LabVIEW™ FPGA Module Release and Upgrade Notes

Version 8.6

These release notes contain instructions for installing the LabVIEW FPGA Module, introduce new features, and provide upgrade information. Refer to the resources listed at the end of this document for information about developing applications with the FPGA Module.

Contents

NATIONAL INSTRUMENTS™

System Requirements

The development computer is a PC or PXI system on which you install the LabVIEW development system and the LabVIEW FPGA Module. You create and edit FPGA VIs and host VIs on the development computer. A host VI is the VI you create to run on Windows or an RT target and to communicate with the FPGA VI on the FPGA target.

Table 1 describes the requirements the development computer must meet to run version 8.6 of the FPGA Module. The FPGA Module system requirements are in addition to the LabVIEW system requirements listed in the *LabVIEW Release Notes*.

Table 1. System Requirements for the FPGA Module 8.6

Supported Platforms	Media and System Requirements	Important Notes
• Windows XP Pro (Service Packs 1 or 2) • Windows Vista (32-bit) • Windows Vista (64-bit) * • Windows 2000 Pro (Service Packs 2, 3, or 4) *	• 1.2 GHz Pentium processor or a compatible processor of equal or higher speed • 2 GB memory or more** • At least 4 GB additional available disk space • LabVIEW 8.6 Full or Professional Development System	*** Windows 2000 and Vista 64-bit**—The Xilinx tools used by the FPGA Module do not officially support Windows 2000 or Windows Vista 64-bit. National Instruments obtained permission from Xilinx to allow FPGA Module customers to use the tools on these platforms, with the disclaimer that Xilinx will not be able to fix any bugs found that are specific to these platforms. National Instruments tested the Xilinx tools, as they are used by the FPGA Module, and did not find any issues related to these platforms. If you encounter problems with the Xilinx tools specific to these platforms, you might be required to compile using Windows XP or Vista 32-bit. In such cases, you might want to install the LabVIEW FPGA Compile Server on a remote computer. Refer to Table 2 for requirements to install the Compile Server on a remote computer. National Instruments will not be liable for any problems or issues related to the use of Xilinx tools with Windows 2000 or Windows Vista 64-bit. **** Memory Requirements**—Although these memory requirements apply to typical designs, the unique characteristics of each design affect the memory requirements. Design complexity and constraints affect whether the FPGA Module can implement the design using more or less memory. You must monitor the memory and adjust the memory, if necessary. You might find it useful to monitor the memory usage of `xst.exe` in the **Windows Task Manager**. Also, memory requirements vary by FPGA target. If you require more than 2 GB of memory, consider installing the Compile Server on a remote computer. Refer to Table 2 for requirements to install the Compile Server on a remote computer. **FPGA Target Hardware and Software**—You also must have an FPGA target and target-specific software, which is typically packaged with the FPGA target. The FPGA target is the hardware on which the FPGA VI runs, such as an NI Reconfigurablee I/O device. You can develop an FPGA VI without an FPGA target, but you cannot run the FPGA VI without an FPGA target.

You can install the Compile Server on the development computer or a remote computer. Table 2 describes the requirements for installing the Compile Serve on a remote computer.

Table 2. System Requirements for Installing the Compile Server on a Remote Computer

Supported Platform	Important Notes
• Windows XP Pro (Service Packs 1 or 2) • Windows XP Pro x64 • Windows Vista (32-bit)	**Memory Requirements**—Some designs and targets might require more than 2 GB of memory required by the development computer, in which case National Instruments recommends installing the Compile Server on a remote computer with Windows XP x64. National Instruments does not support Windows XP x64 for development. However, you can use Windows XP x64 for compiling. With Windows XP x64, the Compile Server can take advantage of up to 4 GB of memory. Refer to the *Compiling an FPGA VI Remotely* topic in the *LabVIEW Help* for more information about running the Compile Server on a remote computer.

Installing the LabVIEW FPGA Module

This section includes information about installing the FPGA Module on a development computer using the CDs included in the FPGA Module kit. If you installed the FPGA Module from the LabVIEW 8.6 Platform DVD, you do not need to reinstall the FPGA Module from the CDs.

Note You must install LabVIEW 8.6 from the LabVIEW 8.6 Platform DVD before installing the FPGA Module. Refer to the *LabVIEW Release Notes* for information about installing the LabVIEW development system.

Complete the following steps to install the FPGA Module.

1. Log in to the development computer as an administrator or as a user with administrative privileges.
2. (Optional) Install the RT Module. Refer to the *LabVIEW Real-Time Module Release and Upgrade Notes* for information about installing the RT Module and the device drivers you need and for information about configuring RT targets.
3. Install the FPGA Module from the first LabVIEW FPGA Module 8.6 installation CD, following the instructions that appear on the screen. Depending on what you choose to install, the FPGA Module installer might prompt you to insert the second installation CD.
4. Select **Complete** when you reach the **Select Installation Type** step. By default, the FPGA Module installer prompts you to insert the NI-RIO CD, which installs driver software for R Series and CompactRIO devices. You might need to install different or additional driver software for the FPGA target you use.

   The FPGA Module installs program files and documentation and copies files from Xilinx ISE to the *x*`:\NIFPGA86` directory, where *x* is the drive on which you installed LabVIEW 8.6. Xilinx ISE is third-party software that the FPGA Module uses to compile FPGA VIs into code that runs on the FPGA target.
5. (Optional) Install NI device drivers available on the NI Device Drivers DVD.

Note The NI Device Drivers DVD might not include drivers for the FPGA target you use. If the NI Device Drivers DVD does not include the driver for the FPGA target you use, install FPGA target drivers from the media available with the targets in step 7.

6. (Optional) Install the appropriate drivers and FPGA Module support files for the FPGA target you will use. Refer to the specific hardware documentation for information about the appropriate drivers and for information about installing and configuring the FPGA target.

Activating the FPGA Module License

The FPGA Module relies on licensing activation. You have a temporary license for a 30-day evaluation period. If you do not activate the FPGA Module license, the FPGA Module operates in evaluation mode, by default, for a 30-day evaluation period. When the evaluation period expires, you must activate a valid FPGA Module license to continue using the FPGA Module. Refer to the *Activation Instructions for National Instruments Software* document for more information about licensing.

LabVIEW 8.6 FPGA Module Features

The FPGA Module 8.6 includes the following new features to help you better manage and implement the components of an FPGA application.

Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help** in LabVIEW, for more information about the new features.

Improved Debugging and Test Bench Creation Capabilities

Simulating I/O

In the FPGA Module 8.6, you can test the logic of an FPGA VI by executing the FPGA VI on the development computer using simulated I/O. You can change where the FPGA VI executes by right-clicking the FPGA target in the **Project Explorer** window and selecting an option from the **Execute VI on** shortcut menu.

In addition to using random data for inputs, you can create a custom VI for read and write operations of FPGA I/O Nodes, FPGA I/O Method Nodes, and FPGA I/O Property Nodes. With the custom VI, you can control values that the FPGA VI reads and monitor values that the FPGA VI writes. Refer to the *Tutorial: Creating Test Benches* topic in the *LabVIEW Help* for an example of using a custom VI.

Use the **Debugging Properties** page to specify the custom VI to use when executing the FPGA VI on a development computer. In the **Project Explorer** window, right-click an FPGA target and select **Properties** from the shortcut menu to display the **FPGA Target Properties** page. Select **Debugging** from the **Category** list to display the **Debugging Properties** page.

The following table shows the FPGA Module 8.6 execution option that corresponds to each of the emulation options in the FPGA Module 8.5.*x* and earlier.

Emulator Mode in FPGA Module 8.5.*x* and earlier	Execution Mode in FPGA Module 8.6
Off	**Execute VI on FPGA Target**
On–Use Random Data for Inputs	**Execute VI on Development Computer with Simulated I/O** with the **Use Random Data for FPGA I/O Read** option selected
On–Use Target Hardware for I/O	**Execute VI on Development Computer with Real I/O**

Using Host VIs to Test FPGA VIs

In the FPGA Module 8.5.*x* and earlier, the host VI controls only the FPGA VI on the FPGA target. In the FPGA Module 8.6, you can use the host VI to control the FPGA VI running on the development computer. Therefore, you can test more of the functionality of the FPGA VI than was possible in previous versions of the FPGA Module. In particular, you can test interactions of the FPGA VI with the host VI by reading and writing controls, handling interrupts, and using DMA FIFOs. You now can repeat tests of FPGA VI and host VI interactions. Refer to the *Communicating with an FPGA VI Running on a Development Computer* topic in the *LabVIEW Help* for special considerations.

You can execute different code on the host VI depending on where the FPGA VI executes using the Invoke Method function configured for the Get FPGA VI Execution Mode method.

Verifying the FPGA VI Compiles for the FPGA

When you execute an FPGA VI on a development computer, consider periodically compiling the FPGA VI to determine whether the FPGA VI compiles with all of the constraints enforced by the FPGA Module, whether the resources required to implement the FPGA VI are available in the fabric of the target FPGA, and whether the logic meets timing constraints set by the FPGA clocks. Right-click the FPGA VI in the **Project Explorer** window and select **Compile** from the shortcut menu to compile the FPGA VI.

FIFO and Memory Improvements

Fixed-Point Data Type Support for FIFO and Memory Items

In the FPGA Module 8.6, FIFO and memory items support the fixed-point data type.

Use the **FPGA FIFO Properties** dialog box to configure a FIFO item to use the fixed-point data type. Right-click a FIFO item in the **Project Explorer** window or a VI-Scoped FIFO Configuration node on the block diagram and select **Properties** from the shortcut menu to display the **FPGA FIFO Properties** dialog box.

Use the **Memory Properties** dialog box to configure a memory item to use the fixed-point data type. Right-click a memory item in the **Project Explorer** window or a VI-Scoped Memory Configuration node on the block diagram and select **Properties** from the shortcut menu to display the **Memory Properties** dialog box.

Note FIFO and memory items do not support overflow for the fixed-point data type.

Additional Data Type Support for DMA FIFOs

In the FPGA Module 8.5.*x* and earlier, Direct Memory Access (DMA) FIFOs support only 32-bit unsigned integers. In the FPGA Module 8.6, DMA FIFOs also support Boolean, fixed-point, and all integer data types. Use the **FPGA FIFO Properties** dialog box to configure the data type for the DMA FIFO. Right-click a FIFO item in the **Project Explorer** window and select **Properties** from the shortcut menu to display the **FPGA FIFO Properties** dialog box.

Note DMA FIFOs do not support overflow for the fixed-point data type.

Automatic DMA Channel Allocation

In the FPGA Module 8.5.*x* and earlier, you must specify the DMA channel to use on the FPGA target. In the FPGA Module 8.6, the FPGA Module selects the DMA channel automatically.

Math and Analysis VIs

New VIs

The FPGA Module 8.6 includes the following new VIs:

- Rational Resampler—Provides a rational resampling filter, which updates the input sample rate by an **L/M** factor where **L** is an interpolation factor and **M** is a decimation factor.
- Scaled Window—Minimizes spectral leakage associated with truncated waveforms.
- FFT—Computes the Fast Fourier Transform (FFT) point by point.

Fixed-Point Support

The FPGA Module 8.6 includes support for the fixed-point data type in the following VIs: Backlash, Dead Zone, Discrete Control Filter, Discrete Normalized Integration, Friction, Initial Condition, Linear Interpolation, Memory Element, Rate Limiter, Relay, Switch, Trigger, Unit Delay, Zero Crossing, and Zero-Order Hold.

Single-Cycle Timed Loop Support

In the FPGA Module 8.6, you can use the following VIs in a single-cycle Timed Loop: Backlash, Dead Zone, Discrete Control Filter, Discrete Normalized Integration, Friction, Initial Condition, Memory Element, Rate Limiter, Relay, Switch, Trigger, Unit Delay, Zero Crossing, and Zero-Order Hold.

Additional Functions on the FPGA Numeric Functions Palette

The FPGA **Numeric** palette now includes the Divide, Reciprocal, and Square Root functions, which support the fixed-point data type.

Running HDL Code on FPGAs alongside FPGA VIs

In the FPGA Module 8.6, some FPGA targets support using component-level IP (CLIP) to instantiate HDL code with a defined interface that occupies a portion of an FPGA. Unlike the HDL Interface Node, you can use CLIP to perform the following tasks:

- Run HDL code in parallel with LabVIEW code.
- Execute HDL code in multiple clock domains.
- Include constraints in the compilation.

Refer to the tutorial in the *Using VHDL as Component-Level IP* book in the *LabVIEW Help* for more information.

Improved Performance Using External Clocks

Some FPGA targets, such as the PXI-795*x*R targets, support using a digital I/O resource as an external clock. External clocks enable tighter synchronization with external hardware and, in some cases, better performance than onboard clocks, which require additional overhead for synchronization. You can use an external clock as the clock for a single-cycle Timed Loop.

Caution If you use an external clock incorrectly, the clock can introduce incorrect block diagram execution within the single-cycle Timed Loop. Refer to the *Using External Clocks for Synchronous Design* topic in the LabVIEW Help for more information about external clocks and issues you must consider to avoid incorrect execution.

Upgrade and Compatibility Issues

Refer to the following sections for upgrade and compatibility issues specific to different versions of the FPGA Module.

Upgrading from FPGA Module 8.5.*x*

Changes to Host VIs

For LabVIEW projects saved in the FPGA Module 8.5.*x* or earlier, if you selected the **On–Use Random Data for Inputs** or **On–Use Target Hardware for I/O** option from the **Emulator** shortcut menu or **FPGA Target Properties** dialog box, the option had no effect on the FPGA Interface functions. In the FPGA Module 8.6, the FPGA Interface functions control the FPGA VI running on the development computer. Use the text underneath the icon for the Open FPGA VI Reference function to determine where the FPGA VI executes.

Configuring Open FPGA VI Reference Functions

The Open FPGA VI Reference function no longer includes the **Select VI**, **Select Bitfile**, **Open and Run**, **Open**, **Bind to Typedef**, and **Select Address** shortcut menu options. Instead, you can configure the Open FPGA VI Reference function using the **Configure Open FPGA VI Reference** dialog box. Right-click the Open FPGA VI Reference function and select **Configure Open FPGA VI Reference** from the shortcut menu to display the dialog box. To specify the FPGA target on which you want to run the FPGA VI, use the **Resource** field on the **FPGA Target Properties** page or the **resource name** input on the Open FPGA VI Reference function.

Passing an FPGA VI Reference between FPGA Interface Functions

The FPGA Interface functions no longer use a cluster to pass information between functions. Instead, the FPGA Interface functions now use an FPGA VI reference. Right-click an FPGA VI reference control, constant, or indicator and select **Configure FPGA VI Reference** from the shortcut menu to configure the reference.

Call VI Function No Longer Supported

The FPGA Module 8.5 removed the Call VI function from the **FPGA Interface** palette. The FPGA Module 8.6 replaces the Call VI function in existing applications with a series of functions that maintain the functionality of the Call VI function.

Saturation Arithmetic VIs No Longer Supported

The Saturation Arithmetic VIs are no longer available on the **Saturation Arithmetic** palette. However, existing applications that use the Saturation Arithmetic VIs continue to work.

Change to the Analog Period Measurement VI

If you wired a signed integer to the **hysteresis** input on the Analog Period Measurement VI in an existing application, the FPGA Module 8.6 changes the input to an unsigned fixed-point terminal and adds a To Fixed-Point function to convert to the unsigned type with saturation. This change provides improved edit-time error checking to prevent inadvertently passing a negative number to the **hysteresis** input. However, this change requires additional resources on the FPGA, and the To Fixed-Point function takes one extra clock cycle to execute. You can restore the original performance by wiring an appropriate unsigned type to the **hysteresis** input and removing the To Fixed-Point function.

Additional FPGA Resources Required

The FPGA Module 8.6 provides improved reliability of the FPGA during a system restart. As a result, if you recompile an existing FPGA VI using the FPGA Module 8.6, the FPGA VI might require additional resources. So, some existing FPGA VIs might not fit on the FPGA.

Path to Compile Tools

You no longer can use the **Configure Server** dialog box to specify the path to the directory containing the Xilinx compile tools. Instead, the FPGA Module installer defines the path.

Increased Minimum Depth for FIFOs

For DMA FIFOs, the default depth is now 15 elements. For target-scoped FIFOs, the default is now 20 elements.

Upgrading from FPGA Module 8.2.*x*

You might encounter the following compatibility issues when you upgrade to FPGA Module 8.6 from FPGA Module 8.2.*x*.

Changes to DMA FIFOs

In the FPGA Module 8.2.*x* and earlier, if you use the Invoke Method function with the *FIFO*»Configure method, the default of the **Depth** parameter is twice the number of elements in the DMA FIFO item in the project. In the FPGA Module 8.6, the default of the **Depth** parameter is 10,000 elements.

In the FPGA Module 8.2.*x* and earlier, if you use the Invoke Method function with the *FIFO*»Stop method, you cannot read data from the host part of the FIFO, but you can read data from the FPGA part of the FIFO. In the FPGA Module 8.6, if you use the *FIFO*»Stop method, you cannot read data from the host or FPGA parts of the FIFO. As a result, FPGA VIs that use DMA FIFOs now require additional FPGA resources. So, some existing FPGA VIs might not fit on the FPGA.

In the FPGA Module 8.6, if an error occurs while the *FIFO*»Stop method executes, it now executes normally and sets its own error status in **error out**.

Discrete PID VI

The FPGA Module 8.6 no longer supports the Discrete PID VI. Use the PID (FPGA) Express VI, available with the LabVIEW PID Control Toolkit, to implement single-channel and multi-channel PID controllers on FPGA targets. The PID Control Toolkit is available with the LabVIEW Real-Time Module. You also can purchase the PID Control Toolkit separately.

Changes to the Scale by Power of 2 Function

In the FPGA Module 8.6, the Scale by Power of 2 function consumes less space on the FPGA. However, the maximum clock rate for the function is now slightly slower for non-constant signed shifts. Therefore, existing applications that include the function might not meet existing timing specifications.

Using the Invoke Method to Force a Download

The Download method on the Invoke Method function no longer includes the **Force Download** parameter. When you call the Download method, the method now always forces an FPGA VI to download to the FPGA target. However, existing applications that use the **Force Download** parameter continue to work, because the FPGA Module replaces the previous version of the Download method with a Case structure. The selector of the Case structure is the value wired to the **Force Download** parameter in the previous version of the Download method. The Case structure executes the new version of the Download method only when the value of the selector is `True`.

FIFO Read and FIFO Write Parameter Renamed

The **Full** output on the FIFO Write function and the **Empty** output on the FIFO Read function were both renamed to **Timed Out?**.

Upgrading from FPGA Module 8.0

You might encounter the following compatibility issues when you upgrade to FPGA Module 8.6 from FPGA Module 8.0. Refer to the *Upgrading from FPGA Module 8.2.x* section of this document for information about other upgrade issues you might encounter.

Memory Read and Memory Write Functions

If you use the Memory Read or Memory Write functions in an FPGA VI from LabVIEW 8.0 or earlier, the FPGA Module 8.6 does not update the Memory Read and Memory Write functions to 8.6 functionality. All Memory Read and Memory Write functions from 8.0 or earlier continue to work in the FPGA Module 8.6.

If you created memory blocks using the FPGA Module Memory Extension Utility on `ni.com`, the FPGA Module 8.6 does not convert the memory blocks to use an 8.6 memory item. The memory blocks continue to work in the FPGA Module 8.6. However, the memory blocks might not be supported in a future release of the FPGA Module. To replace a memory block, create a new memory item with the same configuration as the memory block and configure the Memory Read and Memory Write functions to access the new memory item.

Wait on Occurrence Function

In the FPGA Module 8.0 and earlier, when you use the Wait on Occurrence function in an FPGA VI, the function uses ticks as the unit for the **ms timeout** parameter. When you use the Wait on Occurrence function in a Windows VI, the unit is milliseconds. In the FPGA Module 8.6, the Wait on Occurrence function uses milliseconds in both FPGA and Windows VIs. The FPGA Module 8.6 also includes a Wait on Occurrence with Timeout in Ticks function.

When you open an FPGA VI that is saved in LabVIEW 8.0 or earlier in the application instance for an FPGA target, the FPGA Module 8.6 replaces the Wait on Occurrence function with the new Wait on Occurrence with Timeout in Ticks function. If you use the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility, the FPGA Module 8.6 replaces the Wait on Occurrence function with the new Wait on Occurrence with Timeout in Ticks function in all FPGA VIs.

However, the FPGA Module might not replace the Wait on Occurrence function in the following cases:

- The FPGA VI was last saved on a non-FPGA target in LabVIEW 8.0 and you open and save (or mass compile) the VI in a non-FPGA target in LabVIEW 8.6.
- The FPGA VI was last saved in LabVIEW 7.*x* and you open and save (or mass compile) the VI in a non-FPGA target in LabVIEW 8.6.
- The FPGA VI was saved or mass compiled in LabVIEW 8.6 before installing the FPGA Module.

In the cases listed above, you must manually replace the Wait on Occurrence function with the Wait on Occurrence with Timeout in Ticks function to continue using the ticks unit.

Tunnels and Shift Registers on For Loops

In the FPGA Module 8.0 and earlier, if the value wired to the count (**N**) terminal on a For Loop is `0`, the outputs from the tunnels and shift registers are undefined. In the FPGA Module 8.6, the output tunnels and shift register terminal on the right side of the For Loop contain an extra MUX to handle a `0` value wired to the count terminal in a manner consistent with LabVIEW for Windows. As a result, FPGA VIs that use the output tunnels or right shift register terminal in a For Loop might use more FPGA resources and/or compile at a slightly lower clock rate.

Upgrading from FPGA Module 1.*x*

You might encounter the following compatibility issues when you upgrade to FPGA Module 8.6 from FPGA Module 1.*x*. Refer to the *Upgrading from FPGA Module 8.0* section of this document for information about other upgrade issues you might encounter.

Refer to `ni.com/info` and enter the info code `ex8kk4` for more information about how to perform FPGA Module 1.*x* tasks with the FPGA Module 8.6.

Refer to the *LabVIEW FPGA Module Release and Upgrade Notes* for each version of the LabVIEW FPGA Module between versions 1.*x* and 8.6 at `ni.com/manuals` for more information about the new features and changes in each version.

FPGA Device I/O Functions

The FPGA Module no longer supports the FPGA Device I/O functions. Use the FPGA I/O Node configured to target-specific I/O resources instead.

Embedded Project Manager

The FPGA Module no longer supports the Embedded Project Manager. Use the **Project Explorer** window instead.

Upgrading FPGA VIs, Host VIs, and Embedded Projects from the FPGA Module 1.*x* to 8.6

If you created FPGA VIs or host VIs using LabVIEW 7.*x* and the FPGA Module 1.*x*, you can update the VIs for use with LabVIEW 8.6 and the FPGA Module 8.6. Complete the following steps to update the VIs for use with LabVIEW 8.6 and the FPGA Module 8.6.

Note NI recommends you back up your files before you begin updating them for use with the FPGA Module 8.6.

1. Launch LabVIEW 8.6.
2. Select **Tools»FPGA Module»Import FPGA Files from LabVIEW FPGA Module 1.X** in the **Getting Started** window or in the **Project Explorer** window of an open project to launch the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility. If you do not have a project open, LabVIEW prompts you to create a new project.

- Upgrading from the FPGA Module 1.1 only.

  3. Complete the following steps to add the LEP file you want to import to the **LabVIEW Embedded Project (*.lep) Files** list.

     a. Click the **Add** button below the **LabVIEW Embedded Project (*.lep) Files** list. The **Select an Embedded Project to Add** dialog box appears.

     b. Navigate to the location of the LEP file you want to import.

     c. Click the **Add** button. LabVIEW adds the LEP file and all the VIs associated with the LEP file to the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility. You can see only the LEP file in the **LabVIEW Embedded Project (*.lep) Files** list.

     d. Proceed to step 5.

- Upgrading from the FPGA Module 1.0 only.

  4. Complete the following steps to add the FPGA VIs you want to import to the **FPGA 1.0 VIs** list.

     a. Click the **Add** button below the **FPGA 1.0 VIs** list. The **Select a VI to Add** dialog box appears.

     b. Navigate to the location of the FPGA VI you want to import.

     c. Click the **Add** button. LabVIEW adds the FPGA VI to the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility.

     d. Proceed to step 5.

  5. Add host VIs from the FPGA Module 1.*x* you want to import to the **Host VIs** list.

  6. Click the **Continue** button. The **Import Status** dialog box appears. When LabVIEW finishes importing the 1.*x* files, the **Save VIs** dialog box appears.

  7. Select a location in the **Save Location** section to save the imported VIs. Select **Next to Project** if you want to save the new VIs in the same location as the new project. National Instruments recommends this option. Select **Choose a Location** if you want to specify a different location. Select **Defer Decision - Leave VIs open in memory** if you want to wait until later to save the VIs.

  8. If you select **Choose a Location** or **Next to Project** and you have not saved the project, click the **Continue** button. If you select **Choose a Location**, the **Location to Save** dialog box appears. If you select **Next to Project**, the **Name the Project** dialog box appears. Proceed to step 9.

     If you have saved the project or if you select **Defer Decision - Leave VIs open in memory**, click the **Finish** button. LabVIEW returns you to the project. If you select **Defer Decision - Leave VIs open in**

memory, LabVIEW also opens the front panel windows of the top-level VIs.

9. If you have the **Location to Save** dialog box open, choose an existing folder or create a new folder in which to save the new project and click the **Save** button. If you have the **Name the Project** dialog box open, type a name for the new LabVIEW project file (`.lvproj`) and click the **Save** button to save the new project.

The project file now includes all the VI and I/O information, as well as the FPGA target(s) on which the FPGA VI runs.

Where to Go from Here

National Instruments provides many resources to help you succeed with your NI products. Use the following resources as you start exploring LabVIEW and the FPGA Module.

Examples

The driver software for many FPGA targets includes corresponding examples. Refer to the specific hardware documentation for information about whether the FPGA target you use comes with corresponding examples.

You can start with an existing example and use it as a starting point for developing FPGA VIs and host VIs. From LabVIEW, launch the NI Example Finder by selecting **Help»Find Examples**. Browse the examples by directory or by task.

Related Documentation

Refer to the *LabVIEW Help* for information about LabVIEW and the FPGA Module, and refer to the **Context Help** window for information about using VIs and functions.

- *LabVIEW Help*—Available by selecting **Help»Search the LabVIEW Help** in LabVIEW. Browse the **FPGA Module** book in the **Contents** tab for an overview of the FPGA Module and hardware-specific information included in the *LabVIEW Help*. Browse the **FPGA Interface** book in the **Contents** tab for an overview of the FPGA Interface information included in the *LabVIEW Help*.
- **Context Help** window—Available by selecting **Help»Show Context Help**.

Refer to the specific hardware documentation for information about using the FPGA target with LabVIEW and the FPGA Module and for information about hardware specifications.

NI Web Site

Visit ni.com/fpga for the latest NI Developer Zone articles, examples, and support information for the FPGA Module.

Refer to ni.com/info and enter the info code fpgatrn to access online training for the FPGA Module.

Known Issues

Refer to readme_fpga.html on the LabVIEW FPGA Module installation CD for information about known issues with the FPGA Module. You also can find the readme_fpga.html file installed in the LabVIEW 8.6\readme directory. Select **Start»All Programs»National Instruments»LabVIEW 8.6»Readme** to open the LabVIEW 8.6\readme directory.

371572D-01 Jun08