LabVIEW™ FPGA Module Release and Upgrade Notes

Version 8.5

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 running Windows and 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 in Windows or on an RT target and communicate with the FPGA VI on the FPGA target.

Table 1 describes the requirements the development computer must meet to run version 8.5 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 8.5

Supported Platforms	Media and System Requirements	Important Notes
• Windows 2000 Pro (Service Packs 2, 3, or 4) • XP Pro (Service Packs 1 or 2) • Windows Vista Business (32-bit version)	• 1.2 GHz Pentium processor or a compatible processor of equal or higher speed • at least 512 MB memory for 1 million gate FPGA targets and at least 2 GB memory for 3 million gate FPGA targets • at least 4 GB additional available disk space • LabVIEW 8.5 Full or Professional Development System	**Windows Vista**—The Xilinx tools used by the FPGA Module do not officially support Windows Vista. National Instruments obtained permission from Xilinx to allow FPGA Module customers to use the tools on Windows Vista, with the disclaimer that Xilinx will not be able to fix any bugs found that are specific to the Windows Vista OS. National Instruments tested the Xilinx tools, as they are used by the FPGA Module, and did not find any issues related to Windows Vista. If you encounter problems with the Xilinx tools specific to Windows Vista, you might be required to compile using Windows XP or Windows 2000. National Instruments will not be liable for any problems or issues related to the use of Xilinx tools with Windows Vista. **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**. **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 Reconfigurable I/O device. You can develop an FPGA VI without an FPGA target, but you cannot run the FPGA VI without an FPGA target.

Installing the LabVIEW FPGA Module

Complete the following steps to set up a development computer with LabVIEW, the LabVIEW Real-Time (RT) Module (optional), the FPGA Module, and the appropriate NI device drivers.

1. Log in to the development computer as an administrator or as a user with administrative privileges.
2. Install LabVIEW 8.5 from the LabVIEW 8.5 installation CD. Refer to the *LabVIEW Release Notes* for information about installing the LabVIEW development system.

Note The LabVIEW installer prompts you to insert the NI Device Drivers CD to install certain device drivers. If you plan to install the LabVIEW Real-Time Module, wait until after step 4 to install the NI Device Drivers CD.

3. (Optional) Install the LabVIEW Real-Time Module from the LabVIEW Real-Time Module 8.5 installation CD. 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.
4. Install the FPGA Module from the first LabVIEW FPGA Module 8.5 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.
5. 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 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*:\NIFPGA85 directory, where *x* is the drive on which you installed LabVIEW 8.5. Xilinx ISE is third-party software that the FPGA Module uses to compile FPGA VIs into code that runs on the FPGA target.
6. (Optional) Install NI device drivers available on the NI Device Drivers CD.

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

7. (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.

LabVIEW 8.5 FPGA Module Features

The FPGA Module 8.5 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.

Wizard for Creating New FPGA Projects

The FPGA Module 8.5 includes the FPGA Project Wizard, which you can use to access any installed FPGA target wizards. In the **Getting Started** window, select **FPGA Project** from the **Targets** list and click the **Go** button to display the wizard. You also can select **Tools»FPGA Module»FPGA Project** to display the wizard.

Improved Code Reuse

The FPGA Module 8.5 includes the following features to help you create reusable subVIs:

FPGA I/O Controls

The FPGA Module 8.5 includes a new FPGA I/O control that you can use to specify FPGA I/O items. You can wire FPGA I/O controls to the FPGA I/O Node, I/O Method Node, and I/O Property Node, all of which now include **FPGA I/O In** and **FPGA I/O Out** terminals. FPGA I/O controls are useful for creating reusable subVIs with inputs that specify FPGA I/O items. Refer to the *LabVIEW Help* for information about using FPGA I/O controls to create reusable subVIs.

FPGA Clock Controls

The FPGA Module 8.5 includes a new FPGA clock control that you can use to specify FPGA clocks. You can wire FPGA clock controls to the **Source Name** input terminal on the input node of the single-cycle Timed Loop. FPGA clock controls are useful for creating reusable subVIs with inputs that specify FPGA clocks. Refer to the *LabVIEW Help* for information about using FPGA clock controls to create reusable subVIs.

Feedback Nodes

Use a Feedback Node anywhere on the block diagram to store data from a previous VI or loop execution. You can use the Feedback Node in subVIs inside the single-cycle Timed Loop. You can use objects with internal registers, such as the Memory Read function, in subVIs inside the single-cycle Timed Loop. You also can use the Feedback Node to create reusable subVIs that store state. For example, you can use the Feedback Node with the FPGA I/O control to create a reusable counter. Refer to the *LabVIEW Help* for more information about the Feedback Node.

Reentrant Execution Enabled by Default

When you create a VI under an FPGA target using the FPGA Module 8.5, LabVIEW configures the FPGA VI for reentrant execution by default. Moving a VI to an FPGA target does not change the reentrant execution setting of the VI. Also, adding a VI created using a previous version of the FPGA Module to an FPGA Module 8.5 project does not change the reentrant execution setting of the VI. Refer to the *LabVIEW Help* for information about configuring the reentrant execution setting for FPGA VIs.

New Signal Generation Express VIs

The FPGA Module 8.5 includes the following new VIs:

- Square Wave Generator—Generates a point-by-point square wave using direct digital synthesis.
- White Noise Generator—Generates a white noise signal with uniform or Gaussian distribution.

Filter Improvements

In the FPGA Module 8.5, the Butterworth Filter VI accepts multiple input channels. The FPGA Module 8.5 also includes the Notch Filter VI, which attenuates a specific frequency band in one or more input channels using a second-order IIR notch filter. Both filters also include optional configuration terminals that you can use to tune filter coefficients at run time. Both filters saturate output if the actual result of the calculation exceeds the range of the data type.

Clearing FIFOs

You now can clear target-scoped and VI-scoped FIFOs from the block diagram of an FPGA VI using the FIFO Clear function. When you clear a FIFO, all data from the FIFO is removed.

You cannot use the FIFO Clear function inside a single-cycle Timed Loop. However, when FIFOs are configured as **Block Memory**, you can place the FIFO Clear function in the top-level clock domain, even if the top-level clock domain is different than the clock domain(s) specified by the single-cycle Timed Loop(s) in which the FIFO Write and FIFO Read functions are placed.

Improved Interrupt Handling

In the FPGA Module 8.5, you can wait on different interrupts from different places in a VI at the same time. Use calls to the Wait on IRQ and Acknowledge IRQ methods on the Invoke Node to implement concurrent waiting. National Instruments recommends that you specify non-overlapping interrupts for different calls to the Wait on IRQ method.

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.

Robust Handling of Multiple FPGA VI References in Host VIs

In the FPGA Module 8.2.1 and earlier, if you open multiple references to FPGA VIs or bitfiles on the same target, the FPGA Module might cause erroneous behavior without returning any error messages. In the FPGA Module 8.5, you can simultaneously have more than one FPGA VI reference open on a target, as long as all the references correspond to the same FPGA VI or bitfile on the same target. If you attempt to open a reference to a different FPGA VI or bitfile on the same target without closing the reference to the original FPGA VI or bitfile, the FPGA Module now returns an error.

Reconnecting to an FPGA VI Compilation

After you begin an FPGA VI compilation, you can disconnect from the compilation to continue using LabVIEW to edit and run other VIs. The LabVIEW FPGA Compile Server continues to compile the FPGA VI. In the FPGA Module 8.5, you can view results of the compilation by right-clicking the FPGA VI in the **Project Explorer** window and selecting **Reconnect to Compilation** from the shortcut menu. The **Reconnect to Compilation** option does not run the VI after the compilation is complete. You also can click the **Run** button on an FPGA VI to reconnect to a compilation, as well as run the VI after the compilation is complete.

Fixed-Point Data Type

The FPGA Module 8.5 includes limited support for a fixed-point data type. The fixed-point data type is a numeric data type that represents a subset of rational numbers within a user-specified range and of a user-specified precision. You can configure this data type to specify a set of numbers that map to a finite set of bits that digital hardware can store and manipulate.

Use the **Properties** dialog box to configure fixed-point numbers and functions that support the fixed-point data type. Right-click a numeric constant, control, indicator, or function that accepts fixed-point data and select **Properties** from the shortcut menu to display the **Properties** dialog box. You also can use the **Properties** dialog box to specify how numeric functions handle overflow and quantization for fixed-point numbers.

Note FIFOs, memory, FPGA Math & Analysis VIs, and some functions do not support the fixed-point data type.

Caution If you wire a fixed-point number to an integer input, you might lose significant fractional bits.

Refer to the *Using the Fixed-Point Data Type (FPGA Module)* topic in the **FPGA Module»FPGA Module Concepts»Creating FPGA VIs** book on the **Contents** tab of the *LabVIEW Help* for more information.

Additional Functions on the FPGA Numeric Functions Palette

The FPGA **Numeric** palette now includes the Square, Round To Nearest, Round Toward -Infinity, and Round Toward +Infinity functions, which support the fixed-point data type.

Support for the LabVIEW Statechart Module

The FPGA Module 8.5 supports code generated by the LabVIEW 8.5 Statechart Module. Use the Statechart Module to create state-based applications in LabVIEW.

Documentation Enhancements

The **FPGA Module** book in the *LabVIEW Help* includes the following enhancements:

- New **FPGA VI and Functions Details** book, which includes FPGA-specific details for many of the objects on the FPGA palettes. The details include Timed Loop compatibility, usage, timing, and resource considerations. In the *LabVIEW Help*, use the **Contents** tab to navigate to **FPGA Module»FPGA Module VIs and Functions» FPGA VI and Function Details**.

- Additional content about optimizing FPGA VIs. In the *LabVIEW Help*, use the **Contents** tab to navigate to **FPGA Module»FPGA Concepts» Optimizing FPGA VIs for Speed and Size**.
- New *Getting Started with the FPGA Module* topic, which includes links to topics you might find useful based on your LabVIEW, FPGA Module, and hardware experience. In the *LabVIEW Help*, use the **Contents** tab to navigate to **FPGA Module»Getting Started with the FPGA Module**.

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 8.2.*x*

You might encounter the following compatibility issues when you upgrade to FPGA Module 8.5 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 8.5, 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 8.5, 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 FPGA VIs might not fit on the FPGA.

In the FPGA Module 8.5, 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 8.5 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.

Call VI Function No Longer Supported

The Call VI function is no longer available on the **FPGA Interface** palette. However, existing applications that use the Call VI function continue to work.

If you want to replace the Call VI function, complete the following steps to obtain the same functionality.

1. Use the Open FPGA VI Reference function to open a reference to the FPGA VI.
2. Use the Read/Write Control function to access the controls and indicators on the FPGA VI.
3. Use the Run method on the Invoke Method function with a TRUE constant wired to the **Wait Until Done** input.
4. Use the Read/Write Control function to read indicators.
5. Use the Close FPGA VI Reference function to close the reference to the FPGA VI.

Changes to the Scale by Power of 2 Function

In the FPGA Module 8.5, 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 8.5 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 8.5 does not update the Memory Read and Memory Write functions to 8.5 functionality. All Memory Read and Memory Write functions from 8.0 or earlier continue to work in the FPGA Module 8.5.

If you created memory blocks using the FPGA Module Memory Extension Utility on `ni.com`, the FPGA Module 8.5 does not convert the memory blocks to use an 8.5 memory item. The memory blocks continue to work in the FPGA Module 8.5. 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 8.5, the Wait on Occurrence function uses milliseconds in both FPGA and Windows VIs. The FPGA Module 8.5 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 8.5 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 8.5 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 8.5.

- The FPGA VI was last saved in LabVIEW 7.*x* and you open and save (or mass compile) the VI in LabVIEW 8.5 in a non-FPGA target.
- The FPGA VI was saved or mass compiled in LabVIEW 8.5 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 8.5, 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 8.5 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 8.5.

Refer to the *LabVIEW FPGA Module Release and Upgrade Notes* for each version of the LabVIEW FPGA Module between versions 1.*x* and 8.5 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 8.5

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 8.5 and the FPGA Module 8.5. Complete the following steps to update the VIs for use with LabVIEW 8.5 and the FPGA Module 8.5.

Note NI recommends you back up your files before you begin updating them for use with the FPGA Module 8.5.

1. Launch LabVIEW 8.5.
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 8.5\readme` directory. Select **Start»All Programs»National Instruments»LabVIEW 8.5»Readme** to open the `LabVIEW 8.5\readme` directory.

371572C-01 Aug07