phase.
- Learn Geometric Pattern Setup Data is a string that contains information obtained from the <a href="MAQ Setup Learn Geometric Pattern">IMAQ Advanced Setup Learn Geometric Pattern</a> VI or the <a href="IMAQ Advanced Setup Learn Geometric Pattern">IMAQ Advanced Setup Learn Geometric Pattern</a> VI. If this input is not connected, the VI uses default parameters during the learning phase.
- Learn Mask is an optional, 8-bit image of the same size as the template that specifies where to search for edges in the template. Use the following pixel values when constructing the mask:
  - 0—Maintains the default behavior. The corresponding pixel
    in the template image is considered an edge only if it meets
    the conditions specified by the Curve Parameters in the
    IMAQ Setup Learn Geometric Pattern VI or the IMAQ
    Advanced Setup Learn Geometric Pattern VI.
  - 1—The corresponding pixel in the template image is never

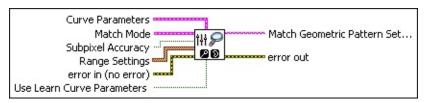
- considered an edge.
- 2—The corresponding pixel in the template image is always considered an edge.
- 4—The corresponding pixel in the template image is not used when computing the **Correlation Score**.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Template Image Out** is a reference to the template. This template image contains the data that defines the template pattern for the matching phase.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Setup Match Geometric Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Sets parameters that are used during the matching process. Run this VI before using the <u>IMAQ Match Geometric Pattern</u> VI.



**Curve Parameters** is a cluster containing information about how curves are extracted from the inspection image during the match phase.



**Note** The VI ignores the **Curve Parameters** settings if **Use Learn Curve Parameters** is set to TRUE.

**Extraction Mode** specifies how the VI identifies curves in the image. Choose from the following options:

Normal (0)	(Default) Select <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background.
Uniform Regions (1)	Select <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.

Edge Threshold specifies the minimum contrast a seed point must have for it to be begin a curve. Valid values range from 1 to 360. The default value is 75.

Edge Filter Size specifies the width of the edge filter that the VI uses to identify curves in the image. The following options are available:

<b>Fine</b> (0)	Sets the <b>Edge Filter Size</b> to fine.
Normal (1)	(Default) Sets the <b>Edge Filter Size</b> to normal.

**Min Length** specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value. The default value is 25.

- Row Search Step Size specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Col Search Step Size specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Max Endpoint Gap specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
- Closed specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
- Subpixel is ignored by the VI.
- Match Mode is a cluster identifying the conditions under which you want the VI to find template matches. Set the values for these conditions in the Range Settings array.
  - Note Disabling any of these conditions or limiting their ranges decreases the search time.
  - **Rotation** When enabled, the VI searches for occurrences of the template in the inspection image, allowing for template matches to be rotated.
  - Scale When enabled, the VI searches for occurrences of the template in the inspection image regardless of whether possible matches are of a different size in the inspection image. The default scale range is 75% to 125%.
  - Occlusion When enabled, the VI searches for occurrences of the template in the inspection image, allowing for a specified percentage of the template to be occluded. The default

occlusion range is 0% to 25%.

- Subpixel Accuracy determines whether to calculate match locations with subpixel accuracy. The default is FALSE.
- Range Settings is an array of the acceptable ranges for each Match Constraints option. To decrease the search time, limit the range of these constraints as much as possible.
  - Match Constraints specifies the match option whose values you want to constrain by the given range. The following options are available:

Rotation Angle (degrees) (0)	Set a range for this option to specify the angles at which you expect the VI to find template matches in the inspection image. The VI ignores this range if the <b>Rotation</b> is not enabled.
Scale Factor (%) (1)	Set a range for this option to specify the sizes at which you expect the VI to find template matches in the inspection image. The size is expressed as the following percentage: the size of the match divided by size of the template image, multiplied by 100. The VI ignores this range if the <b>Scale</b> is not enabled.
Occlusion (%) (2)	Set a range for this option to specify the amount of occlusion you expect for a match in the inspection image. NI Vision can accurately detect matches that have up to 50% occlusion. The VI ignores this range if the <b>Occlusion</b> is not enabled.

- Min is the minimum value of the range for a given Match Constraint.
- Max is the maximum value of the range for a given Match Constraint.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Use Learn Curve Parameters specifies whether to use the Curve Parameters from the <a href="IMAQ Setup Learn Geometric Pattern">IMAQ Setup Learn Geometric Pattern</a> VI or the <a href="IMAQ Advanced Setup Learn Geometric Pattern">IMAQ Advanced Setup Learn Geometric Pattern</a> VI to extract the target curves during the match phase. The default value is TRUE.
- Match Geometric Pattern Setup Data is a string that contains information about the setup parameters for the matching phase.

  Connect this output to the <a href="IMAQ Match Geometric Pattern">IMAQ Advanced Setup Match Geometric Pattern</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

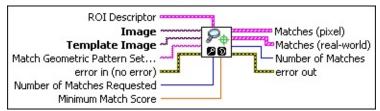
## **IMAQ Match Geometric Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Searches for a pattern, or template image, in the inspection image.

#### **Details**





- ROI Descriptor specifies the region of the inspection image in which to perform geometric matching. This region must be a single rectangle or rotated rectangle contour. By default, the VI uses the entire inspection image for matching.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- **Image** is a reference to the inspection image in which you want to search for the template image.
- **Template Image** is a reference to the color template for which you want to search during the matching phase. The template image is an output of the <a href="IMAQ Learn Geometric Pattern">IMAQ Learn Geometric Pattern</a> VI.
- Match Geometric Pattern Setup Data is a string that contains information from the <a href="MAQ Setup Match Geometric Pattern">IMAQ Setup Match Geometric Pattern</a> VI. If this input is not connected, the VI uses default parameters for all unconfigured options.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number of Matches Requested is the number of valid matches expected.
- **Minimum Match Score** is the smallest score a match can have to be considered valid. The data range for a match is between 0 and 1000.
- Matches (pixel) contains information about a match found in the image.
  - **Position** is a point-coordinate cluster corresponding to the location of the template origin.
  - Angle is the rotation angle, in degrees, of the template at the current match location.
  - **Scale** is the ratio of the size of the matched template relative to the original template image, expressed as a percentage.
  - Score ranks the match results on a scale of 0 to 1000, where 0 equals no match and 1000 equals a perfect match.

- **Bounding Box** is an array of five points that define the boundary of the template.
- Contrast Reversed specifies whether the match is an inverse of the template image. This parameter is TRUE if the match is an inverse of the template image. For example, the match is a white object on a black background but the template image is a black object on a white background. This parameter is FALSE if the match is not the inverse of the template image.
- Occlusion % is the percentage of the match that is occluded.
- Template Target Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all template curves have a corresponding curve in the match region of the inspection image.
- Target Template Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all curves in the match region of the inspection image have a corresponding curve in the template image.
  - D

Note The VI does not compute the Target Template Curve Score by default. Use the IMAQ Advanced Setup Match Geometric Pattern VI to configure the algorithm to compute this value.

- Correlation Score specifies the accuracy of the match obtained by comparing the template image to the match region using a correlation metric that compares the two regions as a function of their pixel values. A score of 1000 indicates a perfect match, and a score of 0 indicates no match.
- Matches (real-world) contains information about a match found in the image.
  - D

Note Matches (real-world) may differ from Matches (pixel) if Image is a calibrated image. If Image is a calibrated

- image, **Position**, **Angle**, and **Bounding Box** are returned in real-world units.
- **Position** is a point-coordinate cluster corresponding to the location of the template origin.
- **Angle** is the rotation angle, in degrees, of the template at the current match location.
- **Scale** is the ratio of the size of the matched template relative to the original template image, expressed as a percentage.
- Score ranks the match results on a scale of 0 to 1000, where 0 equals no match and 1000 equals a perfect match.
- **Bounding Box** is an array of five points that define the boundary of the template.
- Contrast Reversed specifies whether the match is an inverse of the template image. This parameter is TRUE if the match is an inverse of the template image. For example, the match is a white object on a black background but the template image is a black object on a white background. This parameter is FALSE if the match is not the inverse of the template image.
- Occlusion % is the percentage of the match that is occluded.
- Template Target Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all template curves have a corresponding curve in the match region of the inspection image.
- Target Template Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all curves in the match region of the inspection image have a corresponding curve in the template image.



Note The VI does not compute the Target Template Curve Score by default. Use the IMAQ Advanced Setup Match Geometric Pattern VI to configure the algorithm to compute this value.

- Correlation Score specifies the accuracy of the match obtained by comparing the template image to the match region using a correlation metric that compares the two regions as a function of their pixel values. A score of 1000 indicates a perfect match, and a score of 0 indicates no match.
- Number of Matches is the number of template matches found in the inspection image based on the input settings.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

Use one of the following methods to configure the template image for use with this VI.

• Use the <u>IMAQ Read Image And Vision Info</u> VI to load a template image that was created previously using the NI Vision Template Editor.



Note You can access the NI Vision Template Editor from Start»All Programs»National Instruments»Vision»Template Editor.

• Use the <u>IMAQ Learn Geometric Pattern</u> VI to learn the template image.

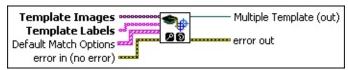
## **IMAQ Learn Multiple Geometric Patterns VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Combines the descriptions of the patterns you want to search for during the matching phase into a multiple geometric template. Use the multiple geometric template to search for these template images in the target image.

#### **Details**





- **Template Images** is an array of template images you want to search for in the target image.
- Template Labels is an array of labels that identify the template images specified in Template Images. Each label in the array identifies the template at the corresponding index in the Template Images array.
- **Default Match Options** are the match options that are used to search for the template images in the target image during the match phase. The options get associated with the templates in the following manner.
  - If you pass an empty array, the algorithm uses an empty string for the Match Setup Data, 1 for Number of Templates, and 800 for the Minimum Match Score.
  - If you pass an array of one element, the algorithm uses the value of this element to match all the templates.
  - If you pass as many elements in this array as there are geometric templates, then the algorithm makes a one to one correspondence between the elements of the array and the geometric templates.
  - Passing any other number of elements in the array results in an error.



Note You can override the match options specified in this VI

- during the match phase by passing new values to the **Match Options** control on the <u>IMAQ Match Multiple Geometric</u> Patterns VI.
- Match Setup Data is a string that contains information from the IMAQ Setup Match Geometric Pattern VI or the IMAQ Advanced Setup Match Geometric Pattern VI. Set this value to an empty string if you want to use default parameters for all options.
- Number of Matches is the number of valid matches expected.
- Minimum Match Score is the smallest score a match can have to be considered valid. The data range for a match is between 0 and 1000.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Multiple Template (out) is a reference to the new multiple geometric template.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

Use one of the following methods to configure the template image for use with this VI.

• Use the <u>IMAQ Read Image And Vision Info</u> VI to load a template image that was created previously using the NI Vision Template Editor.



Note You can access the NI Vision Template Editor from Start»All Programs»National Instruments»Vision»Template Editor.

• Use the <u>IMAQ Learn Geometric Pattern</u> VI to learn the template image.

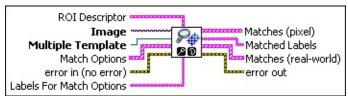
# **IMAQ Match Multiple Geometric Patterns VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Matches the templates specified in the Multiple Template control to the target image.

#### **Details**





- ROI Descriptor is the descriptor of the region of interest specifying the location of the sample in the image. The ROI must be one rectangle or one rotated rectangle. If ROI Descriptor is empty or not connected, the entire image is considered to be the region of interest.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Image is a reference to the source image.
- Multiple Template is the reference to the multiple geometric template on which this VI operates.
- Match Options is an array of match options used to override the Default Match Options set using the IMAQ Learn Multiple Geometric Patterns. This array is used with Labels For Match Options to override match options corresponding to one or more templates specified in Multiple Template. This VI uses the

following rules to determine which match options are overridden:

- If the array is empty, then all templates use the options set during the learn phase.
- If there is only one element in the array, the match options that are overridden depend on the value of Labels For Match Options.
  - If Labels For Match Options is empty, then the algorithm uses the values in the array element for all the templates.
  - If Labels For Match Options is not empty, then the algorithm uses the values in the array element for the templates corresponding to the specified labels.
- If there is more than one element in the array, Labels For Match Options must have the same number of elements as the specified array. The VI makes a one to one correspondence between the Match Options and the Labels for Match Options to determine which template match options are overridden.
- Match Setup Data is a string that contains information from the IMAQ Setup Match Geometric Pattern VI or the IMAQ Advanced Setup Match Geometric Pattern VI. Set this value to an empty string if you want to use default parameters for all options.
- Number of Matches is the number of valid matches expected.
- Minimum Match Score is the smallest score a match can have to be considered valid. The data range for a match is between 0 and 1000.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Labels For Match Options** is an array of labels that identify the templates for which you want to override the default match options. Refer to the values specified for **Match Options** to determine which options are overridden.
- Matches (pixel) contains information about a match found in the image.
  - **Position** is a point-coordinate cluster corresponding to the location of the template origin.
  - **Angle** is the rotation angle, in degrees, of the template at the current match location.
  - Scale is the ratio of the size of the matched template relative to the original template image, expressed as a percentage.
  - Score ranks the match results on a scale of 0 to 1000, where 0 equals no match and 1000 equals a perfect match.
  - **Bounding Box** is an array of five points that define the boundary of the template.
  - Contrast Reversed specifies whether the match is an inverse of the template image. This parameter is TRUE if the match is an inverse of the template image. For example, the match is a white object on a black background but the template image is a black object on a white background. This parameter is FALSE if the match is not the inverse of the template image.

- Occlusion % is the percentage of the match that is occluded.
- Template Target Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all template curves have a corresponding curve in the match region of the inspection image.
- Target Template Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all curves in the match region of the inspection image have a corresponding curve in the template image.
  - Note The VI does not compute the Target Template Curve Score by default. Use the <a href="IMAQ Advanced Setup Match Geometric Pattern">IMAQ Advanced</a> Setup Match Geometric Pattern VI to configure the algorithm to compute this value.
- Correlation Score specifies the accuracy of the match obtained by comparing the template image to the match region using a correlation metric that compares the two regions as a function of their pixel values. A score of 1000 indicates a perfect match, and a score of 0 indicates no match.
- Matched Labels is an array that specifies the template label for corresponding matches in Matches.
- Matches (real-world) contains information about a match found in the image.
  - Note Matches (real-world) may differ from Matches (pixel) if Image is a calibrated image. If Image is a calibrated image, Position, Angle, and Bounding Box are returned in real-world units.
  - **Position** is a point-coordinate cluster corresponding to the location of the template origin.
  - Angle is the rotation angle, in degrees, of the template at the

- current match location.
- **Scale** is the ratio of the size of the matched template relative to the original template image, expressed as a percentage.
- Score ranks the match results on a scale of 0 to 1000, where 0 equals no match and 1000 equals a perfect match.
- **Bounding Box** is an array of five points that define the boundary of the template.
- Contrast Reversed specifies whether the match is an inverse of the template image. This parameter is TRUE if the match is an inverse of the template image. For example, the match is a white object on a black background but the template image is a black object on a white background. This parameter is FALSE if the match is not the inverse of the template image.
- Occlusion % is the percentage of the match that is occluded.
- Template Target Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all template curves have a corresponding curve in the match region of the inspection image.
- Target Template Curve Score specifies how closely the curves in the match region of the inspection, or target, image match the curves in the template. Score values can range from 0 to 1000, where a score of 1000 indicates that all curves in the match region of the inspection image have a corresponding curve in the template image.



Note The VI does not compute the Target Template Curve Score by default. Use the IMAQ Advanced Setup Match Geometric Pattern VI to configure the algorithm to compute this value.

Correlation Score specifies the accuracy of the match obtained by comparing the template image to the match region using a correlation metric that compares the two regions as a function of their pixel values. A score of 1000 indicates a perfect match, and a score of 0 indicates no

match.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Details**

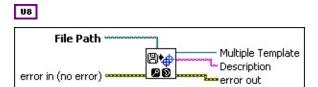
Use one of the following methods to configure multiple geometric templates for use with this VI.

- Use the <u>IMAQ Read Multiple Geometric Template</u> VI to load a multiple geometric template saved using the <u>IMAQ Write Multiple Geometric Template</u> VI.
- Use the <u>IMAQ Learn Multiple Geometric Patterns</u> VI to learn the multiple geometric template.

## **IMAQ Read Multiple Geometric Template VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Creates a new multiple geometric template from the file specified by **File Path**.



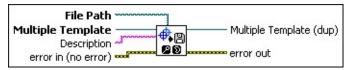
- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Multiple Template is a reference to the new multiple geometric template.
- **Description** describes the multiple geometric template.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write Multiple Geometric Template VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Writes a multiple geometric template to the file specified by **File Path**. This VI saves the exact state of the multiple geometric template.



- **File Path** is the complete pathname, including drive, directory, and filename, of the file to write.
- Multiple Template is the reference to the multiple geometric template on which this VI operates.
- Description describes the multiple geometric template.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Multiple Template (dup) has the same value as Multiple

#### Template.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Dispose Multiple Geometric Template VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Destroys a multiple geometric template and frees the space it occupied in memory.



- Multiple Template is the reference to the multiple geometric template on which this VI operates.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

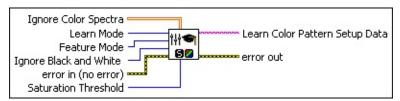
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning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

## **IMAQ Setup Learn Color Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Sets parameters that are used during the color learning stage. Run this VI before using the <u>IMAQ Learn Color Pattern</u> VI.

#### **Examples**



Ignore Color Spectra is an array of color spectra providing a set of colors to exclude from the color features of the template image. Each color to ignore is the dominant color of a color spectrum. Any color excluded during the learning phase is also ignored in the template during the matching phase.



**Note** The dominant color chosen is based on the **Color Sensitivity** set in the <u>IMAQ Setup Match Color Pattern</u> VI.



**Note** To make color spectra, use the <u>IMAQ ColorLearn</u> VI and set **Color Sensitivity** to **High**.

Learn Mode specifies the invariance mode to use when learning the template pattern. Choose from the following values:

<b>All</b> (0)	Extracts information for shift and rotation invariant matching.
Shift Information (1)	(Default) Extracts information for shift invariant matching.
Rotation Information (2)	Extracts information for rotation invariant matching.

Feature Mode specifies the color feature mode to use when setting up a color learn pattern. Choose from the following values:

Color	(Default) Extracts information for distinguishing color
and	and shape features of the template in the matching
Shape (0)	phase.

		Extracts information for distinguishing color features of the template in the matching phase.
		Extracts information for distinguishing shape features of the template in the matching phase.

Ignore Black and White specifies whether black or white colors are excluded from the color features of the template image. Any color excluded during the learning process is also ignored from the template in the match phase. Choose from the following values:

Disabled (0)	(Default) Specifies that black and white pixels are not ignored during the matching phase.
Ignore Black (1)	Specifies that black pixels are ignored during the matching phase.
Ignore White (2)	Specifies that white pixels are ignored during the matching phase.
Ignore Black and White (3)	Specifies that black and white pixels are ignored during the matching phase.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

- Saturation Threshold specifies the threshold used to distinguish between two colors having the same hue values.
- Learn Color Pattern Setup Data is a string that contains information about the setup parameters selected for the learning phase. Connect this output to the <a href="IMAQ Learn Color Pattern">IMAQ Learn Color Pattern</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\Color Pattern Matching\Color Pattern Matching Example.llb

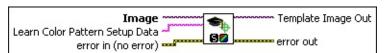
### **IMAQ Learn Color Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Creates a description of the color template image that you are going to look for during the color pattern matching stage. This description data is appended to the input color template image. During the color pattern matching step, the color template descriptor is extracted from the color template image and used to search for the template in the color match image.

#### **Examples**





- Image is a reference to the color template image for which you want to search in the inspection image.
- Learn Color Pattern Setup Data is a string that contains information obtained from <a href="IMAQ Setup Learn Color Pattern">IMAQ Setup Learn Color Pattern</a> VI. If this input is not connected, the VI uses default parameters during the learning phase. The default learns color and shape for shift-invariant pattern matching.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Template Image Out** is a reference to the color template for which you want to search during the matching phase. The template image contains the data that defines the color template pattern for the matching stage.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for an example that uses this VI.

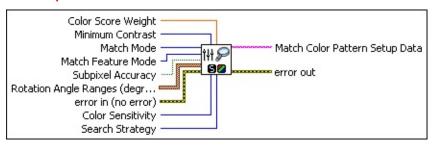
• LabVIEW\examples\Vision\2. Functions\Color Pattern Matching\Color Pattern Matching Example.llb

## **IMAQ Setup Match Color Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Sets parameters that are used during the color matching process. Run this VI before using the <u>IMAQ Match Color Pattern</u> VI.

#### **Examples**



- Color Score Weight (between 0 and 1000) determines the percent contribution of the color score to the final color pattern matching score. The software uses the color score weight for the final match ranking. For example, if you use a weight of 1000, the algorithm finds each match by using both color and shape information and then ranks the matches based on their color scores. If the weight is 0, the matches are ranked based on their shape scores. The default is 500, indicating that the match score uses an equal combination of the color and shape scores.
- Minimum Contrast specifies the minimum expected difference between the maximum luminance value and the minimum luminance value of the color pixels in the inspection image. The default is 0, which allows the algorithm to compute the Minimum Contrast.
- Match Mode specifies the invariance mode to use when looking for the color template pattern in the inspection image. Choose from the following values:

(Default) Searches for the color template pattern in the image, assuming it is not rotated more than 4 degrees.
Searches for the color template pattern in the image with no restriction on the rotation of the template.

**Match Feature Mode** specifies the features to use when looking for the color template pattern in the color image. The following values are valid:

	(Default) Searches for the color template pattern in the image using color and shape features.
Color (1)	Searches for the color template pattern in the image using color features.
Shape (2)	Searches for the color template pattern in the image using shape features.

- Subpixel Accuracy determines whether to return the match results with subpixel accuracy. The default is FALSE. If Match Feature Mode is set to color, the VI ignores Subpixel Accuracy.
- Rotation Angle Ranges (degrees) is an array of angle ranges, where each range specifies how much you expect the color template to rotate in the inspection image. Specify a lower and upper angle, in degrees, for each range.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

- Color Sensitivity specifies the sensitivity of the color information in the image. The default is **Low**. Set this option to **High** when you need to distinguish colors with close hue values.
- Search Strategy specifies how the color features of the image are used during the search phase. Choose from the following strategies:

Conservative (0)	Uses a very small step size, the least amount of subsampling, and all the color information present in the template. The conservative strategy is the most reliable method to look for a template in any image at potentially reduced speed.
Balanced (1)	Uses values in between the aggressive and conservative strategies.
Aggressive (2)	Uses a large step size, a lot of subsampling, and all the color spectrum information from the template.
Very Aggressive (3)	Uses the largest step size, the most subsampling, and only the dominant color from the template to search for the template. Use this strategy when the color in the template is almost uniform, the template is well contrasted from the background and there is a good amount of separation between different occurrences of the template in the image. This strategy is the fastest way to find templates in an image.

- Match Color Pattern Setup Data is a string that contains information about the setup parameters selected for the matching phase. Connect this output to the <a href="MAQ Match Color Pattern">IMAQ Match Color Pattern</a> VI.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for

more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Examples**

Refer to the following for examples that use this VI.

- LabVIEW\examples\Vision\2. Functions\Color Pattern Matching\Color Pattern Matching Example.llb
- LabVIEW\examples\Vision\3. Applications\Fusebox Inspection Example.llb

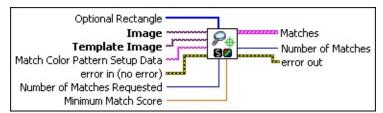
## **IMAQ Match Color Pattern VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Searches for a color pattern, or color template image, in the input Image. Use the <u>IMAQ Learn Color Pattern</u> VI prior to using this VI to ensure that the color template image has been configured for the color match stage.

#### **Examples**





- Optional Rectangle defines a four-element array that contains the left, top, right, and bottom coordinates of the region to process. The VI applies the operation to the entire image if Optional Rectangle is empty or not connected.
- Image is a reference to the color image in which you want to locate the color template image.
- Template Image is a reference to the color template for which you want to search during the matching phase. The template image is an output of the <a href="IMAQ Learn Color Pattern">IMAQ Learn Color Pattern</a> VI.
- Match Color Pattern Setup Data is a string that contains information obtained from the <a href="IMAQ Setup Match Color Pattern">IMAQ Setup Match Color Pattern</a> VI. If this input is not connected, the VI uses default parameters during the matching phase.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number of Matches Requested is the number of valid matches expected.
- Minimum Match Score is the smallest score a match can have to be considered valid. The data range for a match is between 0 and 1000.
- Matches is an array of match clusters. Matches returns the following elements:
  - **Position** is a point-coordinate cluster corresponding to the center of the template.
  - Angle is the rotation angle of the template in the source image at the current match location.
  - Scale is the change in percentage size of the matched template from the original template. Currently this is always set to 1.
  - Score ranks the match results on a scale of 0 to 1,000, where 0 equals no match and 1,000 equals a perfect match.
  - Bounding Box is an array of five points that define the boundary of the template pattern in the image. You can connect this array directly to the <a href="MAQ Overlay Multiple Lines">IMAQ Overlay Multiple Lines</a> VI to overlay the location of a match on your image.
- Number of Matches is the number of template matches found in the inspection image based on the input settings.
- error out contains error information. If error in indicates that an

error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

### **Examples**

Refer to the following for examples that use this VI.

- LabVIEW\examples\Vision\2. Functions\Color Pattern Matching\Color Pattern Matching Example.llb
- LabVIEW\examples\Vision\3. Applications\Fusebox Inspection Example.llb
- LabVIEW\examples\Vision\3. Applications\Blister Pack Inspection Example.llb

# **IMAQ Detect Shapes VI**

Owning Palette: Searching and MatchingInstalled With: NI Vision

**Development Module** 

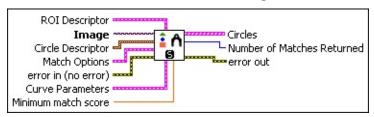
Searches for rectangles, lines, ellipses, or circles in an image.

Use the pull-down menu to select an instance of this VI.

Select an instance
--------------------

#### **IMAQ Detect Circles**

Searches for circles in an image.



- **ROI Descriptor** is the descriptor that defines the rectangle or rotated rectangle within which to detect shapes.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- **Image** is a reference to the image in which you want to search.
- **Circle Descriptor** specifies the sizes of circles to detect.
  - Min Radius specifies the minimum radius describing the circles to detect.
  - Max Radius specifies the maximum radius describing the circles to detect.
- Match Options are the match options used when detecting shapes.
  - **Rotation** specifies whether or not to search for rotated versions of the shape.
  - **Scale** specifies whether or not to search for scaled versions of the shape.
  - Occlusion specifies whether or not to search for occluded versions of the shape.

Range Settings is an array that specifies the allowable ranges for rotation and scale.

Match Constraints specifies the match option whose values you want to constrain by the given range. The following options are available:

Rotation Angle (degrees) (0)	Set a range for this option to specify the angles at which you expect the VI to find template matches in the inspection image. The VI ignores this range if the <b>Rotation</b> is not enabled.
Scale Factor (%) (1)	Set a range for this option to specify the sizes at which you expect the VI to find template matches in the inspection image. The size is expressed as the following percentage: the size of the match divided by size of the template image, multiplied by 100. The VI ignores this range if the <b>Scale</b> is not enabled.
Occlusion (%) (2)	Set a range for this option to specify the amount of occlusion you expect for a match in the inspection image. NI Vision can accurately detect matches that have up to 50% occlusion. The VI ignores this range if the <b>Occlusion</b> is not enabled.

Min is the minimum value of the range for a given Match Constraint.

Max is the maximum value of the range for a given Match Constraint.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Curve Parameters** contains information about how curves are extracted from the inspection image during the match phase.
  - **Extraction Mode** specifies how the VI identifies curves in the image. Choose from the following options:

Normal (0)	(Default) Select <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background.
Uniform Regions (1)	Select <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.

- Edge Threshold specifies the minimum contrast a seed point must have for it to be begin a curve. Valid values range from 1 to 360. The default value is 75.
- Edge Filter Size specifies the width of the edge filter that the VI uses to identify curves in the image. The following options are available:

Fine (0)	Sets the <b>Edge Filter Size</b> to fine.
Normal (1)	(Default) Sets the <b>Edge Filter Size</b> to normal.

Min Length specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any

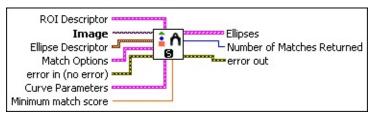
- curves that have a length less than this value. The default value is 25.
- Row Search Step Size specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Col Search Step Size specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Max Endpoint Gap specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
- Closed specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
- **Subpixel** is ignored by the VI.
- Minimum match score is the minimum score a returned shape can have.
- **Circles** is an array of circle clusters that describe each found circle.
  - **Position** is the coordinates of the found circle center.
  - **Radius** is the radius of the found circle.
  - Score is the score of the found circle. Values range from 0 to 1000, where 1000 represents a perfect match.
- Number of Matches Returned is the number of matches found.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - **status** is TRUE (X) if an error occurred or FALSE

(checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ Detect Ellipses**

Searches for ellipses in an image.



- **ROI Descriptor** is the descriptor that defines the rectangle or rotated rectangle within which to detect shapes.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- **Image** is a reference to the image in which you want to search.
- **Ellipse Descriptor** specifies the sizes of ellipses to detect.
  - Min Major Radius specifies the minimum length of the semimajor axis describing the ellipses to detect.
  - Max Major Radius specifies the maximum length of the semimajor axis describing the ellipses to detect.
  - Min Minor Radius specifies the minimum length of the semiminor axis describing the ellipses to detect.
  - Max Minor Radius specifies the maximum length of the semiminor axis describing the ellipses to detect.
- Match Options are the match options used when detecting shapes.
  - **Rotation** specifies whether or not to search for rotated versions of the shape.

- **Scale** specifies whether or not to search for scaled versions of the shape.
- Occlusion specifies whether or not to search for occluded versions of the shape.
- Range Settings is an array that specifies the allowable ranges for rotation and scale.
  - Match Constraints specifies the match option whose values you want to constrain by the given range. The following options are available:

Rotation Angle (degrees) (0)	Set a range for this option to specify the angles at which you expect the VI to find template matches in the inspection image. The VI ignores this range if the <b>Rotation</b> is not enabled.
Scale Factor (%) (1)	Set a range for this option to specify the sizes at which you expect the VI to find template matches in the inspection image. The size is expressed as the following percentage: the size of the match divided by size of the template image, multiplied by 100. The VI ignores this range if the <b>Scale</b> is not enabled.
Occlusion (%) (2)	Set a range for this option to specify the amount of occlusion you expect for a match in the inspection image. NI Vision can accurately detect matches that have up to 50% occlusion. The VI ignores this range if the <b>Occlusion</b> is not enabled.

- Min is the minimum value of the range for a given Match Constraint.
- Max is the maximum value of the range for a given Match Constraint.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to

**error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Curve Parameters** contains information about how curves are extracted from the inspection image during the match phase.
  - **Extraction Mode** specifies how the VI identifies curves in the image. Choose from the following options:

Normal (0)	(Default) Select <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background.
Uniform Regions (1)	Select <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.

- Edge Threshold specifies the minimum contrast a seed point must have for it to be begin a curve. Valid values range from 1 to 360. The default value is 75.
- Edge Filter Size specifies the width of the edge filter that the VI uses to identify curves in the image. The following options are available:

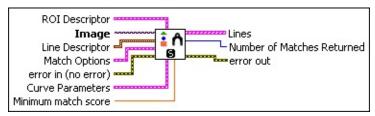
<b>Fine</b> (0)	Sets the <b>Edge Filter Size</b> to fine.
Normal (1)	(Default) Sets the <b>Edge Filter Size</b> to normal.

- Min Length specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value. The default value is 25.
- Row Search Step Size specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Col Search Step Size specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Max Endpoint Gap specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
- Closed specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
- **Subpixel** is ignored by the VI.
- Minimum match score is the minimum score a returned shape can have.
- **Ellipses** is an array of ellipse clusters that describe each found ellipse.
  - **Position** are the coordinates of the found ellipse center.
  - Angle is the orientation of the major axis with respect to the horizontal.
  - Major Radius is the length of the semi-major axis.
  - Minor Radius is the length of the semi-minor axis.
  - Score is the score of the found ellipse. Values range from 0 to 1000, where 1000 represents a perfect match.

- Number of Matches Returned is the number of matches found.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ Detect Lines**

Searches for lines in an image.



- **ROI Descriptor** is the descriptor that defines the rectangle or rotated rectangle within which to detect shapes.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- **Image** is a reference to the image in which you want to search.
- Line Descriptor specifies the lengths of lines to detect.
  - Min Length specifies the minimum length of the described line.
  - Max Length specifies maximum length of the described line.
- Match Options are the match options used when detecting shapes.
  - **Rotation** specifies whether or not to search for rotated versions of the shape.
  - Scale specifies whether or not to search for scaled versions of the shape.
  - Occlusion specifies whether or not to search for occluded versions of the shape.
  - Range Settings is an array that specifies the allowable

ranges for rotation and scale.

Match Constraints specifies the match option whose values you want to constrain by the given range. The following options are available:

Rotation Angle (degrees) (0)	Set a range for this option to specify the angles at which you expect the VI to find template matches in the inspection image. The VI ignores this range if the <b>Rotation</b> is not enabled.
Scale Factor (%) (1)	Set a range for this option to specify the sizes at which you expect the VI to find template matches in the inspection image. The size is expressed as the following percentage: the size of the match divided by size of the template image, multiplied by 100. The VI ignores this range if the <b>Scale</b> is not enabled.
Occlusion (%) (2)	Set a range for this option to specify the amount of occlusion you expect for a match in the inspection image. NI Vision can accurately detect matches that have up to 50% occlusion. The VI ignores this range if the <b>Occlusion</b> is not enabled.

- Min is the minimum value of the range for a given Match Constraint.
- Max is the maximum value of the range for a given Match Constraint.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Curve Parameters** contains information about how curves are extracted from the inspection image during the match phase.
  - **Extraction Mode** specifies how the VI identifies curves in the image. Choose from the following options:

Normal (0)	(Default) Select <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background.
Uniform Regions (1)	Select <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.

- Edge Threshold specifies the minimum contrast a seed point must have for it to be begin a curve. Valid values range from 1 to 360. The default value is 75.
- Edge Filter Size specifies the width of the edge filter that the VI uses to identify curves in the image. The following options are available:

<b>Fine</b> (0)	Sets the <b>Edge Filter Size</b> to fine.
Normal (1)	(Default) Sets the <b>Edge Filter Size</b> to normal.

Min Length specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value. The default

value is 25.

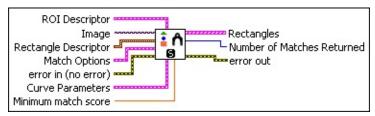
- Row Search Step Size specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Col Search Step Size specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Max Endpoint Gap specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
- Closed specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
- **Subpixel** is ignored by the VI.
- Minimum match score is the minimum score a returned shape can have.
- Lines is an array of line clusters that describe each found line.
  - **Start Point** The starting point of the line.
  - **End Point** The ending point of the line.
  - **Length** The length of the line measured in pixels from the start point to the end point.
  - Angle The orientation of the line with respect to the horizontal.
  - **Score** is the score of the found line. Values range from 0 to 1000, where 1000 represents a perfect match.
- Number of Matches Returned is the number of matches found.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ Detect Rectangles**

Searches for rectangles in an image.



- **ROI Descriptor** is the descriptor that defines the rectangle or rotated rectangle within which to detect shapes.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- **Image** is a reference to the image in which you want to search.
- Rectangle Descriptor specifies the sizes of rectangles to detect.
  - Note Width is defined as the length of the shorter side of a rectangle and height is defined as the longer side of the rectangle you want to search for.
  - Min Width is the minimum width describing the rectangles to detect.
  - Max Width is the maximum width describing the rectangles to detect.
  - Min Height is the minimum height describing the rectangles to detect.
  - Max Height is the maximum height describing the rectangles to detect.
- Match Options are the match options used when detecting

#### shapes.

- **Rotation** specifies whether or not to search for rotated versions of the shape.
- **Scale** specifies whether or not to search for scaled versions of the shape.
- Occlusion specifies whether or not to search for occluded versions of the shape.
- Range Settings is an array that specifies the allowable ranges for rotation and scale.
  - Match Constraints specifies the match option whose values you want to constrain by the given range. The following options are available:

Rotation Angle (degrees) (0)	Set a range for this option to specify the angles at which you expect the VI to find template matches in the inspection image. The VI ignores this range if the <b>Rotation</b> is not enabled.
Scale Factor (%) (1)	Set a range for this option to specify the sizes at which you expect the VI to find template matches in the inspection image. The size is expressed as the following percentage: the size of the match divided by size of the template image, multiplied by 100. The VI ignores this range if the <b>Scale</b> is not enabled.
Occlusion (%) (2)	Set a range for this option to specify the amount of occlusion you expect for a match in the inspection image. NI Vision can accurately detect matches that have up to 50% occlusion. The VI ignores this range if the <b>Occlusion</b> is not enabled.

- Min is the minimum value of the range for a given Match Constraint.
- Max is the maximum value of the range for a given Match Constraint.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Curve Parameters** contains information about how curves are extracted from the inspection image during the match phase.
  - **Extraction Mode** specifies how the VI identifies curves in the image. Choose from the following options:

Normal (0)	(Default) Select <b>Normal</b> if you want the VI to make no assumptions about the uniformity of objects in the image or the image background.
	Select <b>Uniform Regions</b> if you want the VI to assume that either the objects in the image or the image background consist of uniform pixel values. This allows the VI to more accurately calculate the external curves of the objects.

**Edge Threshold** specifies the minimum contrast a seed point must have for it to be begin a curve. Valid values range from 1 to 360. The default value is 75.

Edge Filter Size specifies the width of the edge filter that the VI uses to identify curves in the image. The following options are available:

<b>Fine</b> (0)	Sets the <b>Edge Filter Size</b> to fine.
Normal (1)	(Default) Sets the <b>Edge Filter Size</b> to normal.

- Min Length specifies the length, in pixels, of the smallest curve that you want the VI to identify. The VI ignores any curves that have a length less than this value. The default value is 25.
- Row Search Step Size specifies the distance, in the y direction, between the image rows that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Col Search Step Size specifies the distance, in the x direction, between the image columns that the algorithm inspects for curve seed points. Valid values range from 1 to 255. The default value is 15.
- Max Endpoint Gap specifies the maximum gap, in pixels, between the endpoints of a curve that the VI identifies as a closed curve. If the gap is larger than this value, the VI identifies the curve as an open curve. The default value is 10.
- Closed specifies whether to identify only closed curves in the image. The default value is FALSE, which specifies that the VI should identify both open and closed curves.
- Subpixel is ignored by the VI.
- Minimum match score is the minimum score a returned shape can have.
- **Rectangles** is an array of rectangle clusters that describe each found rectangle.
  - Note Width is defined as the length of the shorter side of a rectangle and height is defined as the longer side of the rectangle you want to search for.
  - Corners returns an array of coordinates that specify the

- corners of the rectangle.
- Angle is the orientation of the rectangle.
- **Width** is the width of the rectangle.
- **Bullet** Height is the height of the rectangle.
- **Score** is the score of the rectangle.
- Number of Matches Returned is the number of matches found.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Shape Match Tool VI**

Owning Palette: <u>Searching and Matching</u>Installed With: NI Vision Development Module

Finds objects in an image whose shape matches the shape of the object specified by a template.



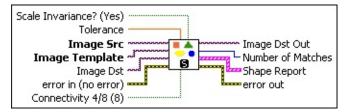
**Note** This VI modifies the source image. If you need the original source image, create a copy of the image using the <u>IMAQ Copy</u> VI before using this VI.



**Note** The input image must be a binary image that contains only pixel values of 0 or 1.

#### **Details**

#### U8 I16



- Scale Invariance? (Yes) specifies whether to use size or scale invariance when matching objects in the source image to the template object.
- **Tolerance** indicates the maximum allowed deviation between the template shape and similar shapes in the image. Valid values are between 0 and 1.
- Image Src is a reference to the source image.
- Image Template is a reference to the 8-bit binary template for which you want to search.
- Image Dst is a reference to an binary image that contains the objects from the Image Src that match the object in the Image Template.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Connectivity 4/8 (8) specifies the type of connectivity used by the algorithm for particle detection. The connectivity mode directly determines whether an adjacent pixel belongs to the same particle or a different particle. The default is 8. The following values are possible:
  - 8 (TRUE) Particle detection is performed in connectivity mode 84 (FALSE) Particle detection is performed in connectivity mode 4
- Image Dst Out is a reference to the destination image. If Image Dst is connected, Image Dst Out is the same as Image Dst.
  Otherwise, Image Dst Out refers to the image referenced by Image Src.
- Number of Matches is the number of template matches found in the inspection image based on the input settings.
- **Shape Report** is an array of clusters in which each cluster contains the following information on the matched objects:
  - Global Rectangle contains the bounding rectangle of the object.
    - **x1Left** indicates the x-coordinate of the top-left corner of the rectangle.

- **y1Top** indicates the y-coordinate of the top-left corner of the rectangle.
- **x2Right** indicates the x-coordinate of the bottom-right corner of the rectangle.
- **y2Bottom** indicates the y-coordinate of the bottom-right corner of the rectangle.
- **Centroid** is a point-coordinate cluster containing the location of the centroid of the object.
- Object Size is the size in pixels of the object.
- Score (between 1 and 1000) specifies how similar the object in the image is to the template. A score of 1000 implies a perfect match.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **Details**

The matching process is invariant to rotation and can be set to be invariant to the scale of the objects. This VI requires that the objects have been separated from the background and that the input image is binary. Before performing the match operation, this VI labels the image to give each object in the image a unique ID.

Refer to the NI Developer Zone document <u>How to Use the Shape</u> <u>Matching Functionality</u> for more information about shape matching.

# Caliper

Owning Palette: Machine VisionInstalled With: NI Vision Development Module

Use the Caliper VIs to detect specific edges, peaks, rotational shifts, and other features in an image along regions of interest that you specify.

Palette Object	Description
IMAQ Simple Edge	Finds step edges along an array of pixel coordinates. This VI can return the first, both the first and the last, or all the edges found.
IMAQ Edge Tool 2	Finds edges along a path defined in the image.
IMAQ Peak- Valley Detector	Finds the location, amplitude, and second derivative of peaks or valleys in the input array.
IMAQ Caliper Tool	Finds edge pairs along a specified path in the image.
IMAQ Line Gauge	Measures the distance between selected edges with high- precision subpixel accuracy.
IMAQ Rake 2	Finds edges along a set of parallel lines defined inside a rectangular region.
IMAQ Spoke 2	Finds edges along radial lines specified inside an annular region.
IMAQ Concentric Rake 2	Finds edges along concentric circular or angular paths in the image.
IMAQ Interpolate 1D	Resamples an array of pixels from an image using multiple interpolation functions. Use this VI to perform subpixel analysis of pixel profiles in the image.
IMAQ Rotation	Detects the rotational shift between two images, generally between a reference image, with the part being inspected at

	a known orientation, and another image that has the part in an unknown position.
IMAQ Find Straight Edges 2	Finds straight edges (lines) within a region in an image.

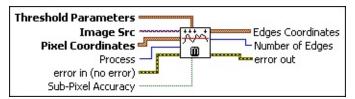
# IMAQ Simple Edge VI

Owning Palette: CaliperInstalled With: NI Vision Development Module

Finds step edges along an array of pixel coordinates. This VI can return the first, both the first and the last, or all the edges found.

#### **Details** Examples





- Threshold Parameters is a cluster containing information used to determine whether a change in pixel value is considered as an edge. This cluster contains the following parameters:
  - Level Type chooses between looking for peaks (positive-going bumps) and valleys (negative-going bumps).
  - Threshold Level can be either absolute or relative. Absolute threshold is based on the pixel values. Relative threshold is expressed as a percentage of the pixel-value range found along the path defined by the pixel coordinates.
  - Hysteresis can be either absolute or relative. This parameter determines the difference in threshold level between a rising and a falling edge, enabling accurate detection in noisy images.
- Image Src is a reference to the source image.
- Pixel Coordinates is an array containing the coordinates of the pixels to inspect. You can obtain the pixel coordinates from an ROI descriptor using the <a href="MAQ ROIProfile">IMAQ ROIProfile</a> VI.
- Process determines the type of search. Choose from the following values:

Get First Edge (0)	Returns the first edge.
Get First + Last Edge (1)	Returns the first and last edge.

Get All Edges (2)	(Default) Returns all edges found along the
	search path(s).

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Sub-Pixel Accuracy determines the accuracy required for the location of the edge coordinates. Setting this control to FALSE enables fast edge detection. A subpixel localization of the edges is obtained when this control is TRUE. The subpixel result is computed using a local quadratic interpolation. The default is FALSE.
- Edges Coordinates returns the coordinates of the detected edges. If the Sub-Pixel Accuracy control is set to TRUE, the result is given with subpixel accuracy.
- Number of Edges specifies the number of edges found.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

#### **Details**

Use the Threshold Parameters to determine whether a change in the pixel values is considered an edge. The main criterion is a Threshold Level on the pixel values. This threshold value can be either relative or absolute. Relative threshold level is specified as a percentage of the pixel range found in the pixel path. Absolute value is based directly on the pixel grayscale intensity. You can compute the location of the edges with subpixel accuracy.

# **Examples**

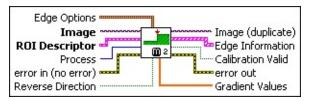
Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Rotating Part Example.llb

# **IMAQ Edge Tool 2 VI**

**Owning Palette:** CaliperInstalled With: NI Vision Development Module Finds edges along a path defined in the image.

U8 I16 SGL



**Edge Options** specifies the parameters that are used to compute the edge gradient information and detect the edges.

**Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
Falling Edges (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- Width specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- Minimum Edge Strength specifies the minimum edge strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the following options:

Zero Order (0)	Rounds to the nearest integral edge location
Bilinear (1)	Uses bilinear interpolation to compute the edge location
Bilinear Fixed (4)	(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge location

**Data Processing Method** is the method used to process the data extracted for edge detection.

• , ,	(Default) Averages the data extracted for edge detection
` '	Takes the median of the data extracted for edge detection

- **Image** is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the rectangle, rotated rectangle, annulus, oval, line, broken line, polygon, or freehand region within which edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Process determines the type of search. Choose from the following values:

Get First Edge (0)	Returns the first edge.
Get First + Last Edge (1)	Returns the first and last edge.
Get All Edges (2)	(Default) Returns all edges found along the search path(s).
Get Best Edge (3)	Returns the strongest edges found along the search path(s).

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or

function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Reverse Direction**, when TRUE, reverses the direction that the ROI traverses to find edges.
- Image (duplicate) is a reference to the source image.
- **Edge Information** is an array that contains information about each detected edge.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
  - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
  - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
  - Strength is the strength (gradient magnitude) at the located edge.

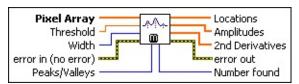
- **Noise** is the strength of the noise associated with the current edge.
- **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- Calibration Valid indicates if the calibration information associated with the edges found along the search line is valid. If the calibration information is invalid for any pixel that is used for edge detection, the Calibration Valid Boolean is FALSE.
- Gradient Values is an array that contains the calculated edge strengths along the user-defined search area.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Peak-Valley Detector VI**

Owning Palette: CaliperInstalled With: NI Vision Development Module

Finds the location, amplitude, and second derivative of peaks or valleys in the input array. This VI is based on an algorithm that fits a quadratic polynomial to sequential groups of data points. The number of data points used in the fit is specified by **Width**. For each peak or valley, the quadratic fit is tested against the **Threshold** level. Peaks with heights lower than the threshold or valleys with troughs higher than the **Threshold** are ignored.

#### **Examples**



- Pixel Array contains the pixel data to be processed.
- Threshold rejects peaks or valleys that are too small. Any peak found with a fitted amplitude that is less than Threshold is ignored. Valleys are ignored if the fitted trough is greater than Threshold.
- Width specifies the number of consecutive data points to use in the quadratic least-squares fit. Width must be greater than or equal to 3 but should be no larger than one-fourth of the approximate width of the peaks or valleys. Large widths can reduce the apparent amplitude of peaks and shift the apparent location. The less noise your data has, the smaller the value you can choose.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Peaks/Valleys specifies whether to look for peaks (positive-going bumps) or valleys (negative-going bumps). The settings for this control are 0 (peaks) and 1 (valleys).
- **Locations** is an array containing the locations of peaks or valleys found in the current block of data. Locations are reported in indices from the beginning of processing.
- Amplitudes is an array containing the amplitudes of peaks or valleys found in the current block of data.
- **2nd Derivatives** is an array containing the second derivatives of peaks or valleys found in the current block of data.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number found is the number of peaks or valleys found in the current block of data. Number found is the size of the Locations, Amplitudes, and 2nd Derivatives arrays.

# **Examples**

Refer to the following for an example that uses this VI.

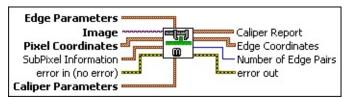
• LabVIEW\examples\Vision\3. Applications\Gauging Example.llb

### **IMAQ Caliper Tool VI**

Owning Palette: CaliperInstalled With: NI Vision Development Module

Finds edge pairs along a specified path in the image. This VI performs an edge extraction and finds edge pairs based on specified criteria such as the distance between the leading and trailing edges and edge polarity.





- **Edge Parameters** is a cluster defining the characteristics of the filter used to detect the edges. This cluster consists of the following parameters:
  - Contrast specifies the threshold for the contrast of the edge.
    Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - Filter width specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
- Image is a reference to the source image.
- Pixel Coordinates is an array containing the coordinates of the pixels to inspect. You can obtain the pixel coordinates from an ROI descriptor using the <a href="MAQ ROIProfile">IMAQ ROIProfile</a> VI.
- SubPixel Information is a cluster containing the following parameters for subpixel analysis:
  - SubPixel Accuracy specifies the number of samples to obtain from a pixel. For example, a subpixel accuracy of One Fourth specifies that each pixel is split into four subpixels.
  - Interpolation Type specifies the method used to perform the

interpolation. The following values are valid:

Quadratic (1)	Uses the quadratic formula to compute the pixel location
Cubic Spline (2)	Uses cubic splines to compute the pixel location

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Caliper Parameters specifies the following parameters for finding edge pairs:
  - Polarity specifies the polarity of the leading and trailing edge in the search process. The following values are possible:

None (0)	Polarity of the edges is ignored.
	The leading edge has a dark-to-light polarity, and the trailing edge has a light-to-dark polarity.
	The leading edge has a light-to-dark polarity, and the trailing edge has a dark-to-light polarity.

The leading edge has a dark-to-light polarity, and the trailing edge has a dark-to-light polarity.
The leading edge has a light-to-dark polarity, and the trailing edge has a light-to-dark polarity.

- Separation Separation specifies the desired distance between the leading and trailing edges of an edge pair. Edge pairs with separation distance outside the range [Separation Separation Dev., Separation + Separation Dev.] are ignored. If you set this parameter to 0, this VI finds all edge pairs.
- Separation Dev. specifies a tolerance value for the separation between the edges. This value influences the score of the detected edge pairs.
- Caliper Report is an array of clusters that contain the following information about the detected edge pairs:
  - **Edge1 Position** is the location of the leading edge.
  - Edge1 Contrast is the contrast of the leading edge.
  - Edge2 Position is the location of the trailing edge.
  - Edge2 Contrast is the contrast of the trailing edge.
  - **Separation** is the computed distance in pixels between the edges in the edge pair.
  - **Score** is unused.
- **Edge Coordinates** is an array of point clusters consisting of the spatial coordinates of the detected edges.
- Number of Edge Pairs returns the number of detected edge pairs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

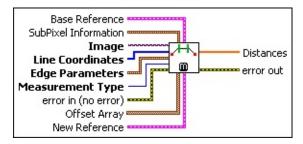
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ Line Gauge VI**

Owning Palette: CaliperInstalled With: NI Vision Development Module

Measures the distance between selected edges with high-precision subpixel accuracy. You can use different measurement modes with this VI, such as measuring distances between points and edges and vice versa. This VI also can step and repeat its measurements across the image. Depending on the type of the measurement, this VI first determines the relevant edges in the image before computing the distances.





- **Base Reference** is the base coordinate system. This cluster contains the following elements:
  - Origin is a point cluster that specifies the origin of the base reference coordinate system.
  - Angle (deg) is the angle the base reference coordinate system makes with the image coordinate system.
  - Axis Reference is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.
- SubPixel Information is a cluster containing the following parameters for subpixel analysis:
  - Interpolation Type specifies the method used to perform the interpolation. The following values are valid:

Quadratic (1)	Uses the quadratic formula to compute the pixel location
	Uses cubic splines to compute the pixel location

SubPixel Accuracy specifies the number of samples to

obtain from a pixel. For example, a subpixel accuracy of **One Fourth** specifies that each pixel is split into four subpixels.

- **Image** is a reference to the source image.
- Line Coordinates is an array specifying the pixel coordinates that form the end points of the line.
- **Edge Parameters** is a cluster defining the characteristics of the filter used to detect the edges. This cluster consists of the following parameters:
  - Contrast specifies the threshold for the contrast of the edge. Only edges with a contrast greater than this value are used in the detection process. Contrast is defined as the difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
  - Filter width specifies the number of pixels that the VI averages to find the contrast at either side of the edge.
  - Steepness specifies the slope of the edge. This value represents the number of pixels that correspond to the transition area of the edge.
- Measurement Type specifies the type of measurement to perform. The following values are valid:
  - Ŋ

**Note** Subpixel information is not used when measuring distances between points.

_	Measures the distance between the first and last edge locations along a line in the image.
	Measures the distance between the first edge in the image and the end point of a line.
	Measures the distance between the start point of the line and the first edge along the line of the image.
Point To Point (3)	Measures the distance between two points in the image.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Offset Array is an array of point clusters that specifies the shift in the start and end points of the pixel coordinates. The Offset Array is used for repeating the measurement across the image.
- New Reference is the new coordinate system.

This cluster contains the following elements:

- **Origin** is a point cluster that specifies the origin of the base reference coordinate system.
- Angle (deg) is the angle the base reference coordinate system makes with the image coordinate system.
- **Axis Reference** is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.
- Distances is an array containing the measured distances.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

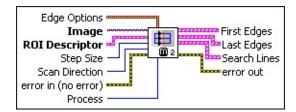
### **IMAQ Rake 2 VI**

Owning Palette: CaliperInstalled With: NI Vision Development Module

Finds edges along a set of parallel lines defined inside a rectangular region. Edges are determined based on their contrast and slope.

#### **Details**





**Edge Options** specifies the parameters that are used to compute the edge gradient information and detect the edges.

**Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
Falling Edges (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- Width specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the following options:

Zero Order (0)	Rounds to the nearest integral edge location
` '	Uses bilinear interpolation to compute the edge location

(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge
location

**Data Processing Method** is the method used to process the data extracted for edge detection.

• • • • • • • • • • • • • • • • • • • •	(Default) Averages the data extracted for edge detection
` '	Takes the median of the data extracted for edge detection

- **Image** is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the rectangle or rotated rectangle within which edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Step Size is the distance, in pixels, between each parallel line in the rectangular region.
- Scan Direction specifies the direction in which the VI searches for edges along the parallel search lines. The following options are valid:

Left to Right (0)	The VI searches from left to right.
Right to Left (1)	The VI searches from right to left.
<b>Top to Bottom</b> (2)	The VI searches from top to bottom.
<b>Bottom to Top</b> (3)	The VI searches from bottom to top

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this

VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Process determines the type of search. Choose from the following values:

Get First Edge (0)	Returns the first edge.
Get First + Last Edge (1)	Returns the first and last edge.
Get All Edges (2)	(Default) Returns all edges found along the search path(s).
Get Best Edge (3)	Returns the strongest edges found along the search path(s).

- First Edges returns information about all edges detected as first edges.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.

- **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
- Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
- **Strength** is the strength (gradient magnitude) at the located edge.
- **Noise** is the strength of the noise associated with the current edge.
- **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- Last Edges returns information about all edges detected as last edges.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
  - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
  - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
  - Strength is the strength (gradient magnitude) at the located edge.
  - **Noise** is the strength of the noise associated with the current edge.
  - **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- Search Lines contains information about the search lines used to

detect the edges.

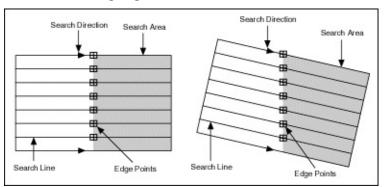
- **Start Point** is the start point of the search line in pixel units.
- **End Point** is the end point of the search line in pixel units.
- **Edges** contains information of all the edges detected along the search line.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
  - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
  - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
  - Strength is the strength (gradient magnitude) at the located edge.
  - **Noise** is the strength of the noise associated with the current edge.
  - **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- **Edge Map** is an array that contains the calculated edge strengths along the search line.
- Calibration Valid indicates if the calibration information associated with the edges found along the search line is valid. If the calibration information is invalid for any pixel that is used for edge detection, the Calibration Valid Boolean is FALSE.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for

more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Details**

The following figure illustrates the rake:



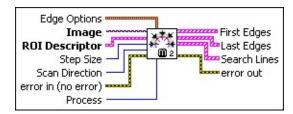
## IMAQ Spoke 2 VI

Owning Palette: CaliperInstalled With: NI Vision Development Module

Finds edges along radial lines specified inside an annular region. The edges are determined based on their contrast and slope.

#### **Details**





**Edge Options** specifies the parameters that are used to compute the edge gradient information and detect the edges.

**Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
Falling Edges (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- Width specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the following options:

Zero Order (0)	Rounds to the nearest integral edge location
` '	Uses bilinear interpolation to compute the edge location

Bilinear	(Default) Uses the fixed-point computation of
Fixed (4)	bilinear interpolation to determine the edge
	location

**Data Processing Method** is the method used to process the data extracted for edge detection.

• • • • • • • • • • • • • • • • • • • •	(Default) Averages the data extracted for edge detection
` '	Takes the median of the data extracted for edge detection

- **Image** is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the annular region within which edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Step Size is the angle, in degrees, between each radial line in the annular region.
- Scan Direction specifies the direction in which the VI searches for edges along the radial search lines. The following options are valid:

Outside to Inside (0)	The VI searches from outside to inside.
<b>Inside to Outside</b> (1)	The VI searches from inside to outside.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or

function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Process determines the type of search. Choose from the following values:

Get First Edge (0)	Returns the first edge.
Get First + Last Edge (1)	Returns the first and last edge.
Get All Edges (2)	(Default) Returns all edges found along the search path(s).
Get Best Edge (3)	Returns the strongest edges found along the search path(s).

- First Edges returns information about all edges detected as first edges.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
  - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.

- Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
- **Strength** is the strength (gradient magnitude) at the located edge.
- **Noise** is the strength of the noise associated with the current edge.
- **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- Last Edges returns information about all edges detected as last edges.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
  - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
  - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
  - **Strength** is the strength (gradient magnitude) at the located edge.
  - **Noise** is the strength of the noise associated with the current edge.
  - **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- Search Lines contains information about the search lines used to detect the edges.
  - **Start Point** is the start point of the search line in pixel units.

**End Point** is the end point of the search line in pixel units.

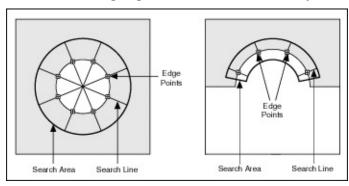
- **Edges** contains information of all the edges detected along the search line.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
  - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
  - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
  - Strength is the strength (gradient magnitude) at the located edge.
  - Noise is the strength of the noise associated with the current edge.
  - **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- **Edge Map** is an array that contains the calculated edge strengths along the search line.
- Calibration Valid indicates if the calibration information associated with the edges found along the search line is valid. If the calibration information is invalid for any pixel that is used for edge detection, the Calibration Valid Boolean is FALSE.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE

(checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Details**

The following figure illustrates the spoke edge search tool:



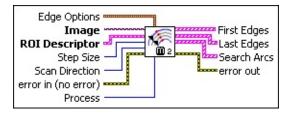
## **IMAQ Concentric Rake 2 VI**

Owning Palette: CaliperInstalled With: NI Vision Development Module

Finds edges along concentric circular or angular paths in the image. Edges are determined based on their contrast and slope.

#### **Details**





**Edge Options** specifies the parameters that are used to compute the edge gradient information and detect the edges.

**Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
Falling Edges (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- Width specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the following options:

Zero Order (0)	Rounds to the nearest integral edge location
• •	Uses bilinear interpolation to compute the edge location

	(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge
` ,	location

**Data Processing Method** is the method used to process the data extracted for edge detection.

Average (0)	(Default) Averages the data extracted for edge detection
Median (1)	Takes the median of the data extracted for edge detection

- **Image** is a reference to the source image.
- ROI Descriptor is a descriptor that defines the circular or annular region within which concentric paths are defined. The ROI must be an annulus.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Step Size is the radial distance, in pixels, between each concentric line in the annular region.
- Scan Direction specifies the direction in which the VI searches for edges along the concentric search lines. The following options are valid:

	Searches for edges in a counterclockwise direction.
( )	Searches for edges in a clockwise direction.

error in (no error) describes the error status before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Process determines the type of search. Choose from the following values:

Get First Edge (0)	Returns the first edge.
Get First + Last Edge (1)	Returns the first and last edge.
Get All Edges (2)	(Default) Returns all edges found along the search path(s).
Get Best Edge (3)	Returns the strongest edges found along the search path(s).

- First Edges returns information about all edges detected as first edges.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.

- **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
- Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
- Strength is the strength (gradient magnitude) at the located edge.
- **Noise** is the strength of the noise associated with the current edge.
- **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- Last Edges returns information about all edges detected as last edges.
  - **Location (Pixel)** is the location of the edge point in the image in pixel units.
  - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
  - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
  - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
  - **Strength** is the strength (gradient magnitude) at the located edge.
  - **Noise** is the strength of the noise associated with the current edge.
  - **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.

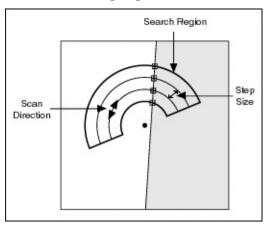
- Search Arcs contains information about the search arcs used to detect edges
  - Center Point is a point-coordinate cluster that describes the (x, y) coordinates of the center of the search arc.
  - **Radius** is the radius, in pixels, of the search arc.
  - Start Angle is the first angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
  - **End Angle** is the second angle of the circular strip. The angle is measured counterclockwise from the x-axis of a coordinate system centered on the center of the circle or annulus.
  - **Edge Info** is an array that contains information about each detected edge along the search arc.
    - **Location (Pixel)** is the location of the edge point in the image in pixel units.
    - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
    - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
    - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
    - Strength is the strength (gradient magnitude) at the located edge.
    - **Noise** is the strength of the noise associated with the current edge.
    - **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
  - **Edge Map** is an array that contains the calculated edge strengths along the search line.
  - **Calibration Valid** indicates if the calibration information

associated with the edges found along the search line is valid. If the calibration information is invalid for any pixel that is used for edge detection, the **Calibration Valid** Boolean is FALSE.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **Details**

The following figure illustrates the concentric rake.

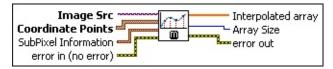


# **IMAQ Interpolate 1D VI**

Owning Palette: CaliperInstalled With: NI Vision Development Module

Resamples an array of pixels from an image using multiple interpolation functions. Use this VI to perform subpixel analysis of pixel profiles in the image.





- Image Src is a reference to the source image.
- **Coordinate Points** is an array consisting of the spatial coordinates of pixels in the image.
- **SubPixel Information** is a cluster containing the following parameters for subpixel analysis:
  - Interpolation Type specifies the method used to perform the interpolation. The following values are possible:

Bi-Linear (0)	Uses linear interpolation in both the x- and y-directions to compute the pixel location
Quadratic (1)	Uses the quadratic formula to compute the pixel location
Cubic Spline (2)	Uses cubic splines to compute the pixel location

- SubPixel Accuracy specifies the method used to perform the interpolation.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Interpolated array contains the resampled pixel data.
- Array Size specifies the size of the interpolated array.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

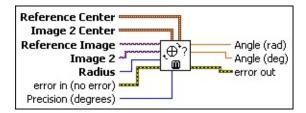
## **IMAQ Rotation Detect VI**

Owning Palette: CaliperInstalled With: NI Vision Development Module

Detects the rotational shift between two images, generally between a reference image, with the part being inspected at a known orientation, and another image that has the part in an unknown position. This VI extracts pixel values around a circular region in the reference image. These values then are compared to the same region in Image 2. The algorithm looks for the rotational shift between those two samples. To speed up the process, adjust the requested precision of the result.

#### **Examples**





- Reference Center contains coordinates of the center of a circular region in the reference image. Pixel values along this region in the reference image are used as features for computing the rotational shift.
- Image 2 Center contains coordinates of the center of a circular region in Image 2. The pixel values along this region in Image 2 are used to compute the rotational shift.
- Reference Image is the reference to the image containing the part to be inspected at a known position.
- Image 2 is the reference to the image that contains the part to be inspected at an unknown rotational shift.
- Radius is the radius of the circular regions in both images. Pixel values along this region in the two images are used to detect the rotation angle between the two images.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Precision (degrees) is the sampling period of the pixel values extracted from the circular region. The speed of this VI is affected directly by the sampling period. If the sampling period is high (the number of samples along the circular region are few), the processing speed of the VI increases at the cost of reduced accuracy in the computed rotational shift. In many cases, a precision higher than five degrees is not needed to position the regions of inspection of a part. The default is 5.
- Angle (rad) returns the angle in radians corresponding to the shift giving the best match between the two samples.
- Angle (deg) returns the angle in degrees corresponding to the shift giving the best match between the two samples.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

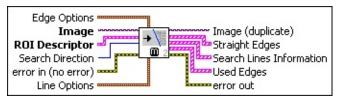
Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Rotating Part Example.llb

# **IMAQ Find Straight Edges 2 VI**

Owning Palette: <u>Caliper</u>Installed With: NI Vision Development Module Finds straight edges (lines) within a region in an image.





**Edge Options** specifies the parameters that are used to compute the edge gradient information and detect the edges.

**Edge Polarity** specifies the polarity of the edges to be found.

All Edges (0)	(Default) Searches for all edges
Rising Edges (1)	Searches for rising edges
Falling Edges (2)	Searches for falling edges

- **Kernel Size** specifies the size of the edge detection kernel. The default is 3.
- Width specifies the number of pixels averaged perpendicular to the search direction to compute the edge profile strength at each point along the search ROI. The default is 3.
- Minimum Edge Strength specifies the minimum edge strength (gradient magnitude) required for a detected edge. The default is 10.
- Interpolation Type specifies the interpolation method used to locate the edge position. Choose from the following options:

Zero Order (0)	Rounds to the nearest integral edge location
Bilinear (1)	Uses bilinear interpolation to compute the edge location
Bilinear Fixed (4)	(Default) Uses the fixed-point computation of bilinear interpolation to determine the edge location

**Data Processing Method** is the method used to process the data extracted for edge detection.

• , ,	(Default) Averages the data extracted for edge detection
` '	Takes the median of the data extracted for edge detection

- **Image** is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the rectangle, rotated rectangle, or other four-side rectangular polygon within which the edge detection is performed.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Search Direction specifies the direction in which the Region of Interest (ROI) is examined when straight edge detection is performed.

Left to Right (0)	(Default) Searches for edges from left to right
Right to Left (1)	Searches for edges from right to left
<b>Top to Bottom</b> (2)	Searches for edges from top to bottom
<b>Bottom to Top</b> (3)	Searches for edges from bottom to top

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to

display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Line Options** specifies the options that are used to detect the straight edges.
  - Number of Lines specifies the number of straight edges to find. The default is 1.
  - **Type** specifies the method used to find the straight edge. Choose from the following options:

First Edge Rake (0)	Fits a straight edge on the first points detected using a Rake
Best Edge Rake (1)	Fits a straight edge on the best points detected using a Rake
Hough Edge Rake (2)	Finds the strongest straight edge using all points detected on a Rake
_	Uses the location of the first projected edge as the straight edge
Best Edge Projection (4)	(Default) Finds the strongest projected edge location to determine the straight edge

- Min. Score specifies the minimum score of a detected straight edge. The default is 10.
- Max. Score specifies the maximum score of a detected straight edge. The default is 1000.

- **Orientation** specifies the angle at which the straight edge is expected to be found. The default is 0.
- Angle Range specifies the positive and negative range around the **Orientation** within which the straight edge is expected to be found. The default is 10.
- Angle Tolerance specifies the expected angular accuracy of the straight edge. The default is 1.
- Step Size specifies the gap in pixels between the search lines used with the Rake-based methods. The default is 3.
- Min. SNR (db) specifies the minimum signal to noise ratio (SNR) of the edge points used to fit the straight edge. The default is 0.
- Min. Points (%) specifies the minimum number for points as a percentage of the number of search lines that need to be included in the detected straight edge. The default is 25%.
- Hough Iterations specify the number of iterations used in the Hough-based method. The default is 5.
- Image (duplicate) is a reference to the source image.
- Straight Edges contains information about the detected straight edges.
  - **Point1 (Pixels)** is the start point of the detected straight edge in pixel units.
  - Point2 (Pixels) is the end point of the detected straight edge in pixel units.
  - **Point1 (Real-World)** is the start point of the detected straight edge in calibrated units.
  - Point2 (Real-World) is the end point of the detected straight edge in calibrated units.
  - Angle (Pixel) is the angle the detected straight edge makes with the axis perpendicular to the search direction.
  - Angle (Real-World) is the angle the detected straight edge makes with the axis perpendicular to the search direction in calibrated units.

- **Score** is the score of the detected straight edge.
- Straightness is the straightness value of the detected straight edge. Straightness is defined as the root mean squared error of the fitted line that represents the detected straight edge. A value of 0 indicates a perfectly straight line.
- Average SNR (dB) is the average signal-to-noise ratio along the detected straight edge.
- Calibration Valid indicates if the calibration information associated with the edges found along the search line is valid. If the calibration information is invalid for any pixel that is used for edge detection, the Calibration Valid Boolean is FALSE.
- Search Lines Information contains information about the search lines that are used to detected the straight edges.
  - **Start Point** is the start point of the search line in pixel units.
  - **End Point** is the end point of the search line in pixel units.
  - **Edges** contains information of all the edges detected along the search line.
    - **Location (Pixel)** is the location of the edge point in the image in pixel units.
    - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
    - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
    - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
    - Strength is the strength (gradient magnitude) at the located edge.
    - **Noise** is the strength of the noise associated with the current edge.

**Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.

- Gradient Values is an array that contains the calculated edge strengths along the search line.
- Calibration Valid indicates if the calibration information associated with the edges found along the search line is valid. If the calibration information is invalid for any pixel that is used for edge detection, the Calibration Valid Boolean is FALSE.
- **Used Edges** contains the edge points that were used to find each returned straight edge. The outer array corresponds to the number of straight edges that were found. The inner array contains the edges points that were used to find each straight edge.
  - **Edge Information** is an array that contains information about each detected edge.
    - **Location (Pixel)** is the location of the edge point in the image in pixel units.
    - Location (Real-World) is the location of the edge point in the image in calibrated units. When the image is not calibrated, this is the location is in pixel units.
    - **Distance (Pixel)** is the location of the edge from the first point of the search line along the boundary of the input ROI in pixels.
    - Distance (Real-World) is the location of the edge from the first point of the search line along the boundary of the input ROI in calibrated units. When the image is not calibrated, this distance is in pixel units.
    - Strength is the strength (gradient magnitude) at the located edge.
    - Noise is the strength of the noise associated with the current edge.
    - **Rising?** indicates the polarity of the edge. If TRUE, the edge is a Rising Edge.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

# **Analytic Geometry**

Owning Palette: Machine VisionInstalled With: NI Vision Development Module

Use Analytic Geometry VIs to derive results from the coordinates of points returned by image analysis and machine vision algorithms. Use these VIs to fit lines, circles, and ellipses to a set of points in the image; compute the area of a polygon represented by a set of points; measure distances between points; and find angles between lines represented by points.

Palette Object	Description
IMAQ Point Distances	Computes the distance between consecutive pairs of points.
IMAQ Get Angles	Computes the angles formed by sets of four points in an image or between sets of two points and a common vertex.
IMAQ Fit Line	Finds the line of such quality that best represents a set of points. The resulting line may only take into account a subset of the input points.
IMAQ Lines Intersection	Computes the intersection point and the angle between two lines. Each line is specified by any two points that lie on the line.
IMAQ Perpendicular Line	Computes a perpendicular line that passes through a point and is perpendicular to a reference line. It also returns the distance from the point to the line.
IMAQ Bisecting Line	Computes a line that bisects the two lines specified by Line 1 and Line 2.
IMAQ Mid Line	Computes the mid line between a point and a reference line. The mid line is the line that is parallel to the reference line and lies midway between the point and the reference line.
IMAQ Polygon Area	Computes the area of a polygon described by the coordinates of its vertices.
IMAQ Fit Circle 2	Finds the circle that best represents a set of points

	and returns the radius, perimeter, and area of the circle.
IMAQ Fit Ellipse 2	Finds the ellipse that best represents a set of points and returns the center, major and minor axes, perimeter, and area of the ellipse. This VI requires a minimum of five or six points, depending on the value of Reject Outliers. The resulting ellipse may take into account only a subset of the points you provide.
IMAQ GetPointsOnLine	Given the endpoints of a line, returns an array of all the points comprising the line.
IMAQ GetPointsOnContour	Finds the number of edge segments in an image and returns the coordinates of the pixels in each segment. Any pixel that is greater than zero is considered an edge location.
IMAQ Build CoordSys (Points)	Builds a reference for any arbitrary coordinate system with respect to the image plane.

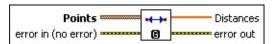
## **IMAQ Point Distances VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Computes the distance between consecutive pairs of points.

#### **Examples**



- **Points** is an array of point-coordinate clusters. This array must contain two or more point-coordinate clusters corresponding to two or more points on the line.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Distances** is an array containing the computed distances. **Distance** i is computed between the points i and (i + 1).
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

# **Examples**

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\Circle Distance Example.vi

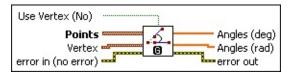
## **IMAQ Get Angles VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

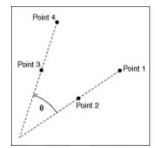
**Development Module** 

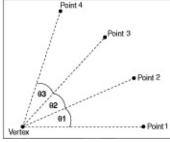
Computes the angles formed by sets of four points in an image or between sets of two points and a common vertex.

#### **Examples**



- **Use Vertex (No)** specifies whether a vertex point is used while computing the angle.
- Points is an array of point-coordinate clusters. If **Use Vertex** is **No**, four points at a time are considered when computing the angle, as shown in the leftmost figure. If **Use Vertex** is **Yes**, the angle made by two consecutive points in the array and the vertex is computed, as shown in the rightmost figure.





- **Vertex** specifies the vertex point to use if the **Use Vertex** option is TRUE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Angles (deg) is an array containing the computed angles in degrees.
- Angles (rad) is an array containing the computed angles in radians.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Examples**

Refer to the following for an example that uses this VI.

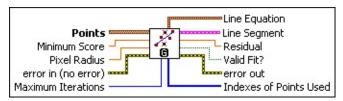
• LabVIEW\examples\Vision\3. Applications\Gauging Example.llb

## **IMAQ Fit Line VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Finds the line of such quality that best represents a set of points. The resulting line may only take into account a subset of the input points.



- **Points** is an array of point-coordinate clusters. This array must contain two or more point-coordinate clusters corresponding to two or more points on the line.
- Minimum Score specifies the required quality of the fitted line. The score can vary from 0 to 1000, where 1000 represents a perfect line. The default is 900.
- Pixel Radius specifies the neighborhood pixel relationship for the initial subset of points being used.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Maximum Iterations specifies the maximum number of refining steps the algorithm takes to ensure that the quality of the fit is better than Minimum Score.
- **Line Equation** is a cluster of three coefficients A, B, and C of the equation in the normal form (Ax + By + C=0) of the best fit line.
- **Line Segment** represents the intersection of the line equation and the bounding rectangle of the input points.
- **Residual** represents the least-square error of the fitted line to the entire set of points.

Residual =  $\sqrt{\sum_{i} e_{i}^{2}}$  where  $e_{i}$  is the distance from the  $i^{th}$  input point to the fit line.

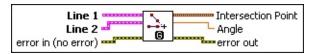
- **Valid Fit?** specifies whether the **Minimum Score** was attained within the given **Maximum Iterations**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Indexes of Points Used is an array that returns the indexes of points in the Points array that were used to fit the line. Some points may be excluded based on the Minimum Score and Pixel Radius specified.

#### **IMAQ Lines Intersection VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Computes the intersection point and the angle between two lines. Each line is specified by any two points that lie on the line.



- Line 1 is a cluster of two point coordinates defining the first line.
- **Line 2** is a cluster of two point coordinates defining the second line.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Intersection Point is a point-coordinate cluster indicating the coordinates of the intersection point of Line 1 and Line 2.
- Angle is the angle in degrees between Line 1 and Line 2. The angle reference (for example, 0 degrees) is the horizontal, right axis. The angle is computed from Line 1 to Line 2 in a

counterclockwise direction.

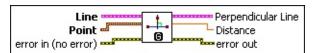
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Perpendicular Line VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Computes a perpendicular line that passes through a point and is perpendicular to a reference line. It also returns the distance from the point to the line.



- **Line** is a cluster of two point coordinates defining the reference line.
- **Point** is a cluster that specifies the coordinates of a point.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Perpendicular Line is a line perpendicular to the reference Line and passing through the Point. A cluster of two point coordinates defines the Perpendicular Line. The first point of this cluster is the input Point. The second point lies on the Line.



**Note** If **Point** lies on **Line**, the first point of **Perpendicular Line** is not the same as **Point**. The second point of **Perpendicular Line** is **Point**, and the first point lies on the line perpendicular to **Line**.

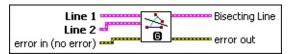
- **Distance** is the shortest (Euclidean) distance from the **Point** from the **Line**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Bisecting Line VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Computes a line that bisects the two lines specified by Line 1 and Line 2.



- **Line 1** is a cluster of two point coordinates defining the first line.
- Line 2 is a cluster of two point coordinates defining the second line.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Bisecting Line** is a cluster of two point coordinates indicating the coordinates of the line bisecting **Line 1** and **Line 2**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

the front panel and select **Explain Error** from the shortcut menu for more information about the error.

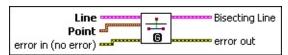
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning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

#### **IMAQ Mid Line VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Computes the mid line between a point and a reference line. The mid line is the line that is parallel to the reference line and lies midway between the point and the reference line.



- **Line** is a cluster of two point coordinates defining the reference line.
- **Point** is a cluster that specifies the coordinates of a point.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Bisecting Line** is a cluster of two point coordinates indicating the coordinates of the mid line between the **Point** and the **Line**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

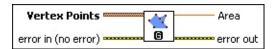
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning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

### **IMAQ Polygon Area VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Computes the area of a polygon described by the coordinates of its vertices.



- Vertex Points is an array of point-coordinate clusters. This array must contain three or more point-coordinate clusters corresponding to three or more points located on the vertices of the polygon.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Area is the area of the polygon defined by the array of vertex points.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the

same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

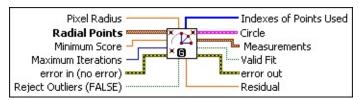
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning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

## **IMAQ Fit Circle 2 VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Finds the circle that best represents a set of points and returns the radius, perimeter, and area of the circle. This VI requires a minimum of three points. The resulting circle may take into account only a subset of the points you provide.



- **Pixel Radius** specifies the acceptable distance, in pixels, that a point determined to belong to the circle can be from the perimeter of the circle.
- Radial Points is an array of point-coordinate clusters that correspond to points on the circumference of the circle. This array must contain three or more point-coordinate clusters.
- Minimum Score specifies the required quality of the fitted circle. The score can vary from 0 to 1000, where 1000 represents a perfect circle. The default is 500.
- Maximum Iterations specifies the maximum number of refining steps the algorithm takes to ensure that the quality of the fit is better than Minimum Score. Maximum Iterations is valid only when Reject Outliers is TRUE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Reject Outliers (FALSE) controls whether to use every given radial point or only a subset of the points to fit the circle. If this control is TRUE, the algorithm determines the best subset of points to use and ignores the outliers (the points outside the subset). If this control is FALSE, the algorithm uses every given point.
- Indexes of Points Used is an array that returns the indexes of points in the Radial Points array that were used to fit the circle. Some points may be excluded based on the Minimum Score, Pixel Radius, or Reject Outliers specified.
- **Circle** is a cluster with the following members:
  - **Center** is the coordinates of the center of the circle.
  - Radius is the radius of the circle.
- **Measurements** is a cluster with the following members:
  - Area is the area of the circle.
  - **Perimeter** is the perimeter of the circle.
- Valid Fit indicates whether the fit is valid. This indicator is TRUE if the quality of the fit is better than Minimum Score and the fit was attained within Maximum Iterations. Valid Fit is a function of Minimum Score, Maximum Iterations, and Pixel Radius.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Residual** indicates the least square error of the fitted circle over every given radial point.

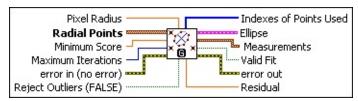
Residual =  $\sqrt{\sum_{i} e_{i}^{2}}$  where  $e_{i}$  represents the distance from the i th input point to the perimeter of the circle.

## **IMAQ Fit Ellipse 2 VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Finds the ellipse that best represents a set of points and returns the center, major and minor axes, perimeter, and area of the ellipse. This VI requires a minimum of five or six points, depending on the value of Reject Outliers. The resulting ellipse may take into account only a subset of the points you provide.



- **Pixel Radius** specifies the acceptable distance, in pixels, that a point determined to belong to the ellipse can be from the perimeter of the ellipse.
- Radial Points is an array of point-coordinate clusters that correspond to points along the perimeter of the ellipse. The minimum number of points needed along the perimeter of the ellipse is 5 points if **Reject Outliers** is TRUE, or 6 points if **Reject Outliers** is FALSE.
- Minimum Score specifies the required quality of the fitted ellipse. The score can vary from 0 to 1000, where 1000 represents a perfect ellipse. The default is 500.
- Maximum Iterations specifies the maximum number of refining steps the algorithm takes to ensure that the quality of the fit is better than Minimum Score. Maximum Iterations is valid only when Reject Outliers is TRUE.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out

to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Reject Outliers (FALSE) controls whether to use every given radial point or only a subset of the points to fit the ellipse. If this control is TRUE, the algorithm determines the best subset of points to use and ignores the outliers (the points outside the subset). If this control is FALSE, the algorithm uses every given point.
- Indexes of Points Used is an array that returns the indexes of points in the Radial Points array that were used to fit the ellipse. Some points may be excluded based on the Minimum Score, Pixel Radius, or Reject Outliers specified.
- **Ellipse** is a cluster with the following members:
  - **Center** is the coordinates of the center of the ellipse.
  - Major Axis is a cluster of two point-coordinate clusters. This cluster specifies the coordinates of the points that compose the major axis of the ellipse.
  - Minor Axis is a cluster of two point-coordinate clusters. This cluster specifies the coordinates of the points that compose the minor axis of the ellipse.
- **Measurements** is a cluster with the following members:
  - Area is the area of the ellipse.
  - Perimeter is the length of the perimeter of the fit ellipse.
- **Valid Fit** indicates whether the fit is valid. This indicator is TRUE if the quality of the fit is better than **Minimum Score** and the fit was

attained within **Maximum Iterations**. **Valid Fit** is a function of **Minimum Score**, **Maximum Iterations**, and **Pixel Radius**.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Residual** indicates the least square error of the fitted ellipse over every given radial point.

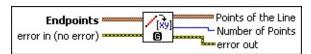
Residual =  $\sqrt{\sum_{i} e_i^2}$  where  $e_i$  is the distance from the  $i^{th}$  input point to the fit line.

#### IMAQ GetPointsOnLine VI

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Given the endpoints of a line, returns an array of all the points comprising the line.



- **Endpoints** specifies the points at each end of the line.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Points of the Line** returns an array of clusters, where each point of the line is represented by its x- and y-coordinates.
- Number of Points returns the number of points found on the line.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

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

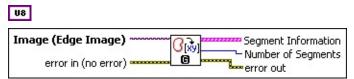
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### IMAQ GetPointsOnContour VI

Owning Palette: Analytic GeometryInstalled With: NI Vision

**Development Module** 

Finds the number of edge segments in an image and returns the coordinates of the pixels in each segment. Any pixel that is greater than zero is considered an edge location. This VI joins adjoining edge pixels into edge segments. An edge segment is considered closed if it forms a loop. Each edge segment is given a weight based on the pixel gray values along that edge. An edge segment with high gray values has a higher weight.



- Image (Edge Image) is the reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **Segment Information** is an array of clusters containing the following information:
- Number of Points is the number of points that make up the edge segment.
- Open/Closed indicates whether the edge segment is open or closed.
- **Weight** indicates the significance of the edge in terms of the gray values that constitute the edge.
- Contour Points is an array of clusters, where each point on the edge is represented by the following coordinates:
  - X returns the x-coordinate value of the edge point in the image.
  - Y returns the y-coordinate value of the edge point in the image.
  - Curvature returns the change in slope at this edge point of the segment.
  - X Displacement returns the x displacement of the current edge pixel from a cubic spline fit of the current edge segment.
  - Y Displacement returns the y displacement of the current edge pixel from a cubic spline fit of the current edge segment.
- Number of Segments returns the number of edge segments in the image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a

warning code.

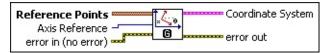
**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Build CoordSys (Points) VI**

Owning Palette: Analytic GeometryInstalled With: NI Vision

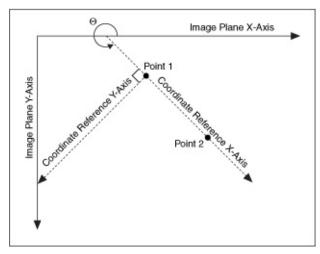
**Development Module** 

Builds a reference for any arbitrary coordinate system with respect to the image plane. The reference of the coordinate system is specified as the position of the origin of the coordinate system, the orientation of its x-axis with respect to that of the image plane, and the direction of the axis.

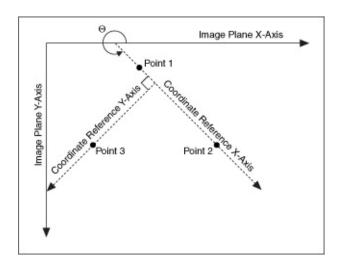


Reference Points is an array of point clusters that define the coordinate system.

If two points are specified, these points are assumed to lie along the x-axis of the coordinate system and the first point is used as the origin of the coordinate axis, as shown in the following figure.



If three points are specified, the first two points are assumed to be along the x-axis, and the third point is assumed to be on the y-axis of the coordinate system, as shown in the following figure.



- Axis Reference specifies the direction of the coordinate system.

  The Axis Reference can be Direct or Indirect. Refer to Axis Type for more information.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Coordinate System specifies the coordinate system defined by Reference Points and the Axis Reference.

- Origin is a point cluster that specifies the origin of the base reference coordinate system.
- Angle (deg) is the angle the base reference coordinate system makes with the image coordinate system.
- Axis Reference is defined as **Direct** or **Indirect**. Refer to Axis Type for more information.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### Classification

Owning Palette: Machine VisionInstalled With: NI Vision Development Module

Use Classification VIs to identify unknown objects by comparing significant features to a set of features that conceptually represent classes of known objects.

Palette Object	Description
IMAQ Read Classifier File	Reads a classifier session from the file specified by <b>File Path</b> .
IMAQ Classify	Classifies the image sample located in the given ROI.
IMAQ Dispose Classifier	Destroys a classifier session and frees the space it occupied in memory. You must call IMAQ Dispose Classifier when the application no longer needs the session. This VI is required for each classifier session created.
IMAQ Add Classifier Sample	Assigns a new image sample to the specified class in <b>Classifier Session</b> . The new sample is based on an ROI in the image. The sample is assigned to a specific class.
IMAQ Get Classifier Sample Info	Retrieves the thumbnail image and class label of the sample at the specified index.
IMAQ Change Sample Class	Labels the sample at the index specified by <b>Sample Index</b> .
IMAQ Delete Classifier Sample	Deletes the sample at the index specified by <b>Sample Index</b> .

Writes a classifier session to the file specified in <b>File Path</b> . This VI saves the exact state of the classifier session.
Provides information about the accuracy and predictive value of the trained classifier.

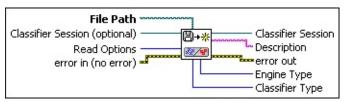
Subpalette	Description
<u>Classifier</u> <u>Types</u>	Use the Classifier Types VIs to create specific types of classifiers and perform classifier-specific options.
<u>Classifier</u> <u>Engines</u>	Use the Classifier Engines VIs to train and configure classifier sessions.

## **IMAQ Read Classifier File VI**

Owning Palette: Classification Installed With: NI Vision Development

Module

Reads a classifier session from the file specified by **File Path**.



- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- Classifier Session (optional) is the reference to the classifier session on which this VI operates. This input is optional only when **Read Options** is set to **Read All**.
- Read Options specifies what to read from Classifier Session.

	T
Read All (0)	Restores the session to its state when it was saved.
Add	Keeps the properties of the passed-in session, and
Samples,	appends samples of the file to the current session. If
Do Not	Classifier Session (optional) is not connected, a
Change	new session of the same type as the session read
Options (1)	from file is connected.
Read	Copies the properties of the saved session without
Options,	changing the samples of the passed-in session . If
Do Not Add	Classifier Session (optional) is not connected, a
Samples (2)	new session of the same type as the session read
	from file is connected.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

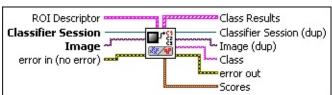
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Classifier Session is the classifier session refnum.
- **Description** is a user-provided description of the classifier session.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Engine Type specifies the type of classification engine associated with Classifier Session.
- Classifier Type specifies the type of classifier session read from file.

# **IMAQ Classify VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Classifies the image sample located in the given ROI.





ROI Descriptor is the descriptor of the region of interest specifying the location of the sample in the image. The ROI must be one or more closed contours. If ROI Descriptor is empty or not connected, the entire image is considered to be the region.



- **Tip** For best performance, use only one rectangle or one rotated rectangle per sample.
- Global Rectangle contains the coordinates of the bounding rectangle.
- Contours are each of the individual shapes that define an ROI.
  - **ID** refers to whether the contour is the external or internal edge of an ROI.
  - **Type** is the shape type of the contour.
  - Coordinates indicates the relative position of the contour.
- Classifier Session is the reference to the classifier session on which this VI operates.
- Image is a reference to the source image.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

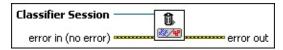
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Class Results is an array with one element for every class in the classifier session.
  - Class is one of the classes in Classifier Session.
  - Distance is the distance from the closest sample in Class to the input sample when performing Nearest Neighbor and K-Nearest Neighbor classification. Distance is the distance between the input sample and the center of each class when performing Minimum Mean Distance classification.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Image (dup) is a reference to the connected input Image.
- Class is the class into which the classifier session categorizes the input sample.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Scores returns estimations of how well the classifier session classified the input. The score can vary from 0 to 1000, where 1000 represents the best possible score.
  - Classification Score indicates how much better the assigned class represents the input sample than other classes represent the input.
  - Identification Score indicates the similarity of the input and the assigned class. Use Identification Score only when you cannot reach a decision about the class of a sample using Classification Score alone.

### **IMAQ Dispose Classifier VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Destroys a classifier session and frees the space it occupied in memory. You must call IMAQ Dispose Classifier when the application no longer needs the session. This VI is required for each classifier session created.



- Classifier Session is the reference to the classifier session on which this VI operates.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for

more information about the error.

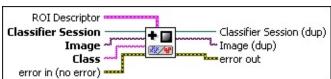
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Add Classifier Sample VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Assigns a new image sample to the specified class in **Classifier Session**. The new sample is based on an ROI in the image. The sample is assigned to a specific class.





ROI Descriptor is the descriptor of the region of interest specifying the location of the sample in the image. The ROI must be one or more closed contours. If ROI Descriptor is empty or not connected, the entire image is considered to be the region.



- **Tip** For best performance, use only one rectangle or one rotated rectangle per sample.
- Global Rectangle contains the coordinates of the bounding rectangle.
- **Contours** are each of the individual shapes that define an ROI.
  - **ID** refers to whether the contour is the external or internal edge of an ROI.
  - **Type** is the shape type of the contour.
  - **Coordinates** indicates the relative position of the contour.
- Classifier Session is the reference to the classifier session on which this VI operates.
- Image is a reference to the source image.
- Class is the class to which you want to assign the sample.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to

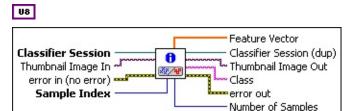
**error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Image (dup) is a reference to the connected input Image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Get Classifier Sample Info VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Retrieves the thumbnail image and class label of the sample at the specified index.



- Classifier Session is the reference to the classifier session on which this VI operates.
- Thumbnail Image In is the reference to the image that will store the thumbnail image of the sample specified by **Sample Index**.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

**Sample Index** is the index of a sample in the classifier session. The samples are indexed in the order in which they were added to the session, starting with 0.



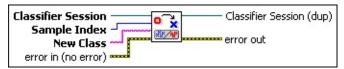
Note Set this parameter to -1 to get only the **Number of Samples** value.

- Feature Vector is the feature vector entered in the <a href="MAQ Add">IMAQ Add</a>
  <a href="Custom Sample">Custom Sample</a> VI.</a>
  - Note Note If the session referenced by Classifier Session is not a custom session, Feature Vector is empty.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Thumbnail Image Out is the reference to the image that stores the thumbnail image of the sample specified by Sample Index.
- Class is the class to which the sample specified by Sample Index belongs.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Number of Samples is the total number of samples in the classifier session.

## **IMAQ Change Sample Class VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Labels the sample at the index specified by **Sample Index**.



- Classifier Session is the reference to the classifier session on which this VI operates.
- Sample Index is the index of a sample in the classifier session. The samples are indexed in the order in which they were added to the session, starting with 0.
- New Class is the class to which you want to assign the sample referenced by Sample Index.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

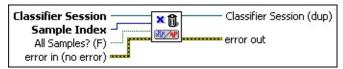
**Classifier Session (dup)** is a reference to the session referenced by **Classifier Session**.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Delete Classifier Sample VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Deletes the sample at the index specified by **Sample Index**.



- Classifier Session is the reference to the classifier session on which this VI operates.
- Sample Index is the index of a sample in the classifier session. The samples are indexed in the order in which they were added to the session, starting with 0.
- All Samples? (F) specifies whether the VI deletes all samples. The default value, FALSE, deletes only the sample at the index specified by Sample Index. Set this value to TRUE to delete all samples in Classifier Session.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

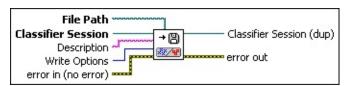
error or warning. The default is an empty string.

- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Write Classifier File VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Writes a classifier session to the file specified in **File Path**. This VI saves the exact state of the classifier session.



- **File Path** is the complete pathname, including drive, directory, and filename, of the file to read.
- Classifier Session is the reference to the classifier session on which this VI operates.
- **Description** is a user-provided description of the classifier session.
- Write Options specifies how to save the classifier session to disk.

Full (Editable) (0)	Creates a classifier session file that saves the exact state of the classifier session. When the IMAQ Read Classifier File VI reads a file created with this write option, an exact copy of the classifier session is created. This option is the default.
Compact (Not Editable) (1)	Creates a smaller classifier session file that contains only enough information to call the <a href="IMAQ Classify">IMAQ Classify</a> VI. Files created with this write option cannot be modified.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

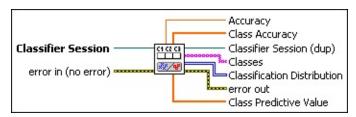
status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Classifier Accuracy VI**

Owning Palette: Classification Installed With: NI Vision Development Module

Provides information about the accuracy and predictive value of the trained classifier.



- Classifier Session is the reference to the classifier session on which this VI operates.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Accuracy is the proportion of the samples in Classifier Session that are properly classified by Classifier Session.
- Class Accuracy is the proportion of samples correctly classified as

- a given class to all samples classified as a given class for each class in the order given in **Classes**.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Classes is the ordered list of each class in Classifier Session.
- Classification Distribution is a table showing the results of the classifier session classifying every sample in Classifier Session. The first dimension corresponds to the classes assigned to samples in Classifier Session. The second dimension corresponds to the classes into which samples are classified. Each value is the total number of samples belonging to the class corresponding to its first dimension that were classified as the class corresponding to its second dimension. The number of correctly classified samples for Class *X* is shown in the table at the intersection of the Class *X* first dimension and Class *X* second dimension (along a diagonal). All other table values indicate the number of incorrectly classified samples.

Refer to Chapter 15, *Binary Particle Classification*, of the *NI Vision Concepts Manual* for more information about the classification distribution table, class predictive value, and class accuracy.

- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Class Predictive Value is the proportion of samples correctly

classified as a given class to all samples in **Classifier Session** of a given class for each class in the order given in **Classes**.

# **Classifier Types**

Owning Palette: ClassificationInstalled With: NI Vision Development

Module

Use the Classifier Types VIs to create specific types of classifiers and perform classifier-specific options.

Palette Object	Description
IMAQ Create Particle Classifier	Creates a particle classifier session.
IMAQ Particle Classifier Options	Configures the particle classifier options for the classifier session.
IMAQ Create Custom Classifier	Creates a custom classifier session.
IMAQ Add Custom Sample	Assigns a class to the custom feature vector given in <b>Feature Vector</b> and adds it to the classifier session.
IMAQ Classify Custom	Classifies the <b>Feature Vector</b> .

## **IMAQ Create Particle Classifier VI**

Owning Palette: Classifier Types Installed With: NI Vision Development Module

Creates a particle classifier session.



**Note** You must call the <u>IMAQ Dispose Classifier</u> VI when the application no longer needs this session.



- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Classifier Session is the reference to the classifier session that this VI creates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on

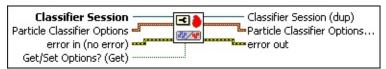
the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Particle Classifier Options VI**

Owning Palette: <u>Classifier Types</u>Installed With: NI Vision Development Module

Configures the particle classifier options for the classifier session.



- Classifier Session is the reference to the classifier session on which this VI operates.
- Particle Classifier Options define the dependence of the particle classifier on shape, scale, and mirror symmetry. By default, when Scale Dependence and Mirror Dependence equal 0, the classifier depends only on variations in shape to classify particles. When Scale Dependence and Mirror Dependence have non-zero values, the dependence on shape is calculated as follows: Shape Dependence = 1000 (Scale Dependence + Mirror Dependence).
  - Scale Dependence determines the relative importance (between 0 and 1000) of scale when classifying particles. If the value is 0, the samples are classified independent of scale.
  - Mirror Dependence determines the relative importance (between 0 and 1000) of mirror symmetry when classifying particles. An example of objects exhibiting mirror symmetry is a lowercase letter *p* and a lowercase letter *q*. If the value is 0, the samples are classified independent of mirror symmetry.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set Options? (Get) specifies whether the VI gets or sets the property value.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Particle Classifier Options out define the dependence of the particle classifier on shape, scale, and mirror symmetry. By default, when Scale Dependence and Mirror Dependence equal 0, the classifier depends only on variations in shape to classify particles. When Scale Dependence and Mirror Dependence have non-zero values, the dependence on shape is calculated as follows: Shape Dependence = 1000 (Scale Dependence + Mirror Dependence).
  - Scale Dependence determines the relative importance (between 0 and 1000) of scale when classifying particles. If the value is 0, the samples are classified independent of scale.
  - Mirror Dependence determines the relative importance (between 0 and 1000) of mirror symmetry when classifying particles. An example of objects exhibiting mirror symmetry is a lowercase letter *p* and a lowercase letter *q*. If the value is 0, the samples are classified independent of mirror symmetry.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that

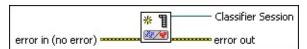
this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Create Custom Classifier VI**

Owning Palette: Classifier Types Installed With: NI Vision Development Module

Creates a custom classifier session. You must call the <u>IMAQ Dispose</u> <u>Classifier</u> VI when the application no longer needs this session.



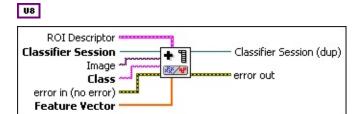
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Classifier Session is the reference to the classifier session that this VI creates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

# **IMAQ Add Custom Sample VI**

Owning Palette: Classifier Types Installed With: NI Vision Development Module

Assigns a class to the custom feature vector given in **Feature Vector** and adds it to the classifier session.



- ROI Descriptor is the descriptor that defines the rectangle specifying the location of the sample in the image. If ROI Descriptor is empty or not connected, the entire image is considered to be the region.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Classifier Session is the reference to the classifier session on which this VI operates.
- Image is a reference to the image to be used as the thumbnail of the sample.
- Class is the class to which you want to assign the sample.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or

function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

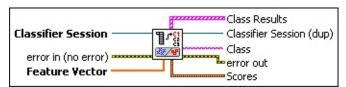
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Feature Vector is a user-defined vector that uniquely describes the sample. All feature vectors added to a custom session must have the same dimensions.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Classify Custom VI**

Owning Palette: Classifier Types Installed With: NI Vision Development

Module

Classifies the Feature Vector.



- Classifier Session is the reference to the classifier session on which this VI operates.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Feature Vector** is a user-defined vector that uniquely describes the sample. **Feature Vector** must have the same dimensions as the feature vectors in the classifier session.
- Class Results is an array with one element for every class in the classifier session.

- Class is one of the classes in Classifier Session.
- Distance is the distance from the closest sample in Class to the input sample when performing Nearest Neighbor and K-Nearest Neighbor classification. Distance is the distance between the input sample and the center of each class when performing Minimum Mean Distance classification.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Class is the class into which the classifier session categorizes the input sample.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Scores returns estimations of how well the classifier session classified the input. The score can vary from 0 to 1000, where 1000 represents the best possible score.
  - Classification Score indicates how much better the assigned class represents the input sample than other classes represent the input.
  - Identification Score indicates the similarity of the input and the assigned class. Use Identification Score only when you cannot reach a decision about the class of a sample using Classification Score alone.

# **Classifier Engines**

Owning Palette: Classification Installed With: NI Vision Development

Module

Use the Classifier Engines VIs to train and configure classifier sessions.

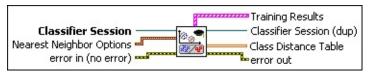
Palette Object	Description
IMAQ Train Nearest Neighbor	Sets the classifier session to use the Nearest Neighbor Classifier engine, and configures the Nearest Neighbor parameters it will use.
IMAQ Get Nearest Neighbor options	Gets the <b>Nearest Neighbor options</b> for <b>Classifier Session</b> .

# **IMAQ Train Nearest Neighbor VI**

Owning Palette: Classifier Engines Installed With: NI Vision

**Development Module** 

Sets the classifier session to use the Nearest Neighbor Classifier engine, and configures the Nearest Neighbor parameters it will use.



- Classifier Session Specifies the classifier session this VI operates on.
- Nearest Neighbor Options are the options to use to train the Nearest Neighbor.
  - **Method** is the nearest neighbor classification method used. The following options are valid:

9	ı
Nearest Neighbor (0)	This is the most direct approach to classification. In Nearest Neighbor classification, the distance of an input sample of unknown class to another class is defined as the distance to the closest samples that are used to represent that class.
K Nearest Neighbor (1)	This is more robust to noise compared with nearest neighbor classification. In K-Nearest Neighbor classification, an input feature vector is classified into a class based on a voting mechanism. The NI Classifier finds $K$ nearest samples from all the classes. The input feature vector of unknown class is assigned to the class with majority of the votes in the $K$ nearest samples.
Minimum Mean Distance (2)	This is most effective in applications that have little or no feature pattern visibility or other corruptive influences. In Minimum Mean Distance classification, an input feature vector of unknown class is classified based on its distance to each class center.

Metric is the distance metric used. The following options are valid:

Maximum (0)	This is the metric most sensitive to small variations between samples. Use <b>Maximum</b> when you need to classify samples with very small differences into different classes.
Sum (1)	(Also known as the Manhattan metric or Taxicab metric) This is the metric used in most classification applications. This is the default value.
Euclidean (2)	This is the metric least sensitive to small variations between samples. Use <b>Euclidean</b> when you need to classify samples with small differences into the same class.

- **k** is the *k* value used if **Method** is set to **K Nearest Neighbor**. If **Method** is not set to **K Nearest Neighbor**, this value is ignored.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in

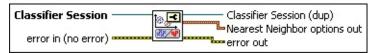
- most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Training Results** is an array of statistical information for each class in the classifier session.
  - Class is the string that the VI read.
  - Standard Deviation is the standard deviation from the mean of all samples in Class.
  - Number of Samples is the number of samples in Class.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Class Distance Table is a table giving the mean distance from each class to each other class. The classes are ordered in each dimension from (0,0) in the order listed in **Training Results**.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **IMAQ Get Nearest Neighbor options VI**

Owning Palette: Classifier Engines Installed With: NI Vision

**Development Module** 

Gets the Nearest Neighbor options for Classifier Session.



- Classifier Session is the reference to the classifier session on which this VI operates.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Classifier Session (dup) is a reference to the session referenced by Classifier Session.
- Nearest Neighbor options out are the options used in the <a href="MAQ">IMAQ</a>
  <a href="IMAQ">Train Nearest Neighbor</a> VI.</a>



**Note** Refer to the *NI Vision Concepts Manual* for more information about these options.

Method is the nearest neighbor classification method used. The following options are valid:

Nearest Neighbor (0)	This is the most direct approach to classification. In Nearest Neighbor classification, the distance of an input sample of unknown class to another class is defined as the distance to the closest samples that are used to represent that class.
K Nearest Neighbor (1)	This is more robust to noise compared with nearest neighbor classification. In K-Nearest Neighbor classification, an input feature vector is classified into a class based on a voting mechanism. The NI Classifier finds $K$ nearest samples from all the classes. The input feature vector of unknown class is assigned to the class with majority of the votes in the $K$ nearest samples.
Minimum Mean Distance (2)	This is most effective in applications that have little or no feature pattern visibility or other corruptive influences. In Minimum Mean Distance classification, an input feature vector of unknown class is classified based on its distance to each class center.

Metric is the distance metric used. The following options are valid:

Maximum (0)	This is the metric most sensitive to small variations between samples. Use <b>Maximum</b> when you need to classify samples with very small differences into different classes.
Sum (1)	(Also known as the Manhattan metric or Taxicab metric) This is the metric used in most classification applications. This is the default value.
Euclidean (2)	This is the metric least sensitive to small variations between samples. Use <b>Euclidean</b>

when you need t	o classify samples with small
differences into t	he same class.

- **k** is the *k* value used if **Method** is set to **K Nearest Neighbor**. If **Method** is not set to **K Nearest Neighbor**, this value is ignored.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **OCR**

Owning Palette: <u>Machine Vision</u>Installed With: NI Vision Development Module

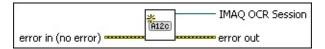
Use OCR VIs to develop optical character recognition (OCR) applications. OCR is the process by which the machine vision software reads text and/or characters in an image.

Palette Object	Description
IMAQ OCR Create Session	Creates an OCR session and returns a refnum associated with the session.
IMAQ OCR Dispose Session	Disposes of the OCR session and frees the resources associated with this session.
IMAQ OCR Property	Gets and sets the value of the property specified by <b>Property</b> . You can use this polymorphic VI to set the value of an I32 property, a string property, or a Boolean property. The data type you wire to the <b>Property Value</b> input determines which polymorphic instance to use.
IMAQ OCR Threshold Data	Gets and sets the threshold data.
IMAQ OCR Train	Assigns character values to the objects that NI Vision identifies in the image. The newly trained characters are appended to the existing trained character set. An image can contain no more than 255 objects.
IMAQ OCR Character Info	Retrieves information about the trained character at the specified index.
IMAQ OCR	Renames the trained character at the index specified by Character Index.

Rename Character	
IMAQ OCR Delete Character	Deletes a character from the trained character set at the index specified by <b>Character Index</b> .
IMAQ OCR Set Valid Characters	Specifies the characters that are valid for each character position in the image.
IMAQ OCR Read Character Set File	Reads a character set and session properties from the character set file specified by <b>File Path</b> .
IMAQ OCR Write Character Set File	Stores the trained character set and properties of the session in the file specified by File Path.
IMAQ OCR Verify Text	Verifies the accuracy of the text in the image.
IMAQ OCR Set Reference Character	Sets a character as the reference character for the character class. If the character class already has a reference character, the new character will replace the old character as the reference character.

## **IMAQ OCR Create Session VI**

**Owning Palette:** OCR Installed With: NI Vision Development Module Creates an OCR session and returns a refnum associated with the session.

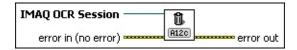


- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- IMAQ OCR Session is a refnum that uniquely identifies the OCR session. Use this value to refer to this OCR session in subsequent VI calls.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

# **IMAQ OCR Dispose Session VI**

**Owning Palette:** OCRInstalled With: NI Vision Development Module Disposes of the OCR session and frees the resources associated with this session.



- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.

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

# **IMAQ OCR Property VI**

Owning Palette: OCRInstalled With: NI Vision Development Module

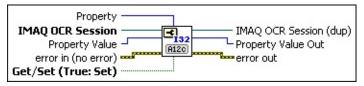
Gets and sets the value of the property specified by **Property**. You can use this polymorphic VI to set the value of an I32 property, a string property, or a Boolean property. The data type you wire to the **Property Value** input determines which polymorphic instance to use.

Use the pull-down menu to select an instance of this VI.

Select an instance	_
--------------------	---

#### **IMAQ OCR I32 Property**

Gets and sets the value of the property specified by Property. This VI works only with I32 properties.



**Property** specifies the property that this VI gets or sets. The following values are valid:

Acceptance Level (0)	The minimum acceptance level at which an object is considered a trained character. The range of valid values for this property is 0 to 1000. The default value is 700.
Min Char Size (1)	The minimum object size, in pixels, at which an object is a potential identifiable character. The minimum value allowed for this property is 1. The value of this property must not exceed <b>Max Char Size</b> . The default is 20 pixels.
Max Char Size (2)	The maximum object size, in pixels, at which an object is a potential identifiable character. Set this value to 65536, which is also the default value, to specify that all values greater than <b>Min Char Size</b> are acceptable. The value of this property must not be less than <b>Min Char Size</b> .
Min Char Spacing (3)	The minimum amount of space, in pixels, that can be between characters. This value must not be less than the <b>Max Horiz Element Spacing</b> . The default value is 1.
Max Horiz Element Spacing (4)	The maximum horizontal spacing, in pixels, between the elements that are combined to form a single character. For stroke characters, set this value to 1 or 2. For segment characters or pin stamp characters, set this value to 5 or higher. This value cannot exceed the actual character spacing value specified by the <b>Min Char</b>

	<b>Spacing</b> property. The horizontal spacing value is required to ensure that OCR correctly trains and reads characters that consist of several elements. The minimum value allowed for this property is 0. The default value is 1.
Max Vert Element Spacing (5)	The maximum vertical spacing, in pixels, between the elements that are combined to form a single character. The default value is 0, which specifies that any object in the ROI should be considered part of a valid object.
Min Bounding Rect Width (6)	The minimum width, in pixels, of a character bounding rectangle. The value of this property cannot exceed <b>Max Bounding Rect Width</b> . The minimum value allowed for this property, which is also the default value, is 1.
Max Bounding Rect Width (7)	The maximum width, in pixels, of a character bounding rectangle. Set this value to 65536, which is also the default value, to specify that all values greater than Min Bounding Rect Width are acceptable. The value of this property must not be less than Min Bounding Rect Width.
Min Bounding Rect Height (8)	The minimum height, in pixels, of a character bounding rectangle. The value of this property cannot exceed <b>Max Bounding Rect Height</b> . The minimum value allowed for this property, which is also the default value, is 1.
Max Bounding Rect Height (9)	The maximum height, in pixels, of a character bounding rectangle. Set this value to 65536, which is also the default value, to specify that all values greater than <b>Min Bounding Rect Height</b> are acceptable. The value of this property must not be less than <b>Min Bounding Rect Height</b> .
Read Strategy (10)	The read strategy determines how closely OCR analyzes images in the reading process to match objects with trained characters. The following list includes valid values:

	<ul> <li>Aggressive—Use few criteria to determine character match.</li> <li>Conservative—Use extensive criteria to determine character match.</li> <li>The default value is Aggressive.</li> </ul>
Aspect Ratio (11)	The maximum aspect ratio variation percentage that the IMAQ OCR Read Text 3 VI allows when it identifies characters. The minimum value for this property is 100, which specifies that the identified object must match the trained character exactly in size and height/width ratio. The default aspect ratio is 400.
Number Of Erosions (12)	The number of <u>erosions</u> to perform to remove small objects. After performing the specified number of erosions, OCR restores remaining objects to their original uneroded size. The default value is 0.
Read Resolution (13)	The level of character criteria that OCR uses to determine if an object matches a trained character. The default is low resolution.
Size of Char Set (14)	The number of characters in the trained character set. This property is read-only.

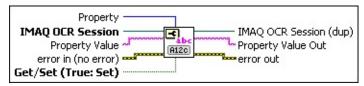
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- Property Value is the new value of the property if **Get/Set** is set to TRUE. Otherwise, **Property Value** is ignored.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set (True: Set) specifies whether the VI gets or sets the property value. TRUE sets the property value. FALSE gets the property value.
- IMAQ OCR Session (dup) is the reference to the OCR session to which this VI operates.
- Property Value Out returns the value of the property specified by Property.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ OCR String Property**

Gets and sets the value of the property specified by Property. This VI works only with string properties.



**Property** specifies the property that this VI gets or sets. The following value is valid:

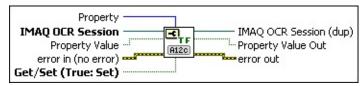
Substitution	The character that the <b>IMAQ OCR Read Text 3</b> VI
Character (15)	substitutes for objects it cannot match with any of
	the trained characters. The default value is a
	question mark (?).

- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- Property Value is the new value of the property if **Get/Set** is set to TRUE. Otherwise, **Property Value** is ignored.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set (True: Set) specifies whether the VI gets or sets the property value. TRUE sets the property value. FALSE gets the property value.
- **IMAQ OCR Session (dup)** is the reference to the OCR session to which this VI operates.
- Property Value Out is the value of the property specified by Property.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ OCR Boolean Property**

Gets and sets the value of the property specified by Property. This VI works only with Boolean properties.



**Property** specifies the property that this VI gets or sets. The following values are valid:

Particles Touching	Set this property to TRUE for OCR to remove objects that are touching the ROI you specified. The removed objects are not considered for training or reading. When you set this property to FALSE, OCR retains objects that are touching the ROI. The default value is FALSE.
Auto Split (17)	Set this property to TRUE to automatically adjust the location of the character bounding rectangle when characters overlap vertically. This property is useful when you are working with an image that contains slanted characters. If the characters are not slanted and/or do not overlap vertically, set this property to FALSE. The default value is FALSE.

- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- **Property Value** is the new value of the property if **Get/Set** is set to TRUE. Otherwise, **Property Value** is ignored.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

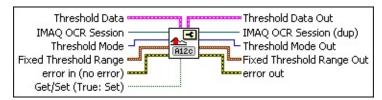
from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set (True: Set) specifies whether the VI gets or sets the property value. TRUE sets the property value. FALSE gets the property value.
- IMAQ OCR Session (dup) is the reference to the OCR session to which this VI operates.
- Property Value Out is the value of the property specified by Property.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ OCR Threshold Data VI**

Owning Palette: OCRInstalled With: NI Vision Development Module

Gets and sets the threshold data.



- Threshold Data specifies parameters that the VI sets if Get/Set is TRUE. The value of this cluster is ignored if Get/Set is FALSE.
  - Dark Characters specifies how the characters appear in the image. FALSE indicates that the image contains light characters on a dark background. TRUE indicates that the image contains dark characters on a light background. The default value is TRUE.
  - Number of Blocks specifies the number of blocks that OCR uses when Threshold Mode is Linear or Non-linear. Valid values range from 4 to 50.
  - Threshold Limits specifies the upper and lower bounds of the threshold values used to calculate the threshold.
    - **Lower Value** is the lower bound of the calculated threshold value.
    - Upper Value is the upper bound of the calculated threshold value.
  - Optimize For Speed TRUE enables OCR to use a faster, less accurate threshold calculation algorithm. FALSE enables OCR to use a slower, more accurate threshold calculation algorithm.
  - Perform Bi-modal Calculation TRUE enables OCR to calculate both the low and high threshold values when Optimize for Speed is set to TRUE. The default is FALSE. If you set this item to TRUE, Dark Characters is ignored.
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.

Threshold Mode specifies the thresholding method that the VI sets when **Get/Set** is TRUE. This value is ignored if **Get/Set** is FALSE. The following values are valid:

Fixed Range (0)	Use this mode when you want to specify the low and high threshold values. OCR performs no automatic
Uniform (1)	calculations. This is the fastest thresholding method.  Computes a single threshold value for the entire ROI.  Uniform is the default.
Linear (2)	Divides the ROI into the number of blocks specified by <b>Number of Blocks</b> , calculates a threshold value for each block, and then uses the value to extract pixel data. Use this mode when the light intensity varies uniformly across the ROI.
Non- linear (3)	Calculates a value on the left side of the ROI, a value on the right size of the ROI, and then fills the middle values in a linear fashion from left to right. The size of the steps from which to calculate the left and right values is based on the value of <b>Number of Blocks</b> .

- Fixed Threshold Range specifies the fixed threshold range when Threshold Mode is set to Fixed Range and Get/Set is TRUE.

  Fixed Threshold Range is ignored for all other values of Threshold Mode or if Get/Set is FALSE.
  - **Lower value** specifies the lower bound of the threshold range.
  - **Upper value** specifies the upper bound of the threshold range.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Get/Set (True: Set) specifies whether the VI gets or sets the property value. TRUE sets the property value. FALSE gets the property value.
- Threshold Data Out is the current value of the advanced threshold data.
  - Dark Characters specifies how the characters appear in the image. FALSE indicates that the image contains light characters on a dark background. TRUE indicates that the image contains dark characters on a light background. The default value is TRUE.
  - Number of Blocks specifies the number of blocks that OCR uses when Threshold Mode is Linear or Non-linear.
  - Threshold Limits specifies the upper and lower bounds of the threshold values used to calculate the threshold.
    - **Lower Value** is the lower bound of the calculated threshold value.
    - Upper Value is the upper bound of the calculated threshold value.
  - Optimize For Speed TRUE enables OCR to use a faster, less accurate threshold calculation algorithm. FALSE enables OCR to use a slower, more accurate threshold calculation algorithm.
  - Perform Bi-modal Calculation TRUE enables OCR to

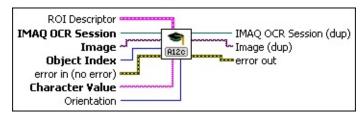
- calculate both the low and high threshold values when **Optimize for Speed** is set to TRUE.
- IMAQ OCR Session (dup) is the reference to the OCR session to which this VI operates.
- Threshold Mode Out is the current value of the thresholding mode.
- **Fixed Threshold Range Out** is the current value of the fixed threshold range.
  - **Lower value** specifies the lower bound of the threshold range.
  - **Upper value** specifies the upper bound of the threshold range.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ OCR Train VI**

Owning Palette: OCRInstalled With: NI Vision Development Module

Assigns character values to the objects that NI Vision identifies in the image. The newly trained characters are appended to the existing trained character set. An image can contain no more than 255 objects.





- ROI Descriptor specifies the ROI in which the VI performs this operation. The ROI must be a rectangle or rotated rectangle. Leave this input unwired to use the entire image for this operation.
  - **Global Rectangle** contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- **Image** is the source image for this operation.
- Object Index specifies the index of the object that you want to train in the set of objects that are identified within the ROI.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

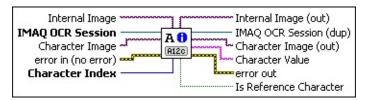
before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Character Value specifies the character value for the object that you want to train. The string must not be longer than 255 characters.
- IMAQ OCR Session (dup) is the reference to the OCR session to which this VI operates.
- Image (dup) has the same value as Image.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

## **IMAQ OCR Character Info VI**

**Owning Palette:** OCRInstalled With: NI Vision Development Module Retrieves information about the trained character at the specified index.



- Internal Image is the internal representation that OCR stores for this character. Internal Image is an 8-bit image.
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the IMAQ OCR Create Session VI.
- Character Image is the image you used to train this character. Character Image can be of type 8 bit, RGB, or HSL.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- Character Index is the index of a character in the character set.
- Internal Image (out) is valid only if you specified an Internal Image. Internal Image returns the internal representation that OCR stores for this character.
- **IMAQ OCR Session (dup)** is the reference to the OCR session to which this VI operates.
- Character Image (out) is valid only if you specified a Character Image. Character Image (out) returns the image that OCR used to train this character.
- **Character Value** is the value of the corresponding character.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Is Reference Character** is TRUE if the character is the reference character for the character class.

## **IMAQ OCR Rename Character VI**

Owning Palette: OCRInstalled With: NI Vision Development Module Renames the trained character at the index specified by Character Index.

IMAQ OCR Session Character Index
New Character Value error in (no error)

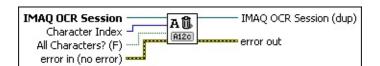
AB 8

- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- Character Index is the index of a character in the character set.
- New Character Value is the new character value that you want to assign to a character in the character set. Character index specifies the index of the character to which you want to assign a new character value. The length of this string must not exceed 255 characters.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

- **IMAQ OCR Session (dup)** is the reference to the OCR session to which this VI operates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ OCR Delete Character VI**

Owning Palette: OCRInstalled With: NI Vision Development Module Deletes a character from the trained character set at the index specified



by **Character Index**.

- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- **Character Index** is the index of a character in the character set.
- All Characters? (F) specifies if the VI deletes all characters. The default value, FALSE, deletes only the character at the index specified by Character Index. Set this value to TRUE to delete all characters in the character set.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

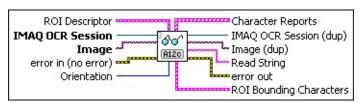
- **IMAQ OCR Session (dup)** is the reference to the OCR session to which this VI operates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ OCR Read Text 3 VI**

Installed With: NI Vision Development Module

Reads the text in the image. The VI identifies all objects in the image based on the properties that you set, and then compares each object with every character in the character set file. For each object, the VI selects the character that most closely matched the object. The VI uses the substitution character for any object that did not match any of the trained characters. You use the **Substitution Character** property to specify the substitution character.





- ROI Descriptor specifies the ROI in which the VI performs this operation. The ROI must be a rectangle or rotated rectangle. Leave this input unwired to use the entire image for this operation.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- **Image** is the source image for this operation.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to

**error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Orientation specifies the orientation of the image data within the circular strip.

(Default) Specifies that the base of the image data is located along the inside edge of the circular strip.
Specifies that the base of the image data is located along the outside edge of the circular strip.

- Character Reports contains additional information about each character that the VI read from the image.
  - **Character Value** specifies the identified character.
  - Classification Score indicates the degree to which the assigned character class represents the object better than the other character classes in the character set.
  - Verification Score returns the verification scores computed against the reference character for the character class. If a reference character does not exist for the character class, the score will be 0.
  - **Verified** is TRUE if the character has been verified against a reference character was found for the character class.

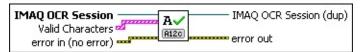
Otherwise, this value is FALSE.

- Character Statistics is cluster containing statistics about the characters OCR segmented in the ROI.
  - Threshold Range specifies the range used to threshold the character.
    - **Lower value** is the lowest pixel value used during a manual threshold. The default is 1.
    - **Upper value** is the highest pixel value used during a manual threshold. The default is 255.
  - Left Offset specifies the left offset of the character bounding rectangles in the current ROI.
  - **Top Offset** specifies the top offset of the character bounding rectangles in the current ROI.
  - **Width** lists the width of each of the characters you trained in the current ROI.
  - Height lists the height of each of the characters you trained in the current ROI.
  - Character Size lists the size of the character in pixels.
- IMAQ OCR Session (dup) is the reference to the OCR session to which this VI operates.
- Image (dup) has the same value as Image.
- Read String is the string that the VI read.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- ROI Bounding Characters is an array that specifies the coordinates of the character bounding ROI.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.

## **IMAQ OCR Set Valid Characters VI**

**Owning Palette:** OCRInstalled With: NI Vision Development Module Specifies the characters that are valid for each character position in the image.



IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the IMAQ OCR Create Session VI.

Valid Characters is the array of characters that are valid for a particular read character position. The valid characters are stored at the array index that corresponds with the index of the read character that OCR is currently evaluating. For example, if OCR is evaluating the read character at index 0, it compares that read character to the valid characters that are stored at index 0 in the array of valid characters. The number of elements in the array is the number of characters that OCR reads. OCR ignores any array elements beyond this number.

Pre-defined characters defines predefined valid character strings. The following characters are the pre-defined characters for the corresponding position:

Any Character (0)	Any ASCII character
User-defined Characters (1)	Uses characters you specify in the <b>User-defined characters</b> string.
Alphabets (2)	A-Z
	a-z
Alphanumeric (3)	A-Z
	a-z
	0-9
Uppercase Letters (4)	A-Z
Lowercase	a-z

Letters (5)	
Decimal Digits (6)	0-9
Hexadecimal Digits (7)	0-9 A-F
Pattern (8)	Any Pattern
Force Space (9)	Ignores the read character and forces a space at the corresponding position

- User-defined characters is the string of characters that is valid for the corresponding character position when Predefined characters is set to User-defined Characters.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **IMAQ OCR Session (dup)** is the reference to the OCR session to which this VI operates.
- error out contains error information. If error in indicates that an

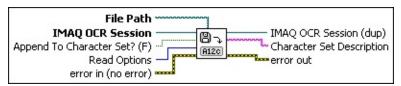
error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

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

## IMAQ OCR Read Character Set File VI

Owning Palette: OCRInstalled With: NI Vision Development Module

Reads a character set and session properties from the character set file specified by **File Path**. The VI adds the character set to the trained character set that the <u>IMAQ OCR Read Text 3</u> VI uses during the reading process.



- File Path is the complete pathname, including drive, directory, and filename, of the file to read.
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- Append To Character Set? (F) specifies whether to append trained characters to the existing character set or replace the existing character set. The characters you append to or copy over the existing character set are those in the character set file you specified in File Path. Set this value to FALSE to replace the existing character set. Set this value to TRUE to append trained characters to the existing character set. The default is FALSE.
- Read Options specifies what to read from the file. You can choose from the following values:

Read All (0)	Read both the character set data and the session properties from the file.
Read Character Set Only (1)	(Default) Read only the character set data from the file.
Read Properties Only (2)	Read only the session properties from the file.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred

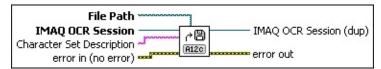
before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- IMAQ OCR Session (dup) is the reference to the OCR session to which this VI operates.
- Example 2 Character Set Description describes the character set.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ OCR Write Character Set File VI**

Owning Palette: OCRInstalled With: NI Vision Development Module

Stores the trained character set and properties of the session in the file specified by File Path.



- **File Path** is the complete pathname, including drive, directory, and filename, of the file to write.
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- Character Set Description describes the character set. The description must be no more than 255 characters.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

**IMAQ OCR Session (dup)** is the reference to the OCR session to which this VI operates.

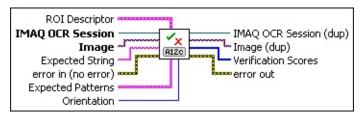
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ OCR Verify Text VI**

Owning Palette: OCRInstalled With: NI Vision Development Module

Verifies the accuracy of the text in the image. For each character, the VI checks for the existence of a reference character for the expected character class and compares the character from the image to the reference character.





- ROI Descriptor specifies the ROI in which the VI performs this operation. The ROI must be a rectangle or rotated rectangle. If the ROI contains multiple contours, each contour is interpreted as a character location. Otherwise the VI searches the ROI for the expected characters. Leave this input unwired to use the entire image for this operation.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the IMAQ OCR Create Session VI.
- **Image** is the source image for this operation.
- **Expected String** contains the expected character values.
- error in (no error) describes the error status before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Expected Patterns** is an array of expected character values and character patterns. The VI only uses this value if **Expected String** is an empty string.
- IMAQ OCR Session (dup) is the reference to the OCR session to which this VI operates.
- Image (dup) has the same value as Image.
- **Verification Scores** is the array of verification scores for the expected characters.
  - Note Verification Scores is set to 0 for a character if the corresponding character class does not contain a reference character.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for

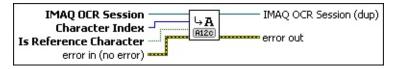
more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **IMAQ OCR Set Reference Character VI**

Owning Palette: OCRInstalled With: NI Vision Development Module

Sets a character as the reference character for the character class. If the character class already has a reference character, the new character will replace the old character as the reference character.



- IMAQ OCR Session specifies the OCR session on which this VI operates. To create a session, use the <a href="IMAQ OCR Create Session">IMAQ OCR Create Session</a> VI.
- Character Index is the index of a character in the character set.
- Is Reference Character specifies that the character is the reference character for the character class. Set this parameter to TRUE if the specified character should be used as a reference character for the character class. Set this parameter to FALSE if you do not want the character to be the reference character for the character class.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **IMAQ OCR Session (dup)** is the reference to the OCR session to which this VI operates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Instrument Readers**

Owning Palette: Machine VisionInstalled With: NI Vision Development Module

Use the Instrument Reader VIs to develop applications that require reading from seven-segment displays, meters or gauges, or one-dimensional barcodes.

Palette Object	Description
IMAQ Get LCD ROI	Use this VI to locate a seven-segment display within an area of interest. The display can contain multiple digits. To find the area of each digit, all of the segments of the seven-segment indicator must be activated.
IMAQ Read LCD	Reads an LCD display in an input image using the descriptor of the ROI calculated by the IMAQ Get LCD ROI. IMAQ Read LCD returns the number as a double and as a string. The Segments status output indicates whether the number is recognized properly or any of the digits are not recognized fully. This output returns the status of each of the digits in the display. This VI reads LCD or LED indicators and tolerates light drift.
IMAQ Read Single Digit	Reads a single seven-segment digit from a rectangular area of interest drawn around this digit. This VI reads LCD and LED indicators. It returns a value as a string and a cluster containing the state of each segment (ON or OFF).
IMAQ Get Meter	Calibrates the meter or gauge using the initial and the full-scale position of the needle. It calculates the position of the base of the needle and the arc traced by the tip of the needle.
IMAQ Read Meter	Reads the position of the needle using the base of the needle and the array of the points of the arc traced by the tip of the needle. This VI reads meters with a dark needle on a light background or with a light needle on a dark background. It returns the location of the tip of the needle as a percentage of the full range of the meter.
IMAQ Get	Calibrates the meter or gauge using three points on the meter: the base of the needle, the tip of the needle at its initial

Meter 2	position, and the tip of the needle at its full-scale position. It calculates the position of the points along the arc covered by the tip of the needle.
IMAQ Read QR Code	Locates and then reads the value encoded in a QR code.
IMAQ Read Barcode	Reads common 1D barcode types, including Codabar, Code 39, Code 93, Code 128, EAN 8, EAN 13, Interleaved 2 of 5, MSI, UPCA, Pharmacode, and RSS Limited.
IMAQ Read Data Matrix Barcode 2	Locates and then reads the value encoded in a Data Matrix barcode.
IMAQ Grade Data Matrix Barcode AIM	Grades a Data Matrix barcode using the AIM Print Quality metrics included in the ISO 16022 specification. You must first prepare this Data Matrix barcode image for grading using IMAQ Read Data Matrix Barcode 2.
IMAQ Read PDF417 Barcode	Reads values encoded into a PDF417 barcode. You can compare the decoded data to a reference string or check whether the data contains a specific pattern.

#### IMAQ Get LCD ROI VI

Owning Palette: Instrument ReadersInstalled With: NI Vision

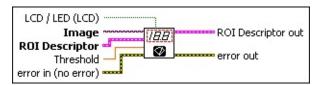
**Development Module** 

Use this VI to locate a seven-segment display within an area of interest. The display can contain multiple digits. To find the area of each digit, all of the segments of the seven-segment indicator must be activated.

This VI is designed for seven-segment displays that use either LCDs or LEDs composed of electroluminescent indicators or light-emitting diodes. It is resistant to light drift. It returns the area of interest containing the global rectangle for each digit. You can pass this ROI Descriptor directly to the <a href="IMAQ Read Meter">IMAQ Read Meter</a> VI.

#### **Examples**





- **LCD / LED (LCD)** is a Boolean that controls the indicator type. FALSE corresponds to LCD indicators, and TRUE corresponds to LED indicators. The default is FALSE.
- **Image** is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the rectangle or rotated rectangle, with a rotation angle of 0, region of interest that contains each digit.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.

- Threshold is the threshold that determines whether a segment is ON or OFF. A segment is considered ON if the standard deviation of the pixels along a line profile across the segment is greater than this threshold. Increase this value when using images with high contrast. Lower this value when using images with low contrast. The default is 8.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- ROI Descriptor out is an ROI descriptor containing bounding rectangles of all the digits found by the algorithm. The bounding rectangle of each located digit is specified as a rectangle contour in the returned ROI descriptor.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.

- **Type** is the shape type of the contour.
- Coordinates indicates the relative position of the contour.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\LCD Example.vi

### **IMAQ Read LCD VI**

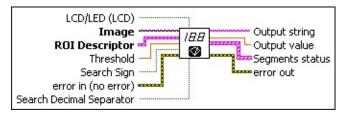
Owning Palette: Instrument Readers Installed With: NI Vision

**Development Module** 

Reads an LCD display in an input image using the descriptor of the ROI calculated by the IMAQ Get LCD ROI. IMAQ Read LCD returns the number as a double and as a string. The Segments status output indicates whether the number is recognized properly or any of the digits are not recognized fully. This output returns the status of each of the digits in the display. This VI reads LCD or LED indicators and tolerates light drift.

#### **Examples**

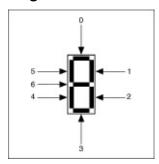




- **LCD/LED (LCD)** is a Boolean that controls the indicator type. FALSE corresponds to LCD indicators, and TRUE corresponds to LED indicators. The default is FALSE.
- **Image** is a reference to the source image.
- ROI Descriptor is a descriptor that defines the region of interest that contains the quadrilaterals surrounding each digit. The ROI is calculated by the <a href="MAQ Get LCD ROI">IMAQ Get LCD ROI</a> VI.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.

- Threshold is the threshold that determines whether a segment is ON or OFF. A segment is considered ON if the standard deviation of the pixels along a line profile across the segment is greater than this threshold. Increase this value when using images with high contrast. Lower this value when using images with low contrast. The default is 8.
- Search Sign indicates whether the algorithm must read the sign of the indicator.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Search Decimal Separator determines whether to look for a decimal separator after each digit. The default is FALSE.
- Output string contains the number that has been read from the display. If the decimal separator is ON and the control Search Decimal Separator is TRUE, it is represented in the string by a dot (.) symbol.
- Output value contains the value of the number represented by the display.

Segments status is an array of clusters containing the state of the segments of each digit. The order of the cluster is illustrated below.



- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\3. Applications\LCD Example.vi

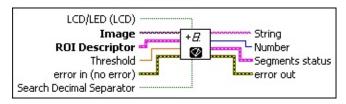
### **IMAQ Read Single Digit VI**

Owning Palette: Instrument Readers Installed With: NI Vision

**Development Module** 

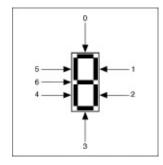
Reads a single seven-segment digit from a rectangular area of interest drawn around this digit. This VI reads LCD and LED indicators. It returns a value as a string and a cluster containing the state of each segment (ON or OFF).





- **LCD/LED (LCD)** is a Boolean that controls the indicator type. FALSE corresponds to LCD indicators, and TRUE corresponds to LED indicators. The default is FALSE.
- Image is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the ROI drawn by the user that corresponds to a quadrilateral surrounding the digit.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Threshold is the threshold that determines whether a segment is ON or OFF. A segment is considered ON if the standard deviation of the pixels along a line profile across the segment is greater than this threshold. Increase this value when using images with high contrast. Lower this value when using images with low contrast. The default is 8.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Search Decimal Separator determines whether to look for a decimal separator after each digit. The default is FALSE.
- String contains the value of the digit and decimal separator. If Search Decimal Separator is enabled, the decimal separator is represented in the string by a period (.) symbol.
- Number is the value represented by the digit.
- Segments status is a cluster of Boolean indicators containing the state of each segment of the digit. The order of the cluster is illustrated below.



- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

#### **IMAQ Get Meter VI**

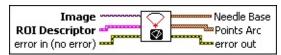
Owning Palette: Instrument ReadersInstalled With: NI Vision

**Development Module** 

Calibrates the meter or gauge using the initial and the full-scale position of the needle. It calculates the position of the base of the needle and the arc traced by the tip of the needle.

#### **Examples**





- **Image** is a reference to the source image.
- ROI Descriptor is a descriptor that defines the ROI drawn by the user to identify the initial position of the needle and the full-scale position of the needle. The origin of each line must be the tip of the needle.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - Coordinates indicates the relative position of the contour.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out

from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Needle Base is a cluster containing the coordinates of the base of the needle.
- **Points Arc** is an array of clusters containing the coordinates of the points of the arc traced by the tip of the needle.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

 $\bullet \quad LabVIEW \land amples \land Vision \land 3. \ Applications \land Meter\ Example. vi$ 

### **IMAQ Read Meter VI**

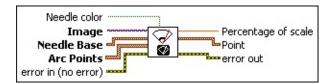
Owning Palette: Instrument Readers Installed With: NI Vision

**Development Module** 

Reads the position of the needle using the base of the needle and the array of the points of the arc traced by the tip of the needle. This VI reads meters with a dark needle on a light background or with a light needle on a dark background. It returns the location of the tip of the needle as a percentage of the full range of the meter.

#### **Examples**





- Needle color is a Boolean that controls the color of the needle. FALSE corresponds to a dark needle on a light background, and TRUE corresponds to a light needle on a dark background. The default is FALSE.
- Image is a reference to the source image.
- Needle Base is a cluster containing the coordinates of the base of the needle.
- Arc Points is an array of clusters containing the coordinates of the points of the arc traced by the tip of the needle.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Percentage of scale contains the angle of the needle as a percentage of the full-scale angle.
- **Point** is a cluster that specifies the coordinates of the needle tip.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

 $\bullet \quad LabVIEW \land amples \land Vision \land 3. \ Applications \land Meter\ Example. vi$ 

#### **IMAQ Get Meter 2 VI**

Owning Palette: Instrument ReadersInstalled With: NI Vision

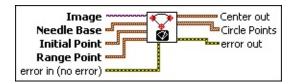
**Development Module** 

Calibrates the meter or gauge using three points on the meter: the base of the needle, the tip of the needle at its initial position, and the tip of the needle at its full-scale position. It calculates the position of the points along the arc covered by the tip of the needle.



**Note** Use this VI with meters or gauges equipped with a dark needle on a light background or with a light needle on a dark background.





- Image is a reference to the source image.
- Needle Base is a cluster containing the coordinates of the base of the needle.
- **Initial Point** is a point cluster containing the coordinates of the tip of the needle in its initial position.
- Range Point is a point cluster containing the coordinates of the tip of the needle in its full-scale position.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Center out** is a point cluster containing the coordinates of the rotation center of the needle.
- Circle Points is an array of point clusters containing the coordinates of the points along the arc traced by the tip of the needle.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

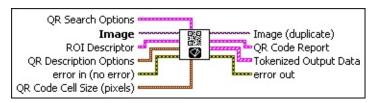
### **IMAQ Read QR Code VI**

Owning Palette: Instrument Readers Installed With: NI Vision

**Development Module** 

Locates and then reads the value encoded in a QR code. You can compare the decoded data to a reference string or check whether the data contains a specific pattern. Many of the options of this VI allow for automatic detection of QR code properties or determine which methods the VI should use to locate and decode the QR code. Selecting specific properties and methods for these options will greatly increase the performance of the VI.

#### **Examples**



**QR Search Options** is a cluster that contains the search options to use when searching for QR codes in the image.

Rotation Mode specifies the amount of QR code rotation the VI should allow for. The following values are valid:

Unlimited (0)	The VI allows for unlimited rotation
0 Degrees (1)	The VI allows for 0 degrees of rotation
90 Degrees (2)	The VI allows for 90 degrees of rotation
<b>180 Degrees</b> (3)	The VI allows for 180 degrees of rotation
<b>270 Degrees</b> (4)	The VI allows for 270 degrees of rotation

Skip Location?, when TRUE, specifies that the VI should assume that the QR code occupies the entire image (or the entire search region). The VI skips the location phase and moves immediately to extracting and decoding the QR code. When FALSE, the VI does not make any assumptions about the percentage of the image occupied by the QR code. The default value is FALSE.

Edge Threshold specifies the minimum contrast a pixel must have in order to be considered part of a matrix cell edge. The

lower this value is set, the more potential edge candidates the VI will examine during the location phase. Setting this value too low will decrease the performance of the VI because the VI will examine too many potential edge candidates. Setting this value too high may also decrease the performance of the VI by removing valid edge candidates, making location more difficult. Setting this value too high may also cause the VI to fail to identify the matrix because all edge candidates are eliminated.

Demodulation Mode specifies the mode the VI should use to demodulate (determine which cells are on and which cells are off) the QR code. The following values are valid:

Auto-detect (-2)	(Default) The VI tries each demodulation mode and use the one which decodes the QR code within the fewest iterations and utilizing the least amount of error correction.
Histogram (0)	The VI uses a histogram of all of the matrix cells to calculate a threshold. This threshold determines if a cell is on or off. This is the fastest method, but it requires QR code images with consistent levels of contrast in the matrix.
Local Contrast (1)	The VI examines each of the cell's neighbors to determine if the cell is on or off. This method is slower, but works with QR code images that have inconsistent levels of contrast in the matrix.
Combination (2)	The VI uses the histogram of the matrix to calculate a threshold. For cells with pixel values that are sufficiently below or above this threshold, the VI will use the threshold to determine if the cell is on or off. If the cell pixel values are close to the threshold, the VI will use the Local Contrast method to determine if the cell is on or off. This method is slower, but works with QR code

images that have extremely low cell fill percentages or gross print growth errors.
The VI tries <b>Histogram</b> , <b>Local Contrast</b> , and <b>Combination</b> , stopping once one mode is successful.

**Cell Sample Size** specifies the sample size, in pixels, the VI should take to determine if each cell is on or off.

Auto- detect (-2)	(Default) The VI will try each sample size and use the one which decodes the QR code within the fewest iterations and utilizing the least amount of error correction.
<b>1×1</b> (1)	The VI will use a 1×1 sized sample from each cell.
<b>2×2</b> (2)	The VI will use a 2×2 sized sample from each cell.
<b>3×3</b> (3)	The VI will use a 3×3 sized sample from each cell.
<b>4×4</b> (4)	The VI will use a 4×4 sized sample from each cell.
<b>5×5</b> (5)	The VI will use a 5×5 sized sample from each cell.
<b>6×6</b> (6)	The VI will use a 6×6 sized sample from each cell.
<b>7×7</b> (7)	The VI will use a 7×7 sized sample from each cell.

Cell Filter Mode specifies the mode the VI uses to determine the pixel value for each cell. Note that if Cell Sample Size is 1×1, the value of the single samples pixel always determines the pixel value for the cell. The following values are valid:

detect (-2)	The VI tries all filter modes and uses the one that decodes the QR code within the fewest iterations and utilizing the least amount of error correction.
Average (0)	The VI sets the pixel value for the cell to the

	average of the sampled pixels.
Median (1)	The VI sets the pixel value for the cell to the median of the sampled pixels.
Central Average (2)	The VI sets the pixel value for the cell to the average of the pixels in the center of the cell sample.
High Average (3)	The VI sets the pixel value for the cell to the average value of the half of the sampled pixels with the highest pixel values.
Low Average (4)	The VI sets the pixel value for the cell to the average value of the half of the sampled pixels with the lowest pixel values.
Very High Average (5)	The VI sets the pixel value for the cell to the average value of the ninth of the sampled pixels with the highest pixel values.
Very Low Average (6)	The VI sets the pixel value for the cell to the average value of the ninth of the sampled pixels with the lowest pixel values.
All Filters (8)	The VI tries each filter mode, starting with <b>Average</b> and ending with <b>Very Low Average</b> , stopping once a filter mode decodes correctly.

- Skew Degrees Allowed specifies the amount of skew in the code the VI should allow for. The default is 5 degrees.
- **Image** is a reference to the source image.
- **ROI Descriptor** is a descriptor that defines the the rectangle, rotated rectangle, or polygon within which the QR code is located.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.

[132]

**Coordinates** indicates the relative position of the contour.

- **QR Description Options** is a cluster that contains the QR code description options to use when searching for QR codes in the image.
  - Matrix Size (cells) specifies the size of QR code.
  - Barcode Polarity specifies the data-to-background contrast for the QR code. The following values are valid:

Auto- detect (-2)	(Default) Sets the VI to determine the code polarity automatically.
Black On White (0)	Sets the VI to read codes with dark data on a bright background.
White On Black (1)	Sets the VI to read codes with bright data on a dark background.

Mirror Mode specifies if the code appears normally in the image or if the code appears mirrored in the image. The following values are valid:

	(Default) Sets the VI to automatically determine if the code is mirrored.
Normal (0)	Sets the VI to read codes that appear normally in the image.
Mirrored (1)	Sets the VI to read codes that appear mirrored in the image.

QR Model is the Model Type to look for. There are three major QR code models: QR Code Model 1, QR Code Model 2, and Micro QR Code. Micro QR Codes have only a single target in the top left, while Model 1 and Model 2 have targets in all corners except for the bottom right. Model 2 codes have alignment patterns inside the data. Model 1 codes have alignment patterns along the edges.

Auto Detect (0)	Automatically selects the QR model.
Micro (1)	Selects the Micro QR Code model.
Model 1 (2)	Selects the QR Code Model 1 model.

<b>Model 2</b> (3)	Selects the QR Code Model 2 model.
--------------------	------------------------------------

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **QR Code Cell Size (pixels)** is a cluster that contains the code size options to use when searching for codes in the image. The size refers to cell size in pixels.
  - Minimum Size specifies the minimum size (in pixels) of a cell in the code.
  - Maximum Size specifies the maximum size (in pixels) of a cell in the code.
- Image (duplicate) is the reference to the image that contains the QR code.
- **QR Code Report** is a cluster that contains the QR code description options to use when searching for QR codes in the image.
  - **Found?** returns TRUE if a QR code was found and decoded in the image. This parameter returns FALSE if the VI failed to

- locate and decode a QR barcode.
- **Data** shows the encoded data in the code.
- **Bounding Box** is an array that define the boundary of the QR code in the image.
- **Errors Corrected** specifies the number of rows in the QR code.
- Dimensions (Rows/Cols) specifies the number of rows and columns in the QR code. This is measured in cells. All QR codes are square by specification.
- **Version** specifies the version of the QR code. This is with respect to the model type.
- **QR Model** is the QR code model found.
- Stream Mode specifies the stream mode or the format of the data that is encoded in the QR code.
- **Barcode Polarity** indicates the data-to-background contrast for the QR code.
- Mirrored? indicates if the QR code appeared mirrored (TRUE) or normally (FALSE) in the image.
- Append Stream Position indicates what position the QR symbol is in with respect to the stream of data in all symbols. It is possible for a QR symbol to be part of a larger array of symbols.
- Size of Append Stream specifies how many QR symbols are part of a larger array of symbols. Sometimes a QR symbol is part of a larger array of symbols. This tells us how many symbols are involved in this array.
- First EAN-128 ID is the first EAN-128 Application ID encountered in the stream. This is only useful for EAN-128 codes. For mixed/appended EAN-128 codes, refer to the tokenized output.
- First Alternate Language ID is the first Regional Language Designator encountered in the stream. This is only useful for ECI codes and for multiple language ECI symbols, refer to the tokenized output.

- Append Stream ID specifies what stream this symbol is in relation to. Sometimes a QR symbol is part of a larger array of symbols. Stream ID can be useful when there are a number of QR streams and a symbol needs to be categorized.
- Minimum Edge Strength specifies the minimum contrast a pixel must have in order to be considered part of a matrix cell edge. The lower this value, the more potential edge candidates the VI will examine during the location phase. Setting this value too low will decrease the performance of the VI because the VI will examine too many potential edge candidates. Setting this value too high may also decrease the performance of the VI by removing valid edge candidates, making location more difficult. Setting this value too high may also cause the VI to fail to identify the matrix because all edge candidates are eliminated.
- **Demodulation Mode** specifies the mode the VI used to demodulate (determine which cells are on and which cells are off) the QR code.
- **Cell Sample Size** specifies the sample size, in pixels, the VI used to determine if each cell is on or off.
- Cell Filter Mode specifies the mode the VI used to determine the pixel value for each cell.
- Tokenized Output Data contains the data tokenized in exactly the way it was encoded in the symbol. This is useful if the symbol is encoded using multiple languages. ModeData will contain the language identifier. This is also useful if the symbol contains multiple EAN-128 application IDs. ModeData will contain the Application Identifiers. Otherwise, the QR Code Report data should suffice.
  - Stream Mode specifies the stream mode or the format of the data that is encoded in the QR code.
  - Mode Data specifies Stream Mode specific data. This parameter uses the following mapping:
    - Numeric Length of Data String
    - AlphaNumeric Length of Data String

- Raw-Byte Length of Data String
- UCC/EAN-128 In Tokens Application ID for the length of the data token
- UCC/EAN-128 In Data Length of Data String
- Extended Character Interpretation Language Identifier for the length of the data token
- Kanji –Length of Data String
- **Data** shows the encoded data in the code.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

Refer to the following for an example that uses this VI.

• LabVIEW\examples\Vision\2. Functions\2D Barcodes\2D Barcode.llb

### **IMAQ Read Barcode VI**

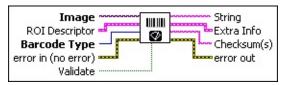
Owning Palette: Instrument ReadersInstalled With: NI Vision

**Development Module** 

Reads common 1D barcode types, including Codabar, Code 39, Code 93, Code 128, EAN 8, EAN 13, Interleaved 2 of 5, MSI, UPCA, Pharmacode, and RSS Limited.

#### **Examples**





- Image is a reference to the source image.
- ROI Descriptor is a descriptor that defines the region of interest within which the code is located. The ROI must be a rectangle or rotated rectangle with a rotation of 0 degrees. If the ROI descriptor is empty or not connected, the entire image is considered to be the region.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - Contours are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.

#### **Barcode Type** is the type of barcode to read.

Codabar (1)	Reads a Codabar barcode.
Code 39 (2)	Reads a Code 39 barcode.
Code 93 (3)	Reads a Code 93 barcode.
Code 128 (4)	Reads a Code 128 barcode.

<b>EAN 8</b> (5)	Reads an EAN 8 barcode.
<b>EAN 13</b> (6)	Reads an EAN 13 barcode.
Interleaved 2 of 5 (7)	Reads an Interleaved 2 or 5 barcode
MSI (8)	Reads an MSI barcode.
<b>UPCA</b> (9)	Reads a UPCA barcode.
Pharmacode (10)	Reads a Pharmacode barcode.
RSS Limited (11)	Reads an RSS Limited barcode.

- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Validate** specifies whether to validate the barcode's data. If the barcode type is Codabar, Code 39, or Interleaved 2 of 5, the error correction information is used to validate the results. For all other barcode types, either no validation is performed or the validation is performed automatically because the type requires it.
- **String** is the decoded barcode data.
- **Extra Info** is a cluster that returns information about the barcode

read. **Extra Info** contains the following elements:

Special Char 1 contains information about the barcode, depending on the type of barcode read.

- **Codabar**—If the VI reads a Codabar barcode, this parameter returns the start character.
- Code 128—If the VI reads a Code 128 barcode, this parameter returns the FNC character.
- EAN 8 and EAN 13—If the VI reads an EAN 8 or EAN 13 barcode, this parameter returns the first country code.

For all other barcodes types, the value of this parameter is set to 0.

- Special Char 2 contains information about the barcode, depending on the type of barcode read.
  - **Codabar**—If the VI reads a Codabar barcode, this parameter returns the stop character.
  - EAN 8 and EAN 13—If the VI reads an EAN 8 or EAN 13 barcode, this parameter returns the second country code.
  - **UPCA**—If the VI reads a UPCA barcode, this parameter returns the system number.

For all other barcodes types, the value of this parameter is set to 0.

Confidence Level is a quality measure of the decoded barcode ranging from 0 to 1000, with 1000 being the best. This value weights the error in the widths of the bars and spaces with the size of the character in the barcode. In general, a confidence level of less than 800 means the decoded string is suspect.



**Note Confidence Level** is particularly useful in decoding EAN 13 barcodes because 11 of the 12 values are encoded as characters in the barcode, and the twelfth value is encoded by the parity of the first 11 encoded characters.

**Barcode Type** is the type of barcode read.

- Checksum(s) is error correction information added to the encoded data that you can use to validate the decoded data. Because Checksum(s) is not part of the input data, it is not returned in the output string.
  - Note Not all barcode types have checksum error correction capabilities.
  - Note In many cases, the text printed below the barcode contains the encoded data, including the checksum. For example, an EAN-13 barcode can display text containing 13 numeric digits. The first 12 digits represent the actual input data and the thirteenth is the checksum.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

### **Examples**

Refer to the following for examples that use this VI.

- LabVIEW\examples\Vision\3. Applications\Barcode Example.vi
- LabVIEW\examples\Vision\2. Functions\Image Management\Unwrap Barcode Example.vi

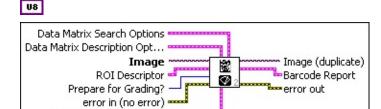
### **IMAQ Read Data Matrix Barcode 2 VI**

Owning Palette: Instrument Readers Installed With: NI Vision

**Development Module** 

Data Matrix Size Options

Locates and then reads the value encoded in a Data Matrix barcode. You can compare the decoded data to a reference string or check whether the data contains a specific pattern. Many of the options of this VI allow for automatic detection of Data Matrix barcode properties or determine which methods the VI should use to locate and decode the Data Matrix barcode. However, selecting specific properties and methods for these options will greatly increase the performance of the VI.



**Data Matrix Search Options** is a cluster that contains the options to use when searching for Data Matrix codes in the image.

Rotation Mode specifies the amount of Data Matrix code rotation the VI should allow for. The following values are valid:

Unlimited (0)	The VI allows for unlimited rotation
0 Degrees (1)	The VI allows for 0 degrees of rotation
90 Degrees (2)	The VI allows for 90 degrees of rotation
<b>180 Degrees</b> (3)	The VI allows for 180 degrees of rotation
<b>270 Degrees</b> (4)	The VI allows for 270 degrees of rotation

Skip Location?, if TRUE, specifies that the VI should assume that the Data Matrix code occupies the entire image, or the entire search region. The VI skips the location phase, moving immediately to extraction and decoding. If FALSE (default), the VI does not make any assumptions about the percentage of the image occupied by the Data Matrix code.

Edge Threshold specifies the minimum contrast a pixel must have in order to be considered part of a matrix cell edge. The lower this value, the more potential edge candidates the VI

will examine during the location phase. Setting this value too low will decrease the performance of the VI because the VI will examine too many potential edge candidates. Setting this value too high may also decrease the performance of the VI by removing valid edge candidates, making location more difficult. Setting this value too high may also cause the VI to fail to identify a Data Matrix barcode because all edge candidates are eliminated.

**Demodulation Mode** specifies the mode the VI should use to demodulate (determine which cells are ON and which cells are OFF) the Data Matrix code. The following values are valid:

Auto-detect (-2)	(Default) The VI tries each demodulation mode and use the one which decodes the Data Matrix code within the fewest iterations and utilizing the least amount of error correction.
Histogram (0)	The VI uses a histogram of all of the cells to calculate a threshold. This threshold determines if a cell is on or off. This is the fastest method, but it requires images with consistent levels of contrast in the matrix.
Local Contrast (1)	The VI examines each of the cell's neighbors to determine if the cell is on or off. This method is slower, but works with images that have inconsistent levels of contrast.
Combination (2)	The VI uses the histogram of the cells to calculate a threshold. For cells with pixel values that are sufficiently below or above this threshold, the VI will use the threshold to determine if the cell is on or off. If the cell pixel values are close to the threshold, the VI will use the Local Contrast method to determine if the cell is on or off. This method is slower, but works with Data Matrix codes that have extremely low cell fill percentages or gross print growth

	errors.
<b>AII</b> (3)	The VI tries <b>Histogram</b> , <b>Local Contrast</b> , and <b>Combination</b> , stopping once one mode is successful.

**Cell Sample Size** specifies the sample size, in pixels, the VI should take to determine if each cell is ON or OFF.

Auto- detect (-2)	(Default) The VI tries each sample size and use the one which decodes the Data Matrix code within the fewest iterations and utilizing the least amount of error correction.
<b>1×1</b> (1)	The VI will use a 1×1 sized sample from each cell.
<b>2×2</b> (2)	The VI will use a 2×2 sized sample from each cell.
<b>3×3</b> (3)	The VI will use a 3×3 sized sample from each cell.
<b>4×4</b> (4)	The VI will use a 4×4 sized sample from each cell.
<b>5×5</b> (5)	The VI will use a 5×5 sized sample from each cell.
<b>6×6</b> (6)	The VI will use a 6×6 sized sample from each cell.
<b>7×7</b> (7)	The VI will use a 7×7 sized sample from each cell.

Cell Filter Mode specifies the mode the VI uses to determine the pixel value for each cell.



**Note** If **Cell Sample Size** is 1×1, then the value of the sample pixel determines the pixel value for the cell.

The following values are valid:

Auto-	The VI tries all filter modes and uses the one
detect (-2)	that decodes the Data Matrix code within the
	fewest iterations and utilizing the least amount

	of error correction.
Average (0)	The VI sets the pixel value for the cell to the average of the sampled pixels.
Median (1)	The VI sets the pixel value for the cell to the median of the sampled pixels.
Central Average (2)	The VI sets the pixel value for the cell to the average of the pixels in the center of the cell sample.
High Average (3)	The VI sets the pixel value for the cell to the average value of the half of the sampled pixels with the highest pixel values.
Low Average (4)	The VI sets the pixel value for the cell to the average value of the half of the sampled pixels with the lowest pixel values.
Very High Average (5)	The VI sets the pixel value for the cell to the average value of the ninth of the sampled pixels with the highest pixel values.
Very Low Average (6)	The VI sets the pixel value for the cell to the average value of the ninth of the sampled pixels with the lowest pixel values.
All Filters (8)	The VI tries each filter mode, starting with <b>Average</b> and ending with <b>Very Low Average</b> , stopping once a filter mode decodes correctly.

- Skew Degrees Allowed specifies the amount of skew in the code the VI should allow for. The default is 5 degrees.
- Maximum Iterations specifies the maximum number of iterations before the VI stops looking for the code. The default is 500.
- Initial Search Vector Width specifies the number of pixels the VI should average together to determine the location of an edge. You may need to increase this value when the Data Matrix code has cells with a low fill percentage. The default width is 5 pixels.

Data Matrix code description options to use when searching for Data Matrix codes in the image.

- Aspect Ratio specifies the ratio of the width of a cell of the Data Matrix code (in pixels) divided by the height of a cell of the Data Matrix code (in pixels). Set this value to 0 (default) to indicate that the VI should determine the aspect ratio.
- Rows specifies the number of rows in the Data Matrix code. Set this value to 0 (default) to indicate that the VI should determine the number of rows.
- Columns specifies the number of columns in the Data Matrix code. Set this value to 0 (default) to indicate that the VI should determine the number of columns.
- **Rectangle** specifies if the Data Matrix code is square (default) or rectangular. If both **Rows** and **Columns** are non-zero, the VI ignores this setting.
- **ECC** specifies the ECC used for the Data Matrix code. The following values are valid:

Auto- detect (-2)	(Default) Sets the VI to determine the barcode ECC automatically.
ECC 000 (0)	Sets the VI to read Data Matrix codes of ECC 000 only.
<b>ECC 050</b> (50)	Sets the VI to read Data Matrix codes of ECC 050 only.
<b>ECC 080</b> (80)	Sets the VI to read Data Matrix codes of ECC 080 only.
ECC 100 (10)	Sets the VI to read Data Matrix codes of ECC 100 only.
ECC 140 (14)	Sets the VI to read Data Matrix codes of ECC 140 only.
<b>ECC 000- 140</b> (19)	Sets the VI to read Data Matrix codes of ECC 000, ECC 050, ECC 080, ECC 100, and ECC 140 only.
ECC 200 (20)	Sets the VI to read Data Matrix codes of ECC 200 only.

Barcode Polarity specifies the data-to-background contrast for the Data Matrix code. The following values are valid:

Auto- detect (-2)	(Default) Sets the VI to determine the code polarity automatically.
Black On White (0)	Sets the VI to read codes with dark data on a bright background.
White On Black (1)	Sets the VI to read codes with bright data on a dark background.

**Cell Fill Percentage** specifies the fill percentage for a cell that is in the ON state. The following values are valid:

	(Default) Sets the VI to determine the Data Matrix code cell fill percentage automatically.
	Sets the VI to read Data Matrix codes with a cell fill percentage of less than 30 percent.
>= <b>30</b> % (1)	sets the VI to read Data Matrix codes with a cell fill percentage greater than or equal to 30 percent.

- Minimum Border Integrity % specifies the minimum percentage of the matrix border the VI should expect in the Data Matrix code. During the location phase, the VI will ignore possible matrix candidates that do not have at least this level of border integrity.
- Mirror Mode specifies if the Data Matrix code appears normally in the image or if the Data Matrix code appears mirrored in the image. The following values are valid:

Auto- detect (-2)	(Default) Sets the VI to automatically determine if the code is mirrored.
Normal (0)	Sets the VI to read codes that appear normally in the image.
Mirrored (1)	Sets the VI to read codes that appear mirrored in the image.

**Image** is a reference to the source image.

**ROI Descriptor** is a descriptor that defines the region of interest

within which the Data Matrix code is located. The first contour of the ROI must be a rectangle, rotated rectangle, or convex polygon. If **Skip Location?** of the **Data Matrix Search Options** input is TRUE, a convex polygon contour has an additional constraint of being four-sided. If the ROI descriptor is empty or not connected, the entire image is considered to be the region.

- Global Rectangle contains the coordinates of the bounding rectangle.
- **Contours** are each of the individual shapes that define an ROI.
  - **ID** refers to whether the contour is the external or internal edge of an ROI.
  - **Type** is the shape type of the contour.
  - Coordinates indicates the relative position of the contour.
- Prepare for Grading? specifies if the VI should make calculations needed to prepare to grade the Data Matrix code. The following are valid values:

Do Not Prepare (0)	(Default) Does not make any preparatory calculations. Attempts to grade the Data Matrix code will generate an error.
•	Prepares the image for grading using the AIM Print Quality metrics.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or

- function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Data Matrix Size Options is a cluster that contains the code size options to use when searching for codes in the image.
  - Data Matrix Size specifies the size range (in pixels) for the code in the image. The VI will ignore potential barcode candidates that are sized outside of this range.
    - Minimum Size specifies the minimum size (in pixels) of the code in the image. Setting this value to 0 (the default) indicates the VI should never exclude a barcode candidate because it is too small.
    - of the code in the image. Setting this value to 0 (the default) indicates the VI should never exclude a barcode candidate because it is too large.
  - Quiet Zone Width specifies the expected minimum size of the quiet zone, in pixels. The VI will ignore barcode candidates whose quiet zones are smaller than this value.
- Image (duplicate) is the reference to the image that contains the Data Matrix code.
- **Data Matrix Report** is a cluster of information about the code that the VI read.
  - **Found?** indicates if the VI located (TRUE) or failed to locate (FALSE) a data matrix code in the image.
  - **Binary?** indicates if the Data Matrix code contains encoded non-ASCII binary data (TRUE) or ASCII text (FALSE).
  - **Data** is the encoded information that the VI read.

- **Bounding Box** is an array of five points that define the boundary of the Data Matrix code in the image.
- Errors Corrected is the number of erasures that the VI corrected using inherent error correction.
- Erasures Corrected is the number of erasures that the VI corrected using inherent error correction.
- Aspect Ratio indicates the aspect ratio of the data matrix code in the image.
- **Rows** is the number of rows in the Data Matrix code.
- **Columns** is the number of columns in the code.
- **ECC** is the type of the Data Matrix code.
- **Barcode Polarity** indicates the polarity of the Data Matrix code.
- **Cell Fill Percentage** indicates the cell fill percentage of the code.
- **Border Integrity** is the percentage of the Data Matrix border that appears correctly in the image.
- **Mirrored?** indicates if the code appeared mirrored (TRUE) or normally (FALSE) in the image.
- Minimum Edge Strength indicates the minimum strength of the edges the VI used to find the coarse location of the Data Matrix code in the image. Use this value as a guide for setting the Edge Threshold value in the Barcode Search Options control.
- Demodulation Mode indicates the demodulation mode the VI used to locate the code. If Demodulation Mode is set to Auto-Detect in Data Matrix Search Options, this output indicates the recommended demodulation mode for this image.
- Cell Sample Size indicates the cell sample size the VI used to locate the Data Matrix barcode. If Cell Sample Size is set to Auto-Detect in Data Matrix Search Options, this output indicates the recommended cell sample size for this image.
- Cell Filter Mode indicates the cell filter mode the VI used to

locate the barcode. If **Cell Filter Mode** is set to Auto-Detect in the **Data Matrix Search Options**, this output indicates the recommended cell filter mode for this image.

- Iterations indicates the number of iterations the VI took in attempting to locate the Data Matrix code. If this number is equal to Maximum Iterations in Data Matrix Search Options and the VI failed to locate the code, you may be able to locate the code by increasing Maximum Iterations.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

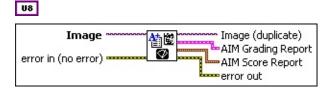
## **IMAQ Grade Data Matrix Barcode AIM VI**

Owning Palette: Instrument ReadersInstalled With: NI Vision Development Module

Grades a Data Matrix barcode using the AIM Print Quality metrics included in the ISO 16022 specification. You must first prepare this Data Matrix barcode image for grading using IMAQ Read Data Matrix Barcode 2.



**Note** Refer to the *NI Vision Concepts Manual* for more information about grading Data Matrix barcodes.



- Image is a reference to the image that contains the Data Matrix code you want to grade. You must first prepare this image for grading using the <a href="IMAQ Read Data Matrix Barcode 2">IMAQ Read Data Matrix Barcode 2</a> VI.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

- Image (duplicate) is the reference to the image that contains the Data Matrix code.
- AIM Grading Report is a cluster of information detailing the grading information for the Data Matrix code. If a Data Matrix code can not be located by the <a href="IMAQ Read Data Matrix Barcode 2">IMAQ Read Data Matrix Barcode 2</a> VI, the VI assigns the Data Matrix code an F for all grades.
  - Overall Grade is the overall letter grade, which is equal to the lowest of the other five letter grades.
  - **Decoding Grade** is the letter grade assigned to a Data Matrix code based on the success of the VI in decoding the Data Matrix code. The VI sets this grade to A if the VI could decode the Data Matrix code, otherwise the VI sets this grade to F.
  - Symbol Contrast Grade is the letter grade assigned to a Data Matrix code based on the symbol contrast raw score.
  - **Print Growth Grade** is the print growth letter grade for the Data Matrix code.
  - **Axial Nonuniformity Grade** is the axial nonuniformity grade for the Data Matrix code.
  - **Unused Error Correction Grade** is the unused error correction letter grade for the Data Matrix code.
- AIM Score Report is a cluster of information detailing the raw AIM scores for the Data Matrix code. If a Data Matrix code could not be located by the <a href="IMAQ Read Data Matrix Barcode 2">IMAQ Read Data Matrix Barcode 2</a> VI, the VI assigns a zero for all raw scores.
  - Symbol Contrast is the symbol contrast raw score representing the percentage difference between the mean of the reflectance of the darkest 10 percent and lightest 10 percent of the Data Matrix code.
  - **Print Growth** is the print growth raw score for the Data Matrix code, which is based on the extent to which dark or light markings appropriately fill their cell boundaries.
  - **Axial Nonuniformity** is the axial nonuniformity raw score for the Data Matrix code, which is based on how much the

sampling point spacing differs from one axis to another.

- Unused Error Correction is the unused error correction raw score for the Data Matrix code, which is based on the extent to which regional or spot damage in the Data Matrix code has eroded the reading safety margin provided by the error correction.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

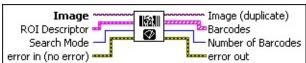
### **IMAQ Read PDF417 Barcode VI**

Owning Palette: Instrument ReadersInstalled With: NI Vision

**Development Module** 

Reads values encoded into a PDF417 barcode. You can compare the decoded data to a reference string or check whether the data contains a specific pattern.





- Image is a reference to the source image.
- ROI Descriptor is a descriptor that defines the region of interest within which the code is located. The first contour of the ROI must be a rectangle, rotated rectangle, oval, annulus, polygon, or freehand region. If the ROI descriptor is empty or not connected, the entire image is considered to be the region.
  - Global Rectangle contains the coordinates of the bounding rectangle.
  - **Contours** are each of the individual shapes that define an ROI.
    - **ID** refers to whether the contour is the external or internal edge of an ROI.
    - **Type** is the shape type of the contour.
    - **Coordinates** indicates the relative position of the contour.
- Search Mode specifies whether the VI searches for one or multiple codes. The following values are valid:

Multiple Barcodes (0)	Searches for multiple 2D barcodes.
Single Barcode,	Searches for 2D barcodes using the same
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	searching algorithm as <b>Multiple Barcodes</b> but
	stops searching after locating one valid barcode.

error in (no error) describes the error status before this VI or

function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Image (duplicate) is the reference to the image that contains the Data Matrix code.
- **Barcodes** is a cluster of information about each of the codes that the VI read.
  - **Type** is the type of code.
  - **Binary** indicates if the barcode contains encoded non-ASCII binary data (TRUE) or ASCII text (FALSE).
  - **Data** is the encoded information that the VI read.
  - **Bounding Box** is an array of five points that define the boundary of the code in the image.
  - Errors Corrected is the number of erasures that the VI corrected using inherent error correction.
  - **Erasures Corrected** is the number of erasures that the VI corrected using inherent error correction.
  - Rows is the number of rows in the barcode.

- **Columns** is the number of columns in the barcode.
- Number of Barcodes is the number of codes that the VI detected and read.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# Inspection

Owning Palette: <u>Machine Vision</u>Installed With: NI Vision Development Module

Use Inspection VIs to compare images to a golden template reference image.

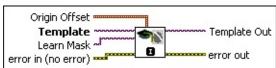
Palette Object	Description
IMAQ Learn Golden Template	Prepares an image for use in the <u>IMAQ</u> <u>Compare Golden Template</u> VI.
IMAQ Compare Golden Template	Compares an image to a template image at a given alignment.

## **IMAQ Learn Golden Template VI**

Owning Palette: <u>Inspection</u>Installed With: NI Vision Development Module

Prepares an image for use in the **IMAQ Compare Golden Template** VI.





- Origin Offset specifies the number of pixels the VI shifts the origin of the golden template from the center of the golden template. The default value is (0, 0), which sets the center of the golden template as the origin.
- **Template** is a reference to the golden template image you want to compare against during the golden template comparison phase.
- Learn Mask is an optional 8-bit image of the same size as the golden template that specifies which regions and edges to ignore when comparing images to a golden template. Use the following pixel values when constructing the mask:
  - 0—Maintains the default behavior.
  - **1**—The corresponding pixel in the golden template should always be ignored.
  - 2—The corresponding pixel in the golden template is an edge and should be dilated according to the value of the **Edge Thickness To Ignore** parameter of the **IMAQ**Compare Golden Template VI.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

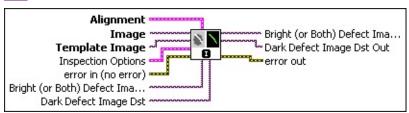
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **Template Out** is a reference to the learned golden template.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

## **IMAQ Compare Golden Template VI**

Owning Palette: <u>Inspection</u>Installed With: NI Vision Development Module

Compares an image to a template image at a given alignment.

U8



- Alignment specifies the area within the image that is compared to the golden template.
  - **Position** is the location of the center of the golden template in the image under inspection.
  - Angle is the rotation of the golden template in the image under inspection, in degrees.
  - **Scale** is the percentage of the size of the area under inspection compared to the size of the golden template.
- Image is a reference to the image to compare to the golden template. If neither Bright (or Both) Defect Image Dst Out or Dark Defect Image Dst Out is connected, Image is used as the destination image for both bright and dark defects.
- **Template Image** is a reference to the golden template image to compare against the image.
- Inspection Options is a cluster specifying the golden template comparison.
  - **Registration Method** specifies whether the VI registers the golden template to the image.
  - Normalization Method specifies how the VI normalizes the golden template to the image. The following values are valid:

None (0)	Specifies no normalization.
	Specifies that the histogram of the image is normalized to match the histogram of the

	template image.	
Average Matching (2)	Specifies that the mean pixel value of the image is normalized to match the mean pixel value of the template image.	

- Edge Thickness to Ignore specifies desired thickness of edges to be ignored. A value of 0 specifies that the algorithm will not ignore edges.
- Bright Threshold specifies the threshold for regions where the image is brighter than the golden template. Pixels that exceed the specified threshold value are assigned a pixel value of 2 in the Bright (or Both) Defect Image Dst Out image.
- Dark Threshold specifies the threshold for regions where the image is darker than the golden template. Pixels that exceed the specified threshold value are assigned a pixel value of 1 in the Dark Defect Image Dst Out or Bright (or Both) Defect Image Dst Out image.
- Grayscale/ Binary? specifies how you want to return the result of the golden template comparison. Choose from the following values:
  - Grayscale—Returns the grayscale image that results from subtracting the golden template image from the inspection image. If you select to return a grayscale image, values specified for Bright Threshold and/or Dark Threshold are ignored.
  - Binary—Returns the binary image that results from subtracting the golden template image from the inspection image, then thresholding the image according to the values specified by Bright Threshold and Dark Threshold.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error

out. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Bright (or Both) Defect Image Dst is a reference to the destination image for bright defects, or both kinds of defects if the same image is also connected to Dark Defect Image Dst.
- **Dark Defect Image Dst** is a reference to the destination image for dark defects.
- Bright (or Both) Defect Image Dst Out is a reference to the destination image for bright defects, or both kinds of defects if the same image is also connected to Dark Defect Image Dst.
- **Dark Defect Image Dst Out** is a reference to the destination image for dark defects.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.

# **Vision Express**

Use the Vision Express VIs to quickly develop common image acquisition and processing applications.

Palette Object	Description
Vision Assistant Express VI	Use NI Vision Assistant from within the LabVIEW environment to perform common image processing tasks.
	Acquires images from cameras using NI-IMAQ or NI-IMAQdx. Refer to the <i>NI Vision Acquisition Express VI Help</i> for more information.

## **Vision Assistant Express**

Creates, edits, and runs vision applications using NI Vision Assistant.

**Dialog Box Options** 

**Block Diagram Inputs** 

**Block Diagram Outputs** 

## **Dialog Box Options**

Refer to the *NI Vision Assistant Help* for information about specific functions.

## **Block Diagram Inputs**

Parameter	Description
Image Src	Is a reference to the source image.
Image Dst	Is a reference to the destination image. If <b>Image Dst</b> is connected, it must be the same type as the <b>Image Src</b> .
error in (no error)	Describes the error status before this VI or function runs. The default is <b>no error</b> . If an error occurred before this VI or function runs, the VI or function passes the <b>error in</b> value to <b>error out</b> . This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <b>Simple Error Handler</b> or <b>General Error Handler</b> VIs to display the description of the error code. Use <b>error in</b> and <b>error out</b> to check errors and to specify execution order by wiring error out from one node to error in of the next node.

## **Block Diagram Outputs**

Parameter	Description
Image Dst Out	Is a reference to the destination image. If <b>Image Dst Out</b> is connected, <b>Image Dst Out</b> is the same as <b>Image Dst</b> . Otherwise, <b>Image Dst Out</b> refers to the image referenced by <b>Image Src</b> .
error out	Contains error information. If <b>error in</b> indicates that an error occurred before this VI or function ran, <b>error out</b> contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the <b>error out</b> indicator on the front panel and select <b>Explain Error</b> from the shortcut menu for more information about the error.

### Kernels

Gradient Kernels

Laplacian Kernels

Smoothing Kernels

Gaussian Kernels

## **Predefined Gradient Kernels**

The following tables list the predefined gradient  $3 \times 3$  kernels.

#### **Prewitt Filters**

The Prewitt filters have the following kernels. The notations West (W), South (S), East (E), and North (N) indicate which edges of bright regions they outline.

#0 W/Edge	#1 W/Image	#2 SW/Edge	#3 SW/Image
-1 0 1	-1 0 1	011	011
-1 <b>0</b> 1	-1 <b>1</b> 1	-1 <b>0</b> 1	-1 <b>1</b> 1
-1 0 1	-1 0 1	-1-1 0	-1-1 0
#4 S/Edge	#5 S/Image	#6 SE/Edge	#7 SE/Image
111	111	110	110
000	010	1 0-1	1 <b>1</b> -1
-1-1-1	-1-1-1	0-1-1	0-1-1
#8 E/Edge	#9 E/Image	#10 NE/Edge	#11 NE/Image
1 0-1	1 0-1	0-1-1	0-1-1
1 <b>0</b> -1	1 <b>1</b> -1	1-0 1	1 <b>1</b> -1
1 0-1	1 0-1	110	110
#12 N/Edge	#13 N/Image	#14 NW/Edge	#15 NW/Image
-1-1-1	-1-1-1	-1-1 0	-1-1 0
000	010	-1 <b>0</b> 1	-1 <b>1</b> 1
111	111	011	011

#### **Sobel Filters**

The Sobel filters are very similar to the Prewitt filters except that they highlight light intensity variations along a particular axis that is assigned a stronger weight. The Sobel filters have the following kernels. The notations West (W), South (S), East (E), and North (N) indicate which edges of bright regions they outline.

	T		
#16 W/Edge	#17 W/Image	#18 SW/Edge	#19 SW/Image
-1 0 1	-1 0 1	012	012
-2 <b>0</b> 2	-2 <b>1</b> 2	-1 <b>0</b> 1	-1 <b>1</b> 1
-1 0 1	-1 0 1	-2-1 0	-2 1 0
#20 S/Edge	#21 S/Image	#22 SE/Edge	#23 SE/Image
121	121	210	210
0 <b>0</b> 0	0 <b>1</b> 0	1 <b>0</b> -1	1 <b>1</b> -1
-1-2-1	-1-2-1	0-1-2	0-1-2
#24 E/Edge	#25 E/Image	#26 NE/Edge	#27 NE/Image
1 0-1	1 0-1	0-1-2	0-1-2
2 <b>0</b> -2	2 <b>1</b> -2	1 <b>0</b> -1	1 <b>1</b> -1
1 0-1	1 0-1	210	210
#28 N/Edge	#29 N/Image	#30 NW/Edge	#31 NW/Image
-1-2-1	-1-2-1	-2-1 0	-2-1 0
000	010	-1 <b>0</b> 1	-1 <b>1</b> 1
121	121	012	012

#### **Other Kernels**

The following table lists the predefined gradient  $5 \times 5$  kernels.

#0 W/Edge	#1 W/Image	#2 SW/Edge	#3 SW/Image
0 -1 0 1 0	0-1 0 1 0	0 0 1 1 1	0 0 1 1 1
	-1 -2 0 2 1	0 0 2 2 1	0 0 2 2 1
-1 -2 <b>0</b> 2 1	-1 -2 <b>1</b> 2 1	-1 -2 <b>0</b> 2 1	-1 -2 <b>1</b> 2 1
	-1 -2 0 2 1	-1 -2 -2 0 0	-1 -2 -2 0 0
0-1 0 1 0	0-1 0 1 0	-1 -1 -1 0 0	-1 -1 -1 0 0
#4 S/Edge	#5 S/Image	#6 SE/Edge	#7 SE/Image
0 1 1 1 0	0 1 1 1 0	1 1 1 0 0	1 1 1 0 0
1 2 2 2 1	1 2 2 2 1	1 2 2 0 0	1 2 2 0 0
0 0 0 0 0	0 0 1 0 0	1 2 <b>0</b> -2 -1	1 2 <b>1</b> -2 -1
-1 -2 -2 -2 -1	-1 -2 -2 -2 -1	0 0 -2 -2 -1	0 0 -2 -2 -1
0 -1 -1 -1 0	0 -1 -1 -1 0	0 0 -1 -1 -1	0 0 -1 -1 -1
#8 E/Edge	#9 E/Image	#10 NE/Edge	#11 NE/Image
#8 E/Edge 0 1 0 -1 0	# <b>9 E/Image</b> 0 1 0 -1 0	# <b>10 NE/Edge</b> 0 0 -1 -1 -1	# <b>11 NE/Image</b> 0 0 -1 -1 -1
0 1 0 -1 0	0 1 0 -1 0	0 0 -1 -1 -1	0 0 -1 -1 -1
0 1 0 -1 0 1 2 0 -2 -1	0 1 0 -1 0 1 2 0 -2 -1	0 0 -1 -1 -1 0 0 -2 -2 -1	0 0 -1 -1 -1 0 0 -2 -2 -1
0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>0</b> -2 -1	0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>1</b> -2 -1	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>0</b> -2 -1	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>1</b> -2 -1
0 1 0 -1 0 1 2 0 -2 -1 1 2 0 -2 -1 1 2 0 -2 -1 0 1 0 -1 0	0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>1</b> -2 -1 1 2 0 -2 -1 0 1 0 -1 0	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>0</b> -2 -1 1 2 2 0 0 1 1 1 0 0	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>1</b> -2 -1 1 2 2 0 0
0 1 0 -1 0 1 2 0 -2 -1 1 2 0 -2 -1 1 2 0 -2 -1 0 1 0 -1 0 #12 N/Edge	0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>1</b> -2 -1 1 2 0 -2 -1 0 1 0 -1 0	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>0</b> -2 -1 1 2 2 0 0 1 1 1 0 0 # <b>14 NW/Edge</b>	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>1</b> -2 -1 1 2 2 0 0 1 1 1 0 0
0 1 0 -1 0 1 2 0 -2 -1 1 2 0 -2 -1 1 2 0 -2 -1 0 1 0 -1 0 #12 N/Edge 0 -1 -1 -1 0	0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>1</b> -2 -1 1 2 0 -2 -1 0 1 0 -1 0 # <b>13 N/Image</b>	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>0</b> -2 -1 1 2 2 0 0 1 1 1 0 0 # <b>14 NW/Edge</b>	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>1</b> -2 -1 1 2 2 0 0 1 1 1 0 0 # <b>15 NW/Image</b>
0 1 0 -1 0 1 2 0 -2 -1 1 2 0 -2 -1 1 2 0 -2 -1 0 1 0 -1 0 #12 N/Edge 0 -1 -1 -1 0	0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>1</b> -2 -1 1 2 0 -2 -1 0 1 0 -1 0 # <b>13 N/Image</b> 0 -1 -1 -1 0	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>0</b> -2 -1 1 2 2 0 0 1 1 1 0 0 # <b>14 NW/Edge</b> -1 -1 -1 0 0	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>1</b> -2 -1 1 2 2 0 0 1 1 1 0 0 # <b>15 NW/Image</b> -1 -1 -1 0 0
0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>0</b> -2 -1 1 2 0 -2 -1 0 1 0 -1 0 # <b>12 N/Edge</b> 0 -1 -1 -1 0 -1 -2 -2 -2 -1	0 1 0 -1 0 1 2 0 -2 -1 1 2 <b>1</b> -2 -1 1 2 0 -2 -1 0 1 0 -1 0 # <b>13 N/Image</b> 0 -1 -1 -1 0 -1 -2 -2 -2 -1	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>0</b> -2 -1 1 2 2 0 0 1 1 1 0 0 # <b>14 NW/Edge</b> -1 -1 -1 0 0 -1 -2 -2 0 0	0 0 -1 -1 -1 0 0 -2 -2 -1 1 2 <b>1</b> -2 -1 1 2 2 0 0 1 1 1 0 0 # <b>15 NW/Image</b> -1 -1 -1 0 0 -1 -2 -2 0 0

The following table lists the predefined gradient  $7 \times 7$  kernels.

#0 W/Edge	#1 W/Image	#2 S/Edge
0-1-1 0 1 1 0	0-1-1 0 1 1 0	0 1 1 1 1 1 0
-1 -2 -2 0 2 2 1	-1 -2 -2 0 2 2 1	1 2 2 2 2 2 1
-1 -2 -3 0 3 2 1	-1 -2 -3 0 3 2 1	1 2 3 3 3 2 1
	-1 -2 -3 <b>1</b> 3 2 1	
-1 -2 -3 0 3 2 1	-1 -2 -3 0 3 2 1	-1 -2 -3 -3 -3 -2 -1
-1 -2 -2 0 2 2 1	-1 -2 -2 0 2 2 1	-1 -2 -2 -2 -2 -1

0-1-1 0 1 1 0	0-1-1 0 1 1 0	0 -1 -1 -1 -1 0
#3 S/Image	#4 E/Edge	#5 E/Image
0 1 1 1 1 1 0 1 2 2 2 2 2 1 1 2 3 3 3 2 1 0 0 0 1 0 0 0 -1 -2 -3 -3 -3 -2 -1 -1 -2 -2 -2 -2 -2 -1 0 -1 -1 -1 -1 0	0 1 1 0 -1 -1 0 1 2 2 0 -2 -2 -1 1 2 3 0 -3 -2 -1 1 2 3 0 -3 -2 -1 1 2 3 0 -3 -2 -1 1 2 2 0 -2 -2 -1 0 1 1 0 -1 -1 0	0 1 1 0 -1 -1 0 1 2 2 0 -2 -2 -1 1 2 3 0 -3 -2 -1 1 2 3 1 -3 -2 -1 1 2 3 0 -3 -2 -1 1 2 2 0 -2 -2 -1 0 1 1 0 -1 -1 0
#6 N/Edge	#7 N/Image	
	0 -1 -1 -1 -1 -1 0 -1 -2 -2 -2 -2 -2 -1 -1 -2 -3 -3 -3 -2 -1 0 0 0 1 0 0 0 1 2 3 3 3 2 1 1 2 2 2 2 2 1 0 1 1 1 1 0	

# **Predefined Laplacian Kernels**

The following table lists the predefined Laplacian  $3 \times 3$  kernels.

#0 Contour 4	#1 +lmage×1	#2 +Image×2
0-1 0	0-1 0	0-1 0
-1 4 -1	-1 <b>5</b> -1	-1 6 -1
0 -1 0	0 -1 0	0 -1 0
#3 Contour 8	#4 +Image×1	#5 +Image×2
-1 -1 -1	-1 -1 -1	-1 -1 -1
-1 8 -1	-1 9 -1	-1 <b>10</b> -1
-1 -1 -1	-1 -1 -1	-1 -1 -1
#6 Contour 12	#7 +Image×1	
-1 -2 -1	-1 -2 -1	
-2 <b>12</b> -2	-2 <b>13</b> -2	
-1 -2 -1	-1 -2 -1	

The following table lists the predefined Laplacian  $5 \times 5$  kernels.

#0 Contour 24	#1 +lmage×1
-1 -1 -1 -1	-1 -1 -1 -1
-1 -1 -1 -1	-1 -1 -1 -1
-1 -1 <b>24</b> -1 -1	-1 -1 <b>25</b> -1 -1
-1 -1 -1 -1	-1 -1 -1 -1
-1 -1 -1 -1	-1 -1 -1 -1

The following table lists the predefined Laplacian  $7 \times 7$  kernels.

#0 Contour 48	#1 +Image×1
-1 -1 -1 -1 -1 -1	
-1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1	
-1 -1 -1 48 -1 -1 -1	-1 -1 -1 <b>49</b> -1 -1 -1
-1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1

# **Predefined Smoothing Kernels**

The following table lists the predefined smoothing  $3 \times 3$  kernels.

010	010	020	0 4 0
101	1 <b>1</b> 1	2 <b>1</b> 2	4 <b>1</b> 4
010	010	020	0 4 0
111	111	222	4 4 4
101	1 <b>1</b> 1	2 <b>1</b> 2	4 <b>1</b> 4
111	111	2 2 2	4 4 4

The following table lists the predefined smoothing  $5 \times 5$  kernels.

The following table lists the predefined smoothing  $7 \times 7$  kernels.

### **Predefined Gaussian Kernels**

The following table lists the predefined Gaussian  $3 \times 3$  kernels.

010	010	111
1 <b>2</b> 1	141	1 <b>2</b> 1
010	010	111
111	121	1 4 1
141	2 <b>4</b> 2	4 <b>16</b> 4
111	121	1 4 1

The following table lists the predefined Gaussian  $5 \times 5$  kernel.

The following table lists the predefined Gaussian  $7 \times 7$  kernel.

#### **Time-Bounded Execution**

The following tables list the NI Vision VIs that support time-bounded execution. Keep the following in mind when developing time-bounded sections of the application.

- Always resize destination images to their proper size before beginning timed execution unless otherwise noted in the endnotes.
- Do not perform array operations that resize the data. These operations request resources from the system, cause excessive jitter, and invalidate timing.

# **VIs that Support Time-Bounded Execution**

The following VIs support time-bounded execution.

IMAQ Absolute	IMAQ FillImage	IMAQ Or
<u>Difference</u>	IMAQ	IMAQ Particle Filter
IMAQ Add	<u>GetColorPixelValue</u>	<u>3</u> 1
IMAQ And	IMAQ GetImageInfo <sup>3</sup>	IMAQ Perpendicular
<u>IMAQ</u>	IMAQ GetImageSize	<u>Line</u>
<u>ArrayToColorImage</u>	IMAQ GetOffset	IMAQ Polygon Area
IMAQ ArrayTolmage	IMAQ GetPixelValue	<u>IMAQ</u>
IMAQ BCGLookup	IMAQ	<u>ReplaceColorPlane</u>
IMAQ Bisecting Line	<u>ImageBorderOperation</u>	IMAQ Resample
IMAQ Build CoordSys	<u>IMAQ</u>	IMAQ ROIToMask <sup>4</sup>
(Points)	<u>ImageBorderSize</u> <sup>2</sup>	<b>IMAQ Rotate</b>
IMAQ Centroid	IMAQ ImageToImage	<b>IMAQ Rotation</b>
IMAQ ColorBCGLookup	IMAQ Inverse	Detect
IMAQ ColorThreshold	IMAQ Is Vision Info	IMAQ
IMAQ ColorToRGB	Present 2	SetColorPixelLine 2
IMAQ ColorUserLookup	IMAQ Label <sup>1</sup>	IMAQ SetColorPixelValue
IMAQ Compare	IMAQ Lines	IMAQ SetOffset
IMAQ Convex Hull <sup>1</sup>	Intersection	_
IMAQ Convolute	IMAQ LogDiff	IMAQ SetPixelLine
IMAQ Divide 2	IMAQ LowPass	IMAQ SetPixelValue
IMAQ EdgeDetection	IMAQ MagicWand	IMAQ SetRowCol
	IMAQ Mask	IMAQ Shift
IMAQ Equalize	IMAQ MathLookup	IMAQ Subtract
IMAQ Expand	IMAQ Mid Line	<b>IMAQ Symmetry</b>
IMAQ Extract	IMAQ Modulo	<b>IMAQ Threshold</b>
IMAQ FytractCalarDlanca	_	IMAQ UserLookup
ExtractColorPlanes	IMAQ MulDiv	IMAQ Xor
<u>IMAQ</u>	IMAQ Multiply	

### ExtractSingleColorPlane IMAQ MultiThreshold

- 1. The destination image must be the source image.
- 2. Only the **Get** option is supported.
- 3. The **Image Name** indicator is not supported.
- 4. The **Image Model** control is required.

#### **Preallocated VIs**

The following VIs have preallocated versions in Preallocated.llb that support time-bounded execution. Use the preallocated version of these VIs in code that requires time-bounded execution.



**Note** Before you save, download, and run these VIs or any timed execution VIs on your RT target, run the Configure LabVIEW for Timed Execution VI (vi.lib\vision\preallocated.llb) at least once on your host machine and close LabVIEW completely. If you do not run the Configure LabVIEW for Timed Execution VI or forget to close LabVIEW completely, LabVIEW may prompt you to resave some of the VIs in preallocated.llb with incorrect settings.

IMAQ AutoMThreshold	<u>IMAQ</u>	IMAQ LineProfile
IMAQ BuildKernel	GetColorPixelLine	IMAQ MaskToROI
IMAQ Caliper Tool	IMAQ GetKernel	IMAQ Match Pattern 2
IMAQ ColorHistogram	IMAQ GetPalette	IMAQ Particle
IMAQ ColorHistograph	IMAQ GetPixelLine	Analysis
IMAQ ColorImageToArray	IMAQ GetPointsOnLine	IMAQ Particle Analysis Report
IMAQ ColorMatch <sup>2</sup>	IMAQ GetRowCol	IMAQ Peak-Valley
<u>IMAQ</u>	IMAQ Histogram	Detector
ColorValueToInteger <sup>1</sup>	IMAQ Histograph	<b>IMAQ Point Distances</b>
IMAQ Convert Pixel To Real World	IMAQ ImageToArray	<b>IMAQ Quantify</b>
IMAQ Convert Real World	IMAQ	<b>IMAQ Refine Matches</b>
To Pixel	IntegerToColorValue <sup>1</sup>	IMAQ ROIProfile
IMAQ Fit Circle 2	IMAQ Interpolate 1D	IMAQ Simple Edge
IMAQ Fit Line	IMAQ Line Gauge	IMAQ TransformROI
IMAQ Get Angles	IMAQ LinearAverages	

- 1. 2D versions show excessive jitter.
- 2. The **ROI Descriptor** is not supported.

## **NI Vision Error Codes**

NI Vision VIs and functions can return the following error codes.

Code	Description
0	No error.
-1074396160	System error.
-1074396159	Not enough memory for requested operation.
-1074396158	Memory error.
-1074396157	Unlicensed copy of NI Vision.
-1074396156	The function requires an NI Vision 5.0 Advanced license.
-1074396155	NI Vision did not initialize properly.
-1074396154	The image is not large enough for the operation.
-1074396153	The barcode is not a valid Codabar barcode.
-1074396152	The barcode is not a valid Code 3 of 9 barcode.
-1074396151	The barcode is not a valid Code93 barcode.
-1074396150	The barcode is not a valid Code128 barcode.
-1074396149	The barcode is not a valid EAN8 barcode.
-1074396148	The barcode is not a valid EAN13 barcode.
-1074396147	The barcode is not a valid Interleaved 2 of 5 barcode.
-1074396146	The barcode is not a valid MSI barcode.
-1074396145	The barcode is not a valid UPCA barcode.
-1074396144	The Code93 barcode contains invalid shift encoding.
-1074396143	The barcode type is invalid.
-1074396142	The image does not represent a valid linear barcode.
-1074396141	The FNC value in the Code128 barcode is not located before the first data value.
-1074396140	The starting code set in the Code128 barcode is not valid.
-1074396139	Not enough reserved memory in the timed environment for the requested operation.
-1074396138	The function is not supported when a time limit is active.
-1074396137	Quartz.dll not found. Install DirectX 8.1 or later.

Í.	,
-1074396136	The filter quality you provided is invalid. Valid quality values range from -1 to 1000.
-1074396135	Invalid button label.
-1074396134	Could not execute the function in the separate thread because the thread has not completed initialization.
-1074396133	Could not execute the function in the separate thread because the thread could not initialize.
-1074396132	The mask must be the same size as the template.
-1074396130	The ROI must only have either a single Rectangle contour or a single Rotated Rectangle contour.
-1074396129	During timed execution, you must use the preallocated version of this operation.
-1074396128	An image being modified by one process cannot be requested by another process while a time limit is active.
-1074396127	An image with pattern matching, calibration, or overlay information cannot be manipulated while a time limit is active.
-1074396126	An image created before a time limit is started cannot be resized while a time limit is active.
-1074396125	Invalid contrast threshold. The threshold value must be greater than 0.
-1074396124	NI Vision does not support the calibration ROI mode you supplied.
-1074396123	NI Vision does not support the calibration mode you supplied.
-1074396122	Set the foreground and background text colors to grayscale to draw on a U8 image.
-1074396121	The value of the saturation threshold must be from 0 to 255.
-1074396120	Not an image.
-1074396119	They custom data key you supplied is invalid. The only valid character values are decimal 32-126 and 161-255. There must also be no repeated, leading, or trailing

	spaces.
-1074396118	Step size must be greater than zero.
-1074396117	Invalid matrix size in the structuring element.
-1074396116	Insufficient number of calibration feature points.
-1074396115	The operation is invalid in a corrected image.
-1074396114	The ROI contains an invalid contour type or is not contained in the ROI learned for calibration.
-1074396113	The source/input image has not been calibrated.
-1074396112	The number of pixel and real-world coordinates must be equal.
-1074396111	Unable to automatically detect grid because the image is too distorted.
-1074396110	Invalid calibration information version.
-1074396109	Invalid calibration scaling factor.
-1074396108	The calibration error map cannot be computed.
-1074396107	Invalid calibration template image.
-1074396106	Invalid calibration template image.
-1074396105	Invalid calibration template image.
-1074396104	Invalid calibration template image.
-1074396103	Invalid calibration template image.
-1074396102	Invalid calibration template image.
-1074396101	Invalid calibration template image.
-1074396100	Invalid calibration template image.
-1074396099	Invalid calibration template image.
-1074396098	You must pass NULL for the reserved parameter.
-1074396097	You entered an invalid selection in the particle parameter
-1074396096	Not an object.
-1074396095	The reference points passed are inconsistent. At least tw similar pixel coordinates correspond to different real-worl coordinates.
-1074396094	A resource conflict occurred in the timed environment.

	Two processes cannot manage the same resource and be time bounded.
	A resource conflict occurred in the timed environment. Two processes cannot access the same resource and be time bounded.
-1074396092	Multiple timed environments are not supported.
	A time limit cannot be started until the timed environment is initialized.
-1074396090	Multiple timed environments are not supported.
-1074396089	The timed environment is already initialized.
	The results of the operation exceeded the size limits on the output data arrays.
-1074396087	No time limit is available to stop.
-1074396086	A time limit could not be set.
-1074396085	The timed environment could not be initialized.
-1074396084	No initialized timed environment is available to close.
-1074396083	The time limit has expired.
	Only 8-bit images support the use of palettes. Either do not use a palette, or convert your image to an 8-bit image before using a palette.
-1074396081	Incorrect password.
-1074396080	Invalid image type.
-1074396079	Invalid metafile handle.
-1074396077	Incompatible image type.
-1074396076	Unable to fit a line for the primary axis.
-1074396075	Unable to fit a line for the secondary axis.
-1074396074	Incompatible image size.
	When the mask's offset was applied, the mask was entirely outside of the image.
-1074396072	Invalid image border.
-1074396071	Invalid scan direction.
-1074396070	Unsupported function.

-1074396069	NI Vision does not support the color mode you specified.
-1074396068	The function does not support the requested action.
-1074396067	The source image and destination image must be different.
-1074396066	Invalid point symbol.
-1074396065	Cannot resize an image in an acquisition buffer.
-1074396064	This operation is not supported for images in an acquisition buffer.
-1074396063	The external buffer must be aligned on a 4-byte boundary. The line width and border pixels must be 4-byte aligned, as well.
-1074396062	The tolerance parameter must be greater than or equal to 0.
-1074396061	The size of each dimension of the window must be greater than 2 and less than or equal to the size of the image in the corresponding dimension.
-1074396060	Lossless compression cannot be used with the floating point wavelet transform mode. Either set the wavelet transform mode to integer, or use lossy compression.
-1074396059	Invalid maximum number of iterations. Maximum number of iterations must be greater than zero.
-1074396058	Invalid rotation mode.
-1074396057	Invalid search vector width. The width must be an odd number greater than zero.
-1074396056	Invalid matrix mirror mode.
-1074396055	Invalid aspect ratio. Valid aspect ratios must be greater than or equal to zero.
-1074396054	Invalid cell fill type.
-1074396053	Invalid border integrity. Valid values range from 0 to 100.
-1074396052	Invalid demodulation mode.
-1074396051	Invalid cell filter mode.
-1074396050	Invalid ECC type.
I	

-1074396049	Invalid matrix polarity.
-1074396048	Invalid cell sample size.
-1074396047	Invalid linear average mode.
-1074396046	When using a region of interest that is not a rectangle, you must specify the contrast mode of the barcode as either black on white or white on black.
-1074396045	Invalid 2-D barcode Data Matrix subtype.
-1074396044	Invalid 2-D barcode shape.
-1074396043	Invalid 2-D barcode cell shape.
-1074396042	Invalid 2-D barcode contrast.
-1074396041	Invalid 2-D barcode type.
-1074396040	Cannot access NI-IMAQ driver.
-1074396039	I/O error.
-1074396038	When searching for a coordinate system, the number of lines to fit must be 1.
-1074396037	Trigger timeout.
-1074396036	The Skeleton mode you specified is invalid.
-1074396035	The template image does not contain enough information for learning the aggressive search strategy.
-1074396034	The template image does not contain enough edge information for the sample size(s) requested.
-1074396033	Invalid template descriptor.
-1074396032	The template descriptor does not contain data required for the requested search strategy in rotation-invariant matching.
-1074396026	Invalid process type for edge detection.
-1074396025	Angle range value should be equal to or greater than zero.
-1074396024	Minimum coverage value should be greater than zero.
-1074396023	The angle tolerance should be equal to or greater than 0.001.
-1074396022	Invalid search mode for detecting straight edges

-1074396021	Invalid kernel size for edge detection. The minimum kernel size is 3, the maximum kernel size is 1073741823 and the kernel size must be odd.
-1074396020	Invalid grading mode.
-1074396019	Invalid threshold percentage. Valid values range from 0 to 100.
-1074396018	Invalid edge polarity search mode.
-1074396017	The AIM grading data attached to the image you tried to open was created with a newer version of NI Vision. Upgrade to the latest version of NI Vision to read this file.
-1074396016	No video driver is installed.
-1074396015	Unable to establish network connection with remote system.
-1074396014	RT Video Out does not support displaying the supplied image type at the selected color depth.
-1074396013	Invalid video mode.
-1074396012	Unable to display remote image on network connection.
-1074396011	Unable to establish network connection.
-1074396010	Invalid frame number.
-1074396009	An internal DirectX error has occurred. Try upgrading to the latest version of DirectX.
-1074396008	An appropriate DirectX filter to process this file could not be found. Install the filter that was used to create this AVI. Upgrading to the latest version of DirectX may correct this error. NI Vision requires DirectX 8.1 or higher.
-1074396007	Incompatible compression filter.
-1074396006	Unknown compression filter.
-1074396005	Invalid AVI session.
-1074396004	A software key is restricting the use of this compression filter.
-1074396003	The data for this frame exceeds the data buffer size specified when creating the AVI file.
-1074396002	Invalid line gauge method.

-1074396001	There are too many AVI sessions open. You must close a session before you can open another one.
-1074396000	Invalid file header.
-1074395999	Invalid file type.
-1074395998	Invalid color table.
-1074395997	Invalid parameter.
-1074395996	File is already open for writing.
-1074395995	File not found.
-1074395994	Too many files open.
-1074395993	File I/O error.
-1074395992	File access denied.
-1074395991	NI Vision does not support the file type you specified.
-1074395990	Could not read Vision info from file.
-1074395989	Unable to read data.
-1074395988	Unable to write data.
-1074395987	Premature end of file.
-1074395986	Invalid file format.
-1074395985	Invalid file operation.
-1074395984	NI Vision does not support the file data type you specified.
-1074395983	Disk full.
-1074395982	The frames per second in an AVI must be greater than zero.
-1074395981	The buffer that was passed in is not big enough to hold all of the data.
-1074395980	Error initializing COM.
-1074395979	The image has invalid particle information. Call imagCountParticles on the image to create particle information.
-1074395978	Invalid particle number.
-1074395977	The AVI file was created in a newer version of NI Vision. Upgrade to the latest version of NI Vision to read this AVI

	file.
-1074395976	The color palette must have exactly 0 or 256 entries.
-1074395975	DirectX has timed out reading or writing the AVI file. When closing an AVI file, try adding an additional delay. When reading an AVI file, try reducing CPU and disk load.
-1074395974	NI Vision does not support reading JPEG2000 files with this colorspace method.
-1074395973	NI Vision does not support reading JPEG2000 files with more than one layer.
-1074395972	DirectX is unable to enumerate the compression filters. This is caused by a third-party compression filter that is either improperly installed or is preventing itself from being enumerated. Remove any recently installed compression filters and try again.
-1074395971	The offset you specified must be size 2.
-1074395960	Initialization error.
-1074395959	Unable to create window.
-1074395958	Invalid window ID.
-1074395957	The array sizes are not compatible.
	The quality you provided is invalid. Valid quality values range from -1 to 1000.
-1074395955	Invalid maximum wavelet transform level. Valid values range from 0 to 255.
	The quantization step size must be greater than or equal to 0.
-1074395953	Invalid wavelet transform mode.
-1074395920	Invalid number of classes.
-1074395880	Invalid particle.
-1074395879	Invalid measure number.
-1074395878	The Image Display control does not support writing this property node.
-1074395877	The specified color mode requires the use of imaqChangeColorSpace2.

-1074395876	This function does not currently support the color mode you specified.
-1074395875	The barcode is not a valid Pharmacode symbol
-1074395840	Invalid handle table index.
-1074395837	The compression ratio must be greater than or equal to 1.
-1074395801	The ROI contains too many contours.
-1074395800	Protection error.
-1074395799	Internal error.
-1074395798	The size of the feature vector in the custom sample must match the size of those you have already added.
-1074395797	Not a valid classifier session.
-1074395796	You requested an invalid Nearest Neighbor classifier method.
-1074395795	The k parameter must be greater than two.
-1074395794	The k parameter must be <= the number of samples in each class.
-1074395793	This classifier session is compact. Only the Classify and Dispose functions may be called on a compact classifier session.
-1074395792	This classifier session is not trained. You may only call this function on a trained classifier session.
-1074395791	This classifier function cannot be called on this type of classifier session.
-1074395790	You requested an invalid distance metric.
-1074395789	The classifier session you tried to open was created with a newer version of NI Vision. Upgrade to the latest version of NI Vision to read this file.
-1074395788	This operation cannot be performed because you have not added any samples.
-1074395787	You requested an invalid classifier type.
-1074395786	The sum of Scale Dependence and Symmetry Dependence must be less than 1000.
-1074395785	The image yielded no particles.

1	
-1074395784	The limits you supplied are not valid.
-1074395783	The Sample Index fell outside the range of Samples.
-1074395782	The description must be <= 255 characters.
-1074395781	The engine for this classifier session does not support this operation.
-1074395780	You requested an invalid particle type.
-1074395779	You may only save a session in compact form if it is trained.
-1074395778	The Kernel size must be smaller than the image size.
-1074395777	The session you read from file must be the same type as the session you passed in.
-1074395776	You can not use a compact classification file with read options other than Read All.
-1074395775	The ROI you passed in may only contain closed contours.
-1074395774	You must pass in a label.
-1074395773	You must provide a destination image.
-1074395772	You provided an invalid registration method.
-1074395771	The golden template you tried to open was created with a newer version of NI Vision. Upgrade to the latest version of NI Vision to read this file.
-1074395770	Invalid golden template.
-1074395769	Edge Thickness to Ignore must be greater than zero.
-1074395768	Scale must be greater than zero.
-1074395767	The supplied scale is invalid for your template.
-1074395766	This backwards-compatibility function can not be used with this session. Use newer, supported functions instead.
-1074395763	You must provide a valid normalization method.
-1074395762	The deviation factor for Niblack local threshold must be between 0 and 1.
-1074395760	Board not found.
-1074395758	Board not opened.

-1074395757	DLL not found.
-1074395756	DLL function not found.
-1074395754	Trigger timeout.
-1074395728	NI Vision does not support the search mode you provided.
-1074395727	NI Vision does not support the search mode you provided for the type of 2D barcode for which you are searching.
-1074395726	matchFactor has been obsoleted. Instead, set the initialMatchListLength and matchListReductionFactor in the MatchPatternAdvancedOptions structure.
-1074395725	The data was stored with a newer version of NI Vision. Upgrade to the latest version of NI Vision to read this data.
-1074395724	The size you specified is out of the valid range.
-1074395723	The key you specified cannot be found in the image.
-1074395722	Custom classifier sessions only classify feature vectors. They do not support classifying images.
-1074395721	NI Vision does not support the bit depth you supplied for the image you supplied.
-1074395720	Invalid ROI.
-1074395719	Invalid ROI global rectangle.
-1074395718	The version of LabVIEW or BridgeVIEW you are running does not support this operation.
-1074395717	The range you supplied is invalid.
-1074395716	NI Vision does not support the scaling method you provided.
-1074395715	NI Vision does not support the calibration unit you supplied.
-1074395714	NI Vision does not support the axis orientation you supplied.
-1074395713	Value not in enumeration.
-1074395712	You selected a region that is not of the right type.
-1074395711	You specified a viewer that does not contain enough

	regions.
-1074395710	The image has too many particles for this process.
-1074395709	The AVI session has not been opened.
-1074395708	The AVI session is a write session, but this operation requires a read session.
-1074395707	The AVI session is a read session, but this operation requires a write session.
-1074395706	This AVI session is already open. You must close it before calling the Create or Open functions.
-1074395705	The data is corrupted and cannot be read.
-1074395704	Invalid compression type.
-1074395703	Invalid type of flatten.
-1074395702	The length of the edge detection line must be greater than zero.
-1074395701	The maximum Data Matrix barcode size must be equal to or greater than the minimum Data Matrix barcode size.
-1074395700	The function requires the operating system to be Microsoft Windows 2000 or newer.
-1074395656	You must specify the same value for the smooth contours advanced match option for all templates you want to match.
-1074395655	You must specify the same value for the enable calibration support advanced match option for all templates you want to match.
-1074395654	The source image does not contain grading information. You must prepare the source image for grading when reading the Data Matrix, and you cannot change the contents of the source image between reading and grading the Data Matrix.
-1074395653	The multiple geometric matching template you tried to open was created with a newer version of NI Vision. Upgrade to the latest version of NI Vision to read this file.
-1074395652	The geometric matching template you tried to open was created with a newer version of NI Vision. Upgrade to the

	latest version of NI Vision to read this file.
-1074395651	You must specify the same edge filter size for all the templates you want to match.
-1074395650	You must specify the same curve extraction mode for all the templates you want to match.
-1074395649	The geometric feature type specified is invalid.
-1074395648	You supplied a template that was not learned.
-1074395647	Invalid multiple geometric template.
-1074395646	Need at least one template to learn.
-1074395645	You supplied an invalid number of labels.
-1074395644	Labels must be <= 255 characters.
-1074395643	You supplied an invalid number of match options.
-1074395642	Cannot find a label that matches the one you specified.
-1074395641	Duplicate labels are not allowed.
-1074395640	The number of zones found exceeded the capacity of the algorithm.
-1074395639	The hatch style for the window background is invalid.
-1074395638	The fill style for the window background is invalid.
-1074395637	Your hardware is not supported by DirectX and cannot be put into NonTearing mode.
-1074395636	DirectX is required for this feature. Please install the latest version
-1074395635	The passed shape descriptor is invalid.
-1074395634	Invalid max match overlap. Values must be between -1 and 100.
-1074395633	Invalid minimum match separation scale. Values must be greater than or equal to -1.
-1074395632	Invalid minimum match separation angle. Values must be between -1 and 360.
-1074395631	Invalid minimum match separation distance. Values must be greater than or equal to -1.

-1074395630	Invalid maximum number of features learn. Values must be integers greater than zero.
-1074395629	Invalid maximum pixel distance from line. Values must be positive real numbers.
-1074395628	Invalid geometric matching template image.
-1074395627	The template does not contain enough features for geometric matching.
-1074395626	The template does not contain enough features for geometric matching.
-1074395625	You specified an invalid value for the match constraint value of the range settings.
-1074395624	Invalid occlusion range. Valid values for the bounds range from 0 to 100 and the upper bound must be greater than or equal to the lower bound.
-1074395623	Invalid scale range. Values for the lower bound must be a positive real numbers and the upper bound must be greater than or equal to the lower bound.
-1074395622	Invalid match geometric pattern setup data.
-1074395621	Invalid learn geometric pattern setup data.
-1074395620	Invalid curve extraction mode.
-1074395619	You can specify only one occlusion range.
-1074395618	You can specify only one scale range.
-1074395617	The minimum number of features must be less than or equal to the maximum number of features.
-1074395616	Invalid edge filter size.
-1074395615	Invalid minimum strength for features. Values must be positive real numbers.
-1074395614	Invalid aspect ratio for rectangular features. Values must be positive real numbers in the range 0.01 to 1.0.
-1074395613	Invalid minimum length for linear features. Values must be integers greater than 0.
-1074395612	Invalid minimum radius for circular features. Values must be integers greater than 0.

-1074395611	Invalid minimum rectangle dimension. Values must be integers greater than 0.
-1074395610	Invalid initial match list length. Values must be integers greater than 5.
-1074395609	Invalid subpixel tolerance. Values must be positive real numbers.
-1074395608	Invalid number of subpixel iterations. Values must be integers greater 10.
-1074395607	Invalid maximum number of features used per match. Values must be integers greater than or equal to zero.
-1074395606	Invalid minimum number of features used for matching. Values must be integers greater than zero.
-1074395605	Invalid maximum end point gap. Valid values range from 0 to 32767.
-1074395604	Invalid column step. Valid range is 1 to 255.
-1074395603	Invalid row step. Valid range is 1 to 255.
-1074395602	Invalid minimum length. Valid values must be greater than or equal to zero.
-1074395601	Invalid edge threshold. Valid values range from 1 to 360.
-1074395600	You must provide information about the subimage within the browser.
-1074395598	The acceptance level is outside the valid range of 0 to 1000.
-1074395597	Not a valid OCR session.
-1074395596	Invalid character size. Character size must be >= 1.
-1074395595	Invalid threshold mode value.
-1074395594	Invalid substitution character. Valid substitution characters are ASCII values that range from 1 to 254.
-1074395593	Invalid number of blocks. Number of blocks must be >= 4 and <= 50.
-1074395592	Invalid read strategy.
-1074395591	Invalid character index.

-1074395590	Invalid number of character positions. Valid values range from 0 to 255.
-1074395589	Invalid low threshold value. Valid threshold values range from 0 to 255.
-1074395588	Invalid high threshold value. Valid threshold values range from 0 to 255.
-1074395587	The low threshold must be less than the high threshold.
-1074395586	Invalid lower threshold limit. Valid lower threshold limits range from 0 to 255.
-1074395585	Invalid upper threshold limit. Valid upper threshold limits range from 0 to 255.
-1074395584	The lower threshold limit must be less than the upper threshold limit.
-1074395583	Invalid minimum character spacing value. Character spacing must be >= 0 pixels.
-1074395582	Invalid maximum horizontal element spacing value.  Maximum horizontal element spacing must be >= 0.
-1074395581	Invalid maximum vertical element spacing value.  Maximum vertical element spacing must be >= 0.
-1074395580	Invalid minimum bounding rectangle width. Minimum bounding rectangle width must be >= 1.
-1074395579	Invalid aspect ratio value. The aspect ratio must be zero or >= 100.
-1074395578	Invalid or corrupt character set file.
-1074395577	The character value must not be an empty string.
-1074395576	Character values must be <=255 characters.
-1074395575	Invalid number of erosions. The number of erosions must be $\geq 0$ .
-1074395574	The character set description must be <=255 characters.
-1074395573	The character set file was created by a newer version of NI Vision. Upgrade to the latest version of NI Vision to read the character set file.
-1074395572	You must specify characters for a string. A string cannot

	contain integers.
-1074395571	This attribute is read-only.
-1074395570	This attribute requires a Boolean value.
-1074395569	Invalid attribute.
-1074395568	This attribute requires integer values.
-1074395567	String values are invalid for this attribute. Enter a boolear value.
-1074395566	Boolean values are not valid for this attribute. Enter an integer value.
-1074395565	Requires a single-character string.
-1074395564	Invalid predefined character value.
-1074395563	This copy of NI OCR is unlicensed.
-1074395562	String values are not valid for this attribute. Enter a Boolean value.
-1074395561	The number of characters in the character value must match the number of objects in the image.
-1074395560	Invalid object index.
-1074395559	Invalid read option.
-1074395558	The minimum character size must be less than the maximum character size.
-1074395557	The minimum character bounding rectangle width must b less than the maximum character bounding rectangle width.
-1074395556	The minimum character bounding rectangle height must be less than the maximum character bounding rectangle height.
-1074395555	The maximum horizontal element spacing value must not exceed the minimum character spacing value.
-1074395554	Invalid read resolution.
-1074395553	Invalid minimum bounding rectangle height. The minimur bounding rectangle height must be >= 1.
-1074395552	Not a valid character set.

-1074395551	A trained OCR character cannot be renamed while it is a reference character.
-1074395550	A character cannot have an ASCII value of 255.
-1074395549	The number of objects found does not match the number of expected characters or patterns to verify.
-1074395410	NI Vision does not support less than one icon per line.
-1074395409	Invalid subpixel divisions.
-1074395408	Invalid detection mode.
-1074395407	Invalid contrast value. Valid contrast values range from 0 to 255.
-1074395406	The coordinate system could not be found on this image.
-1074395405	NI Vision does not support the text orientation value you supplied.
-1074395404	UnwrapImage does not support the interpolation method value you supplied. Valid interpolation methods are zero order and bilinear.
-1074395403	The image was created in a newer version of NI Vision. Upgrade to the latest version of NI Vision to use this image.
-1074395402	The function does not support the maximum number of points that you specified.
-1074395401	The function does not support the matchFactor that you specified.
-1074395400	The operation you have given Multicore Options is invalid. Please see the available enumeration values for Multicore Operation.
-1074395399	You have given Multicore Options an invalid argument.
-1074395397	A complex image is required.
-1074395395	The input image must be a color image.
-1074395394	The color mask removes too much color information.
-1074395393	The color template image is too small.
-1074395392	The color template image is too large.

-1074395391	The contrast in the hue plane of the image is too low for learning shape features.
-1074395390	The contrast in the luminance plane of the image is too low to learn shape features.
-1074395389	Invalid color learn setup data.
-1074395388	Invalid color learn setup data.
-1074395387	Invalid color match setup data.
-1074395386	Invalid color match setup data.
-1074395385	Rotation-invariant color pattern matching requires a feature mode including shape.
-1074395384	Invalid color template image.
-1074395383	Invalid color template image.
-1074395382	Invalid color template image.
-1074395381	Invalid color template image.
-1074395380	Invalid color template image.
-1074395379	Invalid color template image.
-1074395378	Invalid color template image.
-1074395377	Invalid color template image.
-1074395376	The color template image does not contain data required for shift-invariant color matching.
-1074395375	Invalid color template image.
-1074395374	Invalid color template image.
-1074395373	Invalid color template image.
-1074395372	The color template image does not contain data required for rotation-invariant color matching.
-1074395371	Invalid color template image.
-1074395370	Invalid color template image.
-1074395369	Invalid color template image.
-1074395368	Invalid color template image.
-1074395367	Invalid color template image.

-1074395366	The color template image does not contain data required for color matching in shape feature mode.
-1074395365	The color template image does not contain data required for color matching in color feature mode.
-1074395364	The ignore color spectra array is invalid.
-1074395363	Invalid subsampling ratio.
-1074395362	Invalid pixel width.
-1074395361	Invalid steepness.
-1074395360	Invalid complex plane.
-1074395357	Invalid color ignore mode.
-1074395356	Invalid minimum match score. Acceptable values range from 0 to 1000.
-1074395355	Invalid number of matches requested. You must request a minimum of one match.
-1074395354	Invalid color weight. Acceptable values range from 0 to 1000.
-1074395353	Invalid search strategy.
-1074395352	Invalid feature mode.
-1074395351	NI Vision does not support rectangles with negative widths or negative heights.
-1074395350	NI Vision does not support the vision information type you supplied.
-1074395349	NI Vision does not support the SkeletonMethod value you supplied.
-1074395348	NI Vision does not support the 3DPlane value you supplied.
-1074395347	NI Vision does not support the 3DDirection value you supplied.
-1074395346	imaqRotate does not support the InterpolationMethod value you supplied.
-1074395345	NI Vision does not support the axis of symmetry you supplied.
-1074395343	You must pass a valid file name. Do not pass in NULL.

-1074395340	NI Vision does not support the SizeType value you supplied.
-1074395336	You specified the dispatch status of an unknown algorithm.
-1074395335	You are attempting to set the same algorithm to dispatch and to not dispatch. Remove one of the conflicting settings.
	NI Vision does not support the Conversion Method value you supplied.
-1074395333	NI Vision does not support the VerticalTextAlignment value you supplied.
	NI Vision does not support the CompareFunction value you supplied.
-1074395331	NI Vision does not support the BorderMethod value you supplied.
-1074395330	Invalid border size. Acceptable values range from 0 to 5
-1074395329	NI Vision does not support the OutlineMethod value you supplied.
-1074395328	NI Vision does not support the InterpolationMethod value you supplied.
-1074395327	NI Vision does not support the ScalingMode value you supplied.
-1074395326	imaqDrawLineOnImage does not support the DrawMode value you supplied.
-1074395325	NI Vision does not support the DrawMode value you supplied.
-1074395324	NI Vision does not support the ShapeMode value you supplied.
-1074395323	NI Vision does not support the FontColor value you supplied.
-1074395322	NI Vision does not support the TextAlignment value you supplied.
-1074395321	NI Vision does not support the MorphologyMethod value

	you supplied.
-1074395320	The template image is empty.
-1074395319	NI Vision does not support the interpolation type you supplied.
-1074395318	You supplied an insufficient number of points to perform this operation.
-1074395317	You specified a point that lies outside the image.
-1074395316	Invalid kernel code.
-1074395313	Writing files is not supported on this device.
-1074395312	The input image does not seem to be a valid LCD or LED calibration image.
-1074395311	The color spectrum array you provided does not contain enough elements or contains an element set to not-anumber (NaN).
-1074395310	NI Vision does not support the PaletteType value you supplied.
-1074395309	NI Vision does not support the WindowThreadPolicy value you supplied.
-1074395308	NI Vision does not support the ColorSensitivity value you supplied.
-1074395307	The precision parameter must be greater than 0.
-1074395306	NI Vision does not support the Tool value you supplied.
-1074395305	NI Vision does not support the ReferenceMode value you supplied.
-1074395304	NI Vision does not support the MathTransformMethod value you supplied.
-1074395303	Invalid number of classes for auto threshold. Acceptable values range from 2 to 256.
-1074395302	NI Vision does not support the threshold method value you supplied.
-1074395301	The ROI you passed into imaqGetMeterArc must consist of two lines.
-1074395300	NI Vision does not support the MeterArcMode value you

	supplied.
-1074395299	NI Vision does not support the ComplexPlane value you supplied.
-1074395298	You can perform this operation on a real or an imaginary ComplexPlane only.
	NI Vision does not support the ParticleInfoMode value you supplied.
	NI Vision does not support the BarcodeType value you supplied.
-1074395295	imaqInterpolatePoints does not support the InterpolationMethod value you supplied.
-1074395294	The contour index you supplied is larger than the number of contours in the ROI.
-1074395293	The supplied ContourID did not correlate to a contour inside the ROI.
-1074395292	Do not supply collinear points for this operation.
-1074395291	Shape Match requires the image to contain only pixel values of 0 or 1.
-1074395290	The template you supplied for ShapeMatch contains no shape information.
-1074395287	The line you provided contains two identical points, or one of the coordinate locations for the line is not a number (NaN).
-1074395286	Invalid concentric rake direction.
-1074395285	Invalid spoke direction.
-1074395284	Invalid edge process.
-1074395283	Invalid rake direction.
-1074395282	Unable to draw to viewer. You must have the latest version of the control.
-1074395281	Your image must be larger than its border size for this operation.
-107/305280	The ROI must only have a single Rectangle contour.

-1074395279	ROI is not a polygon.
-1074395278	LCD image is not a number.
-1074395277	The decoded barcode information did not pass the checksum test.
-1074395276	You specified parallel lines for the meter ROI.
-1074395275	Invalid browser image.
-1074395270	Cannot divide by zero.
-1074395269	Null pointer.
-1074395268	The linear equations are not independent.
-1074395267	The roots of the equation are complex.
-1074395265	The barcode does not match the type you specified.
-1074395263	No lit segment.
-1074395262	The LCD does not form a known digit.
-1074395261	An internal error occurred while attempting to access an invalid coordinate on an image.
-1074395260	An internal memory error occurred.
-1074395258	The filter width must be odd for the Canny operator.
-1074395257	You supplied an invalid edge direction in the Canny operator.
-1074395256	The window size must be odd for the Canny operator.
-1074395253	Invalid learn mode.
-1074395252	Invalid learn setup data.
-1074395251	Invalid match mode.
-1074395250	Invalid match setup data.
-1074395249	At least one range in the array of rotation angle ranges exceeds 360 degrees.
-1074395248	The array of rotation angle ranges contains too many ranges.
-1074395247	Invalid template descriptor.
-1074395246	Invalid template descriptor.
-1074395245	Invalid template descriptor.
I	

-1074395244	Invalid template descriptor.
-1074395243	The template descriptor was created with a newer version of NI Vision. Upgrade to the latest version of NI Vision to use this template.
-1074395242	Invalid template descriptor.
-1074395241	The template descriptor does not contain data required fo rotation-invariant matching.
-1074395240	Invalid template descriptor.
-1074395239	Invalid template descriptor.
-1074395238	The template descriptor does not contain data required fo shift-invariant matching.
-1074395237	Invalid template descriptor.
-1074395235	The template image does not contain enough contrast.
-1074395234	The template image is too small.
-1074395233	The template image is too large.
-1074395153	Invalid font file format.
-1074395152	Font file not found.
-1074395150	NI Vision does not support the RoundingMode value you supplied.
-1074395149	Found a duplicate transform type in the properties array. Each properties array may only contain one behavior for each transform type.
-1074395148	Overlay Group Not Found.
-1074395147	The barcode is not a valid RSS Limited symbol
-1074395146	Couldn't determine the correct version of the QR code.
-1074395145	Invalid read of the QR code.
-1074395144	The barcode that was read contains invalid parameters.
-1074395143	The data stream that was demodulated could not be read because the mode was not detected.
-1074395142	Couldn't determine the correct model of the QR code.
-1074395139	Invalid Dimensions.
-250399	Internal Error: The DSP device is not operating correctly

	and cannot be initialized.
-250398	Not enough memory on the DSP to perform the operation.
-250397	The DSP device was not found.
-250396	The pointer provided is either NULL when it should not be, or is non-NULL when it should be NULL.
-250395	The Base Address Register window number is incorrect.
-250394	The Base Address Register window offset is incorrect.
-250393	Internal Error: The attempt to download the program to the DSP was unsuccessful.
-250392	The requested file could not be found.
-250391	The requested file is corrupt.
-250390	Internal Error: The installed components of the NI Vision DSP driver are incompatible. Reinstall NI Vision DSP.
-250389	The output data block for the operation was too small to hold all of the output data.
-250388	The operation did not complete before the specified timeout expired.
-250387	The number of operations in progress exceeds the maximum allowable number of concurrent operations.
-250386	The requested operation type is unknown.
-250385	Internal Error: The operation header address is invalid.
-250384	Internal Error: The host has lost communication with the DSP.
-250383	The input data size was invalid.
-250382	The allocation from heap memory failed.
-250381	A failure occured while freeing heap memory.
-250380	Internal Error: The DMA transfer failed.
-250379	Internal Error: The DSP cannot boot from PCI.
-250378	The specified Abort Callback could not be found.
-250377	The requested feature or function is not supported.
-250376	An operation with the same identifier is already in use.

-250375	The DSP device has not been programmed.
-250374	The requested INI file could not be found
-250373	The requested INI file key could not be found.
-250372	The data byte alignment is invalid.
-250371	Internal Error: The memory transfer header address is invalid.

# Glossary

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

Α

AIPD National Instruments internal image file format used for

saving complex and HSL images and calibration

information associated with an image. AIPD images have

the file extension APD.

alignment The process by which a machine vision application

determines the location, orientation, and scale of a part

being inspected.

area Detects objects based on their size, which can fall within

threshold a user-specified range.

arithmetic The image operations multiply, divide, add, subtract, and

operators remainder.

asynchronous Property of a function or operation that begins an

operation and returns control to the program before the

completion or termination of the operation.

auto-median

function

A function that uses dual combinations of opening and

closing operations to smooth the boundaries of objects.

В

barycenter The barycenter of a range of an image's grayscale values

is the grayscale value representing the centroid of that

range in the image histogram.

binary An image containing objects usually represented with a

image pixel intensity of 1(or 255) and the background of 0.

binary Functions that perform morphological operations on a

morphology binary image.

blob Binary large object. A particle, or object, present in a binary

image.

blurring Reduces the amount of detail in an image. Blurring

commonly occurs because the camera is out of focus. You

can blur an image intentionally by applying a lowpass

frequency filter.

BMP Bitmap. Image file format commonly used for 8-bit and

color images. BMP images have the file extension BMP.

border Removes objects (or particles) that touch the image border

function in a binary image.

		-
1	r	,
ı	l	
٦		

caliper Finds edge pairs along a specified path in the image. This

function performs an edge extraction and then finds edge pairs based on specified criteria such as the distance between the leading and trailing edges, edge contrasts,

and so forth.

cell A single module that encodes one bit of data in a 2D

barcode.

CIE L\*a\*b\* Color encoding scheme that classifies colors according to

the human vision system by mimicking the logarithmic

response of the eye.

CIE XYZ Color encoding scheme that classifies colors according to

the human vision system.

circle Detects circular objects in a binary image.

function

class A category representing a collection of similar samples.

classification An operation that assigns samples to classes based on

predefined features.

classification Probability that a sample is classified into the class to

accuracy which it belongs.

classification Degree of certainty that a sample is assigned to one class

confidence instead of other classes. See also class and sample.

classification Probability that a sample classified into a given class

predictive belongs to that class.

value

classifier A function or VI that assigns a sample to a class.

closed An ROI that describes an inclusive area in an image.

contour Types of closed contours include the following: Rectangle,

Oval, Polygon, Freehand Region, Annulus, and Rotated

Rectangle.

closing A dilation followed by an erosion. A closing fills small holes

in objects and smooths the boundaries of objects.

CLUT Color lookup table. Table for converting the value of a

	pixel in an image into a red, green, and blue (RGB) intensity.
code word	Numeric value of the printed bar/space pattern in a 1D or 2D barcode.
color images	Images containing color information, usually encoded in the RGB form.
complex images	Save information obtained from the FFT of an image. The complex numbers that compose the FFT plane are encoded in 64-bit floating-point values: 32 bits for the real part and 32 bits for the imaginary part.
connectivity	Defines which of the surrounding pixels of a given pixel constitute its neighborhood.
connectivity-	Only pixels adjacent in the horizontal and vertical directions are considered neighbors.
connectivity-8	All adjacent pixels are considered as neighbors.
convex hull function	Computes the convex hulls of objects in a binary image.
convolution	See linear filter.
convolution kernel	2D matrices (or templates) used to represent the filter in the filtering process. The contents of these kernels are a discrete two-dimensional representation of the impulse response of the filter that they represent.
curve extraction	The process of finding curves or connected edge points in a grayscale image. Curves usually represent the boundaries of objects in the image.

,

Danielsson function	Similar to the distance functions, but with more accurate results.
densitometry	Determination of optical or photographic density.
density function	For each gray level in a linear histogram, it gives the number of pixels in the image that have the same gray level.
device	Plug-in data acquisition board that can contain multiple channels and conversion devices.
differentiation filter	Extracts the contours (edge detection) in gray level.
digital image	An image f (x, y) that has been converted into a discrete number of pixels. Both spatial coordinates and brightness are specified.
dilation	Increases the size of an object along its boundary and removes tiny holes in the object.
distance calibration	Determination of the physical dimensions of a pixel by defining the physical dimensions of a line in the image.
distance function	Assigns, to each pixel in an object, a gray-level value equal to its shortest Euclidean distance from the border of the object.

Ε

function

erosion

edge Defined by a sharp change (transition) in the pixel intensities in an image or along an array of pixels.

edge contrast The difference between the average pixel intensity

before and the average pixel intensity after the edge.

edge The difference in threshold levels between a rising and a hysteresis falling edge.

edge The number of pixels that correspond to the slope or steepness transition area of an edge.

entropy A measure of the randomness in an image. An image with high entropy contains more pixel value variation

than an image with low entropy.

equalize See histogram equalization.

erasure Missing or undecodable codeword at a known position in

a 2D barcode.

Reduces the size of an object along its boundary and

eliminates isolated points in the image.

exponential Expand the high gray-level information in an image while and gamma suppressing low gray-level information.

exponential Decreases the brightness and increases the contrast in bright regions of an image and decreases contrast in

dark regions.

F

feature A measurement from or attribute of a sample.

feature An operation that computes features of a sample.

extraction

feature A 1D array in which each element represents a different

vector feature of a sample.

FFT Fast Fourier Transform. A method used to compute the

Fourier Transform of an image.

fiducial A reference pattern on a part that helps a machine vision

application find the part's location and orientation in an

image.

Fourier The magnitude information of the Fourier Transform of an

spectrum image.

filters

Fourier Transforms an image from the spatial domain to the

Transform frequency domain.

frequency Counterparts of spatial filters in the frequency domain. For

images, frequency information is in the form of spatial

frequency.

G

gauging	Measurement of an object or distances between objects.
Gaussian filter	A filter similar to the smoothing filter, but using a Gaussian kernel in the filter operation. The blurring in a Gaussian filter is more gentle than a smoothing filter.
geometric matching	The technique used to locate a grayscale template that is characterized by distinct geometric or shape information within a grayscale image.
geometric features	The information extracted from a grayscale template that is used to locate the template in the target image. Geometric features in an image range from low-level features, such as edges or curves detected in the image, to high-level features, such as the geometric shapes made by curves in the image.
golden template	An image containing an ideal representation of an object under inspection.
gradient convolution filter	See gradient filter.
gradient filter	Extracts the contours (edge detection) in gray-level values. Gradient filters include the Prewitt and Sobel filters.
gray level	The brightness of a point (pixel) in an image.
gray-level dilation	Increases the brightness of pixels in an image that are surrounded by other pixels with a higher intensity.
gray-level erosion	Reduces the brightness of pixels in an image that are surrounded by other pixels with a lower intensity.
gray-level images	Images with monochrome information.
gray-level morphology	Functions that perform morphological operations on a gray-level image.

Н

highpass attenuation	Applies a linear attenuation to the frequencies in an image, with no attenuation at the highest frequency and full attenuation at the lowest frequency.
highpass FFT filter	Removes or attenuates low frequencies present in the FFT domain of an image.
highpass filter	Emphasizes the intensity variations in an image, detects edges (or object boundaries), and enhances fine details in an image.
highpass frequency filter	Attenuates or removes (truncates) low frequencies present in the frequency domain of the image. A highpass frequency filter suppresses information related to slow variations of light intensities in the spatial image.
highpass truncation	Removes all frequencies below a certain frequency.
histogram	Indicates the quantitative distribution of the pixels of an image per gray-level value.
histogram equalization	Transforms the gray-level values of the pixels of an image to occupy the entire range (0 to 255 in an 8-bit image) of the histogram, increasing the contrast of the image.
histogram inversion	Finds the photometric negative of an image. The histogram of a reversed image is equal to the original histogram flipped horizontally around the center of the histogram.
hit-miss function	Locates objects in the image similar to the pattern defined in the structuring element.
hole filling function	Fills all holes in objects that are present in a binary image.
HSI	Color encoding scheme in Hue, Saturation, and, Intensity.
HSL	Color encoding scheme using Hue, Saturation, and Luminance information where each image in the pixel is encoded using 32-bits: 8 bits for hue, 8 bits for saturation, 8 bits for luminance, and 8 unused bits.

HSV Color encoding scheme in Hue, Saturation, and Value.

I

image A two-dimensional light intensity function f(x, y), where,

x and y denote spatial coordinates and the value f at any point (x, y) is proportional to the brightness at that point.

image file A file containing image information and data.

image Encompasses various processes and analysis functions

processing that you can apply to an image.

image A technique that interprets the content of the image at a

understanding symbolic level rather than a pixel level.

image The presentation (display) of an image (image data) to

visualization the user.

inner gradient Finds the inner boundary of objects.

inspection The process by which parts are tested for simple defects

such as missing parts or cracks on part surfaces.

inspection Detects specific features in an image, including edges,

functions peaks, and rotational shifts.

intensity Assigning user-defined quantities such as optical

calibration densities or concentrations to the gray-level values in an

image.

intensity The gray-level distribution of the pixels along an ROI in

profile an image.

intensity Defines the range of gray-level values in an object of an

range image.

intensity Characterizes an object based on the range of gray-level threshold values in the object. If the intensity range of the object

falls within the user-specified range, it is considered an object; otherwise it is considered part of the background.

interpolation The technique used to find values between known

values when resampling an image or array of pixels.

invariant A feature vector that is invariant to variations such as the

feature scale, rotation, and mirror symmetry of samples.

#### J

JPEG Joint Photographic Experts Group. Image file format for

storing 8-bit and color images with lossy compression.

JPEG images have the file extension JPG.

JPEG2000 An image file format for storing 8-bit, 16-bit, or color images

with either lossy or lossless compression. JPEG2000

images have the file extension *JP2*.

L

labeling A morphology operation that identifies each object in a

binary image and assigns a unique pixel value to all the pixels in an object. This process is useful for identifying the number of objects in the image and giving each object

a unique pixel intensity.

Laplacian filter

Extracts the contours of objects in the image by

highlighting the variation of light intensity surrounding a

pixel.

line gauge Measures the distance between selected edges with high-

precision subpixel accuracy along a line in an image. For example, this function can be used to measure distances between points and edges and vice versa. This function also can step and repeat its measurements across the

image.

line profile Represents the gray-level distribution along a line of pixels

in an image.

linear filter A special algorithm that calculates the value of a pixel

based on its own pixel value as well as the pixel values of its neighbors. The sum of this calculation is divided by the sum of the elements in the matrix to obtain a new pixel

value.

local threshold A method of image segmentation that categorizes a pixel as part of a particle or the background based on the intensity statistics of the particle's neighboring pixels.

logarithmic and inverse gamma corrections

Expand low gray-level information in an image while compressing information from the high gray-level ranges.

logarithmic function

Increases the brightness and contrast in dark regions of an image and decreases the contrast in bright regions of the image.

logic operators The image operations AND, NAND, OR, NOR, XOR, XNOR, difference, mask, mean, max, and min.

lossless Compression in which the decompressed image is compression identical to the original image.

lossy Compression in which the decompressed image is visually compression similar but not identical to the original image.

lowpass Applies a linear attenuation to the frequencies in an attenuation at the lowest frequency and full attenuation at the highest frequency.

lowpass Removes or attenuates high frequencies present in the compression FFT domain of an image.

lowpass FFT Attenuates intensity variations in an image. You can use filter these filters to smooth an image by eliminating fine details and blurring edges.

lowpass frequency frequency domain of the image. A lowpass frequency filter suppresses information related to fast variations of light intensities in the spatial image.

lowpass Removes all frequency information above a certain truncation frequency.

LUT

L-skeleton Uses an L-shaped structuring element in the Skeleton function function.

Lookup table. Table containing values used to transform the gray-level values of an image. For each gray-level value in the image, the corresponding new value is obtained from the lookup table.

M

machine vision An inspection or measurement application that uses application

images acquired from a 2D sensor (typically a CCD camera) to help with inspection or measurement.

mask Isolates parts of an image for further processing.

mask FFT filter Removes frequencies contained in a mask (range)

specified by the user.

mask image An image containing a value of 1 and values of 0.

> Pixels in the source image with a corresponding mask image value of 1 are processed, while the others are

left unchanged.

match score A number ranging from 0 to 1000 that indicates how

> closely an acquired image matches the template image. A match score of 1000 indicates a perfect match. A

match score of 0 indicates no match.

median filter A lowpass filter that assigns to each pixel the median

value of its neighbors. This filter effectively removes isolated pixels without blurring the contours of objects.

**MMX** Multimedia Extensions. Intel chip-based technology that

allows parallel operations on integers, which results in

accelerated processing of 8-bit images.

morphological Extract and alter the structure of objects in an image. transformations You can use these transformations for expanding

(dilating) or reducing (eroding) objects, filling holes, closing inclusions, or smoothing borders. They mainly are used to delineate objects and prepare them for

quantitative inspection analysis.

M-skeleton Uses an M-shaped structuring element in the skeleton

function.

matching

The technique used to simultaneously locate multiple multiple template

grayscale templates within a grayscale image.

neighbor A pixel whose value affects the values of nearby pixels

when an image is processed. The neighbors of a pixel

are usually defined by a kernel.

neighborhood Operations on a point in an image that take into

operations consideration the values of the pixels neighboring that

point.

nonlinear Replaces each pixel value with a nonlinear function of its

filter surrounding pixels.

nonlinear A highpass edge-extraction filter that favors vertical

gradient filter edges.

nonlinear A highpass edge-extraction filter that favors horizontal

Prewitt filter and vertical edges in an image.

nonlinear A highpass edge-extraction filter that favors horizontal

Sobel filter and vertical edges in an image.

Nth order Filters an image using a nonlinear filter. This filter orders filter (or classifies) the pixel values surrounding the pixel being

processed. The pixel being processed is set to the *N*th

pixel value, where N is the order of the filter.

0

open contour An ROI that describes a point or area in an image. Types

of open contours include the following: Point, Line,

Broken Line, and Free Hand Line.

opening An erosion followed by a dilation. An opening removes

small objects and smooths boundaries of objects in the

image.

operators Allow masking, combination, and comparison of images.

You can use arithmetic and logic operators in NI Vision.

occlusion invariant matching

A geometric matching technique in which the reference pattern can be partially obscured in the target image.

OCR Opticle character recognition. The process of analyzing

an image to detect and recognize characters/text in the

image.

OCV Optical character verification. A machine vision

application that inspects the quality of printed characters.

optical Contains the low-frequency information at the center and

representation the high-frequency frequency information at the corners

of an FFT-transformed image.

outer gradient Finds the outer boundary of objects.

	_
н	_
н	_

palette The gradation of colors used to display an image on screen,

usually defined by a color lookup table.

particle A connected region or grouping of pixels in an image in

which all pixels have the same intensity level.

particle A classifier that classifies particles. See also classifier and

classifier particle.

pattern The technique used to locate quickly known reference

matching patterns or fiducials in an image.

picture An element of a digital image.

element

pixel Picture element.

pixel Directly calibrating the physical dimensions of a pixel in an

calibration image.

pixel The number of bits used to represent the gray level of a

depth pixel.

PNG Portable Network Graphic. Image file format for storing 8-bit,

16-bit, and color images with lossless compression.

Power 1/Y Similar to a logarithmic function but with a weaker effect.

function

Power Y See exponential function.

function

Prewitt Extracts the contours (edge detection) in gray-level values

filter using a 3 × 3 filter kernel.

probability Defines the probability that a pixel in an image has a certain

function gray-level value.

proper- A finite combination of successive closing and opening

closing operations that you can use to fill small holes and smooth

the boundaries of objects.

proper- A finite combination of successive opening and closing

opening operations that you can use to remove small particles and

smooth the boundaries of objects.

pyramidal A technique used to increase the speed of a pattern matching algorithm by matching subsampled versions of the image and the reference pattern.

Q

quantitative analysis Obtaining various measurements of objects in an image.

#### R

Reverse Inverts the pixel values in an image, producing a photometric function negative of the image.

RGB Color encoding scheme using red, green and blue (RGB) color information where each pixel in the color image is encoded using 32 bits: 8 bits for red, 8 bits for green, 8 bits for blue, and 8 bits for the alpha value (unused).

Color encoding scheme using red, green, and blue (RGB) color information where each pixel in the color image is encoded using 64 bits: 16 bits for red, 16 bits for green, 16 bits for blue, and 16 bits for the alpha value (unused).

Roberts Extracts the contours (edge detection) in gray level, favoring filter diagonal edges.

ROI Region of interest. An area of the image that is graphically selected from a window displaying the image. This area can be used to focus further processing. This region can also be defined programmatically.

rotation- A pattern matching technique in which the reference pattern invariant can be at any orientation in the test image.

matching

rotational The amount by which one image is rotated with respect to a shift reference image. This rotation is computed with respect to the center of the image.

S

sample	An object in an image that you want to classify.
scale- invariant matching	A pattern matching technique in which the reference pattern can be any size in the test image.
segmentation function	Fully partitions a labeled binary image into non- overlapping segments, with each segment containing a unique object.
separation function	Separates objects that touch each other by narrow isthmuses.
shape descriptor	A feature vector that describes the shape of a sample. See also feature vector and particle analysis.
shape matching	Finds objects in an image whose shape matches the shape of the object specified by a template. The matching process is invariant to rotation and can be set to be invariant to the scale of the objects.
shift-invariant matching	A pattern matching technique in which the reference pattern can be located anywhere in the test image but cannot be rotated or scaled.
Sigma filter	A highpass filter that outlines edges.
skeleton function	Applies a succession of thinning operations to an object until its width becomes one pixel.
skiz	Obtains lines in an image that separate each object from the others and are equidistant from the objects that they separate.
smoothing filter	Blurs an image by attenuating variations of light intensity in the neighborhood of a pixel.
Sobel filter	Extracts the contours (edge detection) in gray-level values using a $3 \times 3$ filter kernel.
spatial calibration	Assigning physical dimensions to the area of a pixel in an image.
spatial filters	Alter the intensity of a pixel with respect to variations in intensities of its neighboring pixels. You can use these

filters for edge detection, image enhancement, noise reduction, smoothing, and so forth.

spatial resolution

function

The number of pixels in an image, in terms of the number of rows and columns in the image.

Square

See exponential function.

Square Root function

See logarithmic function.

standard

Contains the low-frequency information at the corners representation and high-frequency information at the center of an FFTtransformed image.

structuring element

A binary mask used in most morphological operations. A structuring element is used to determine which

neighboring pixels contribute in the operation.

subpixel analysis

Used to find the location of the edge coordinates or pattern matching results in terms of fractions of a pixel.

substitution error

Erroneously decoded codeword at an unknown position

in a 2D barcode.

Property or operation that begins an operation and synchronous

returns control to the program only when the operation is

complete.

Set of rules to which statements must conform in a syntax

particular programming language.

thickening Alters the shape of objects by adding parts to the object that match the pattern specified in the structuring element.

thinning Alters the shape of objects by eliminating parts of the object that match the pattern specified in the structuring element.

threshold Separates objects from the background by assigning all pixels with intensities within a specified range to the object and the rest of the pixels to the background. In the resulting binary image, objects are represented with a pixel intensity of 255 and the background is set to 0.

threshold Two parameters, the lower threshold gray-level value and the interval upper threshold gray-level value.

TIFF Tagged Image File Format. Image format commonly used for encoding 8-bit and color images. TIFF images have the file extension *TIF*.

truth table A table associated with a logic operator that describes the rules used for that operation.

#### V

virtual corner A corner that would be created if two non-intersecting lines are extended until they intersect.

#### W

watershed A method of image segmentation that partitions an image transform based on the topographic surface of the image. The image is separated into non-overlapping segments with each segment containing a unique particle.

## **Important Information**

**Warranty** 

Copyright

**Trademarks** 

<u>Patents</u>

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 National Instruments trademarks.

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 $^{\text{TM}}$  and Target Language Compiler $^{\text{TM}}$  are trademarks of The MathWorks, Inc.

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

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

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

## **Patents**

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the patents.txt file on your CD, or <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 National Instruments Web site at ni.com for technical support and professional services:

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

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.

#### **Vision and Motion**

Installed With: NI Vision Development Module

Refer to the NI Vision, NI Vision Acquisition Software, and NI-Motion help files, which you can access from the **Help** menu, for more information about those VIs. If NI Vision Acquisition Software and NI-Motion are not installed, you cannot access their help files. Install these additional packages from the Driver CD included with LabVIEW. You must purchase NI Vision separately.

Subpalette	Description
<u>Vision</u> <u>Utilities</u>	Use the Vision Utilities VIs to create and manipulate images in NI Vision.
Image Processing	Use the Image Processing VIs to analyze, filter, and process images in NI Vision.
Machine Vision	Use the Machine Vision VIs to perform common machine vision inspection tasks, including the presence or absence of parts in an image and measuring the dimensions of parts to see if they meet specifications.

## **Casting Images**

The following table lists the way in which <u>IMAQ Cast Image</u> casts pixel values if the **Image Src** type and **Image Type** parameter are not the same.

Image Src Type	Image Type	Result
		If you provide a lookup table, the destination pixel will have the lookup value of the source pixel. If you do not provide a lookup table, the VI copies the source value to the destination unmodified.
U16		Each color component of the destination is set to the source value. If the source value is greater than 255, the VI sets each color component to 255. If the source value is less than 0, the VI sets each color component to 0. If the source image has a specified bit depth, the VI uses the bit depth when performing this conversion.
U16		The VI sets the luminance component of the destination to the source value. If the source value is greater than 255, the VI sets the luminance to 255. If the source value is less than 0, the VI sets the luminance to 0. The VI sets hue and saturation component of the destination to 0. If the source image has a specified bit depth, the VI uses the bit depth when performing this conversion.
U16		The VI sets the real component of the destination to the source value. The VI sets the imaginary component of the destination to 0.
U16		The VI shifts the source value to the right by the given shift value (divides each source pixel value by 2^shift) and stores the value in the destination. If the shifted value is greater than 255, the VI sets the destination value to 255. If the shift value is zero, the VI uses the specified bit depth of the source image.

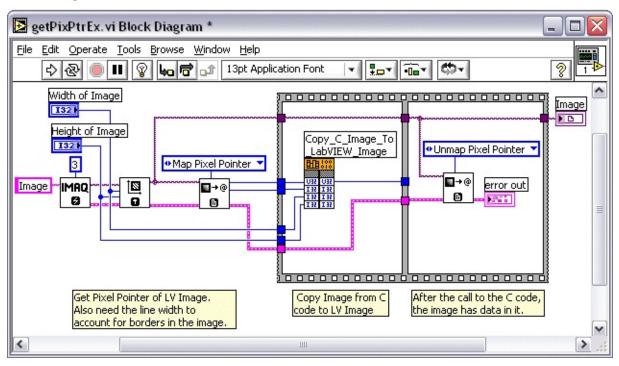
	ļ	
U16		Each color component of the destination is set to the source value. If the source value is greater than 65535, the VI sets each color component to 65535. If the source value is less than 0, the VI sets each color component to 0.
RGB US4		The VI shifts the source value to the 8-bit range using the specified bit depth of the source image. Then the VI sets the destination value to the average of the three color components of the source.
RGB U64	U16	The VI sets the destination value to the average of the three color components of the source. If the average of the source color components is out of the range of the destination, the VI coerces the average to the range.
RGB U64	RGB U32	The VI shifts the source value to the 8-bit range using the specified bit depth of the source image. Then the VI sets each color component in the destination value to the corresponding component in the source value.
RGB U64	HSL US2	The VI shifts the source value to the 8-bit range using the specified bit depth of the source image. Then the VI converts each pixel from the RGB color space to the HSL color space.
U16		If you provide a lookup table, the destination pixel will have the lookup value of the source pixel. If you do not provide a lookup table, the VI copies the source value to the destination unmodified.
	U16	The VI sets the destination value to the source value. If the source value is out of the range of the destination, the VI coerces the source to the range.
RGB U32	U16	The VI sets the destination value to the average of the three color components of the source.
RGB U32	HSL U32	The VI converts each pixel from the RGB color

		space to the HSL color space.
RGB U32 RGB U64		The VI sets the real portion of the destination value to the average of the three color components of the source, and it sets the imaginary portion of the destination to 0.
HSL U32	U16	The VI sets the destination value to the luminance component of the source value.
HSL U32	RGB U32 RGB U64	The VI converts each pixel from the HSL color space to the RGB color space.
HSL <sub>2</sub>		The VI sets the real portion of the destination value to the value of the luminance component of the source, and it sets the imaginary portion of the destination to 0. If the source value is out of the range of the destination, the VI coerces the source to the range.
	016	The VI sets the destination value to the magnitude of the source value. If the source value is out of the range of the destination, the VI coerces the source to the range.
	RGB U6-4	The VI sets each color component of the destination value to the magnitude of the source value. If the source value is out of the range of the destination, the VI coerces the source to the range.
		The VI sets the luminance component of the destination value to the magnitude of the source value, and it sets the hue and saturation components to 0.
U16		The VI sets the destination value to the source value. If the source value is out of the range of the destination, the VI coerces the source to the range.
		The VI sets the destination value to the source value. If the source value is out of the range of the destination, the VI coerces the source to the range.

ı	I	I		ı

### **IMAQ GetImagePixelPtr Example**

The following figure illustrates a typical implementation scheme for <u>IMAQ</u> <u>GetImagePixelPtr</u>.



This VI enables the user to set an image size in LabVIEW. The VI then gets the pixel pointer, line width, image width, and image height.



**Note** The line width is the width, in pixels, of an image including the image border.

The Call Library Function Copy\_C\_Image\_To\_LabVIEW\_Image takes in the above parameters, fills a C image with user-defined data, and copies the data to the LabVIEW image. When Copy\_C\_Image\_To\_LabVIEW\_Image finishes execution, the LabVIEW image contains the data that the C code calculated.

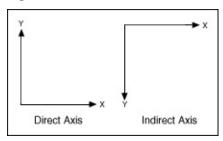
The C code calculates the following:

```
//
                      128, and copies it to a passed LV image
                      which is then returned to the calling VI.
//
//
// Parameters:
                   - the passed LV image
//
    LVImagePtr
    LVLineWidth - the width (including the border) of the passed LV imag
//
                   - the width (excluding the border) of the passed LV image
    LVWidth
//
                   - the height of the passed LV image
//
    LVHeight
//
void __declspec(dllexport) __stdcall Copy_C_Image_To_LabVIEW_Image (char
int LVWidth, int LVHeight)
ImageInfo testImageInfo;
Image *testImage;
int y;
PixelValue pixelValue;
// Create the image as IMAQ_IMAGE_U8
testImage = imaqCreateImage (IMAQ_IMAGE_U8, 3);
// Set the image size
imagSetImageSize (testImage, LVWidth, LVHeight);
// Fill the image with 128's
pixelValue.grayscale = 128;
imagFillImage (testImage, pixelValue, NULL);
// Get the image info -- this should be that same as the height and width
// the image was created from
imaqGetImageInfo (testImage, &testImageInfo);
// Copy, line by line, the C image into the LV image.
for (y=0; y<LVHeight; ++y) {
  memcpy (LVImagePtr + y * LVLineWidth, (char*)testImageInfo.imageStart +
  y * testImageInfo.pixelsPerLine, LVWidth);
}
// Dispose of the C image
```

```
imaqDispose(testImage);
}
```

# **Axis Type**

You can specify an axis to be Direct or Indirect, as shown in the following figure:

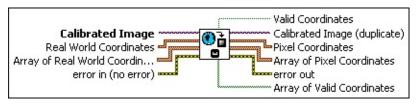


### **IMAQ Convert Real World to Pixel VI**

Owning Palette: <u>Calibration</u>Installed With: NI Vision Development Module

Transforms real-world coordinates to pixel coordinates, according to the calibration information acquired from the <u>IMAQ Learn Calibration</u> <u>Template</u> VI, the <u>IMAQ Set Simple Calibration</u> VI, or copied from a calibrated image using the <u>IMAQ Set Calibration Info</u> VI.





- **Calibrated Image** is the reference to the input image that has calibration information attached to it.
- Real World Coordinates is a cluster of real-world point coordinates to convert to pixel coordinates.
- Array of Real World Coordinates is an array of clusters of real-world point coordinates to convert to pixel coordinates.
- error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
  - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero **error code**. If **status** is FALSE, **code** is 0 or a

warning code.

- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Valid Coordinates returns TRUE if the transformed coordinate lies inside the Corrected Image Learn ROI of the IMAQ Learn Calibration Template VI. In the case of a simple calibration or a corrected image, Valid Coordinates is always TRUE. If one of the point coordinates is defined outside the ROI learned for calibration, this output is set to FALSE.
- Calibrated Image (duplicate) is a reference to the output image. This image is identical to the input Calibrated Image.
- **Pixel Coordinates** is a cluster of pixel coordinates to convert to real-world coordinates.
- Array of Pixel Coordinates is an array of clusters containing the xand y-coordinates of the pixels corresponding to the real-world points.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code** is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
  - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Array of Valid Coordinates is an array of Boolean values specifying whether each transformation of the Array of Real World Coordinates is valid, according to the region selected in the Corrected Image Learn ROI of the IMAQ Learn Calibration

<u>Template</u> VI. In the case of a corrected image, the elements of this array are always TRUE.

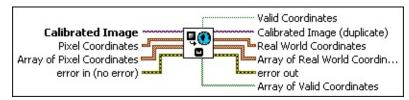
### **IMAQ Convert Pixel to Real World VI**

Owning Palette: <u>Calibration</u>Installed With: NI Vision Development Module

Transforms pixel coordinates to real-world coordinates, according to the calibration information contained in the image. Calibration information is attached to this image by the <a href="IMAQ Learn Calibration Template">IMAQ Learn Calibration Template</a> VI, the <a href="IMAQ Set Simple Calibration">IMAQ Set Calibration Info</a> VI.

#### **Examples**





- **Calibrated Image** is the reference to the input image that has calibration information attached to it.
- **Pixel Coordinates** is a cluster of pixel coordinates to convert to real-world coordinates.
- Array of Pixel Coordinates is an array of pixel coordinate clusters to convert to real-world coordinates.
  - D

Note If both Pixel Coordinates and Array of Pixel Coordinates are connected, the VI uses Array of Pixel Coordinates.

error in (no error) describes the error status before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the <a href="Simple Error Handler">Simple Error Handler</a> or <a href="General Error Handler">General Error Handler</a> VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or

- that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- Valid Coordinates returns TRUE if the transformed coordinate lies inside the Corrected Image Learn ROI of the IMAQ Learn Calibration Template VI. In the case of a simple calibration or a corrected image, Valid Coordinates is always TRUE. If one of the point coordinates is defined outside the ROI learned for calibration, this output is set to FALSE.
- Calibrated Image (duplicate) is a reference to the output image. This image is identical to the input Calibrated Image.
- Real World Coordinates returns a cluster of real-world point coordinates that correspond to the points specified by Point Coordinates.
- Array of Real World Coordinates is an array of real-world point coordinates corresponding to the array of pixel coordinates specified by Array of Pixel Coordinates.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - code is the error or warning code. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
  - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the

error or warning. The default is an empty string.

Array of Valid Coordinates is an array of Boolean values specifying whether each transformation of the Array of Pixel Coordinates is valid, according to the region selected in the Corrected Image Learn ROI of the IMAQ Learn Calibration Template VI. Depending on which areas of the image were calibrated, not all pixel positions can be converted to real-world coordinates.

### **Examples**

Refer to the following for examples that use this VI.

- LabVIEW\examples\Vision\2. Functions\Calibration\Simple Calibration Example.vi
- LabVIEW\examples\Vision\2. Functions\Calibration\Perspective Calibration Example.vi

### **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 413091
Japan	81 3 5472 2970
Korea	82 02 3451 3400
Lebanon	961 (0) 1 33 28 28
Malaysia	1800 887710
Mexico	01 800 010 0793
Netherlands	31 (0) 348 433 466
New Zealand	0800 553 322
Norway	47 (0) 66 90 76 60
Poland	48 22 3390150
Portugal	351 210 311 210
Russia	7 495 783 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