

USER GUIDE

NI Vision Builder for Automated Inspection Development Toolkit

Version 3.0

The Vision Builder for Automated Inspection (Vision Builder AI) Development Toolkit features a wizard and an application program interface (API) you can use to create custom steps for use in any Vision Builder AI script.



Note This toolkit is designed for advanced LabVIEW users who have experience developing LabVIEW applications with NI Vision.

You can create custom steps to provide functionality that is not already included in Vision Builder AI. For example, you can develop a custom step that drives a camera or other hardware not currently supported in Vision Builder AI. You also can create a custom step to include a customized image processing algorithm.



Note The Vision Builder AI Development Toolkit does not support the creation of acquisition steps for remote LabVIEW Real-Time targets, such as the NI CVS-1450 Series compact vision system.

You can distribute custom steps in addition to the released version of Vision Builder AI, providing your end users with a customized Vision Builder AI package.

System Requirements

The following list includes minimum system requirements for the Vision Builder AI Development Toolkit:

- Pentium III or greater or Celeron 866 MHz or equivalent processor (Pentium 4/M or equivalent processor recommended)
- Microsoft Windows 2000/XP

- Microsoft Internet Explorer 5.0 or later
- 1024 × 768 resolution video adapter with a 16-bit display (65,536 colors)
- 256 MB of RAM (512 MB recommended)
- 10 MB of available hard disk space
- LabVIEW 8.0.1
- NI Vision 8.0 (optional, but recommended)
- Vision Builder for Automated Inspection 3.0

Installation Instructions

Complete the following steps to install the Vision Builder AI Development Toolkit.



Note You must install LabVIEW 8.0.1 before installing the Vision Builder AI Development Toolkit.



Note To install the Vision Builder AI Development Toolkit on a Windows 2000/XP system, you must be logged in with administrator privileges.

1. When installing Vision Builder AI, select the **Custom** installation option instead of the Complete installation option.
2. In the tree under Vision Builder for Automated Inspection, select **Development Toolkit** and select **Install this feature to the local drive**.
3. Click **Next**.
4. Follow the setup instructions on your screen.

The Vision Builder AI Development Toolkit files are installed in the <LabVIEW>\project\Vision Builder AI 3.0 directory.

Installing the Toolkit When Vision Builder AI is Already Installed

Complete the following steps to install the Vision Builder AI Development Toolkit on a system that already has Vision Builder AI installed.



Note You must install LabVIEW 8.0.1 before installing the Vision Builder AI Development Toolkit.

1. Navigate to the Windows **Add or Remove Programs** utility, which is located on the Control Panel.
2. Select **National Instruments Software** and click **Change**.

3. Select **NI Vision Builder AI 3.0** and click **Modify**.
4. In the tree under Vision Builder for Automated Inspection, select **Development Toolkit** and select **Install this feature to the local drive**.
5. Click **Next**.
6. Follow the setup instructions on your screen.

Related Documentation

The following documents contain information that you may find helpful as you work with the Vision Builder AI Development Toolkit:

- *NI Vision Concepts Manual*
- *NI Vision for LabVIEW User Manual*
- *NI Vision for LabVIEW VI Reference Help*
- *NI Vision Builder for Automated Inspection: Configuration Help*
- *NI Vision Builder for Automated Inspection: Inspection Help*

Creating a Custom Step

Complete the following steps to create a custom step:

1. Launch LabVIEW 8.0.1.
2. Close any open VIs.
3. Select **Tools»Vision Builder AI 3.0»Create Custom Step**.
4. Type a **Step Name** for the custom step.
5. Enter a **Description** for the custom step. The description is displayed next to the step name in the Vision Builder AI Inspection Steps palette.
6. Select one of the following template sources:
 - **Custom step template**
 - **Existing custom step you previously created**
7. Click **Next**.
8. If you selected **Custom step template** for the template source, select one of the following step types based on the demonstrated feature you want to use:
 - **Simulated Acquisition Step**—Acquires or creates an image and makes it available to other steps for processing. This step type demonstrates how to create an image acquisition step.



Note You must always start from this step template when you create a custom step that acquires images.

- **Simple Processing Step**—Processes the image from the previous step but does not return measurements other than the pass/fail information. This step type demonstrates image processing.
- **Processing Step that Logs Measurements**—Processes an image from the previous step, makes measurements in the image, and makes the measurements available to subsequent steps. This step type demonstrates image processing and the creation of measurements.
- **Generate a Report**—Provides access to the measurements produced by previous steps. You can use previous measurements to create new measurements and create a customized pass/fail condition. This step type demonstrates accessing measurements produced by previous steps.
- **Global Step Status**—This step passes inspection if all of the previous steps passed. Otherwise, the step fails inspection. This step does not require NI Vision. If you are developing a custom step, and you do not have NI Vision installed, select this template. This step demonstrates how to use the Vision Builder AI Development Toolkit without NI Vision.
- **Coordinate System**—This step demonstrates using coordinate systems to reposition regions of interest.



Tip In addition to using the custom step templates to create your custom step, you can use the custom step templates to help determine how you design your custom step. For example, if you are creating a custom step that combines image processing and creating image masks, you might create the custom step using the Simple Processing template and refer to the Coordinate System template for ideas about how to fill out the remainder of the code to design your custom step to include creating coordinate system support.

9. If you selected **Existing custom step you previously created** for the template source, select the previously created step you want to copy.
10. Click **Next** and verify the setup information.



Tip Close all open VIs before you finish creating the custom step to ensure that the custom step is created successfully.

11. Click **Finish** to create the custom step.

The Create Custom Step wizard opens the XXXX - All VIs VI, where XXXX is the step name you entered for the custom step. This VI provides a project window that contains the VIs you can modify for your custom step. Refer to the [XXXX - All VIs VI](#) section for information about the XXXX - All VIs VI.



Note Your custom step is not available in Vision Builder AI until you save it for distribution. National Instruments recommends that you debug your custom step before you save it for distribution. Refer to the [Debugging Custom Steps](#) section for information about debugging your custom step. Refer to the [Distributing Custom Steps](#) section for information about saving your custom step for distribution.

Custom Step Files

The Create Custom Step wizard creates the following files in the <Vision Builder AI 3.0>\UserPlugins folder:

- 59 XXXX.llb—Source code for your custom step
- XXXX.tif—Icon for your custom step

In each of these file names, XXXX represents the name of your custom step.

Modifying the Custom Step Icon

Complete the following steps to modify the custom step icon:

1. Launch a graphics application, such as Microsoft Paint.
2. Navigate to <Vision Builder AI 3.0>\UserPlugins\XXXX.tif.
3. Open the image.
4. Modify the image as necessary and save it.



Note You must save the file with the .tif file extension.



Note The icon image must be 32 × 32 pixels.

Customizing the Generated VIs

When you create a custom step, the Create Custom Step wizard creates several VIs that provide the functionality of the custom step and installs them to the <Vision Builder AI 3.0>\UserPlugins\59 XXXX.llb file. Table 1 lists the custom step VIs you can modify. Use the block diagram of the XXXX - All VIs VI to access these VIs.



Note Do *not* modify any custom step VIs not listed in Table 1.

Table 1. Custom Step VIs

VI	Description
XXXX - All VIs.vi	<p>Project VI that contains all the VIs you can customize as well as utility VIs you can use in your step. This VI is a starting point to easily access all of the VIs for the custom step.</p> <p>Note: Add any VIs you create and call dynamically to the block diagram of the XXXX - All VIs VI.</p> <p>Tip: This VI helps you organize your VIs and is not designed to be executed.</p>
XXXX - Init Globals.vi	<p>Contains the parameters of the custom step, such as the step name and description, supported image types, available tools, tab location, and help files. The step name for the custom step is the name displayed in the Vision Builder AI tools palette.</p>
XXXX - User Programming.vi	<p>Contains the code called by the Vision Builder AI engine. One of the following three modes is specified for this VI:</p> <ul style="list-style-type: none"> • Setup • Execution • Cleanup <p>Refer to the XXXX - User Programming VI section for information about using the VI in each of these modes.</p>
XXXX - User Interface.vi	<p>Called by the Vision Builder AI engine when the user clicks the custom step icon in the Inspection Steps palette or edits the step already inserted in the script. This VI contains the user interface of the custom step, which is where the user specifies the name and parameters of the step.</p>
XXXX - Parameters.ctl	<p>Contains the type definitions of the user-defined parameters.</p> <p>Note: The parameters the user selects in the user interface are stored in a variant that is passed to the XXXX - User Programming VI. You must use the Variant To Data VI to extract the data from the variant.</p> <p>National Instruments recommends you store the user-defined parameters of the custom step in the XXXX - Parameters type definition so that you can easily update parameters in the custom step code. When you make changes in the type definition, every instance of the parameter is updated in the LabVIEW code.</p>

Complete the following steps to open the custom step VIs:

1. Launch LabVIEW.
2. Click **File>Open**.
3. Navigate to <Vision Builder AI 3.0>\UserPlugins\
59 XXXX.llb, where XXXX is the name of the custom step.
4. Open the LLB.
5. Select XXXX - All VIs.vi and click **OK**.
6. Open the block diagram. The block diagram contains each of the customizable VIs included in the custom step.
7. Modify the custom step VIs as necessary. Refer to the following sections for information about each VI.

Data Flow Among Custom Step VIs

The XXXX - User Programming VI is the core of your custom step. The Vision Builder AI engine sets the value of the Setup variant to the value that you previously set in the custom step User Interface VI.

Figure 1 shows how the custom step configuration data flows from the user interface to the execution of the custom step, which occurs in a loop. The outputs of the XXXX - User Interface VI are saved in the Vision Builder AI engine, which then dispatches the outputs to the inputs of the XXXX - User Programming VI.

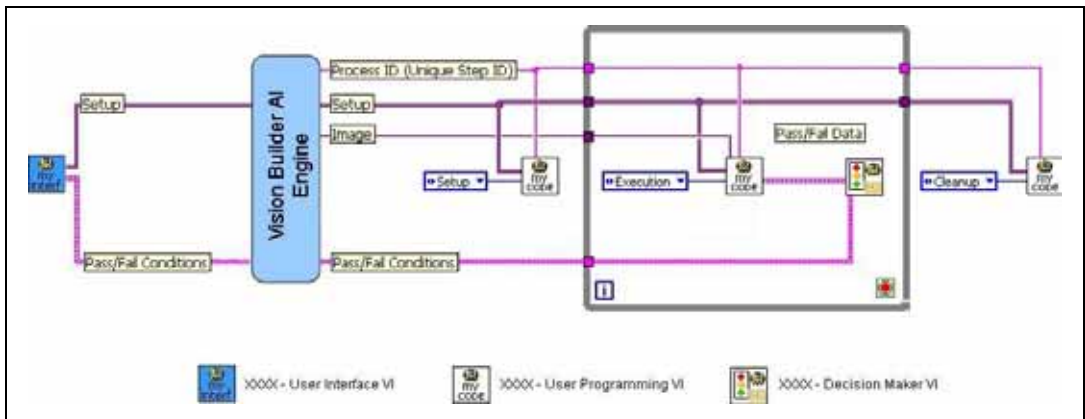


Figure 1. Decision Making Custom Step Block Diagram

The setup mode of the XXXX - User Programming VI is executed when the user validates the user interface of the custom step that is being inserted into the inspection, or when you open an inspection that contains your custom step.

The image, the value of the Setup variant, and the Previous Measurements array are set by the Vision Builder AI engine when the step executes.

When the user closes the script, the Vision Builder AI engine calls the cleanup mode of the XXXX - User Programming VI.

The XXXX - User Interface VI is called when the user selects the step to be inserted or edited. The front panel of this VI is the interface the user sees when configuring a step.

In the XXXX - User Interface VI, you must call the XXXX - User Programming VI in setup mode to perform any initialization before you call the execution mode of the XXXX - User Programming VI. If the user validates a custom step by clicking **OK** in the user interface of the custom step, Vision Builder AI automatically calls the setup and execution modes of the XXXX - User Programming VI, inserting the step in the script.

Complete the following steps to view a block diagram that shows this relationship:

1. Create a custom step using the Simple Processing Step template.
2. Open the XXXX - User Interface VI.
3. View the block diagram.
4. Select frame 3 of the sequence structure.

Setup Variant

The Setup variant stores the parameters of the step. If the step requests more than one parameter, bundle the parameters in a cluster, and use the To Variant VI to store the parameters in the variant. Use the Variant to Data VI to retrieve these parameters in the XXXX - User Programming VI.

Complete the following steps to view a block diagram that shows this relationship:

1. Create a custom step using the Simple Processing Step template.
2. Open the XXXX - User Interface VI.
3. View the block diagram.
4. Select frame 1 of the sequence structure.

Process ID

When an instance of the custom step is created, the Vision Builder AI engine provides a Process ID, which is a string that uniquely identifies the custom step. This string can be used to differentiate multiple instances of the same step.

VI Descriptions

This section contains descriptions of the VIs included in a custom step as well as descriptions of the VI controls, indicators, and parameters that the Vision Builder AI engine uses.

XXXX - All VIs VI

This VI is designed to help you locate and organize all of the VIs you can customize. It contains all the VIs you must modify for your custom step as well as utility VIs.



Tip This VI is not designed to be executed.

The block diagram of this VI is divided into the following four sections:

- **Set Up Global**—Contains the XXXX - Init Globals VI. Open this global first to modify the basic properties of the custom step. Refer to the *XXXX - Init Globals VI* section for more information about this VI.



Note The XXXX - Init Globals VI is connected to a string indicator to ensure that the All VIs VI has no errors. This string indicator serves no other purpose.

- **Custom VIs**—This section contains the VIs you must modify to achieve the processing you want.
- **Utility VIs**—This section contains utility VIs you can use to perform specific functions, such as displaying an image in the main Vision Builder AI window. You cannot modify these VIs.
- **Dynamic VIs**—This section is designed to contain any VIs that you call dynamically. This design helps you organize your custom step and ensure that the dynamic VIs are saved correctly when you build your custom step for distribution.

XXXX - Init Globals VI

This VI contains the following initialization elements.

Table 2. Init Globals VI Elements

Element	Description
Local Name	Name of the step as it appears in the Vision Builder AI Inspection Steps palette. You can modify this name.
Description	Descriptive text that appears next to the custom step in the Vision Builder AI tools palette.

Table 2. Init Globals VI Elements (Continued)

Element	Description
Version	String to help keep track of the custom step version. To check the version, the developer must read the global variable.
Tab Location	Tab of the Vision Builder AI Inspection Tools palette where your custom step is located.
Context Help	File path to your custom help file, which is in the form of compiled HTML help. The file extension is .chm. Refer to the Creating HTML Documentation for the Custom Step section for information about creating help for your custom step.
ROI Tools and Image Support	ROI tools available. Disable the tools you do not want to use. A green LED indicates that the tool appears in the toolbar. A red LED indicates that the tool does not appear in the toolbar.
Localized ROI Tool Name	Display name of the tools selected to be available in the Region of Interest selection. This element is useful for localizing tool names.
Default Tool	Default ROI tool. Choose the tool you want the custom step to select by default.
Show Dynamic ROIs	Show/hide previously created ROIs in the Region of Interest control on the User Interface VI front panel.
Supported Image Types	<p>Image types supported by the custom step. Disable the image types you do not want to use. A green LED indicates that the image type is supported by the step. A red LED indicates that the image type is not supported.</p> <p>Note: An error will occur if the user attempts to apply the step to an unsupported image type.</p> <p>Note: The Supported Image Types control does not appear in the Simulate Acquisition Step.</p>



Note When you modify the controls and indicators in these VIs, you must make the new values the default for the control or indicator and save your changes. To make the current values the default for a control or indicator, select **Edit»Make Current Values Default**, and save the VI.

XXXX - User Programming VI

This VI contains the main processing code of your step. The front panel is never displayed, so it does not have to be modified. Do not modify the connector pane of this VI.

The Vision Builder AI engine sets the value of the Setup variant control, which contains the setup parameters defined by the user in the user interface. Use the Variant To Data VI to access this data. Use the same data structure that you used in XXXX - User Interface VI when you packaged the data in the Setup variant. To ensure you use the same datatype, always use the XXXX - Parameters type definition with the Variant to Data VI.

The VI has three different modes:

- **Setup**—Executed once when the user opens an inspection that contains the custom step or inserts the custom step in the script.
Use this mode to perform any one-time initialization routines, such as reading a pattern matching template or initializing a DAQ or image acquisition device.
- **Execution**—Executed each time the step is called within the Vision Builder AI inspection.
This mode performs the main function of the custom step.
- **Cleanup**—Executed when the user closes an inspection or when the step is removed from the script.
Use this mode to dispose of any resources allocated in the Setup mode. Each instance of the custom step in a script has a unique Process ID. You can use the Process ID to allocate resources specific to an instance of a custom step. For example, you can pass a Process ID to the IMAQ Create VI to allocate a unique image for an instance of a custom step.



Tip The XXXX - Resource Manager VI is designed to help you create, get, and deallocate step specific resources.



Tip You can use a global variable to pass information between the different modes of the VI. If the values of the global variable change between multiple instances of the same custom VI, the global variable is overwritten with the latest value.

Save any global variables created in the 59 XXXX.llb with the step name as the prefix. For example, if you created a global variable called MyGlobal for a step you named Image Processing, save the global as Image Processing - MyGlobals.vi.

Tables 3 and 4 show the User Programming VI controls and indicators.

Table 3. XXXX - User Programming VI Controls

Controls	Description
Image	Processed image from the previous step in the inspection. Tip: This control is used only in Execution mode.
Process ID	Unique ID provided from the Vision Builder AI engine.
Setup	Parameters the user sets in the XXXX - User Interface VI. Use the Variant To Data VI to extract the values stored in this variant.
Mode	The mode of the VI.
ROI Descriptor	The constant ROI to be saved. The array contains the following elements: <ul style="list-style-type: none">• Global Rectangle—Contains the coordinates of the bounding rectangle.• Contours—Each of the individual shapes that define an ROI.

Table 4. XXXX - User Programming VI Indicators

Indicators	Description
Image Out	<p>Processed image to be provided to the next step in the script. If you are not processing the image, you must still connect Image In to Image Out to ensure that the other steps in the script receive the image.</p> <p>Tip: This control is used only in Execution mode.</p>
Measurements	<p>Measurements created by your step, which can be passed to subsequent steps. This cluster contains the following elements:</p> <ul style="list-style-type: none"> • Value—Flattened string of the Measurements value. You can store measurements only in the following three LabVIEW data types: double, Boolean, and string. If you choose a double, Value must be the flattened string of a double. If you choose Pass Fail or Boolean for the Type element, Value must be the flattened string of a Boolean. If you choose ID or String in the Type element, Value must be the flattened string of a LabVIEW string. • Type—Data type of the data. • Name—Measurement name. • Unit—Measurement unit. <p>Tip: This control is used only in Execution mode.</p>
Pass/Fail Data	<p>Pass/fail data that the Decision Maker VI uses to make the pass/fail decision for the step. The Pass/Fail Data array must contain the same number of elements as the Pass/Fail Conditions out indicator on the user interface. The Decision Maker VI applies the pass/fail condition information from the user interface to the pass/fail condition information of the XXXX - User Programming VI.</p> <ul style="list-style-type: none"> • Pass/Fail Data—Number that the Decision Maker VI compares to the pass/fail condition information from the user interface. The Vision Builder AI engine compares the Pass/Fail Data number to Value 1 and Value 2 of the Pass/Fail Condition out cluster of the XXXX - User Interface VI. The engine bases this comparison on the function specified in Pass/Fail Condition out. • Fail String—Strings that are displayed in the script window of the user interface when a step fails. <p>Tip: This control is used only in Execution mode.</p>

XXXX - User Interface VI

The XXXX - User Interface VI is executed when a user selects the custom step from the Vision Builder AI Inspection Steps palette or when the user opens a step already inserted in the script to edit parameters. Table 5 lists the controls for this VI.

Table 5. XXXX - User Interface VI Controls

Controls	Description
Name in	Name of the step.
Process ID	Unique ID provided from the Vision Builder AI engine.
Image	Processed image from the previous step in the script.
Setup in	Variant that contains the step parameters that were entered during editing and saved when the user clicked OK and inserted the step into the script. National Instruments recommends you use XXXX - Parameters to save a type definition of the parameters in the step because the type definition is used in several places. If you use a type definition, you make the change in just one place when you need to change the parameter type. Use the Variant To Data VI to access data stored in Setup in . Bundle multiple parameters in a cluster and use the To Variant VI to package the cluster in the Setup out indicator.
Edit Mode	False indicates a new instance of the custom step. True indicates that the step is being edited by the user. For example, if a user double-clicks a step that is already in the script, Edit Mode is True.

Table 5. XXXX - User Interface VI Controls (Continued)

Controls	Description
<p>Pass/Fail Condition in</p>	<p>Pass/fail conditions of the custom step. This array contains the following elements:</p> <ul style="list-style-type: none"> • Used—Boolean that indicates if the pass/fail condition is used. • Value 1, Value 2—Limit values. These values are compared against the Pass/Fail Data produced by the XXXX - User Programming VI in execution mode, depending on the relational operator function selected. • Function—The relational operator function. The following list includes valid values: <ul style="list-style-type: none"> – Less—Compares the pass/fail data set in the XXXX - User Programming VI to Value 1 to determine if it is less than Value 1. – Less or Equal—Compares the pass/fail data set in the XXXX - User Programming VI to Value 1 to determine if it is less than or equal to Value 1. – Greater—Compares the pass/fail data set in the XXXX - User Programming VI to Value 1 to determine if it is greater than Value 1. – Greater or Equal—Compares the pass/fail data set in the XXXX - User Programming VI to Value 1 to determine if it is greater than or equal to Value 1. – In Range—Determines if the pass/fail data set in the XXXX - User Programming VI is in between Value 1 and Value 2. <p>The user can specify the pass/fail condition on the custom step user interface. This information is passed to the XXXX - Decision Maker (Float) VI, which compares the function results to the pass/fail criteria.</p> <p>The pass/fail condition is passed to the custom step user interface so that you can display the information in the appropriate controls on the front panel and store new values when the user modifies the pass/fail condition on the user interface.</p> <p>The Create Custom Step wizard provides two templates that demonstrate the functionality of the Pass/Fail Conditions controls and indicators for the following types of custom steps:</p> <ul style="list-style-type: none"> • Processing Step that Logs Measurements. • Global Step Status. <p>You can modify other types of custom steps to use the Pass/Fail Conditions controls and indicators. Refer to the Creating a Custom Step section for more information about the template types.</p>

Table 5. XXXX - User Interface VI Controls (Continued)

Controls	Description
ROI Event	List of available ROI tools and created ROIs. The ROI tool will automatically change when selected.
Previous Measurements	<p>Measurements of the previous steps. This array of clusters has as many elements as there are steps in the script. Each element of the array describes the measurements logged by the corresponding step.</p> <p>A cluster contains the following elements:</p> <ul style="list-style-type: none"> • Step Name—Name of the step. • Measurement IDs—Array of clusters describing the measurements logged by the step named Step Name. <p>Note: Use the VBAI SDK Get Result VI to access information about the measurement, such as the value, type, name, and unit. If you need to use the Previous Measurements control in your step, save the Measurement ID in the Setup variant, which you can access from the User Programming VI. Use the VBAI SDK Get Result VI to get the value of the Measurement ID when the User Programming VI is executed. For examples on how to use Previous Measurements, create the Generate a Report template or the Global Step Status template.</p>
Reference CoordSys Array	List of available coordinate systems. Use this in conjunction with the Coordsys Utilities to manage coordinate systems.
ROI Descriptor	<p>The constant ROI to be saved. The array contains the following elements:</p> <ul style="list-style-type: none"> • Global Rectangle—Contains the coordinates of the bounding rectangle. • Contours—Each of the individual shapes that define an ROI.
Cancelled	Boolean that indicates if the user cancelled the step.

The XXXX - User Interface VI returns the following data.

Table 6. XXXX - User Interface VI Indicators

Indicators	Description
Name out	Name the user entered for the step.
Setup out	Variant that contains the parameters of the step. If the step requires more than one parameter, bundle the parameters in a cluster, and use the To Variant VI to package the parameters in the variant.

Table 6. XXXX - User Interface VI Indicators (Continued)

Indicators	Description
Pass/Fail Condition out	Pass/fail conditions of the custom step.
ROI Descriptor Out	The constant ROI to be saved. If a programmable ROI was selected, the ROI will be saved automatically.



Note National Instruments recommends that you do not resize the tab control on the user interface or change its position. Changing the tab control on the custom step user interface might move the user interface controls and indicators to a position that the user cannot access.



Note When the user is configuring the custom step in Vision Builder AI and has the custom step in Edit mode, the Run Once button does not update the image display or step parameters. To update the display and parameters, run the custom step outside of Edit mode.

XXXX - Parameters.ctl

This is the type definition of the parameters of the custom step.

National Instruments recommends you store the user-defined parameters of the custom step in a type definition so that you can easily update parameters in the custom step code. When you make changes in the type definition, every instance of the parameter is updated in the custom step code.



Tip Bundle multiple parameters in a cluster.

Utility VIs

Utility VIs are available to help complete necessary tasks in your custom step. These VIs can be accessed in the XXXX - All VIs block diagram and are located at `<Vision Builder AI 3.0>\Plugins\Common SDK VIs.llb`.

VBAI Display in Main Window VI

Use this VI in the XXXX - User Interface VI to display an image in the Main window of Vision Builder AI. This VI is used only in the XXXX - User Interface VI to interactively show changes in the image after parameter values are changed.

VBAI Resource Manager VI

This VI creates a new image for each instance of a custom step. The name of the image is the unique identifier of the step. The VI also allows you to retrieve the image and dispose it. If an image is not created from a unique name, the same image buffer will be accessed for each instance of the custom step.

VBAI CoordSys Name Utility VI

Use this VI to retrieve the coordinate system ID, coordinate system index, or coordinate system ID from the coordinate system name.

When outputting the chosen coordinate system from the user interface, pass the coordinate system ID through the process setup. Passing the coordinate system ID ensures the correct coordinate system is selected, even when the coordinate system is renamed.

VBAI CoordSys ID Utility VI

Use this VI to retrieve the coordinate system ID, coordinate system index, or coordinate system ID from the coordinate system name. This VI is also used to update the user interface from the saved coordinate system ID.

VBAI - Decision Maker (Float) VI

This VI returns pass/fail information by applying the pass/fail conditions of the custom step user interface to the pass/fail conditions data of the XXXX - User Programming VI. Use the VBAI - Decision Maker VI only in the XXXX - User Interface VI so that the user can see immediate pass/fail results when modifying the pass/fail conditions. Outside of the user interface, the decision making functionality is performed automatically.

VBAI Get Global Variables VI

This VI returns the user-defined globals and system globals that are currently defined.

VBAI SDK Get Result VI

This VI returns result information, such as the name and value of the result, that corresponds to a Measurement ID.

Setting Up Limit Conditions for a Custom Step

In the **Processing Step that Logs Measurements** template, you can set up limit conditions that define when an object under inspection fails the inspection. The pass/fail conditions specified by the XXXX - User Interface VI are compared to the pass/fail data returned from the execution mode of the XXXX - User Programming VI. You must modify the following VIs to set up pass/fail conditions for your custom step:

- XXXX - User Interface VI
- XXXX - User Programming VI

XXXX - User Interface VI

Complete the following steps to set up a limit condition in the XXXX - User Interface VI:

1. Open the XXXX - User Interface VI.
2. Click the **Limits** tab to display it. The **Pass Inspection if** control is a cluster of data necessary to configure your pass/fail condition.
3. (Optional) In the control, enter values for **Value 1** and **Value 2**.
4. Set the **Function** control to one of the following values:
 - Less
 - Less or Equal
 - Greater
 - Greater or Equal
 - In Range
 - Equal
 - Not Equal
5. To set up more limit conditions, copy and paste the **Pass Inspection if** control.
6. Build an array with all the **Pass/Fail Condition** clusters to handle exiting the custom step if the user does not cancel the custom step. Set the **Pass/Fail Conditions out** indicator on the last frame of the XXXX - User Interface VI block diagram, as shown in Figure 2.

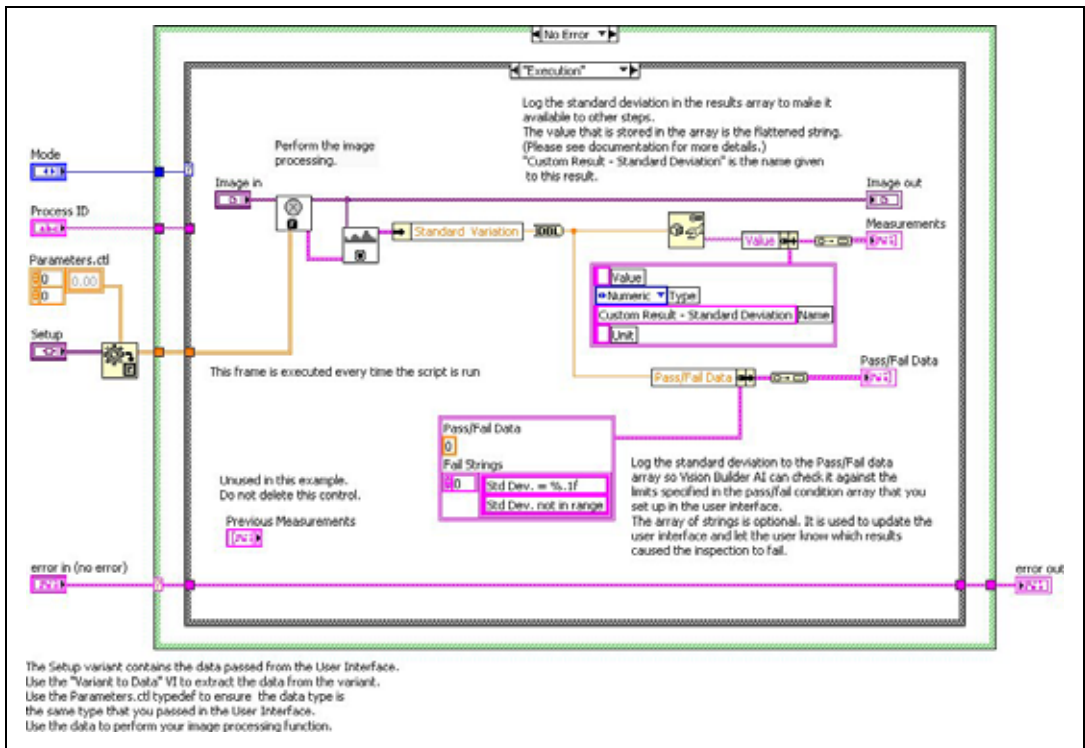


Figure 3. XXXX - User Programming Block Diagram for Setting Up Pass/Fail Handling

3. Build all of the clusters into a single array and pass the array to the Pass/Fail Data indicator. Match the array order with the array order set by the Pass/Fail Conditions in the XXXX - User Interface VI.

Logging Measurements from Custom Steps

You can use the Measurements indicator in the Execution mode of the XXXX - User Programming VI to log double, Boolean, and string measurements of the custom step for use in subsequent steps. The Measurements indicator is an array of clusters that contain the elements listed in Table 7.

Table 7. Measurements Indicator

Element	Description/Data Type
Value	Flattened string of a double, Boolean, or string. Use the Flatten to String VI to create this parameter.
Type	<p>One of the following predefined Vision Builder AI types:</p> <ul style="list-style-type: none"> • Pass Fail—Do not use. • X Position (Pix)—Use this double to log an x-coordinate point. • Y Position (Pix)—Use this double to log a y-coordinate point. • X Position (World)—Do not use. Vision Builder AI automatically creates the Real World coordinates of points if the image is calibrated. • Y Position (World)—Do not use. Vision Builder AI automatically creates the Real World coordinates of points if the image is calibrated. • Score—Double • Distance (Pix)—Double • Distance (World)—Double • Angle—Double • Straightness—Double • Radius (Pix)—Double • Radius (World)—Double • Roundness—Double • Average Intensity—Double • Std Dev Intensity—Double • Minimum Intensity—Double • Maximum Intensity—Double • # of Holes—Double • Boolean—Boolean • Numeric—Double • String—String • Area (Pix)—Double • Area (World)—Double • Percentage of Area—Double • ID—String • # of Matches—Double • # of Objects—Double • # of Edges—Double • Orientation—Double • Aspect Ratio—Double
Name	Description of the measurement.
Unit (optional)	Unit of measure for the measurement.

Importing Vision Builder AI 2.6 Custom Steps

Complete the following steps to import steps created for Vision Builder 2.6.

1. Launch LabVIEW 8.0.1.
2. Close any open VIs.
3. Select **Tools»Vision Builder AI 3.0»Import Vision Builder AI 2.6 Custom Step**.
4. Custom steps found in the <Vision Builder 2.6>\UserPlugins directory will be listed in the **Existing Custom Steps Previously Created** textbox. Select the step you want to import.
5. Click **Next**.
6. Click **Finish**.

The Import Vision Builder AI 2.6 Step wizard creates the following files in the <Vision Builder AI 3.0>\UserPlugins folder:

- 59 xxxx.llb—Source code for your custom step
- xxxx.tif—Icon for your custom step

In each of these file names, xxxx represents the name of your custom step.

Debugging Custom Steps

Complete the following steps to debug a custom step:

1. Launch LabVIEW.
2. Open the custom step VIs you want to debug.
3. To debug custom steps, you must enable TCP/IP support for the LabVIEW VI Server. Complete the following steps to enable TCP/IP support.
 - a. In LabVIEW, select **Tools»Options**.
 - b. Select **VI Server: Configuration** from the Category list.
 - c. Enable the **TCP/IP** checkbox.
 - d. Enable all of the **Accessible Server Resources** options.
 - e. Select **VI Server: Machine Access** from the Category list.
 - f. Verify that the **Machine access list** includes the localhost, or use * to allow any machine to use the VI Server.
 - g. Select **VI Server: Exported VIs** from the Category list.
 - h. Verify that the **Exported VIs** list includes *Proxy(Target).vi, or use * to allow all VIs to be exported.
 - i. Click **OK**.

4. Click **Tools»Vision Builder AI 3.0»Test Your Custom Step in Vision Builder AI** to launch Vision Builder AI and test the step.



Note You must open Vision Builder AI from LabVIEW to test a custom step that has not been saved for distribution. Refer to the *Distributing Custom Steps* section for information about saving custom steps for use with the Vision Builder AI executable.



Note When Vision Builder AI is launched from LabVIEW, do not create a new custom step or open an existing custom step while Vision Builder AI is open.

5. Open the Vision Builder AI Configuration interface.
6. Open your custom step. Your custom step is located in the tab you specified in the **Tab Location** control of the XXXX - Init Globals VI. The default location for custom steps is the **Use Additional Tools** tab.



If you did not modify the custom step icon, the custom step displays the default custom step icon.

7. Once you open your custom step, Vision Builder AI will begin operating in debugging mode. Clicking the **Exit** button will close debugging mode and you will not be able to debug your custom step unless you open your custom step again.
8. The front panel of the User Interface VI for your custom step will open in LabVIEW. Press <Ctrl-E> or select **Window»Show Block Diagram** to open the block diagram and begin debugging your custom step.



Note The front panel of the User Interface VI may open behind the Vision Builder AI window. Open the User Interface VI by clicking the VI window in the Taskbar.

9. Standard LabVIEW debugging tools are available to debug your custom step. Select the appropriate debugging options and breakpoints in the custom step VI.



Tip You can exit debugging mode by clicking **Abort Execution** in the LabVIEW toolbar, by clicking **Cancel** in the custom step User Interface VI front panel, or by clicking the **Exit** button in Vision Builder AI.

Distributing Custom Steps

You can save a custom step for distribution. This allows your customers to call the custom step from the executable version of Vision Builder AI.

Complete the following steps to prepare a custom step for distribution:

1. Launch LabVIEW.
2. Click **Tools»Vision Builder AI 3.0»Save Your Custom Step for Distribution**.
3. Select the custom step you want to save for distribution.
4. Enable the **Build Step for RT Target** control if you want to use the custom step with a LabVIEW Real-Time system, such as the NI CVS-1450 Series compact vision system.

You can set the Save Your Custom Step for Distribution wizard to automatically download the custom LLB to the real-time target after the LLB is created by enabling the **Download Step to RT Target** control.



Note After downloading the step to the RT target, you must reboot the device to register the new step.

5. If you selected **Download Step to RT Target**, specify the IP address of the real-time target.
6. Click **Build** to save the custom step for distribution.
7. Click **OK** to close the Save Your Custom Step for Distribution wizard.

The Save Your Custom Step for Distribution wizard builds 59 XXXX.bin in the <Vision Builder AI 3.0>\UserPlugins folder. When distributing this step, distribute this BIN file with its corresponding .tif file.



Note When you build the plug-in step for an RT Target, the built LLB is saved under <Vision Builder AI 3.0>\UserPlugins\RT\59 XXXX.llb. If you did not enable the Download Step to RT Target control, you must manually copy the LLB to the c:\VBAI\Plugins folder of the RT Target.

Creating Documentation for the Custom Step

You may want to provide documentation for your custom step so that the users can access information about how to use the step. To add HTML documentation for your custom step, complete the following steps:

1. Create HTML documentation for the custom step.
2. Download and install HTML Help Workshop.
3. Modify the XXXX - Init Globals VI to call the HTML file(s) you created for the custom step.

Refer to the following sections for instructions for each of these procedures.

Creating HTML Documentation for the Custom Step

Use any text editor or HTML editor to create HTML documentation for the custom step. Save the file with an `.html` extension.

Although not required, you can create three help files for the custom step to be consistent with the Vision Builder AI online help. These three help files include one describing configuration procedures, one describing the controls, and one listing frequently asked questions regarding the custom step.

Downloading and Installing HTML Workshop

Use Microsoft HTML Help Workshop to create a compiled HTML help file that you can hook into your custom step. Adding documentation to your custom step provides a professional appearance for the step, and helps your users use the step correctly.



Tip A compiled HTML help file has a file extension of `*.chm`.

Parts of HTML Help Workshop are components of the Windows operating system. However, unless you routinely develop HTML documentation, you probably do not have the HTML Help Workshop components necessary to develop HTML help.

To determine if you have the components, select **Start>Run**, type `hhw.exe`, and click **OK**. If you get an error indicating the application could not be found, download and install Microsoft HTML Workshop from <http://msdn.microsoft.com>. If HTML Help Workshop loads, go to the steps in the *Modifying the XXXX - Init Globals VI to Call the Custom Step Documentation* section.



Tip For information about creating compiled HTML help, refer to *Help for HTML Help*, which is available from the **Help** menu in HTML Help Workshop.

Modifying the XXXX - Init Globals VI to Call the Custom Step Documentation

You can add the custom step documentation to the Vision Builder AI user interface so your customers can access the custom step documentation the same way they access the other Vision Builder AI help files.

Complete the following steps to add the custom step documentation to the Vision Builder AI user interface:

1. Save the `.chm` file to the `<Vision Builder AI 3.0>\Help` folder.

2. Launch LabVIEW.
3. Click **Open VI**. Navigate to <Vision Builder AI 3.0>\UserPlugins\59 XXXX.llb, where XXXX represents the name of the custom step.
4. Select **XXXX - Init Globals.vi** and click **OK**.
5. Type the .chm file name and the HTML page in the elements of the **Context Help** array. For example, if your help file name is myhelp.chm and the HTML file name of the page you that contains configuration information is config.html, type myhelp.chm\
config.html in the first element of the array.

Each element of the array is the relative path of the HTML page for the associated tab. For example, the first element is the relative path of the HTML file you want to display for the **Configuration** tab. These tabs appear in the context help window of Vision Builder AI. Refer to Figure 4 for an example of the context help window.

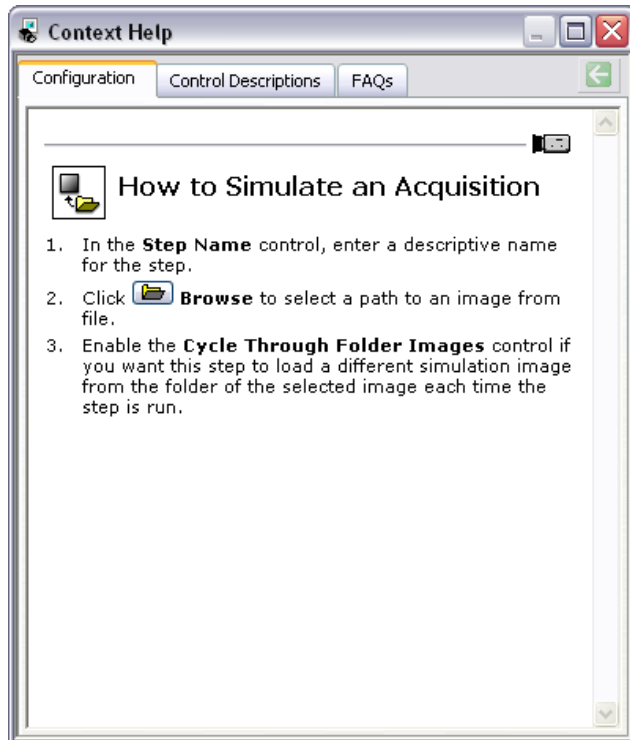


Figure 4. Context Help Window in Vision Builder AI



Note If you add at least one file to the **Context Help** array, make sure all three elements of the array are populated, duplicating the file name if necessary. If you add a file name to one element and leave one or more of the remaining elements blank, Vision Builder AI displays an error indicating that a page was not found for the tabs that are represented by the blank elements.

For example, if you created an HTML file that covers configuring the custom step and you add that file name to the first element in **Context Help** but leave the other two elements blank, the **Control Descriptions** and **FAQs** tabs in Vision Builder AI display an error indicating a page was not found for the custom step.

6. Click **Edit»Make Current Values Default**.
7. Save and close the VI.
8. Click **Tools»Vision Builder AI 3.0»Test Your Custom Step in Vision Builder AI**.
9. Navigate to the appropriate tab and click the custom step to view the newly added documentation.