Getting Started with the LabVIEW™ Real-Time Module

This document provides exercises designed that 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 will also learn concepts and practices for programming a real-time operating system.

Contents

NATIONAL INSTRUMENTS™

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 an NI 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.
- **Real-Time (RT) Installation/Configuration Troubleshooter**— Refer to the National Instruments Web site at `ni.com/info` and enter the Info Code `rtconfig` to access the *Real-Time (RT) Installation/Configuration Troubleshooter* page that contains links to instructional resources to help you install and configure RT targets.

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**.

Installing Multiple-CPU Support

If your RT target has a multi-core CPU, use the LabVIEW Real-Time Software Wizard in MAX to install the NI RT Extensions for SMP. 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.

Note Only install the NI RT Extensions for SMP on multi-core CPU systems. Refer to the National Instruments Web site at `ni.com/info` and enter the Info Code `rtsmp` for further details about optimizing RT applications for systems with multiple CPUs.

Launching the Real-Time Project

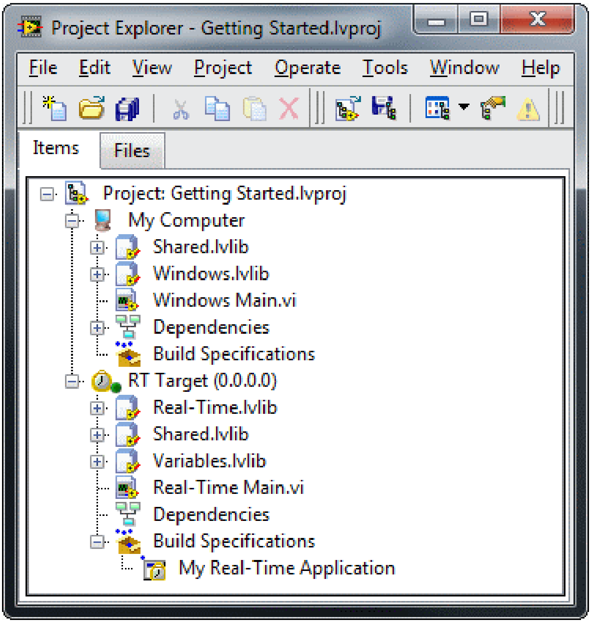
You must use a LabVIEW project to build real-time applications and communicate with RT targets. You can also use LabVIEW projects to deploy files and stand-alone real-time applications to an RT target.

In this tutorial, you will analyze the Getting Started project. This project 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 Getting Started project.

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. Figure 1 shows the **Project Explorer** window for `Getting Started.lvproj`.

Figure 1. The **Project Explorer** Window

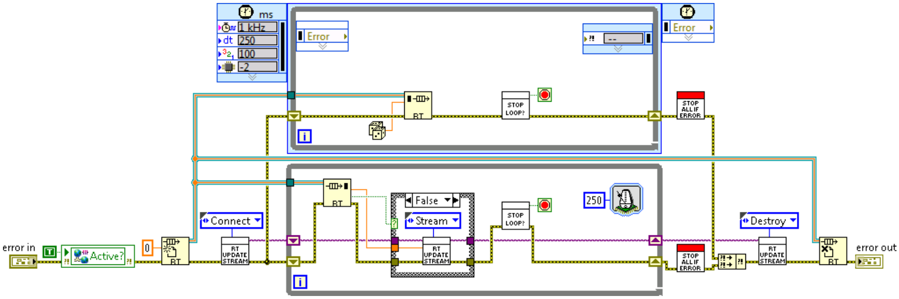


- Project: Getting Started.lvproj
  - My Computer
    - Shared.lvlib
    - Windows.lvlib
    - Windows Main.vi
    - Dependencies
    - Build Specifications
  - RT Target (0.0.0.0)
    - Real-Time.lvlib
    - Shared.lvlib
    - Variables.lvlib
    - Real-Time Main.vi
    - Dependencies
    - Build Specifications
      - My Real-Time Application

Explore the VI on the RT Target

Start by exploring the VI that runs on the RT target. Double-click `Real-Time Main.vi` in the **Project Explorer** window to open the VI.

Figure 2. The Real-Time Main VI



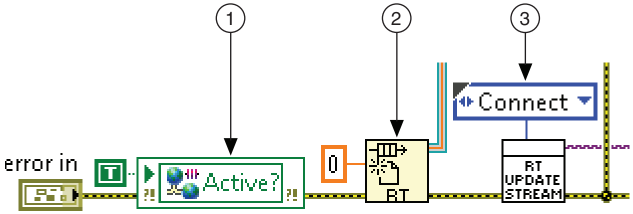
The Real-Time Main VI generates and streams single points of data from a simulated data acquisition device. The following sections describe aspects of this VI that highlight common RT programming practices. Double-click the subVIs in the Real-Time Main VI to analyze their designs.

The Real-Time Main VI performs the following tasks:

- Initializes the data transfer components
- Acquires data with a Timed Loop
- Transfers data over the network with a While Loop
- Closes the data transfer components

Initialize the Data Transfer Components

Figure 3. Initializing the Data Transfer Components

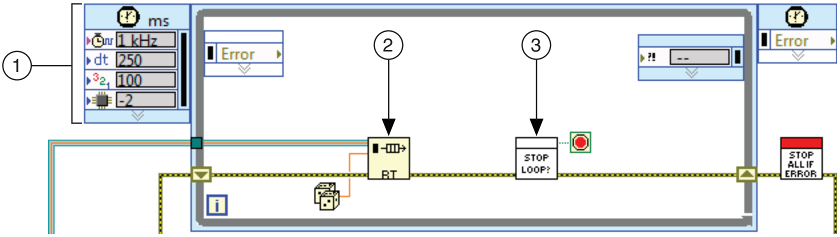


1. The **Active?** shared variable is initialized with a value of TRUE to prevent the shared variable from reading incorrect values when the application starts.
2. The RT FIFO Create function creates the RT FIFO, and the **type** input specifies the type of data the RT FIFO stores. This example does not specify a size for the RT FIFO, so the RT FIFO Create function uses the default of 10 elements.

   National Instruments recommends defining the number and size of the RT FIFO elements when designing your own applications. Specifying the number of elements in RT FIFOs helps prevent data logging errors. Refer to the *LabVIEW Help* for more information about defining the size and number of elements in an RT FIFO.
3. The Connect case of the RT Update Stream subVI creates the writer endpoint of a network stream, `RTAcqData`.

Acquire Data with a Timed Loop

Figure 4. Acquiring Data with a Timed Loop



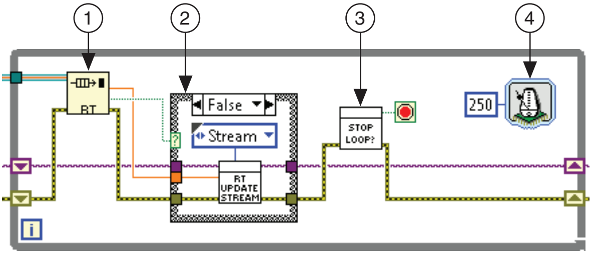
1. This configuration tool, known as the Input Node, displays terminals for customizing the Timed Loop. Double-click the Input Node to display the **Configure Timed Loop** dialog box, where you can configure additional Timed Loop settings. For example, this Timed Loop uses a period of 250 ms.

> **Note** Matching the Timed Loop period to the rest of the application equalizes the write rate and the read rate. The read rate of the application must be equal to or greater than the write rate of the application to ensure all data is read.

2. The RT FIFO Write function writes a single element from the Random Number (0-1) function to the RT FIFO, simulating data acquisition.
3. The Stop Loop? subVI stops the While Loop if an error occurs or if you click the **Stop Application** button on the front panel of the VI running on the host computer.

Transfer Data over the Network with a While Loop

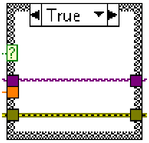
Figure 5. Transferring Data over the Network with a While Loop



1. The RT FIFO Read function reads the oldest element in the RT FIFO and sends it to the Case structure that contains the RT Update Stream subVI.
2. If the RT FIFO contains an element, this Case structure executes the FALSE case. The Stream case of the RT Update Stream SubVI writes the element from the RT FIFO Read

function to the RTAcqData network stream. If the RT FIFO is empty, this Case structure executes the blank TRUE case and no data enters the RTAcqData network stream.

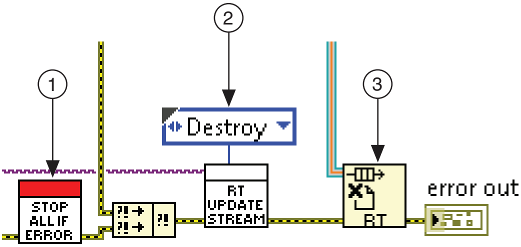
Figure 6. TRUE Case



3. The Stop Loop? subVI stops the While Loop if an error occurs or if you click the **Stop Application** button on the front panel of the VI running on the host computer.
4. The Wait Until Next Multiple VI changes the period of the While Loop to 250 ms. Use a While Loop with the Wait Until Next Multiple VI to control the execution rate of functions that require less determinism than a Timed Loop, such as transferring data over a network.

Close the Data Transfer Components

Figure 7. Closing the Data Transfer Components

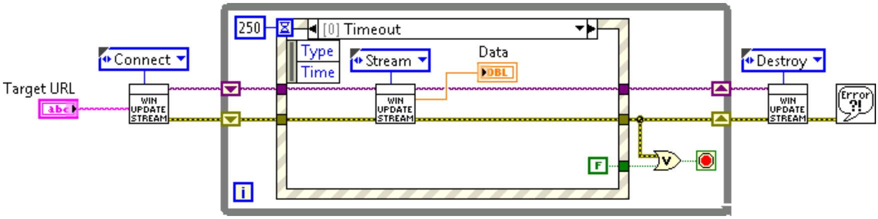


1. If an error stops one loop, the Stop All If Error subVI stops the other loop. National Instruments recommends implementing similar functionality in all multiple-loop applications.
2. The Destroy case of the RT Update Stream subVI destroys the endpoint of the RTAcqData network stream. Destroying both the reading and writer endpoints of a network stream frees the memory allocated for the stream.
3. The RT FIFO Delete function closes the RT FIFO reference and frees the memory allocated for the FIFO.

Explore the VI on the Host Computer

After you examine the Real-Time Main VI, double-click `Windows Main.vi` in the **Project Explorer** window to open the VI that runs on the host computer.

Figure 8. The Windows Main VI



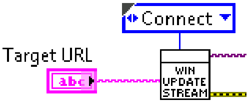
The Windows Main VI reads the data streaming from the RT target and displays it on a Waveform Chart. The following sections describe aspects of this VI that highlight common RT programming practices. Double-click the subVIs in the Windows Main VI to analyze their designs.

The Windows Main VI performs the following tasks:

- Connects to the network stream
- Reads from the network stream
- Disconnects from the network stream

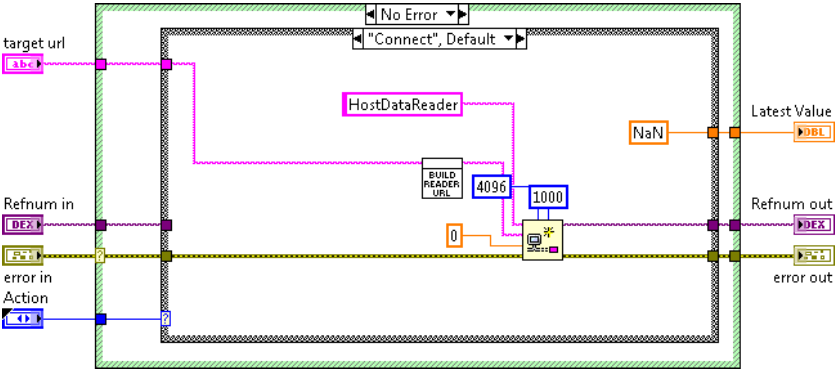
Connect to the Network Stream

Figure 9. Connecting to the Network Stream



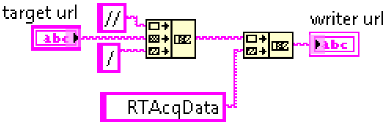
The Connect case in the Win Update Stream subVI uses the Create Network Stream Reader Endpoint function to create a reader endpoint for the network stream writer endpoint in the RT Main VI, as shown in Figure 10.

Figure 10. The Connect Case of the Win Update Stream SubVI



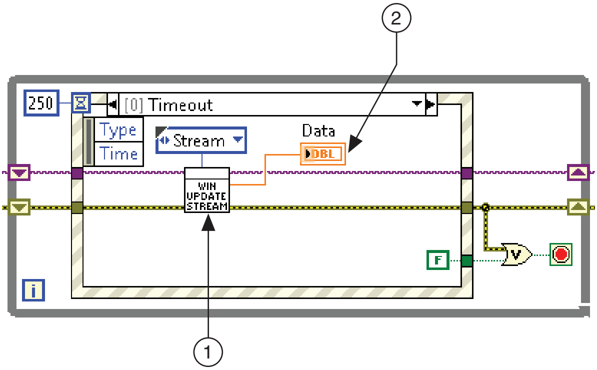
The Build Reader URL subVI uses Concatenate Strings functions to combine strings into the writer URL for the network stream, as shown in Figure 11.

Figure 11. Build Reader URL SubVI



Read from the Network Stream

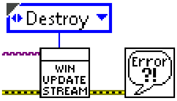
Figure 12. Reading from the Network Stream



1. The Stream case in the Win Update Stream subVI uses the Read Single Element from Stream function to read data the RT target streams over the network. Double-click the Win Update Stream subVI and select the Stream case to view this function.
2. The data from the RT target displays on the Waveform Chart on the front panel of the Windows Main VI.

Disconnect from the Network Stream

Figure 13. Disconnecting from the Network Stream



1. The Destroy case in the Win Update Stream subVI uses the Destroy Stream Endpoint function to close the reference to the network stream. Destroying both the reader and writer endpoints of a network stream frees the memory allocated for the stream. Double-click the Win Update Stream subVI and select the Destroy case to view this function.

Note Always close network stream references that are no longer needed.

Building and Deploying an Application

Now that you have explored the Getting Started project, you can build a stand-alone real-time application and deploy it to an RT target.

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.
- **Unsupported LabVIEW Features**—Information about LabVIEW features that are not supported by the Real-Time Module.
- **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. Select **Start»All Programs»National Instruments»LabVIEW»LabVIEW Manuals** to open the `labview\manuals` directory.
2. Double-click `RT_Release_Upgrade_Notes.pdf` to open this manual.

371375D-01 Jun12