

COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

 Sell For Cash  Get Credit  Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New**, **New Surplus**, **Refurbished**, and **Reconditioned** NI Hardware.



Bridging the gap between the manufacturer and your legacy test system.

 1-800-915-6216

 www.apexwaves.com

 sales@apexwaves.com

All trademarks, brands, and brand names are the property of their respective owners.

Request a Quote

 **CLICK HERE**

PXI-8108

Getting Started with the LabVIEW™ Real-Time Module

This document provides exercises designed to introduce you to the LabVIEW Real-Time Module. Over the course of these exercises, you will examine, modify, and deploy a real-time application. You also will learn concepts and practices for programming a real-time operating system.

Contents

Key Concepts.....	1
Installing and Configuring the Real-Time Module	2
Installing the Real-Time Module.....	2
Configuring the RT Target	3
Installing RT Target Software	3
Optimizing Multiple-CPU Support	3
Creating a Real-Time Application.....	4
Building and Deploying an Application.....	4
Configuring RT Target Properties	4
Deploying a Build Specification to the RT Target	5
Debugging RT Target VIs and Stand-Alone Applications.....	5
Debugging RT Target VIs	5
Debugging Stand-Alone Real-Time Applications.....	6
Where to Go from Here	7
LabVIEW Help.....	7
LabVIEW Real-Time Module Release and Upgrade Notes.....	7

Key Concepts

The following key concepts provide the basic information you need to start using the Real-Time Module:

- **Real-time (RT) application**—An application designed for stable execution and precise timing.
- **Determinism**—The characteristic of a real-time application that describes how consistently the application responds to external events or performs operations within a given time limit. Maximizing determinism is often a priority when designing real-time applications.
- **Jitter**—The time difference between the fastest and slowest executions of the application. Minimizing jitter is often a priority when designing real-time applications.
- **Real-time operating system (RTOS)**—An operating system designed to run applications with increased determinism and reduced jitter. A general-purpose operating system, like

Microsoft Windows, completes operations at unpredictable times. In contrast, each operation an RTOS performs has a known maximum completion time. By designing an application for an RTOS, you can make sure an application will run deterministically.

- **RT target**—A controller, such as an NI PXI-8108 or a cRIO-9012, that runs an RTOS.
- **Stand-alone RT application**—An RT application that runs automatically when you power on an RT target.
- **Device driver software**—A software component that translates commands from LabVIEW into a format appropriate for a particular RT target and any installed I/O devices. You install the appropriate device driver software as a part of configuring your RT target.
- **Host computer**—The computer you use to design a real-time application. You deploy a real-time application from the host computer to the RT target. You can also communicate with the RT target through a user interface running on the host computer.
- **NI Measurement & Automation Explorer (MAX)**—The software you use to configure RT targets. After you install the Real-Time Module on the host computer, you can use MAX to install the Real-Time Module, the RTOS, and device driver software on the RT target.
- **Subnet**—A subdivision of a network over which devices can communicate using TCP/IP protocol. MAX automatically detects RT targets connected to the same subnet as the host computer.
- **Shared variable**—A memory space that you can read data from and write data to. You can read and write shared variables on a single computer with single-process shared variables or on multiple computers with network-published shared variables. Use shared variables to publish only the latest values in a data set to one or more computers.
- **RT FIFO**—Acts like a fixed-size queue, where the first value you write to the FIFO queue is the first value that you can read from the FIFO queue. An RT FIFO ensures deterministic behavior by imposing a size restriction on the data you share and by pre-allocating memory for the data. Use RT FIFO functions to share data between VIs or parallel loops running on an RT target.
- **Network stream**—A lossless, unidirectional, one-to-one communication channel that consists of a writer endpoint and a reader endpoint. Use network streams to stream lossless data over a network.

Installing and Configuring the Real-Time Module

You must perform the following tasks before developing a real-time application:

1. Install the Real-Time Module on the host computer
2. Configure the RT target
3. Install the RTOS and device driver software on the RT target

Installing the Real-Time Module

Refer to the *LabVIEW Real-Time Module Release and Upgrade Notes*, available in your software kit or at ni.com/manuals, for the Real-Time Module and device driver installation instructions.

Configuring the RT Target

After you install the Real-Time Module on the host computer, use MAX to configure the general settings of the target.



Note To perform the initial configuration, you must connect networked RT targets to the same subnet as the host computer from which you launch MAX.

Refer to the *MAX Remote Systems Help* book in the **Contents** tab of the *Measurement & Automation Explorer Help*, available from MAX by selecting **Help»MAX Help**, for information about configuring RT targets.

Refer to the following additional resources for information about configuring RT targets:

- **LabVIEW Real-Time Target Configuration Tutorial**—Refer to the *LabVIEW Real-Time Target Configuration Tutorial* topic in the *Measurement & Automation Explorer Help* for a step-by-step tutorial about configuring networked RT targets.
- **Miscellaneous and Advanced Settings**—Refer to the *Configuring RT Target Settings* topic of the *LabVIEW Help* for information about configuring additional settings for an RT target.

Installing RT Target Software

After you configure the RT target, use the LabVIEW Real-Time Software Wizard in MAX to install the software and drivers from the host computer on the RT target. Complete the following steps to launch the LabVIEW Real-Time Software Wizard:

1. Launch MAX.
2. Find and expand your target in the **Remote Systems** tree.
3. Right-click **Software** and select **Add/Remove Software**.

When you select **Add/Remove Software**, MAX launches the LabVIEW Real-Time Software Wizard. This displays all the National Instruments software and drivers installed on the host computer that you can install on a selected RT target.

4. Click the icon next to **LabVIEW Real-Time**.
5. Select **Install the feature**.

Optimizing Multiple-CPU Support

If your RT target has a multi-core CPU, refer to the *Optimizing RT Applications for Multiple-CPU Systems* book in the **Contents** tab of the *LabVIEW Help* for information about optimizing RT applications for multi-core CPU systems.

You also can refer to the National Instruments Web site at ni.com/info and enter the Info Code `rt_smp` for further details about optimizing RT applications for systems with multiple CPUs.

Creating a Real-Time Application

Refer to the *LabVIEW Help* for a tutorial that guides you through creating a real-time application. The application you create contains VIs that acquire and process data on the RT target and display that data on the host computer. Complete the following steps to open the tutorial:

1. Launch LabVIEW.
2. Select **Help»LabVIEW Help**.
3. Navigate to the *Tutorial: Creating a Real-Time Application* topic in the *Real-Time Module How-To* book of the *Real-Time Module Help*.
4. Follow the instructions in the tutorial to create a real-time application.

Building and Deploying an Application

Now that you have completed the tutorial, you can build a stand-alone real-time application and deploy it to an RT target. For this section, you use a pre-built example of a real-time application similar to the one described in the tutorial. Complete the following steps to open the example:

1. Launch LabVIEW.
2. Open the Example Finder by selecting **Help»Find Examples**.
3. Browse to **Toolkits and Modules»Real-Time»Tutorial**.
4. Double-click `Getting Started.lvproj` to open the **Project Explorer** window.

Configuring RT Target Properties

Use the RT target **Properties** dialog box to configure the project settings of an RT target.

Complete the following steps to enter the IP address of an RT target in the **Project Explorer** window:

1. Right-click the RT Target and select **Properties** to open the RT target **Properties** dialog box.
2. Enter the IP address of your RT target in the **IP Address / DNS Name** field and click the **OK** button.

You must deploy the RT target before the settings take effect. Right-click the RT target and select **Deploy** from the shortcut menu to deploy settings to the target.



Note Deploying an RT target overwrites the current target settings with the latest settings specified in the RT target **Properties** dialog box.

Deploying a Build Specification to the RT Target

The Getting Started project includes the `My Real-Time Application` build specification. Using this build specification, you will build and run a stand-alone real-time application on an RT target. You will also set the application to launch when you power on the RT target.

Complete the following steps to build and run a stand-alone real-time application on an RT target:

1. In the **Project Explorer** window, right-click `My Real-Time Application` and select **Build** from the shortcut menu to build the stand-alone real-time application. The Application Builder builds the stand-alone real-time application configured with the settings defined in the build specification. When the build completes, click **Done** to close the **Build Status** dialog box.
2. Right-click the build specification and select **Run as startup** from the shortcut menu. The **Run as startup** shortcut menu item sets the application as the startup application, deploys the application to the target, and prompts you to reboot the RT target.

When you set a build specification as the startup application, LabVIEW displays the build specification with a green border around the specification icon in the **Project Explorer** window. Setting a build specification as startup enables the application generated from the build specification to launch automatically when you power on the RT target.

3. (Optional) If you plan to monitor the application using a host VI, prevent shared variables from automatically deploying by right-clicking the RT target in the **Project Explorer** window and selecting **Disable Autodeploy Variables**. If variable autodeploy is enabled when you run the host VI, LabVIEW closes the stand-alone application running on the target and redeploys shared variable settings.
4. From the **Project Explorer** window, open the `Windows Main.vi` under **My Computer**.
5. Enter the IP address of the RT target in the **Target URL** text box.
6. Run the VI to display a Waveform Chart of the data acquired and sent from the stand-alone real-time application.

Debugging RT Target VIs and Stand-Alone Applications

You can use the LabVIEW tools and utilities to debug RT target VIs and stand-alone real-time applications that are in memory on an RT target.

Debugging RT Target VIs

You can open a front panel connection with an RT target and use the LabVIEW debugging tools to debug the VIs running on the target.

Use the LabVIEW debugging tools, such as execution highlighting and single-stepping, while the host computer is connected to an RT target to step through LabVIEW code. Refer to the *LabVIEW Help* for more information about using the LabVIEW debugging tools to debug VIs.

Debugging Stand-Alone Real-Time Applications

Use the **Debug Application or Shared Library** dialog box to debug stand-alone real-time applications running on an RT target.

You must enable debugging for the stand-alone real-time application to connect to the application using the **Debug Application or Shared Library** dialog box. Complete the following steps to enable debugging for the stand-alone real-time application in the Getting Started project:

1. In the **Project Explorer** window, right-click **My Real-Time Application** under **Build Specifications** on the RT target and select **Properties**.
2. Click **Advanced** in the **Category** list.
3. Place a checkmark in the **Enable debugging** checkbox and click **OK**.
4. Right-click *My Real-Time Application* and select **Build** from the shortcut menu to build the stand-alone real-time application.

Complete the following steps to debug a stand-alone real-time application using the **Debug Application or Shared Library** dialog box:

1. Right-click *My Real-Time Application* and select **Run as startup** from the shortcut menu to run the application on the RT target.
2. From the **Project Explorer** window, select **Operate»Debug Application or Shared Library** to open the **Debug Application or Shared Library** dialog box.
3. Enter the IP address of the RT target in the **Machine name or IP address** text box and click the **Refresh** button to view the list of stand-alone real-time applications that are in memory on the RT target.
4. Select *startup.rtexe* from the **Application or shared library** pull-down menu and click the **Connect** button to connect to the startup application for the RT target.

LabVIEW opens the front panel of the startup VI for debugging. You can access the block diagram and use probes, breakpoints, and other LabVIEW debugging techniques on the block diagram to identify errors in the application.

5. Close the startup VI and the **Debug Application or Shared Library** dialog box.
6. Disable the stand-alone real-time application to launch when you power on the RT target and then deploy the build specification to apply the changes to the startup property.

Where to Go from Here

The Real-Time Module includes a comprehensive documentation set designed to help you create deterministic applications to run on RT targets.

LabVIEW Help

The *LabVIEW Help*, available by selecting **Help»LabVIEW Help** in LabVIEW, contains the following information that is specific to the Real-Time Module:

- **Real-Time Module Best Practices**—Information about best practices for designing, developing, and deploying applications with the Real-Time Module.
- **Real-Time Module Concepts**—Information about programming concepts, application architectures, and Real-Time Module features you can use to create deterministic applications.
- **Real-Time Module How-To**—Step-by-step instructions for using Real-Time Module features.
- **Real-Time VIs**—Reference information about Real-Time Module VIs, functions, and error codes.
- **Real-Time Operating Systems**—Information about using LabVIEW on real-time operating systems.
- **Real-Time Module Error Codes**—Information about error codes specific to the Real-Time Module.

LabVIEW Real-Time Module Release and Upgrade Notes

The *LabVIEW Real-Time Module Release and Upgrade Notes* contains information to help you install and configure the Real-Time Module and a list of upgrade issues and new features.

Complete the following steps to access this document:

1. Open the `labview\manuals` directory.
2. Double-click `RT_Release_Upgrade_Notes.pdf` to open this manual.

Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for more information on National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patents Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the readme file for your NI product. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data. NI MAKES NO EXPRESS OR IMPLIED WARRANTIES AS TO THE ACCURACY OF THE INFORMATION CONTAINED HEREIN AND SHALL NOT BE LIABLE FOR ANY ERRORS. U.S. Government Customers: The data contained in this manual was developed at private expense and is subject to the applicable limited rights and restricted data rights as set forth in FAR 52.227-14, DFAR 252.227-7014, and DFAR 252.227-7015.

© 2003–2015 National Instruments. All rights reserved.