FPGA Interface Functions

June 2008, 370960E-01

Installed With: FPGA Interface. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the FPGA Interface functions to communicate with an FPGA VI from a host VI. The VI that runs on an FPGA target is called the FPGA VI. A host VI is a VI that communicates with the FPGA VI to control the FPGA target. A host VI can run on a Windows computer or on an RT target.

You can use the FPGA Interface functions to programmatically control and communicate with an FPGA VI. Use the FPGA Interface functions to perform the following operations in host VIs:

- Establish and terminate communication with the FPGA VI.
- Download, abort, reset, and run the FPGA VI on the FPGA target.
- Read and write data to the FPGA VI.
- Wait for and acknowledge FPGA VI interrupts.
- Read DMA FIFOs.



Note This palette is available only when you edit a VI under **My Computer** or an RT target in the **Project Explorer** window.

The functions on this palette can return <u>general LabVIEW error codes</u>, specific FPGA Interface error codes, or error codes specific to the FPGA target.

Palette Object	Description
Close FPGA VI Reference	Closes the reference to the FPGA VI and, optionally, resets or aborts execution of the VI.
Invoke Method	Invokes an FPGA Interface method or action from a host VI on an FPGA VI. Use methods to do the following: download, abort, reset, and run the FPGA VI on the FPGA target, wait for and acknowledge FPGA VI interrupts, read DMA FIFOs, and write to DMA FIFOs. The methods you can choose from depend on the target hardware and the FPGA VI. You must wire the FPGA VI Reference In input to view the available

	methods in the shortcut menu.
Open FPGA VI Reference	Opens a reference to the FPGA VI or bitfile and FPGA target you specify. Right-click the Open FPGA VI Reference function and select Configure Open FPGA VI Reference from the shortcut menu to display the Configure Open FPGA VI Reference dialog box.
	Reads a value from or writes a value to a control or indicator in the FPGA VI on the FPGA target.

Subpalette	Description
Advanced Function	Includes the Up Cast function.

 $[\]hfill @$ 2003–2008 National Instruments Corporation. All rights reserved.

Close FPGA VI Reference Function

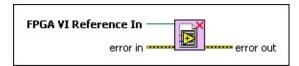
Owning Palette: FPGA Interface Functions

Installed With: FPGA Interface

Closes the reference to the FPGA VI and, optionally, resets or aborts

execution of the VI.

The Close FPGA VI Reference function also stops all <u>DMA FIFOs</u> on the FPGA.



- Place on the block diagram Find on the Functions palette
- **FPGA VI Reference In** is the reference to the FPGA VI running on the FPGA target. You must open a reference to the FPGA VI to use this parameter.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally even if an error occurred before this VI or function runs. 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 exception control to treat what is normally an error as no error or to treat a warning as an error. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

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

Close FPGA VI Reference Details

The Close FPGA VI Reference function always closes the reference to the FPGA VI. To configure this function to only close the reference, right-click the function and select **Close** from the shortcut menu.

When the FPGA target is set to **Execute VI on FPGA Target**, you also can right-click the Close FPGA VI Reference function and select **Close** and **Reset if Last Reference** from the shortcut menu to stop and reset the FPGA VI running on the FPGA target if the function is invoked for the last reference to the VI.

When the FPGA target is set to **Execute VI on Development Computer**, you also can right-click the Close FPGA VI Reference function and select **Close and Abort without Reference Counting** from the shortcut menu to abort the FPGA VI. Reference counting is not available in this case, so the function always aborts the FPGA VI.

When you change the target configuration from execution on the FPGA target to execution on the development computer, the FPGA Module translates Close and Reset if Last Reference as Close and Abort without Reference Counting. When you change the target configuration from execution on the development computer to execution on the FPGA target, the FPGA Module translates Close and Abort without Reference Counting as Close and Reset if Last Reference.

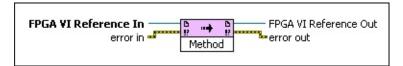
There are differences in behavior between resetting the VI executing on the FPGA target and aborting the VI executing on the development computer. Resetting the VI on an FPGA target sets VI controls and indicators to their default states, sets uninitialized shift registers to their default values, clears FIFOs, and sets global variables to their default values. Aborting the VI on the development computer does not reset the default values in the FPGA VI. The behavior of aborting the VI on the development computer depends on whether that VI remains in memory after a reference to the VI is closed. If the VI is not in memory, the next time you open a reference to that VI, the VI behaves as if it was just loaded from disk, with all of the relevant values set to defaults. If the VI is in memory, the next time you open a reference to the VI, the VI is in with whatever internal state it happens to be in. The VI remains in memory if the front panel is open, the VI is a subVI of another VI, or there are other references to the VI.

Invoke Method Function

Owning Palette: FPGA Interface Functions

Installed With: FPGA Interface

Invokes an FPGA Interface method or action from a host VI on an FPGA VI. Use methods to do the following: download, abort, reset, and run the FPGA VI on the FPGA target, wait for and acknowledge FPGA VI interrupts, read DMA FIFOs, and write to DMA FIFOs. The methods you can choose from depend on the target hardware and the FPGA VI. You must wire the FPGA VI Reference In input to view the available methods in the shortcut menu.



- Place on the block diagram Find on the **Functions** palette
- **FPGA VI Reference In** is the reference to the FPGA VI running on the FPGA target. You must open a reference to the FPGA VI to use this parameter.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs except where noted otherwise. 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 exception control to treat what is normally an error as no error or to treat a warning as an error. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.

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

Invoke Method Details

You can invoke the following FPGA Interface methods on most FPGA targets to control the FPGA VI. You might have fewer or more methods available depending on the FPGA target. Refer to the specific FPGA target hardware documentation for information about the FPGA Interface methods you can use.

- FIFO—Displays one of the following methods you can use to read from or write to a <u>Direct Memory Access (DMA) FIFO</u> in the FPGA VI. FIFO is the name of the FIFO item in the project.
 - **Configure**—Allows you to specify the depth of the host memory part of the DMA FIFO. This method is optional.
 - Depth specifies the number of elements in the host memory part of the DMA FIFO. If you do not wire this parameter, the Invoke Method function uses a default of 10,000 elements. If you place the FIFO Configure method after a FIFO Start or FIFO Read method in the data flow, the Invoke Method function sets the new depth when the next FIFO Start or FIFO Read method executes.
 - Start—Begins DMA data transfer between the FPGA target and the host computer. This method is optional. The FIFO Read and FIFO Write methods automatically start DMA data transfer. You might want to use this method if you want to start data transfer with the DMA FIFO before you read the first element of the FIFO.
 - Read—Reads elements of the DMA FIFO from the host memory part of the FIFO. The Read method returns Data when the Number of Elements is available or when the Timeout (ms) period ends.
 - Note The Read method is available only if the FIFO Type is Target to Host—DMA in the General FPGA FIFO Properties page.
 - Number of Elements determines the number of elements you read from the DMA FIFO.
 - Timeout (ms) specifies the number of milliseconds

the Invoke Method function waits before timing out. The default is 5000 milliseconds. Set this parameter to -1 if you want the Invoke Method function to wait indefinitely for the number of elements.

- **Data** returns the data contained in the host memory part of the DMA FIFO.
- Elements Remaining returns the number of elements remaining in the host memory part of the DMA FIFO.
- Write—Writes elements to the DMA FIFO from the host
 VI. The Write method returns Empty Elements
 Remaining when the data is written or when the Timeout (ms) period ends.
 - Note The Write method is available only if the FIFO Type is Host to Target—DMA in the General FPGA FIFO Properties page.
 - **Data** specifies the data that you want to transfer to the FPGA target.
 - Timeout (ms) specifies the number of milliseconds the Invoke Method function waits before timing out. The Invoke Method function times out if the host part of the FIFO does not contain enough space to write **Data** to by the time the number of milliseconds you specify elapse. The default is 5000 milliseconds. Set this parameter to –1 if you want the Invoke Method function to wait indefinitely.
 - Empty Elements Remaining returns the number of empty elements remaining in the host memory part of the DMA FIFO.
- Stop—Stops the DMA data transfer between the FPGA target and the host computer. This method empties all data from the host memory and FPGA parts of the FIFO. This method is optional. Most applications do not require using the Stop method. If an error occurred before this

method runs, the method runs normally and passes the **error in** value to **error out**. If an error occurs while this method runs, the method runs normally and merges **error in** and its own error status to produce **error out**.

- **Run**—Runs the FPGA VI on the FPGA target. If the FPGA VI is already running on the FPGA target, the Run method does nothing.
 - **Wait Until Done (F)** makes the Invoke Method function wait until the FPGA VI finishes running. If you set this parameter to TRUE, make sure the FPGA VI terminates execution on its own.
- Abort—Aborts the opened and running FPGA VI on the FPGA target. This method does not reset the default values in the FPGA VI. If an error occurred before this method runs, the method runs normally and passes the error in value to error out. If an error occurs while this method runs, the method runs normally and merges error in and its own error status to produce error out.
- Reset—Aborts and resets the FPGA VI on the FPGA target to the
 default state of the VI. This method sets the FPGA VI controls
 and indicators to their default states, sets uninitialized shift
 registers to their default values, clears FIFOs, and sets global
 variables to their default values. This method does not reset
 memory. If an error occurred before this method runs, the method
 runs normally and passes the error in value to error out. If an
 error occurs while this method runs, the method runs normally
 and merges error in and its own error status to produce error
 out.
- Wait on IRQ—Waits for any number of interrupt requests you included in the compiled FPGA VI running on the FPGA target. Always acknowledge interrupts after they occur using the Acknowledge IRQ method. You can use several calls to the Wait on IRQ method to implement waiting on different interrupts from different places in a VI. If you do so, National Instruments recommends that you specify non-overlapping interrupts for different Wait on IRQ method calls.



Note The Wait on IRQ method consumes threads. If you use too many calls to the Wait on IRQ method, other code

in the application might stop executing until an interrupt occurs. If you notice unexpected execution behavior, try reducing the number of calls or put the Wait on IRQ methods in subVIs in different execution systems. Use the Execution Properties page to specify the execution system.

- IRQ Number(s) specifies the logical interrupt or array of logical interrupts for which the function waits. The default is 0. Typical supported values are 0 through 31.
- Timeout (ms) specifies the number of milliseconds the VI waits before timing out. Wire a –1 for an infinite timeout. The default is 0. If you do not wire this parameter and no interrupt is received, the Invoke Method function times out immediately.
- Timed Out returns TRUE if this method has timed out.
- IRQ(s) Asserted returns the asserted interrupts. If you are waiting for a single interrupt, a value of -1 indicates that the interrupt was not received. If you are waiting for multiple interrupts, an empty array indicates that no interrupts were received.
- Acknowledge IRQ—Acknowledges and resets to the default value any interrupts that occur. After a successful Wait on IRQ method, use the Acknowledge IRQ method to acknowledge the source of the interrupt.
 - IRQ Number(s) specifies the logical interrupt or array of logical interrupts the function acknowledges.
- Download—Downloads the most recent version of the compiled FPGA VI or bitfile to the FPGA target. Make sure the front panel of the FPGA VI matches the front panel of the FPGA VI you open a reference to.
- **Get FPGA VI Execution Mode**—Returns the FPGA VI execution mode. Use this method if you want to execute different code depending on where the FPGA VI executes.
 - FPGA VI Execution Mode indicates where the FPGA VI executes. This enum can return FPGA Target,

 Development Computer with Simulated I/O, or

Development Computer with Real I/O.



Note If you use a host VI to communicate with an FPGA VI that is running on a development computer, you must be aware of special considerations for the host VI.

Open FPGA VI Reference Function

Owning Palette: FPGA Interface Functions

Installed With: FPGA Interface

Opens a reference to the FPGA VI or bitfile and FPGA target you specify. Right-click the Open FPGA VI Reference function and select **Configure**Open FPGA VI Reference from the shortcut many to display the

Open FPGA VI Reference from the shortcut menu to display the

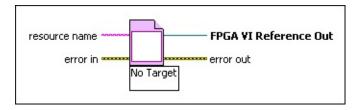
Configure Open FPGA VI Reference dialog box.

You must open a reference to the FPGA target before you can communicate between the host VI and the FPGA VI. You can download and run only one FPGA VI at a time on a single FPGA target. If you attempt to download a second VI to the FPGA target while the first FPGA VI is still in use, the LabVIEW FPGA Module reports an error and the download fails.



Note You can use the Open FPGA VI Reference function to interface with an FPGA bitfile even if you do not have the FPGA Module installed.

Details



- Place on the block diagram
 Find on the Functions palette
- resource name specifies the FPGA target on which you want to run the FPGA VI. LabVIEW automatically might determine the FPGA target on which the FPGA VI runs based on the information in the project, depending on the FPGA target. The data type of the resource name input varies by FPGA target.



Note You must compile the FPGA VI before the Open FPGA VI Reference function can download or run the VI on the FPGA target.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value

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

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. The default is 0. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
- source specifies the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- FPGA VI Reference Out returns a reference to the FPGA VI running on the FPGA target. You can bind the FPGA VI Reference Out parameter to type definitions so that LabVIEW automatically propagates configuration changes to subsequent subVIs in the data flow.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out front panel indicator and select Explain Error from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - source describes the origin of the error or warning and is, in

most cases, the name of the VI or function that produced the error or warning.

Open FPGA VI Reference Details

Use the Open FPGA VI Reference function to do the following:

- Select the FPGA VI or bitfile with which the host VI communicates.
- Select the FPGA target on which the FPGA VI runs.
- Determine whether the host VI opens and runs the FPGA VI or just opens the FPGA VI.

Place and wire a <u>Close FPGA VI Reference</u> function for every Open FPGA VI Reference function in a host VI. When the Open FPGA VI Reference function first executes on the block diagram, the function checks whether the compiled FPGA VI already exists on the FPGA target. If the compiled FPGA VI is not on the FPGA target, the Open FPGA VI Reference function downloads the compiled FPGA VI to the FPGA target. If you place a checkmark in the **Run the FPGA VI** checkbox in the **Configure Open FPGA VI Reference** dialog box, the FPGA VI starts running if it is not already running.

The Open FPGA VI Reference function does not affect the FPGA VI if the FPGA VI is already downloaded and running. If you want to restart the FPGA VI, you can use the Invoke Method function to abort or reset the FPGA VI and use the Invoke Method function to run the FPGA VI again.

If you want to open references to different FPGA VIs or bitfiles on one target, open only one reference at a time and close the reference before opening another. You can have more than one FPGA VI reference simultaneously open on a target, as long as all the references correspond to the same FPGA VI or bitfile on the same target.



Note The Open FPGA VI Reference function extracts the bitstream associated with the compiled FPGA VI or bitfile and stores the bitstream when you save the host VI. The bitstream contains the programming instructions LabVIEW downloads to the FPGA target.

If you double-click an Open FPGA VI Reference function that is not yet configured for an FPGA VI, the function displays the **Configure Open FPGA VI Reference** dialog box. If you double-click an Open FPGA VI Reference function that is configured for an FPGA VI, the function opens the front panel of the FPGA VI.

Selecting the FPGA VI or Bitfile

To determine the VI that the Open FPGA VI Reference function opens, drag the FPGA VI from the **Project Explorer** window onto the Open FPGA VI Reference function. You also can right-click the Open FPGA VI Reference function, select **Configure Open FPGA VI Reference** from the shortcut menu, and specify an FPGA VI or bitfile.



Note You cannot open a reference to an FPGA VI if you do not have the LabVIEW FPGA Module installed. However, you can open a reference to a bitfile.

Selecting the Open Behavior

By default for some FPGA targets, this function opens and runs the compiled FPGA VI on the FPGA target if the FPGA VI is not already running. To open a reference to the FPGA VI without running it, remove the checkmark from the **Run the FPGA VI** checkbox in the **Configure Open FPGA VI Reference** dialog box. You then can run the FPGA VI using the <u>Invoke Method</u> function.

Some targets do not allow you to run the FPGA VI when you open it. You can always run the FPGA VI from the host VI using the Invoke Method function.

Determining Where the FPGA VI Executes

The text underneath the icon for the Open FPGA VI Reference function indicates where the FPGA VI executes. You can change where the FPGA VI executes by right-clicking the FPGA target and selecting an option from the **Execute VI on** shortcut menu. You also can use the <u>Debugging Properties</u> page of the <u>FPGA Target Properties</u> dialog box to specify where the FPGA VI executes.

You can use the Open FPGA VI Reference function when the FPGA target is configured to execute on the FPGA target or the development computer with simulated I/O. If you use this function when the FPGA target is configured to execute on the development computer with real I/O, LabVIEW returns a run-time error.

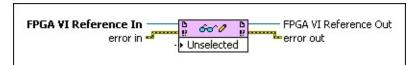
Read/Write Control Function

Owning Palette: FPGA Interface Functions

Installed With: FPGA Interface

Reads a value from or writes a value to a control or indicator in the FPGA

VI on the FPGA target.



- Place on the block diagram Find on the Functions palette
- **FPGA VI Reference In** is the reference to the FPGA VI running on the FPGA target. You must open a reference to the FPGA VI to use this parameter.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. 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 exception control to treat what is normally an error as no error or to treat a warning as an error. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.
 - source specifies the origin of the error or warning and is, in most cases, the name of the VI or function that produced

the error or warning. The default is an empty string.

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

Read/Write Control Details

A host VI can control and monitor data passed through the FPGA VI front panel. You cannot access values on any wires on the FPGA VI block diagram that do not have controls or indicators unless the data is stored in a DMA FIFO.

First use the Open FPGA VI Reference function to open a reference to the FPGA target. Then wire the FPGA VI Reference Out parameter to the Read/Write Control function to access controls and indicators on the FPGA VI. You can read indicators and write controls. You also can write indicators and read controls. You can expand the Read/Write Control function to read or write multiple controls and indicators. When you run the host VI, the Read/Write Control function reads and writes controls and indicators in the order they appear in the Read/Write Control function on the block diagram.



Tip The Read/Write Control function supports scalar data, such as numeric and Boolean controls, and complex data, such as arrays and clusters. You can program the FPGA VI to bundle scalar data into arrays or clusters and then read or write the arrays or clusters of data as a single block with the host VI to make sure all data is read or written at the same time. You can use the Read/Write Control function to read whole clusters or an individual element of a cluster. If you need to read multiple elements of a cluster, read the whole cluster. You can write to a whole cluster, but you cannot write to individual elements of a cluster. Be careful not to overuse arrays because arrays use space on an FPGA target.

The FPGA Module creates a register map, specific to the FPGA VI, that includes a hardware register for every control and indicator. LabVIEW uses the register map internally to communicate with the FPGA VI directly with Interactive Front Panel Communication and using the host VI with Programmatic FPGA Interface Communication.

Advanced Function

Owning Palette: FPGA Interface Functions

Installed With: FPGA Module. This topic might not match its

corresponding palette in LabVIEW depending on your operating system,

licensed product(s), and target.

Includes the Up Cast function.

The function on this palette can return <u>general LabVIEW error codes</u>, specific FPGA Interface error codes, or error codes specific to the FPGA target.

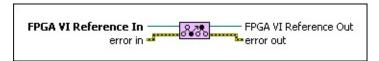
Palette Object	Description
<u>Up</u> <u>Cast</u>	Converts an FPGA VI-specific reference to a more generic reference. You then can use common code to interact with different FPGA VIs. The FPGA targets must be of the same class. You might want to use this function if you want to evaluate different algorithms in a host VI without rewriting the host VI. Some FPGA targets might not support this function.

Up Cast Function

Owning Palette: Advanced Function

Installed With: FPGA Interface

Converts an FPGA VI-specific reference to a more generic reference. You then can use common code to interact with different FPGA VIs. The FPGA targets must be of the same class. You might want to use this function if you want to evaluate different algorithms in a host VI without rewriting the host VI. Some FPGA targets might not support this function.



- Place on the block diagram Find on the **Functions** palette
- **FPGA VI Reference In** is the reference to the FPGA VI running on the FPGA target. You must open a reference to the FPGA VI to use this parameter.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. 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 exception control to treat what is normally an error as no error or to treat a warning as an error. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a nonzero error code. If status is FALSE, code is 0 or a warning code.

- source specifies the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- FPGA VI Reference Out returns a reference to the FPGA target. If you select Interface from the shortcut menu, FPGA VI Reference Out also returns information about the VI, including controls, indicators, data types, and the connector pane, but not including the VI filename. You can bind the FPGA VI Reference Out parameter to type definitions so that LabVIEW automatically propagates configuration changes to subsequent subVIs in the data flow.
- error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out front panel indicator and select Explain Error from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a nonzero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Up Cast Details

Right-click the Up Cast function and select **Interface** from the shortcut menu to output VI information and target information. Use the **Interface** option to pass data to the **Read/Write Control** function or **Invoke Method** function. If you select the **Interface** option, the FPGA VIs and targets must meet the following requirements: the front panel windows of the FPGA VIs are the same, the control and indicator <u>tab order</u> matches, and you use only one clock domain on the FPGA VI block diagram.

Right-click the Up Cast function and select **Target** from the shortcut menu to output only target information. Use the **Target** option to pass data to the Invoke Method function. The default is **Target**.



Note When you pass references to <u>subVIs</u>, consider binding the output of the Up Cast function to a type definition to propagate changes to the subVIs. Right-click the Up Cast function and select **Bind Reference Output to Type Definition** from the shortcut menu to bind the output.