LABVIEW™ REAL-TIME

Version 6.1

These release notes provide information about LabVIEW Real-Time (RT), related documentation, special installation considerations, compatibility with NI-DAQ, examples of LabVIEW RT applications, new features in version 6.1, and issues known to affect the performance of LabVIEW RT.

Contents

About LabVIEW Real-Time (RT)

LabVIEW RT combines the user-friendliness of LabVIEW with the power of real-time systems so you can use graphical programming to create deterministic applications. You create virtual instruments (VIs) and applications with LabVIEW RT the same way you do with LabVIEW for Windows. Using a host PC or PXI system running a Windows operating system, you can create LabVIEW RT VIs and applications. You also use the host PC to download VIs to the RT Engine on an RT Series device, such as a PCI/PXI-7030 RT Series DAQ device, a PXI controller, or a FieldPoint 2000 system. VIs downloaded to the RT Engine run in real time.

You create LabVIEW RT applications by using the same functions you use to create applications in LabVIEW for Windows. However, LabVIEW RT includes the following two additional sets of VIs you can use in real-time applications:

- Shared Memory VIs in both host and RT Engine applications communicate data using the shared memory on PCI/PXI-7030 real-time boards.

- Watchdog Timer VIs access the watchdog timer hardware on PXI controllers using LabVIEW RT applications.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function & How-To Help**, for more information about the Shared Memory and Watchdog Timer VIs. You can find *LabVIEW Real-Time Help* in the **VI and Function Reference** book.

Related Documentation

The following documents contain information you might find helpful as you work with LabVIEW RT:

- *LabVIEW Real-Time User Manual*

- *LabVIEW Real-Time Help*, available by selecting **Help»VI, Function, & How-To Help** and then selecting **VI and Function Reference» Real-Time VIs**.

- *RT Series DAQ Device User Manual*

- *RT Series PXI/CompactPCI Controller User Manual*

- *FieldPoint 2000/2010 User Manual*

- Your RT Series hardware user manual and documentation

- *Getting Started with LabVIEW*

- *LabVIEW User Manual*

- *LabVIEW Application Builder Release Notes*

- Measurement & Automation Explorer (MAX), available by selecting **Tools»Measurement & Automation Explorer** in LabVIEW

- *MAX Help*, available by selecting **Help»Help Topics** in MAX

Refer to your RT Series device user manual and the *LabVIEW Real-Time User Manual* for more information about programming LabVIEW RT.

Installation

You must install LabVIEW RT on a PC or PXI system running Windows to use LabVIEW RT with either the PCI/PXI-7030 plug-in board or a networked RT Series device. The Windows-operated PC or PXI system serves as the host PC for LabVIEW RT applications. Refer to Chapter 2, *Installation*, of the *LabVIEW Real-Time User Manual* for more information about installing LabVIEW RT and drivers, such as NI-DAQ, on the host PC.

When you install LabVIEW RT on the host PC, LabVIEW RT automatically installs Measurement & Automation Explorer (MAX). Use MAX to configure PCI or PXI 7030 plug-in boards or networked RT Series PXI systems for use with LabVIEW RT. Use FieldPoint Explorer to configure networked RT Series FieldPoint systems.

Driver Compatibility

Because LabVIEW RT uses driver software on the host PC and in the RT Engine, only certain driver software versions are compatible with LabVIEW RT. The following table summarizes LabVIEW RT and driver software version compatibility.

LabVIEW RT Version	Compatible NI-DAQ Versions	NI-CAN	FlexMotion	NI-Serial	NI-VISA	NI-488.2
5.1	6.5.2, 6.6	—	—	—	—	—
5.1.2	6.7 or later	—	—	1.0	—	1.0
6.0.3	6.9.1	1.5	5.1	2.0	2.52	1.0
6.1	6.9.1	1.5	5.1	2.0	2.52	1.0

The following table summarizes the drivers that LabVIEW RT supports on the FieldPoint platform.

LabVIEW RT Version	IA Control Environment	NI Serial	FieldPoint VI Manager
5.1	—	—	—
5.1.2	—	—	—
6.0.3	4.0	2.0	1.0
6.1	4.0	2.0	1.0

Examples

Refer to the VIs in examples\rt for examples of real-time programming and control applications. The LabVIEW RT examples are grouped into the following VI libraries:

- RT Communication.llb
- RT Control (7030).llb
- RT Control.llb
- RT Tutorial.llb
- RT Watchdog (PXI-8156B).llb

RT Communication

RT Communication examples demonstrate how to communicate data between RT Engine VIs and host PC VIs. You can use the VI Server and TCP/IP examples with the RT Engine on RT Series DAQ devices or networked RT Series devices. You can use the shared memory examples only with the RT Engine on RT Series DAQ devices.

RT Control (7030)

RT Control (7030) examples are similar to the Proportional Integral Derivative (PID) control examples in RT Control.llb. However, RT Control (7030) examples use shared memory to communicate with VIs on the host PC. You can use the RT Control (7030) VIs only with PCI/PXI-7030 plug-in boards.

RT Control

RT Control examples demonstrate various PID control applications that you can run on RT Series devices. The examples use VI Server and TCP/IP for communication. Therefore, you can use them with either PCI/PXI-7030 plug-in boards or networked RT Series devices.

RT Tutorial

RT Tutorial examples illustrate various techniques for programming real-time applications in LabVIEW RT. The examples demonstrate VI priority settings, benchmarking utilities, and use of the LEDs on RT Series DAQ devices.

RT Watchdog

RT Watchdog examples demonstrate how to use the watchdog timer VIs to safely shut down or restart a LabVIEW RT application if an application crashes. These examples work only on PXI controllers.

LabVIEW RT 6.1 Features

LabVIEW RT 6.1 includes the following new features:

- LabVIEW RT supports embedded remote panels so you can view and control your applications from a Web browser.

- LabVIEW RT supports new LabVIEW RT hardware targets.

- You can use LabVIEW RT to switch execution targets from the **LabVIEW** dialog box. For LabVIEW RT users, the **LabVIEW Tutorial** button on the dialog box is replaced with a **Switch Target** button. You also can click the target history link to reconnect to a recently used target.

- *LabVIEW Help* includes information about LabVIEW RT VIs. You can now search *LabVIEW Help* and *LabVIEW Real-Time Help* in one window.

LabVIEW RT 6.0.3 included the following new features:

- MAX replaced Remote System Explorer as the tool used to configure networked RT Series devices for use with LabVIEW RT.

- LabVIEW RT supports FlexMotion PXI motion controllers.

- LabVIEW RT supports NI plug-in PXI Serial communication modules.

- LabVIEW RT includes NI-VISA 2.5.2. You can use NI-VISA 2.5.2 to communicate with PXI, GPIB, and serial instruments.

- You can use Point-by-Point Analysis VIs for online analysis of continuously acquired signals.

Known Issues

The following are known issues with LabVIEW RT 6.1:

- If you use LabVIEW RT with a PXI controller, you *must* ensure that the hard drive of the PXI controller is not formatted with Windows NT File System (NTFS). If your PXI controller has an NTFS system, you must convert the file system to FAT16 or FAT32 to use LabVIEW RT.

- You can programatically change the properties of front panel controls and indicators in LabVIEW by using Property Nodes on a block diagram. Similarly, when you target, or connect, LabVIEW RT to the RT Engine on an RT Series device, Property Nodes affect the front panel of the RT Development System on the host computer. However, if you disconnect the RT Development System from the RT Engine and subsequently reconnect it, you must reset the properties for them to take effect on the reopened front panel. The same applies to embedded remote panels.

- The RT Series PXI Controller cannot use the internal SCXI control bus in a PXI/MXI-III system without modification to the `PXISYS.INI` file on the RT Series PXI Controller. This issue only affects users who connect the PXI chassis to the downstream side of the MXI-III bridge and want to use the internal SCXI bus to control the SCXI modules. Refer to `ni.com\support` for more information about the RT Series PXI Controller.

- Time-critical LabVIEW RT applications take priority over real-time operating system tasks. Because the real-time operating system maintains the date and time information, the Get Date/Time in Seconds function might return inaccurate values. Timing functions such as Tick Count (ms) are not affected by this problem.

- PXI floppy disk utilities created using LabVIEW RT 5.1.2 are not compatible with LabVIEW RT 6.1. You must use MAX to create new PXI utility disks compatible with LabVIEW RT 6.1. PXI floppy disks created in LabVIEW RT 6.0.3 are supported in LabVIEW RT 6.1.

- LabVIEW RT supports only ISA-based GPIB controllers. LabVIEW RT does not support PXI controllers with on-board PCI-based GPIB interfaces. However, certain PXI controllers with ISA-based on-board GPIB interfaces, such as the PXI 8156B, are supported in LabVIEW RT.

- The General Error Handler VI changed in LabVIEW 6.1. The correct files are copied automatically to an RT engine at bootup. If you are using custom errors, place the files in `C:\NI-RT\ SYSTEM\ERRORS` on the RT engine. This directory also contains specific language folders you can use if you want to localize your custom errors. To conform to the 8.3 filename structure, name the error with a `.err` extension rather than `-errors.txt`.

  Two configuration tokens are available for error files. Set `RTDownloadErrors=TRUE` in the embedded system's `NI-RT.INI` file to cause the RT engine to request a list of available errors the first time it is targeted. This request only occurs if the RT engine cannot find any error files locally, such as an RT engine that does not have disk I/O. The host machine searches for all available error files, and transfers the parsed error data back to the RT target. If you want to specify the actual filenames that the host system should parse, you can create a list using the `RTDownloadErrorList` token in the RT Engine's `NI-RT.INI` file. Refer to *LabVIEW Help* for more information about the changes to the General Error Handler VI and using error code files.