

## NI Vision Builder for Automated Inspection: ActiveX Component Reference Help

August 2008, 371484D-01

You can use the Vision Builder for Automated Inspection (Vision Builder AI) ActiveX Component to embed Vision Builder AI inspections in your ActiveX applications. You can use this ActiveX interface to open a Vision Builder AI inspection, change the limits associated with each step, run the inspection, and retrieve the results of each run. You can run an inspection on either the local machine or on an NI Vision remote targets, such as the NI 17xx Smart Camera or NI CVS-1450 Series Compact Vision System. This interface also allows you to display, in your ActiveX application, images used in the inspection.

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

Using Help

**Related Documentation** 

**Important Information** 

Technical Support and Professional Services

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

© 2005–2008 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 product activation and licensing refer to ni.com/activate.

# **Upgrade Considerations**

The following information is important to those upgrading to a newer version of Vision Builder AI.

- If you are using the Vision Builder AI ActiveX Component from LabVIEW, you must use LabVIEW 8.2 or later.
- Installing the Vision Builder AI ActiveX Component will upgrade previously installed versions of the ActiveX Component. Upgrading will not cause existing applications to break, but you may have to link LabVIEW applications again. Use the <u>NIVBAITargetControl</u> to communicate with and control specific versions of Vision Builder AI.

## **Related Documentation**

Most Vision Builder AI 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</u> <u>Product Manuals Library</u> for updated documentation resources.

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

- <u>Vision Builder for Automated Inspection Tutorial</u>—This printed manual contains lessons that teach you how to use the Vision Builder AI Configuration Interface and create various machine vision applications.
- *NI Vision Builder for Automated Inspection: Configuration Help—* This help file contains information about using the Vision Builder AI Configuration interface to create a machine vision application.
- *NI Vision Builder for Automated Inspection: Inspection Help* This help file contains information about running the machine vision application you created in the Vision Builder AI Configuration Interface.

# **Using Help**

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

# Conventions

This help file uses the following formatting and typographical conventions:

The » symbol leads you through nested menu items and » dialog box options to a final action. The sequence **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. P 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 in the software, such as menu items and dialog box options. Bold text also denotes parameter names. Underlined text in this color denotes a link to a help topic, <u>green</u> help file, or Web address. italic Italic text denotes variables, emphasis, cross-references, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply. monospace Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names,

functions, operations, variables, filenames, and extensions.

# **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.

## **Vision Builder AI ActiveX Objects**

**NIVBAIError** 

**NIVBAIInspection** 

**NIVBAIInspections** 

**NIVBAIInspectionEngine** 

**NIVBAIInspectionImage** 

**NIVBAIInspectionStatistics** 

**NIVBAIState** 

<u>NIVBAIStates</u>

<u>NIVBAIStep</u>

**NIVBAISteps** 

**NIVBAIStepLimit** 

**NIVBAIStepLimits** 

**NIVBAIStepMeasurement** 

**NIVBAIStepMeasurements** 

<u>NIVBAIStepResult</u>

**NIVBAIStepResults** 

**NIVBAITarget** 

**NIVBAITargets** 

**NIVBAITargetControl** 

<u>NIVBAILimitType</u>

<u>NIVBAITargetMode</u>

**NIVBAITargetStatus** 

<u>NIVBAITargetType</u>

**NIVBAITargetVersion** 

## NIVBAIError

Information associated with an error.

## **Properties**

CodeError code.DescriptionDescription of the error.TitleTitle of the error.

# **NIVBAIInspection**

Single inspection.

## **Properties**

<b>CreationDate</b>	Date the inspection was created on the target.
<u>DefaultImage</u>	Image returned from the inspection.
<b>Description</b>	Description of the inspection.
LastModifiedDate	Date the inspection was last changed and saved.
<u>Name</u>	Name of the inspection.
<u>States</u>	Collection of states in the inspection.
<u>Target</u>	Target that contains the inspection.

## Methods

<b>CloseInspection</b>	Closes the inspection on the target.
<u>GetStatistics</u>	Retrieves the latest inspection statistics.
<u>Inspect</u>	Runs the inspection once synchronously.
<u>RefreshLimits</u>	Refreshes the <u>NIVBAIStep.Limits</u> information for each step, passing in the most recent limits.
<u>RefreshResults</u>	Refreshes <u>NIVBAISteps.Results</u> with the most recent inspection results.
<u>ResetStatistics</u>	Resets the inspection statistics.
StartInspection	Runs the inspection asynchronously in a loop.
StopInspection	Stops the inspection if it is looping asynchronously.

# NIVBAIInspections

Collection of inspections.

### **Properties**

<u>Count</u> Number of items in the collection.

Item Returns the specified object from the collection.

Target that contains the inspections.

# NIVBAIInspectionEngine

Main engine.

PropertiesActiveTargetsCollection of open targets.InspectionEngineReturns a reference to the global<br/>NIVBAIInspectionEngine object.

### Methods

<u>ConnectToTarget</u>	( <b>Obsolete</b> —Use <u>ConnectToTarget2</u> instead) Connects to either a local or remote target.
ConnectToTarget2	Connects to either a local or remote target.
<u>GetTimeout</u>	Retrieves the current timeout for synchronous methods.
<u>SetTimeout</u>	Sets the timeout for all synchronous methods.
<u>ShowOpenInspectionDialog</u>	Displays a dialog box with a list of all inspections available on the target.

### **Events**

InspectionComplete	Triggered after each inspection when looping asynchronously.
<u>RefreshImageComplete</u>	Triggered when an asynchronous request for the latest image has completed.

# NIVBAIInspectionImage

Image returned from an inspection.

Note If you want to use the NI Vision VIs or Display control with an image in a LabVIEW application, you must use the VBAI Get Image VI. Refer to the <u>VBAI Get Image</u> topic for information about this VI.

### **Properties**

ImageImage data.InspectionInspection that contains the image.NameName of the image.

### Methods

**Refresh** Updates <u>NIVBAIInspectionImage.Image</u> with the latest image from the inspection.

<u>StartRefresh</u> Requests the latest image from the inspection.

# NIVBAIInspectionStatistics

Statistics about an active inspection.

## **Properties**

<u>ActiveRatio</u>	Percentage of the time that the inspection has been active.
<u>ActiveTime</u>	Number of seconds the inspection has been active.
<u>IdleTime</u>	Number of seconds the inspection has been idle.
Inspection	Inspection tracked by the statistics.
NumFailed	Number of times the inspection failed.
NumPassed	Number of times the inspection passed.
<u>TimeStamp</u>	Date and time the statistics were last updated.
<u>TimeStampUTC</u>	Date and time the statistics were last updated in Coordinated Universal Time (UTC).
<u>Yield</u>	Percentage of the total inspections that passed.

1

## NIVBAIState

Single state in an inspection.

#### **Properties**

Inspection The inspection that contains the state.

Name of the state.

<u>Steps</u> Collection of steps in the state.

<u>Terminal</u> Specifies whether the state is a terminal state.

## **NIVBAIStates**

Collection of inspection states.
CountNumber of items in the collection.InspectionThe inspection that contains the states.ItemReturns the specified object from the collection.

### NIVBAIStep

Single step in an inspection.

Inspection Inspection that contains the step.

<u>Limits</u> Collection of limits used to verify the result of the step.

Name of the step.

**<u>Result</u>** Result of running the step after the most recent inspection.

<u>State</u> The state that contains the step.

<u>StepType</u> Action the step performs.

### **NIVBAISteps**

Collection of inspection steps.

<u>Count</u> Number of items in the collection.

Item Returns the specified object from the collection.

**<u>Results</u>** Results from the steps.

<u>State</u> The state that contains the steps.

# NIVBAIStepLimit

Single limit placed on a result.

<u>LimitType</u> Type of limit placed on the result.

LowerLimit Lower bound in a range, or the only limit for single value limit types.

<u>Step</u> Step that contains the limit.

<u>UpperLimit</u> Upper bound in a range.

<u>UseLimit</u> Value that indicates whether to use the limit when inspecting.

# NIVBAIStepLimits

Collection of limits for a step.

<u>Count</u> Number of items in the collection.

- <u>Item</u> Returns the specified object from the collection.
- <u>Step</u> Step that contains the limits.

# NIVBAIStepMeasurement

Measurement returned by an inspection step.

Name of the measurement.

<u>Result</u> Result that contains the measurements.

<u>Unit</u> Unit type of the NIVBAIStepMeasurement.Value.

<u>Value</u> Value of the measurement.

# NIVBAIStepMeasurements

Collection of measurements for a step.

Properties	
<u>Count</u>	Number of items in the collection.
<u>ltem</u>	Returns the specified object from the collection.
<u>Result</u>	Result that contains the measurements.
<u>TimeStamp</u>	Date and time the measurements were last updated internally.
<u>TimeStampUTC</u>	Date and time the measurements were last updated in Coordinated Universal Time (UTC).

\_

# NIVBAIStepResult

Result returned by an inspection step.

<u>FailureText</u>	Explanation of why a step failed.
<u>Measurements</u>	Measurements calculated by the step.
Passed	Value that specifies whether the step passed.
<u>Report</u>	Text returned by the step.
<u>Step</u>	Step that contains the result.
<u>TimeStamp</u>	Date and time the results were last updated internally.
<u>TimeStampUTC</u>	Date and time the results were last updated internally in Coordinated Universal Time (UTC).
Value	Measured value returned by the step.

# **NIVBAIStepResults**

Collection of results in an inspection.

<u>Count</u>	Number of items in the collection.
<u>ltem</u>	Returns the specified object from the collection.
<u>Steps</u>	Steps that contain the results.
<u>TimeStamp</u>	Date and time the results were last updated internally.
<u>TimeStampUTC</u>	Date and time the results were last updated internally in Coordinated Universal Time (UTC).

\_

### **NIVBAITarget**

Single local or remote execution target.

Note If you open a target using this object, avoid using <u>NIVBAITargetControl</u> to access the target. Using both the object and the control for the target may result in the inspection attempting to access a target that has been closed unknowingly.

ActiveInspection	Currently-loaded inspection.
Inspections	Collection of inspections available on the target.
IPAddress	IP address of the target.
<u>TargetType</u>	Execution type for the target.
<u>TargetVersion</u>	Version of Vision Builder AI on the target.
<u>UpdateMeasurementsOnRefresh</u>	Tells the target to retrieve or ignore step measurements.

### Methods

<u>Disconnect</u>	Disconnects from the target and leaves it in its current state.
<u>GetDateTime</u>	Returns the current date, time, and timezone on the target.
<u>GetDateTimeUTC</u>	Returns the current date, time, and timezone on the target in Coordinated Universal Time (UTC).
<u>GetStatus</u>	Returns the current status of the target.
<b>OpenInspection</b>	Opens the specified inspection on the target.
<u>RefreshInspections</u>	Updates <u>NIVBAITarget.Inspections</u> to contain the latest collection of inspections available on the target.
<u>SetDateTime</u>	Sets the date, time, and timezone on the target.
<u>SetDateTimeUTC</u>	Sets the date, time, and timezone on the target in Coordinated Universal Time (UTC).

### NIVBAITargets

Collection of execution targets.

<u>Count</u> Number of items in the collection.

Item Returns the specified object from the collection.

\_

# **NIVBAITargetControl**

Control that represents a single local or remote execution target.

- Note If the connection to a local target fails the first time you try to connect to the local target, restart the computer and retry the connection.
- $\overline{\mathbb{N}}$

**Note** If you open a target using this control, avoid using the <u>NIVBAITarget</u> object to access the target. Using both the object and the control for the target may result in the inspection attempting to access a target that has been closed unknowingly.

Caution When you run a LabVIEW VI that contains the NIVBAITargetControl, National Instruments strongly recommends that you avoid using the Abort Execution button to stop running the VI. When you use the Abort Execution button, the VI stops, but the connection to the target is not closed. When the connection to the target is still open, you cannot modify any of the control properties. The property page items appear unavailable.

ActiveInspection	Currently-loaded inspection.
Inspections	Collection of inspections available on the target.
IPAddress	IP address of the target.
IsConnected	Indicates whether a connection is open.
Mode	Level of access to request from the target.
<u>TargetType</u>	Execution type for the target.
<u>UpdateMeasurementsOnRefresh</u>	Tells the target to retrieve or ignore step measurements.
Version	Specifies the version of Vision Builder AI that you expect to be running on a target.

#### **Methods** Connect Connects to either a local or remote target. Disconnects from the target and leaves it in Disconnect its current state. Returns the current date, time, and GetDateTime timezone on the target. **GetDateTimeUTC** Returns the current date, time, and timezone on the target in Coordinated Universal Time (UTC). Returns the current status of the target. GetStatus **OpenInspection** Opens the specified inspection on the target. **RefreshInspections** Updates NIVBAITargetControl.Inspections to contain the latest collection of inspections available on the target. Sets the date, time, and timezone on the **SetDateTime** target. SetDateTimeUTC Sets the date, time, and timezone on the target in Coordinated Universal Time (UTC). ShowOpenInspectionDialog Displays a dialog box with a list of all inspections available on the target.

### **Events**

InspectionComplete	Triggered after each inspection when looping asynchronously.
<u>RefreshImageComplete</u>	Triggered when an asynchronous request for the latest image has completed.

### **NIVBAILimitType Enumeration**

Type of limit applied to the inspection results.

- vbaiLessThan—Result must be less than the limit.
- vbaiLessThanOrEqualto—Result must be less than or equal to the limit.
- vbaiGreaterThan—Result must be greater than the limit.
- vbaiGreaterThanOrEqualTo—Result must be greater than or equal to the limit.
- vbaiEqualTo—Result must be equal to the limit.
- vbaiNotEqualTo—Result must not be equal to the limit.
- vbailnRange—Result must be between the values specified in the limit.
- vbaiMatchPattern—Result must have the same pattern as the limit.

### NIVBAITargetMode Enumeration

Levels of access available on a target.



Note When using the Vision Builder AI ActiveX component, the NIVBAITargetMode setting is not available. Connection to a target always allows full control.

- vbaiMonitor—Indicates read-only access.
- vbaiFullControl—Indicates unrestricted access.

### **NIVBAITargetStatus Enumeration**

Current state of the target.

- vbaiNotRunning—Target either cannot be found or is not available.
- vbaiRunning—Target is available but not ready for Vision Builder AI.
- vbailnitialized—Target is ready for Vision Builder AI, but nothing is currently loaded.
- vbaiConfigured—Target has an inspection loaded, but the inspection is not running.
- vbailnspecting—Target is running an inspection.

# NIVBAITargetType Enumeration

Execution type of a single target.

- vbaiLocalTarget—Indicates a target on the local machine.
- vbaiRemoteTarget—Indicates a target on a network.

### **NIVBAITargetVersion Enumeration**

Version of a target.

- vbaiOldestAvailableVersion—Target is oldest available version of Vision Builder AI.
- vbaiLatestAvailableVersion—Target is latest available version of Vision Builder AI.
- vbaiVersion261—Target is Vision Builder AI version 2.6.1.
- vbaiVersion300—Target is Vision Builder AI version 3.0.0.
- vbaiVersion350—Target is Vision Builder AI version 3.5.0.
- vbaiVersion360—Target is Vision Builder AI version 3.6.0.

# **Important Information**

Warranty Copyright Trademarks Patents Warning Regarding Use of NI Products

### Warranty

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

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

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

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

# Copyright

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

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

# Trademarks

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

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

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

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

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>, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - Free Technical Support—All registered users receive free Basic Service, which includes access to hundreds of Applications Engineers worldwide in the <u>NI Discussion</u> 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.

# **Code Property (Read Only)**

### Syntax

NIVBAIError.Code

### Data Type

<u>Long</u>

Error code.

# **Description Property (Read Only)**

### Syntax

NIVBAIError.Description

# Data Type

<u>String</u>

Description of the error.

## **Title Property (Read Only)**

### Syntax

NIVBAIError.Title

# Data Type

<u>String</u>

Title of the error.

# **CreationDate Property (Read Only)**

#### Syntax

NIVBAIInspection.CreationDate

# Data Type

<u>String</u>

Date the inspection was created on the target.

# **DefaultImage Property (Read Only)**

### Syntax

NIVBAIInspection.DefaultImage

### Data Type NIVBAIInspectionImage

Image returned from the inspection.

# **Description Property (Read Only)**

### Syntax

NIVBAIInspection.Description

# Data Type

<u>String</u>

Description of the inspection.

# LastModifiedDate Property (Read Only)

#### Syntax

NIVBAIInspection.LastModifiedDate

# Data Type

<u>String</u>

Date the inspection was last changed and saved.

# Name Property (Read Only)

### Syntax

NIVBAIInspection.Name

# Data Type

<u>String</u>

Name of the inspection.

## **States Property (Read Only)**

### Syntax

NIVBAIInspection.States

#### Data Type NIVBAIStates

Collection of states in the inspection.

# **Target Property (Read Only)**

### Syntax

NIVBAIInspection.Target

### Data Type NIVBAITarget

Target that contains the inspection.

# **CloseInspection Method**

### Syntax

NIVBAIInspection.CloseInspection

Closes the inspection on the target.
### **GetStatistics Method**

### Syntax

NIVBAIInspection.GetStatistics

### **Return Type**

<u>NIVBAIInspectionStatistics</u>

Retrieves the latest inspection statistics.

#### **Return Value**

Object that contains the inspection statistics.

## **Inspect Method**

### Syntax

NIVBAIInspection.Inspect

### **Return Type**

<u>Boolean</u>

Runs the inspection once synchronously.

### **Return Value**

Indicates whether the inspection passed or failed.

### **RefreshLimits Method**

### Syntax

NIVBAIInspection.RefreshLimits

Refreshes the <u>NIVBAIStep.Limits</u> information for each step, passing in the most recent limits.

# Limits Property (Read Only)

### Syntax

NIVBAIStep.Limits

# Data Type

**NIVBAIStepLimits** 

Collection of limits used to verify the result of the step.

## **RefreshResults Method**

### Syntax

NIVBAIInspection.RefreshResults

Refreshes <u>NIVBAISteps.Results</u> with the most recent inspection results.

# **Results Property (Read Only)**

### Syntax

NIVBAISteps.Results

### Data Type

**NIVBAIStepResults** 

Results from the steps.

## **ResetStatistics Method**

### Syntax

NIVBAIInspection.ResetStatistics

Resets the inspection statistics.

# **StartInspection Method**

### Syntax

NIVBAIInspection.StartInspection

Runs the inspection asynchronously in a loop. Every time an inspection finishes, an InspectionComplete event is triggered.

#### Remarks

This method performs the inspection asynchronously, which might result in out-of-date data. If your application must run synchronously, use <u>NIVBAIInspection.Inspect</u>.

# **StopInspection Method**

### Syntax

NIVBAIInspection.StopInspection

Stops the inspection if it is looping asynchronously.

#### Remarks

This method performs the inspection asynchronously, which might result in out-of-date data. If your application must run synchronously, use <u>NIVBAIInspection.Inspect</u>.

# **Count Property (Read Only)**

### Syntax

NIVBAIInspections.Count

### Data Type

<u>Long</u>

Number of items in the collection.

## **Item Property**

### Syntax

NIVBAIInspections.Item Index

### Data Type

**NIVBAlInspection** 

Returns the specified object from the collection.

#### **Return Value**

Item found at the specified index.

#### **Parameters**

#### Index as <u>Variant</u>

Index of the object in the collection or the name of the inspection. The index is one-based.

# **Target Property (Read Only)**

### Syntax

NIVBAIInspections. Target

### Data Type NIVBAITarget
Target that contains the inspections.

## **ActiveTargets Property (Read Only)**

## Syntax

NIVBAIInspectionEngine.ActiveTargets

## Data Type <u>NIVBAITargets</u>

Collection of open targets.

## InspectionEngine Property

## Syntax

NIVBAIInspectionEngine.InspectionEngine

# Data Type <u>NIVBAIInspectionEngine</u>

Returns a reference to the global NIVBAIInspectionEngine object.

## **ConnectToTarget Method**

### Syntax

<u>NIVBAIInspectionEngine</u>.ConnectToTarget [IPAddress = ""], [Mode = vbaiFullControl], [Password = ""], [Status]

## Return Type

**NIVBAITarget** 

(**Obsolete**—Use <u>ConnectToTarget2</u> instead) Connects to either a local or remote target. A connection must be established before inspections can run. If an inspection is already running on a target, the <u>InspectionComplete</u> events are received.

## **Return Value**

Target opened.

#### Remarks

Vision Builder AI currently supports having only one target open at a time. An error occurs if you try to open a target while another is open, but this error is not guaranteed behavior. Use ActiveTargets.Count to determine if a target is currently open.

#### Parameters

#### IPAddress as Variant

The IP address of the target. Use one of the following three methods to open the local target:

- Do not specify this parameter.
- Use an empty string as the IP address
- Use 127.0.0.1.

#### Mode as <u>Variant</u>

Level of access to request from the target.

#### Password as Variant

Password required to gain access to the target. This parameter is ignored on the local target.

#### Status as <u>Variant</u>

Current state of the target.

## ConnectToTarget2 Method

## Syntax

NIVBAIInspectionEngine.ConnectToTarget2 [IPAddress], [Mode], [Version], Password, Status

## Return Type

**NIVBAITarget** 

Connects to either a local or remote target. A connection must be established before inspections can run. If an inspection is already running on a target, the <u>InspectionComplete</u> events are received.

## **Return Value**

Target opened.

#### Remarks

Vision Builder AI currently supports having only one target open at a time. An error occurs if you try to open a target while another is open, but this error is not guaranteed behavior. Use ActiveTargets.Count to determine if a target is currently open.

#### Parameters

#### IPAddress as Variant

The IP address of the target. Use one of the following three methods to open the local target:

- Do not specify this parameter.
- Use an empty string as the IP address
- Use 127.0.0.1.

#### Mode as <u>Variant</u>

Level of access to request from the target.

Version as Variant

The target version of Vision Builder AI to use.

#### Password as String

Password required to gain access to the target. This parameter is ignored on the local target.

Status as <u>NIVBAITargetStatus</u>

Current state of the target.

## **GetTimeout Method**

## Syntax

NIVBAIInspectionEngine.GetTimeout

## **Return Type**

<u>Long</u>

Retrieves the current timeout for synchronous methods.

## **Return Value**

Timeout in milliseconds.

## **SetTimeout Method**

## Syntax

NIVBAIInspectionEngine.SetTimeout Timeout

Sets the timeout for all synchronous methods.

#### Parameters

Timeout as Long

Timeout in milliseconds.

## ShowOpenInspectionDialog Method

## Syntax

<u>NIVBAIInspectionEngine</u>.ShowOpenInspectionDialog Target

## **Return Type**

**NIVBAlInspection** 

Displays a dialog box with a list of all inspections available on the target. The user chooses which inspection to open.

### **Return Value**

Inspection opened. The method returns Nothing if the user cancelled.

#### Parameters

Target as **NIVBAITarget** 

Target that contains the inspections.

## InspectionComplete Event

## Syntax

<u>NIVBAIInspectionEngine</u>.InspectionComplete Target, Inspection, Passed, Error

Triggered after each inspection when looping asynchronously.

#### **Parameters**

Target as <u>NIVBAITarget</u>

The target containing the inspection.

Inspection as **NIVBAIInspection** 

The inspection that was run.

Passed as Boolean

Indicates whether the inspection passed or failed.

Error as **NIVBAIError** 

Contains error information if an error occurred, or Nothing if everything was successful.

## RefreshImageComplete Event

## Syntax

NIVBAIInspectionEngine.RefreshImageComplete Target, Inspection, Image, Error

Triggered when an asynchronous request for the latest image has completed.
#### Parameters

Target as <u>NIVBAITarget</u>

The target containing the inspection.

Inspection as **NIVBAIInspection** 

The inspection containing the image.

Image as **NIVBAIInspectionImage** 

The image that was refreshed.

Error as **NIVBAIError** 

Contains error information if an error occurred, or Nothing if everything was successful.

# VBAI Get Image VI

Retrieves an image from Vision Builder AI. Details



- **NIVBAILib.\_NIVBAIInspectionImage** is the image you are passing in from a Vision Builder AI inspection.
- **Image In** is a reference to the image from the Vision Builder AI inspection.
- **ZoomX** is the location on the x-axis where you want to zoom in on the image.
- **ZoomY** is the location on the y-axis where you want to zoom in 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 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 if an error occurred before this VI was called, or FALSE if not. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code.
  - **code** is a number identifying an error or warning. If **status** is TRUE, **code** is a nonzero error code. If **status** is FALSE, **code** is zero or a warning code. Use the error handler VIs to look up the meaning of this code and display the corresponding error message.
  - **source** is a string that indicates the origin of the error, if any. Typically, **source** is the name of the VI in which the error occurred.
- Image Out is a reference to the image from the Vision Builder AI

# Using This VI

Use this VI to use an image acquired in Vision Builder AI with your LabVIEW code.

Refer to the SimpleExample VI for a demonstration of using an image from a Vision Builder AI inspection in a LabVIEW VI. This shipping example is installed to <Vision Builder AI>\ActiveX Support\Examples\LabVIEW, where <Vision Builder AI> is the location to which you installed Vision Builder AI.

# Image Property (Read Only)

# Syntax

NIVBAIInspectionImage.Image

# Data Type CWIMAQImage

Image data.

# **Inspection Property (Read Only)**

# Syntax

NIVBAIInspectionImage.Inspection

**NIVBAlInspection** 

Inspection that contains the image.

# Name Property (Read Only)

# Syntax

NIVBAIInspectionImage.Name

<u>String</u>

Name of the image.

# **Refresh Method**

# Syntax

<u>NIVBAIInspectionImage</u>.Refresh [XZoom = 0], [YZoom = 0]

Updates <u>NIVBAIInspectionImage.Image</u> with the latest image from the inspection.

#### Parameters

#### XZoom as Variant

Indicates how much to zoom the image horizontally. Values can be zero or less, and negative numbers indicate how far to zoom out. If this value is anything other than zero, overlays are lost.

#### YZoom as Variant

Indicates how much to zoom the image vertically. Values can be zero or less, and negative numbers indicate how far to zoom out. If this value is anything other than zero, overlays are lost.

# StartRefresh Method

### Syntax

NIVBAIInspectionImage.StartRefresh XZoom, YZoom

Requests the latest image from the inspection.

NIVBAllnspectionImage.Image is updated and a RefreshImageComplete event is triggered when this request is done.

#### Parameters

#### XZoom as Long

Indicates how much to zoom the image horizontally. Values can be zero or less, and negative numbers indicate how far to zoom out. If this value is anything other than zero, overlays are lost.

#### YZoom as Long

Indicates how much to zoom the image vertically. Values can be zero or less, and negative numbers indicate how far to zoom out. If this value is anything other than zero, overlays are lost.

# **ActiveRatio Property (Read Only)**

### Syntax

NIVBAIInspectionStatistics.ActiveRatio

<u>double</u>

Percentage of the time that the inspection has been active.

# **ActiveTime Property (Read Only)**

### Syntax

NIVBAIInspectionStatistics.ActiveTime

<u>double</u>

Number of seconds the inspection has been active.

# IdleTime Property (Read Only)

# Syntax

NIVBAIInspectionStatistics.IdleTime

<u>double</u>

Number of seconds the inspection has been idle.

# **Inspection Property (Read Only)**

# Syntax

NIVBAIInspectionStatistics.Inspection

**NIVBAlInspection** 

Inspection tracked by the statistics.

# NumFailed Property (Read Only)

# Syntax

NIVBAIInspectionStatistics.NumFailed

<u>Long</u>

Number of times the inspection failed.

# NumPassed Property (Read Only)

### Syntax

NIVBAIInspectionStatistics.NumPassed

<u>Long</u>

Number of times the inspection passed.
# **TimeStamp Property (Read Only)**

### Syntax

NIVBAIInspectionStatistics.TimeStamp

<u>Date</u>

Date and time the statistics were last updated.

# TimeStampUTC Property (Read Only)

### Syntax

NIVBAIInspectionStatistics.TimeStampUTC

<u>double</u>

Date and time the statistics were last updated in Coordinated Universal Time (UTC). UTC is represented by the number of seconds that have elapsed since 12:00 AM on January 1, 1970.

## Yield Property (Read Only)

### Syntax

NIVBAIInspectionStatistics.Yield

<u>double</u>

Percentage of the total inspections that passed.

# **Inspection Property (Read Only)**

### Syntax

NIVBAIState.Inspection

**NIVBAlInspection** 

The inspection that contains the state.

## Name Property (Read Only)

### Syntax

NIVBAIState.Name

<u>String</u>

Name of the state.

## **Steps Property (Read Only)**

### Syntax

NIVBAIState.Steps

### Data Type <u>NIVBAISteps</u>

Collection of steps in the state.

# **Terminal Property (Read Only)**

### Syntax

NIVBAIState.Terminal

<u>Boolean</u>

Specifies whether the state is a terminal state.

## **Count Property (Read Only)**

### Syntax

NIVBAIStates.Count

<u>Long</u>

Number of items in the collection.

# **Inspection Property (Read Only)**

### Syntax

NIVBAIStates.Inspection

**NIVBAlInspection** 

The inspection that contains the states.

## Item Property

### Syntax

NIVBAIStates.Item Index

### Data Type NIVBAIState

Returns the specified object from the collection.

#### **Return Value**

Item found at the specified index.

#### Parameters

#### Index as <u>Variant</u>

Index of the object in the collection or the name of the state. The index is one-based.

# **Inspection Property (Read Only)**

### Syntax

NIVBAIStep.Inspection

NIVBAIStepResult

Inspection that contains the step.

## Name Property (Read Only)

### Syntax

NIVBAIStep.Name
<u>String</u>

Name of the step.

## **Result Property (Read Only)**

#### Syntax

NIVBAIStep.Result

NIVBAIStepResult

Result of running the step after the most recent inspection.

## State Property (Read Only)

#### Syntax

NIVBAIStep.State

The state that contains the step.

## StepType Property (Read Only)

#### Syntax

NIVBAIStep.StepType

<u>String</u>

Action the step performs.

## **Count Property (Read Only)**

#### Syntax

NIVBAISteps.Count

<u>Long</u>

Number of items in the collection.

## Item Property

#### Syntax

NIVBAISteps.Item Index

### Data Type NIVBAIStep

Returns the specified object from the collection.

#### **Return Value**

Item found at the specified index.

#### Parameters

#### Index as <u>Variant</u>

Index of the object in the collection or the name of the step. The index is one-based.

## State Property (Read Only)

#### Syntax

NIVBAISteps.State

The state that contains the steps.

## LimitType Property

#### Syntax

NIVBAIStepLimit.LimitType

#### Data Type NIVBAILimitType

Type of limit placed on the result.

## LowerLimit Property

#### Syntax

NIVBAIStepLimit.LowerLimit

<u>Variant</u>

Lower bound in a range, or the only limit for single value limit types.

## **Step Property (Read Only)**

#### Syntax

NIVBAIStepLimit.Step

### Data Type NIVBAIStep

Step that contains the limit.

## **UpperLimit Property**

#### Syntax

NIVBAIStepLimit.UpperLimit

<u>Variant</u>

Upper bound in a range.

## **UseLimit Property**

#### Syntax

NIVBAIStepLimit.UseLimit

<u>Boolean</u>
Value that indicates whether to use the limit when inspecting.

# **Count Property (Read Only)**

# Syntax

NIVBAIStepLimits.Count

<u>Long</u>

Number of items in the collection.

# **Item Property**

# Syntax

NIVBAIStepLimits.Item Index

**NIVBAIStepLimit** 

Returns the specified object from the collection.

### **Return Value**

Item found at the specified index.

#### Parameters

#### Index as <u>Variant</u>

Index of the object in the collection or the name of the step. The index is one-based.

# **Step Property (Read Only)**

# Syntax

NIVBAIStepLimits.Step

# Data Type NIVBAIStep

Step that contains the limits.

# Name Property (Read Only)

# Syntax

NIVBAIStepMeasurement.Name

<u>String</u>

Name of the measurement.

# **Result Property (Read Only)**

# Syntax

NIVBAIStepMeasurement.Result

NIVBAIStepResult

Result that contains the measurements.

# **Unit Property (Read Only)**

# Syntax

NIVBAIStepMeasurement.Unit

<u>String</u>

Unit type of the NIVBAIStepMeasurement.Value.

# Value Property (Read Only)

# Syntax

NIVBAIStepMeasurement.Value

<u>Variant</u>

Value of the measurement.

# **Count Property (Read Only)**

# Syntax

NIVBAIStepMeasurements.Count

<u>Long</u>

Number of items in the collection.

# Item Property

# Syntax

NIVBAIStepMeasurements.Item Index

NIVBAIStepMeasurement

Returns the specified object from the collection.

### **Return Value**

Item found at the specified index.

#### Parameters

#### Index as Variant

Index of the object in the collection. The index is one-based.

# **Result Property (Read Only)**

# Syntax

NIVBAIStepMeasurements.Result

NIVBAIStepResult

Result that contains the measurements.

# **TimeStamp Property (Read Only)**

# Syntax

NIVBAIStepMeasurements.TimeStamp
<u>Date</u>

Date and time the measurements were last updated internally.

# TimeStampUTC Property (Read Only)

## Syntax

NIVBAIStepMeasurements.TimeStampUTC

<u>double</u>

Date and time the measurements were last updated in Coordinated Universal Time (UTC). UTC is represented by the number of seconds that have elapsed since 12:00 AM on January 1, 1970.

# FailureText Property (Read Only)

## Syntax

NIVBAIStepResult.FailureText

<u>String</u>

Explanation of why a step failed. If the step passed, this property is empty.

# **Measurements Property (Read Only)**

#### Syntax

NIVBAIStepResult.Measurements

<u>NIVBAIStepMeasurements</u>

Measurements calculated by the step.

# **Passed Property (Read Only)**

## Syntax

NIVBAIStepResult.Passed

<u>Boolean</u>

Value that specifies whether the step passed.

# **Report Property (Read Only)**

## Syntax

NIVBAIStepResult.Report

<u>String</u>

Text returned by the step.

# **Step Property (Read Only)**

## Syntax

NIVBAIStepResult.Step

## Data Type NIVBAIStep

Step that contains the result.

# **TimeStamp Property (Read Only)**

## Syntax

NIVBAIStepResult.TimeStamp

<u>Date</u>

Date and time the results were last updated internally.

# TimeStampUTC Property (Read Only)

## Syntax

NIVBAIStepResult.TimeStampUTC

<u>double</u>

Date and time the results were last updated internally in Coordinated Universal Time (UTC).

# Value Property (Read Only)

## Syntax

NIVBAIStepResult.Value

<u>Variant</u>

Measured value returned by the step.

# **Count Property (Read Only)**

## Syntax

NIVBAIStepResults.Count

<u>Long</u>

Number of items in the collection.

# **Item Property**

## Syntax

NIVBAIStepResults.Item Index

NIVBAIStepResult

Returns the specified object from the collection.

#### **Return Value**

Item found at the specified index.
#### Parameters

#### Index as <u>Variant</u>

Index of the object in the collection or the name of the step. The index is one-based.

# **Steps Property (Read Only)**

## Syntax

NIVBAIStepResults.Steps

## Data Type <u>NIVBAISteps</u>

Steps that contain the results.

# **TimeStamp Property (Read Only)**

## Syntax

NIVBAIStepResults.TimeStamp

<u>Date</u>

Date and time the results were last updated internally.

# TimeStampUTC Property (Read Only)

## Syntax

NIVBAIStepResults.TimeStampUTC

<u>double</u>

Date and time the results were last updated internally in Coordinated Universal Time (UTC). UTC is represented by the number of seconds that have elapsed since 12:00 AM on January 1, 1970.

# **ActiveInspection Property (Read Only)**

## Syntax

NIVBAITarget. ActiveInspection

**NIVBAlInspection** 

Currently-loaded inspection. This property contains Nothing if no inspection is loaded.

# **Inspections Property (Read Only)**

## Syntax

NIVBAITarget.Inspections

**NIVBAIInspections** 

Collection of inspections available on the target.

# **IPAddress Property (Read Only)**

## Syntax

NIVBAITarget.IPAddress

<u>String</u>

IP address of the target.

# TargetType Property (Read Only)

## Syntax

NIVBAITarget.TargetType

<u>NIVBAITargetType</u>

Execution type for the target.

# **TargetVersion Property (Read Only)**

## Syntax

NIVBAITarget.TargetVersion

**NIVBAITargetVersion** 

Version of Vision Builder AI on the target.

# UpdateMeasurementsOnRefresh Property (Read Only)

## Syntax

NIVBAITarget.UpdateMeasurementsOnRefresh

<u>Boolean</u>

Tells the target to retrieve or ignore step measurements.

#### Remarks

When this value is True, calls to NIVBAISteps.RefreshResults will also update NIVBAIStepResult.Measurements.

## **Disconnect Method**

## Syntax

NIVBAITarget.Disconnect

Disconnects from the target and leaves it in its current state. If an inspection is running on a remote target, it continues to do so. If an inspection is running on the local target, it stops.

## **GetDateTime Method**

## Syntax

**<u>NIVBAITarget</u>**.GetDateTime [TimeZone = 0]

## **Return Type**

<u>Date</u>

Returns the current date, time, and timezone on the target.

#### **Return Value**

Current date and time.

#### Parameters

#### TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the timezone on the target.
# GetDateTimeUTC Method

#### Syntax

NIVBAITarget.GetDateTimeUTC [TimeZone]

## **Return Type**

<u>double</u>

Returns the current date, time, and timezone on the target in Coordinated Universal Time (UTC).

#### **Return Value**

Current date and time in Universal Time (UTC). UTC is represented by the number of seconds that have elapsed since 12:00 AM on January 1, 1970.

#### Parameters

#### TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the timezone on the target.

# **GetStatus Method**

## Syntax

NIVBAITarget.GetStatus

#### **Return Type**

NIVBAITargetStatus

Returns the current status of the target.

## **Return Value**

Current status of the target.

# **OpenInspection Method**

## Syntax

NIVBAITarget.OpenInspection Inspection

## **Return Type**

**NIVBAlInspection** 

Opens the specified inspection on the target.

#### **Return Value**

Inspection that was opened.

#### Parameters

#### Inspection as Variant

Inspection to open. This inspection must be from the <u>NIVBAITarget.Inspections</u> collection on the target or it must be a valid path to an inspection on the target machine.

# **RefreshInspections Method**

## Syntax

NIVBAITarget.RefreshInspections

Updates <u>NIVBAITarget.Inspections</u> to contain the latest collection of inspections available on the target.

# SetDateTime Method

#### Syntax

**<u>NIVBAITarget</u>**.SetDateTime DateTime, [TimeZone]

Sets the date, time, and timezone on the target.

#### Parameters

#### DateTime as Date

New date and time for the target.

TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the new timezone on the target.

# SetDateTimeUTC Method

#### Syntax

NIVBAITarget.SetDateTimeUTC DateTime, [TimeZone]

Sets the date, time, and timezone on the target in Coordinated Universal Time (UTC).

#### Parameters

#### DateTime as Long

New date and time for this target in Coordinated Universal Time (UTC). UTC is represented by the number of seconds that have elapsed since 12:00 AM on January 1, 1970.

#### TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the new timezone on the target.

# **Count Property (Read Only)**

## Syntax

NIVBAITargets.Count

## Data Type

<u>Long</u>

Number of items in the collection.

# **Item Property**

## Syntax

NIVBAITargets.Item Index

## Data Type NIVBAITarget

Returns the specified object from the collection.

#### **Return Value**

Item found at the specified index.

#### **Parameters**

#### Index as <u>Variant</u>

Index of the object in the collection or the IP address of the target. The index is one-based.

# **ActiveInspection Property (Read Only)**

## Syntax

NIVBAITargetControl.ActiveInspection

## Data Type

**NIVBAlInspection** 

Currently-loaded inspection. This property contains Nothing if no inspection is loaded.

# **Inspections Property (Read Only)**

## Syntax

NIVBAITargetControl.Inspections

## Data Type

**NIVBAIInspections** 

Collection of inspections available on the target.
# **IPAddress Property**

### Syntax

NIVBAITargetControl.IPAddress

# Data Type

<u>NIVBAITargetType</u>

IP address of the target. This value cannot be set when a connection is active.

# **IsConnected Property (Read Only)**

### Syntax

NIVBAITargetControl.IsConnected

### Data Type

NIVBAITargetMode

Indicates whether a connection is open.

# Mode Property

### Syntax

NIVBAITargetControl.Mode

### Data Type

NIVBAITargetMode

Level of access to request from the target. This value cannot be set when a connection is active.

# TargetType Property (Read Only)

# Syntax

NIVBAITargetControl.TargetType

# Data Type

<u>NIVBAITargetType</u>

Execution type for the target.

# UpdateMeasurementsOnRefresh Property (Read Only)

# Syntax

NIVBAITargetControl.UpdateMeasurementsOnRefresh

# Data Type

<u>Boolean</u>

Tells the target to retrieve or ignore step measurements.

### Remarks

When this value is True, calls to NIVBAISteps.RefreshResults will also update NIVBAIStepResult.Measurements.

# **Version Property**

### Syntax

NIVBAITargetControl.Version

### Data Type

**NIVBAITargetVersion** 

Specifies the version of Vision Builder AI that you expect to be running on a target. This value cannot be set when a connection is active.

### Remarks

By default, the NIVBAITargetControl will target the earliest available version of Vision Builder AI.

You can specify the following values for the Version property:

Value	Description
vbaiOldestAvailableVersion	Uses the earliest available version of Vision Builder AI on the target.
vbaiLatestAvailableVersion	Uses the latest available version of Vision Builder AI on the target.
vbaiVersion261	Uses Vision Builder AI 2.6.1 on the target.
vbaiVersion300	Uses Vision Builder AI 3.0.0 on the target.
vbaiVersion350	Uses Vision Builder AI 3.5.0 on the target.
vbaiVersion360	Uses Vision Builder AI 3.6.0 on the target.
vbaiOldestAvailableVersion + vbaiVersion261	Uses the earliest available version of Vision Builder AI greater than or equal to Vision Builder AI 2.6.1 on the target.
vbaiOldestAvailableVersion + vbaiVersion300	Uses the earliest available version of Vision Builder AI greater than or equal to Vision Builder AI 3.0.0 on the target.
vbaiOldestAvailableVersion + vbaiVersion350	Uses the earliest available version of Vision Builder AI greater than or equal to Vision Builder AI 3.5.0 on the target.
vbaiOldestAvailableVersion + vbaiVersion360	Uses the earliest available version of Vision Builder AI greater than or equal to Vision Builder AI 3.6.0 on the target.
vbaiLatestAvailableVersion + vbaiVersion261	Uses the latest available version of Vision Builder AI greater than or equal to Vision Builder AI 2.6.1 on the target.
vbaiLatestAvailableVersion + vbaiVersion300	Uses the latest available version of Vision Builder AI greater than or equal to Vision Builder AI 3.0.0 on the target.
vbaiLatestAvailableVersion + vbaiVersion350	Uses the latest available version of Vision Builder AI greater than or equal to Vision

	Builder AI 3.5.0 on the target.
vbaiLatestAvailableVersion + vbaiVersion360	Uses the latest available version of Vision Builder AI greater than or equal to Vision Builder AI 3.6.0 on the target.

# **Connect Method**

### Syntax

<u>NIVBAITargetControl</u>.Connect [Password], [Status]

### **Return Type**

NIVBAITargetStatus

Connects to either a local or remote target. Uses the current values in <u>NIVBAITargetControl.IPAddress</u> and <u>NIVBAITargetControl.Mode</u>. A connection must be established before inspections can run. If an inspection is already running on a target, the <u>InspectionComplete</u> events are received.

#### Parameters

#### Password as Variant

Password required to gain access to the target. This parameter is ignored on the local target.

#### Status as <u>Variant</u>

Current state of the target.

# **Disconnect Method**

### Syntax

NIVBAITargetControl.Disconnect

Disconnects from the target and leaves it in its current state. If an inspection is running on a remote target, it continues to do so. If an inspection is running on the local target, it stops.

# **GetDateTime Method**

### Syntax

<u>NIVBAITargetControl</u>.GetDateTime [TimeZone = 0]

# **Return Type**

<u>Date</u>

Returns the current date, time, and timezone on the target.

### **Return Value**

Current date and time.

#### Parameters

#### TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the timezone on the target.

# GetDateTimeUTC Method

### Syntax

<u>NIVBAITargetControl</u>.GetDateTimeUTC [TimeZone]

# **Return Type**

<u>double</u>

Returns the current date, time, and timezone on the target in Coordinated Universal Time (UTC).

### **Return Value**

Current date and time in Universal Time (UTC). UTC is represented by the number of seconds that have elapsed since 12:00 AM on January 1, 1970.
#### TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the timezone on the target.

## **GetStatus Method**

## Syntax

NIVBAITargetControl.GetStatus

## **Return Type**

NIVBAITargetStatus

Returns the current status of the target.

## **Return Value**

Current status of the target.

# **OpenInspection Method**

## Syntax

NIVBAITargetControl.OpenInspection Inspection

## **Return Type**

**NIVBAlInspection** 

Opens the specified inspection on the target.

## **Return Value**

Inspection that was opened.

#### Inspection as Variant

Inspection to open. This inspection must be from the <u>NIVBAITarget.Inspections</u> collection on the target or it must be a valid path to an inspection on the target machine.

# **RefreshInspections Method**

## Syntax

NIVBAITargetControl.RefreshInspections

## **Return Type**

NIVBAITargetMode

Updates <u>NIVBAITargetControl.Inspections</u> to contain the latest collection of inspections available on the target.

## SetDateTime Method

## Syntax

<u>NIVBAITargetControl</u>.SetDateTime DateTime, [TimeZone]

Sets the date, time, and timezone on the target.

#### DateTime as Date

New date and time for the target.

TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the new timezone on the target.

## SetDateTimeUTC Method

## Syntax

**<u>NIVBAITargetControl</u>**.SetDateTimeUTC DateTime, [TimeZone]

Sets the date, time, and timezone on the target in Coordinated Universal Time (UTC).

#### DateTime as Long

New date and time for this target in Coordinated Universal Time (UTC). UTC is represented by the number of seconds that have elapsed since 12:00 AM on January 1, 1970.

#### TimeZone as Variant

Offset, in hours, from Coordinated Universal Time (UTC), which indicates the new timezone on the target.

# ShowOpenInspectionDialog Method

## Syntax

<u>NIVBAITargetControl</u>.ShowOpenInspectionDialog Target

## **Return Type**

**NIVBAlInspection** 

Displays a dialog box with a list of all inspections available on the target. The user chooses which inspection to open.

### **Return Value**

Inspection opened. The method returns Nothing if the user cancelled.

Target as **NIVBAITarget** 

Target that contains the inspections.

# InspectionComplete Event

## Syntax

**<u>NIVBAITargetControl</u>**.InspectionComplete Inspection, Passed, Error

Triggered after each inspection when looping asynchronously.

Inspection as **NIVBAIInspection** 

The inspection that was run.

Passed as **Boolean** 

Indicates whether the inspection passed or failed.

Error as <u>NIVBAIError</u>

Contains error information if an error occurred, or Nothing if everything was successful.

# RefreshImageComplete Event

## Syntax

<u>NIVBAITargetControl</u>.RefreshImageComplete Inspection, Image, Error

Triggered when an asynchronous request for the latest image has completed.

**Inspection** as <u>NIVBAIInspection</u> The inspection containing the image.

Image as **NIVBAIInspectionImage** 

The image that was refreshed.

Error as **NIVBAIError** 

Contains error information if an error occurred, or Nothing if everything was successful.

## **Branch Offices**

Office	Telephone Number
Australia	1800 300 800
Austria	43 662 457990-0
Belgium	32 (0) 2 757 0020
Brazil	55 11 3262 3599
Canada	800 433 3488
China	86 21 5050 9800
Czech Republic	420 224 235 774
Denmark	45 45 76 26 00
Finland	358 (0) 9 725 72511
France	33 (0) 1 57 66 24 24
Germany	49 89 7413130
India	91 80 41190000
Israel	972 0 3 6393737
Italy	39 02 41309277
Japan	0120 108492 / 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

## Data Types for the NI Vision Builder AI ActiveX Component

<b>Visual Basic</b>	C/C++	Description
Boolean	VARIANT_BOOL	Has the value True (-1) or False (0).
Date	double	A VB date.
Long	long	32-bit signed integer.
String	BSTR	A string.
Variant	LPVARIANT	A variant.
double	double	64-bit floating point number.

## **JavaScript Disabled**

The HTML file you are trying to access uses JavaScript. If you are viewing the file from your computer or from a CD and you have Internet Explorer 4.0 or later installed, JavaScript is enabled by default. If you are viewing the file from a network, such as on an intranet or on the Web, or if you do not have Internet Explorer 4.0 or later installed, you must enable JavaScript to view the file.