

Simulink® Real-Time™

Getting Started Guide



MATLAB® & SIMULINK®

R2019b



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Simulink® Real-Time™ Getting Started Guide

© COPYRIGHT 2000–2019 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

November 2000	First printing	New for Version 1 (Release 12)
June 2001	Online only	Revised for Version 1.2 (Release 12.1)
September 2001	Online only	Revised for Version 1.3 (Release 12.1+)
July 2002	Second printing	Revised for Version 2 (Release 13)
September 2003	Online only	Revised for Version 2.0.1 (Release 13SP1)
June 2004	Third printing	Revised for Version 2.5 (Release 14)
August 2004	Online only	Revised for Version 2.6 (Release 14+)
October 2004	Fourth printing	Revised for Version 2.6.1 (Release 14SP1)
November 2004	Online only	Revised for Version 2.7 (Release 14SP1+)
March 2005	Online only	Revised for Version 2.7.2 (Release 14SP2)
September 2005	Online only	Revised for Version 2.8 (Release 14SP3)
March 2006	Online only	Revised for Version 2.9 (Release 2006a)
May 2006	Fifth printing	Revised for Version 3.0 (Release 2006a+)
September 2006	Online only	Revised for Version 3.1 (Release 2006b)
March 2007	Online only	Revised for Version 3.2 (Release 2007a)
September 2007	Online only	Revised for Version 3.3 (Release 2007b)
March 2008	Online only	Revised for Version 3.4 (Release 2008a)
October 2008	Sixth printing	Revised for Version 4.0 (Release 2008b)
March 2009	Online only	Revised for Version 4.1 (Release 2009a)
September 2009	Online only	Revised for Version 4.2 (Release 2009b)
March 2010	Online only	Revised for Version 4.3 (Release 2010a)
September 2010	Seventh printing	Revised for Version 4.4 (Release 2010b)
April 2011	Online only	Revised for Version 5.0 (Release 2011a)
September 2011	Online only	Revised for Version 5.1 (Release 2011b)
March 2012	Online only	Revised for Version 5.2 (Release 2012a)
September 2012	Online only	Revised for Version 5.3 (Release 2012b)
March 2013	Online only	Revised for Version 5.4 (Release 2013a)
September 2013	Online only	Revised for Version 5.5 (Release 2013b)
March 2014	Online only	Revised for Version 6.0 (Release 2014a)
October 2014	Online only	Revised for Version 6.1 (Release 2014b)
March 2015	Online only	Revised for Version 6.2 (Release 2015a)
September 2015	Online only	Revised for Version 6.3 (Release 2015b)
March 2016	Online only	Revised for Version 6.4 (Release 2016a)
September 2016	Online only	Revised for Version 6.5 (Release 2016b)
March 2017	Online only	Revised for Version 6.6 (Release 2017a)
September 2017	Online only	Revised for Version 6.7 (Release 2017b)
March 2018	Online only	Revised for Version 6.8 (Release 2018a)
September 2018	Online only	Revised for Version 6.9 (Release 2018b)
March 2019	Online only	Revised for Version 6.10 (Release 2019a)
September 2019	Online only	Revised for Version 6.11 (Release 2019b)

1 Introduction

Simulink Real-Time Product Description	1-2
Key Features	1-2
Target Computer Configuration and Control Methods	1-3
MATLAB Command-Line Interface	1-6
Simulink External Mode Interface	1-8
Add Simulink Real-Time Blocks to Simulink Model	1-9
Target Computer Command-Line Interface	1-10
Custom UI with Simulink Real-Time API for Microsoft .NET Framework	1-11
Custom UI with Simulink Real-Time C API	1-12

2 Installation and Configuration

Development Computer Setup and Configuration	2-2
Target Computer Setup and Configuration	2-3
Development Computer Requirements	2-4
Peripherals	2-4
Communication with Target Computer	2-4

Development Computer Communication Setup	2-6
Development Computer Software Installation	2-7
License Requirements	2-7
Location of Files on Development Computer	2-8
PCI Bus Ethernet Setup	2-9
Target Computer Settings	2-12
Target Computer Boot Methods	2-14
Network Boot Method	2-15
DOS Loader Boot Method	2-19
Standalone Boot Method	2-20
Standalone Settings	2-20
Real-Time Application Build	2-21
Real-Time Application Transfer and Boot Configuration	2-21
Application Transfer and Boot Configuration with USB Flash Drive	2-22
Standalone Mode	2-24
Run Confidence Test on Configuration	2-26

Basic Workflows

3

Real-Time Simulation and Testing	3-2
Real-Time Simulation and Testing Workflow	3-2
Special Purpose Tasks	3-6

Set Up and Configure Simulink Real-Time	4-2
Configure Link Between Development and Target Computers	4-2
Configure Target Settings	4-4
Configure Boot Configuration	4-5
Run the Confidence Test	4-7
Create and Run Real-Time Application from Simulink Model	4-9
Transform Simulink Model to Real-Time Application	4-10
Start Target Computer	4-17
Build and Download Real-Time Application by Using Run on Target	4-18
Execute Real-Time Application in Simulink External Mode by Using Step by Step Commands	4-19
Apply Simulink Real-Time Model Template to Create Real-Time Application	4-23
Configure and Control a Real-Time Application	4-26
Execute Real-Time Application with Simulink Real-Time Explorer	4-26
Change Stop Time and Sample Time	4-29
Process a Real-Time Application by Using Simulink Real-Time Explorer	4-31
Load a Preexisting Real-Time Application	4-31
Unload a Real-Time Application from the Target Computer ..	4-31
Simulate Simulink Model by Using MATLAB Language	4-33
Prepare Real-Time Application by Using MATLAB Language	4-36
Execute Real-Time Application with MATLAB Language	4-37
Application and Driver Scripts	4-40
General Examples	4-40
Driver Examples	4-41

Edit Scripts	4-42
--------------------	------

Introduction

- “ Simulink Real-Time Product Description” on page 1-2
- “Target Computer Configuration and Control Methods” on page 1-3
- “MATLAB Command-Line Interface” on page 1-6
- “Simulink External Mode Interface” on page 1-8
- “Add Simulink Real-Time Blocks to Simulink Model” on page 1-9
- “Target Computer Command-Line Interface” on page 1-10
- “Custom UI with Simulink Real-Time API for Microsoft .NET Framework” on page 1-11
- “Custom UI with Simulink Real-Time C API” on page 1-12

Simulink Real-Time Product Description

Build, run, and test real-time applications

Simulink Real-Time lets you create real-time applications from Simulink models and run them on dedicated Speedgoat target computer hardware connected to your physical system. It supports real-time simulation and testing, including rapid control prototyping, DSP and vision system prototyping, and hardware-in-the-loop (HIL) simulation.

With Simulink Real-Time you can extend your Simulink models with driver blocks, automatically generate real-time applications, define instrumentation, and perform interactive or automated runs on a dedicated target computer equipped with a real-time kernel, multicore CPU, I/O and protocol interfaces, and FPGAs.

Simulink Real-Time and Speedgoat target computer hardware are expressly designed to work together to create real-time systems for desktop, lab, and field environments.

Key Features

- Automatic generation of real-time applications from Simulink models targeting dedicated CPUs, I/O and protocol hardware, and FPGAs (with HDL Coder™)
- Multitasking and multicore real-time kernel with microsecond granularity and concurrent execution support
- Speedgoat target computer hardware integration for turnkey desktop, lab, and field use
- Driver blocks for I/O, including analog, digital, pulse train, encoders, transformers, passive components, serial, audio, shared memory, and reconfigurable FPGA
- Driver blocks for protocols and data buses, including Raw Ethernet, real-time UDP, CAN, EtherCAT®, Ethernet/IP, Lin, SAE J1939, FlexRay™, Camera Link®, USB video, ARINC 429, and MIL-STD-1553
- Simulink Real-Time Explorer with gigabit Ethernet connection to multiple target computers for management, execution, and instrumentation of real-time applications
- Standalone operation of real-time applications with high-resolution signal display
- MATLAB® functions for test scripting, and APIs for developing standalone client applications and user interfaces (Visual Basic®, C/C++, Java®, and .NET)

Target Computer Configuration and Control Methods

The Simulink Real-Time environment has a modifiable interface to the target computer. You can use this interface from MATLAB or Simulink. You can use other development environments to create custom standalone client applications independent of MATLAB. Because of this open environment, there are several ways to interact with your real-time application from the development and target computers.

This table compares the interfaces supported by the Simulink Real-Time product.

Interface	Environment Properties	Control	Signal Acquisition	Parameter Tuning	Runnable Outside MATLAB
Simulink Real-Time Explorer	Yes	Yes	Yes	Yes	Yes
“MATLAB Command-Line Interface” on page 1-6	Yes	Yes	Yes	Yes	N/A
“Simulink External Mode Interface” on page 1-8	N/A	Yes	Yes	Yes	N/A
“Add Simulink Real-Time Blocks to Simulink Model” on page 1-9	N/A	N/A	Yes	N/A	N/A
“Target Computer Command-Line Interface” on page 1-10	N/A	Yes	Yes	Yes	Yes

Interface	Environment Properties	Control	Signal Acquisition	Parameter Tuning	Runnable Outside MATLAB
“Explorer Configuration Exported to Run Outside MATLAB”	N/A	Yes	Yes	Yes	Yes
“Deploy MATLAB Application to Control Real-Time Application”	N/A	Yes	Yes	Yes	Yes
“Custom UI with Simulink Real-Time API for Microsoft .NET Framework” on page 1-11	N/A	Yes	Yes	Yes	Yes
“Custom UI with Simulink Real-Time C API” on page 1-12	N/A	Yes	Yes	Yes	Yes

See Also

More About

- **Simulink Real-Time Explorer**
- “MATLAB Command-Line Interface” on page 1-6
- “Simulink External Mode Interface” on page 1-8
- “Add Simulink Real-Time Blocks to Simulink Model” on page 1-9

- “Target Computer Command-Line Interface” on page 1-10
- “Explorer Configuration Exported to Run Outside MATLAB”
- “Deploy MATLAB Application to Control Real-Time Application”
- “Custom UI with Simulink Real-Time API for Microsoft .NET Framework” on page 1-11
- “Custom UI with Simulink Real-Time C API” on page 1-12

MATLAB Command-Line Interface

You can interact with the Simulink Real-Time environment through the MATLAB command-line interface. On your development computer, enter Simulink Real-Time functions in the MATLAB window. You can also write your own MATLAB scripts that use Simulink Real-Time functions for batch processing.

The Simulink Real-Time software has more than 90 MATLAB functions for controlling the real-time application from the development computer. These functions define command line operations that you can do with the Simulink Real-Time environment.

Complete your most common tasks by using the Simulink Real-Time UIs. Through the UIs, you have access to the Simulink Real-Time capabilities, but not their extended functionality. The command-line interface provides an interactive environment that you can extend.

The MATLAB command-line interface includes:

- Environment — Create a target boot kernel and directly change the environment properties without using a graphical interface.
- Control — Restart the target computer, download a real-time application, start the real-time application, change start and sample times without regenerating code, and stop the real-time application. Record task execution time during or after the last run. Add and remove scopes, add and remove signals to scopes, and define triggers for scope display.
- Signal acquisition — Trace signals for viewing while the real-time application is running and monitor signal values without time information. Transfer logged signal data to the MATLAB workspace by uploading data from the target computer to the development computer between runs. For standalone target computers, if you write signal data to a file, use `SimulinkRealTime.fileSystem` functions to copy that file to the development computer.
- Parameter tuning — Change tunable block parameters and tunable global parameters while the real-time application is running. Use Simulink Real-Time capabilities to change parameters in between runs.

See Also

More About

- “Command-Line PCI Bus Ethernet Setup”
- “Command-Line Target Computer Settings”
- “Command-Line Target Computer Boot Methods”
- “Execute Real-Time Application with MATLAB Language” on page 4-37
- “Monitor Signals with MATLAB Language”
- “Configure Target Scopes with MATLAB Language”
- “Configure File Scopes with MATLAB Language”
- “Log Signal Data with Outport Block and MATLAB Language”
- “Tune Parameters with MATLAB Language”

Simulink External Mode Interface

To connect your Simulink block diagram to your real-time application, use Simulink in external mode. The block diagram becomes a user interface to the real-time application. By changing parameters in the Simulink blocks, you also change parameters in the real-time application.

The Simulink external mode interface includes:

- **Control** — Control is limited to connecting the Simulink block diagram to the real-time application and to starting and stopping the real-time application.
- **Signal acquisition** — Establish a communication channel between your Simulink block diagram and your real-time application. The block diagram becomes a user interface to your real-time application. Simulink scopes can acquire signal data from the real-time application.
- **Parameter tuning** — Specify external mode and change parameters in the real-time application by changing parameters in the Block Parameters dialog boxes. When you change a value and click **OK**, the new value is downloaded to the target computer while the real-time application continues to run.

See Also

More About

- “Execute Real-Time Application in Simulink External Mode by Using Step by Step Commands” on page 4-19
- “Trace Signals with Simulink External Mode”
- “Tune Parameters with Simulink External Mode”

Add Simulink Real-Time Blocks to Simulink Model

An alternative to interactively adding scopes to the target computer is to add Simulink Real-Time Scope blocks to your Simulink model. After the download process, these blocks create scopes on the target computer during initialization of the real-time application. You can display data on either the development computer or target computer. You can also save signal data (log real-time data stream) to a file in the target computer file system and transfer that file to another computer. Finally, you can use To and From blocks to transfer data to and from a Simulink AppDesigner user interface.

Signal acquisition — Add scopes to the target computer by adding Simulink Real-Time Scope blocks to your Simulink model. In the Block Parameters dialog box, select the scope mode and set the trigger.

See Also

More About

- “Add Simulink Real-Time Scope Block” on page 4-10
- “Set Target Scope Block Parameters” on page 4-11
- “Signal Tracing Basics”
- “Simulink Real-Time Interface Blocks to Simulink Models”

Target Computer Command-Line Interface

You can interact with the Simulink Real-Time environment through the target computer command window. On the target computer, enter commands at the command line by using the target computer keyboard or console. This interface is useful for working with standalone real-time applications that are not connected to the development computer.

Activate the target computer keyboard or console for target computer commands by typing **C**. The target computer command-line interface includes:

- Control — Start and stop the real-time application and change the stop time and sample time.
- Signal acquisition — Acquiring signal data is limited to viewing signal traces and signal monitoring on the target computer screen.
- Parameter tuning — You can change only scalar parameters in your model.

See Also

More About

- “Control Real-Time Application at Target Computer Command Line”
- “Target Computer Commands”

Custom UI with Simulink Real-Time API for Microsoft .NET Framework

To develop custom applications such as human-machine interface (HMI) software and batch run scripts that use the Simulink Real-Time software, use the .NET API Simulink Real-Time framework. The Simulink Real-Time .NET object model provides objects that you can interact with. The Simulink Real-Time software arranges the Simulink Real-Time .NET objects in a hierarchical order. Each of these objects has methods and properties that enable you to manipulate and interact with it. This document presents this reference using the C# language.

Note Support for the API for Microsoft® .NET Framework will be removed in a future release.

See Also

More About

- “Using the Simulink Real-Time API for Microsoft .NET Framework”

Custom UI with Simulink Real-Time C API

To create a user interface to a real-time application, use the C API by using a development environment that can link in a DLL.

To control the real-time application, tune parameters, and acquire signal data, use the user interface application. A user interface application runs on the development computer and communicates with the real-time application on the target computer by using TCP/IP communication. A user interface application can be a console or Windows® application using ActiveX® components.

Note Support for the Custom UI with Simulink Real-Time C API will be removed in a future release.

See Also

More About

- “Using the C API”

Installation and Configuration

- “Development Computer Setup and Configuration” on page 2-2
- “Target Computer Setup and Configuration” on page 2-3
- “Development Computer Requirements” on page 2-4
- “Development Computer Communication Setup” on page 2-6
- “Development Computer Software Installation” on page 2-7
- “PCI Bus Ethernet Setup” on page 2-9
- “Target Computer Settings” on page 2-12
- “Target Computer Boot Methods” on page 2-14
- “Network Boot Method” on page 2-15
- “DOS Loader Boot Method” on page 2-19
- “Standalone Boot Method” on page 2-20
- “Standalone Mode” on page 2-24
- “Run Confidence Test on Configuration” on page 2-26

Development Computer Setup and Configuration

The setup and configuration steps for using Simulink Real-Time on your development computer are:

- “Development Computer Requirements” on page 2-4
- “Development Computer Communication Setup” on page 2-6
- “Development Computer Software Installation” on page 2-7
- “Command-Line C Compiler Configuration”

Target Computer Setup and Configuration

The setup and configuration steps for using Simulink Real-Time on your target computer are:

- 1 “PCI Bus Ethernet Setup” on page 2-9
- 2 “Target Computer Settings” on page 2-12
- 3 “Target Computer Boot Methods” on page 2-14

To check out the setup, see “Run Confidence Test on Configuration” on page 2-26.

For more information about target computers, see “Speedgoat Target Computers and Support”.

Development Computer Requirements

To install and run Simulink Real-Time, MATLAB, Simulink, and other required and optional software, the development computer must be a 64-bit, PC-compatible system with Microsoft .NET Framework 4.5 or later installed. See [Installing the .NET Framework \(msdn.microsoft.com/en-us/library/5a4x27ek\(v=vs.110\).aspx\)](http://msdn.microsoft.com/en-us/library/5a4x27ek(v=vs.110).aspx).

In the Simulink Real-Time software environment, the development computer is usually your desktop computer. A notebook computer is also a viable development computer.

Peripherals

To install and run the Simulink Real-Time product, the development computer requires one hard disk drive with 60 MB of free space.

For producing target computer boot media, the development computer requires:

- CD-RW drive or DVD-RW drive
- USB drive (Universal Serial Bus drive)

You can use a USB drive as a removable boot drive and for data transfer to and from the target computer. For data storage, Simulink Real-Time supports 1-LUN (logical unit) USB drives, 2-LUN USB drives, and 4-LUN card readers.

- SD (Compact) flash drive
- Removable hard drive
- 3.5-inch floppy disk drive

Communication with Target Computer

For communicating with the target computer, the development computer requires one available Ethernet interface (PCI or USB) connected to a network. Configure the development computer Ethernet port to use Internet Protocol Version 4 (TCP/IPv4) only. Specify a nonroutable static IP address of the form `10.x.x.x`.

See Also

More About

- “Development Computer Communication Setup” on page 2-6
- “Development Computer Software Installation” on page 2-7
- “Command-Line C Compiler Configuration”
- “PCI Bus Ethernet Setup” on page 2-9

External Websites

- www.microsoft.com

Development Computer Communication Setup

To communicate with the target computer by using Ethernet, perform a one-time configuration of your development computer Ethernet port to use Internet Protocol Version 4 (TCP/IPv4) only. Specify a nonroutable static IP address of the form 10.x.x.x.

- 1 On your development computer, from the **Control Panel**, navigate to **Network and Sharing Center**.
- 2 Click the **Local Area Connection** for the Ethernet adapter that you are using. It is often listed as **Unidentified network**.
- 3 Click the **Properties** button.
- 4 In the **Networking** pane, clear all check boxes.
- 5 Select the **Internet Protocol Version 4 (TCP/IPv4)** check box.
- 6 Click the **Properties** button.
- 7 To specify a static IP address of the form 10.x.x.x, select the **Use the following IP address** check box.
- 8 Configure the IP address boxes to:

IP address — 10.10.10.100

Subnet mask — 255.255.255.0

Default gateway — leave blank
- 9 Click the **OK** button.

See Also

More About

- “Development Computer Requirements” on page 2-4
- “Development Computer Software Installation” on page 2-7
- “Command-Line C Compiler Configuration”

Development Computer Software Installation

You install the Simulink Real-Time software only on the development computer. The development computer downloads the kernel software and real-time application to the target computer at run time. The Simulink Real-Time software is distributed on a DVD or as a file that you download from the Internet.

Prerequisites:

- You must have uninstalled prior installations of the Simulink Real-Time product.
- You must have a valid license for the required products. See “License Requirements” on page 2-7.
- You must have Microsoft .NET Framework 4.5 or later installed on the development computer. If the framework is not installed, the installation software prints an error message. See [Installing the .NET Framework \(msdn.microsoft.com/en-us/library/5a4x27ek\(v=vs.110\).aspx\)](http://msdn.microsoft.com/en-us/library/5a4x27ek(v=vs.110).aspx).
- You must have a Microsoft C compiler installed on the development computer.

By default, the Microsoft Visual Studio® 2015 installer does not install the C++ compiler that Simulink Real-Time requires. To install the C++ compiler, perform a custom install and select the C++ compiler. If you have already installed Microsoft Visual Studio with the default configuration, rerun the installer and select the modify option.

See “Command-Line C Compiler Configuration”.

For more information, see “Location of Files on Development Computer” on page 2-8.

License Requirements

Before you install the Simulink Real-Time product, you must have a valid File Installation Key and License file. The File Installation Key identifies the products that you purchased from MathWorks®. The License file activates the installation.

If you have not received either of these items, go to the License Center at the MathWorks website.

Simulink Real-Time includes standalone mode. With standalone mode, you can create standalone real-time applications that run separately from the development computer.

Location of Files on Development Computer

Select a working folder outside the MATLAB root. The files that are located there include:

- MATLAB working folder — Simulink models (`model`), Simulink Real-Time applications (`model.mldatx`)
- Simulink Coder™ Build folder — The Simulink Coder C code files (`model.c`, `model.h`) are in a subfolder called `modelname_slrt_rtw`.

The Simulink Real-Time software uses files located under these folders:

- `matlabroot\toolbox\slrt\`
 - `blocks` — Drivers and associated blocks
 - `slrt` — Development computer functions related to the Simulink Real-Time software, methods for target objects, and methods for scope objects
 - `slrtexamples` — Simulink models and MATLAB code examples
- `matlabroot\toolbox\rtw\targets\xpc\`
 - `target` — Files and functions related to the Simulink Real-Time kernel and build process, including drivers to support I/O blocks
 - `xpc` — Development computer functions related to the Simulink Real-Time software, methods for target objects, and methods for scope objects
 - `xpcdemos` — Simulink models and MATLAB code examples

See Also

More About

- “Development Computer Requirements” on page 2-4
- “Development Computer Communication Setup” on page 2-6
- “Command-Line C Compiler Configuration”

PCI Bus Ethernet Setup

Speedgoat target machines include an onboard (PCI bus) Ethernet controller. The PCI bus and slot information for the Ethernet controllers are available in the Speedgoat documentation. To get a list of these Ethernet cards, after you connect and turn on the target computer, in the MATLAB Command Window, type:

```
tg = slrt;  
getPCIInfo(tg, 'ethernet')
```

To build and download a real-time application, first specify the environment properties for the development and target computers.

- 1 Open a Simulink Real-Time model by creating a model from the instructions on the Simulink start page. In the Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page, and then create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.

- 2 Ask your system administrator for the following information for your target computer:
 - IP address
 - Subnet mask address
 - Port number (optional)
 - Gateway (optional)
- 3 In the Simulink Editor, open the Target Computer Manager. In the **Real-Time** tab, from the targets list, click **Target Computer Manager**.
- 4 From the **Target Computers** list, select the target computer node.
 - To add a node representing another target computer, in the **Targets** pane, click the **Add target** button.
 - To remove a node representing a target computer, select the node and click the **Remove target** button.
- 5 From the **Advanced settings** list, set **IP address** to the IP address for your target computer (for example, Speedgoat target machines are preconfigured to address `192.168.7.1`).

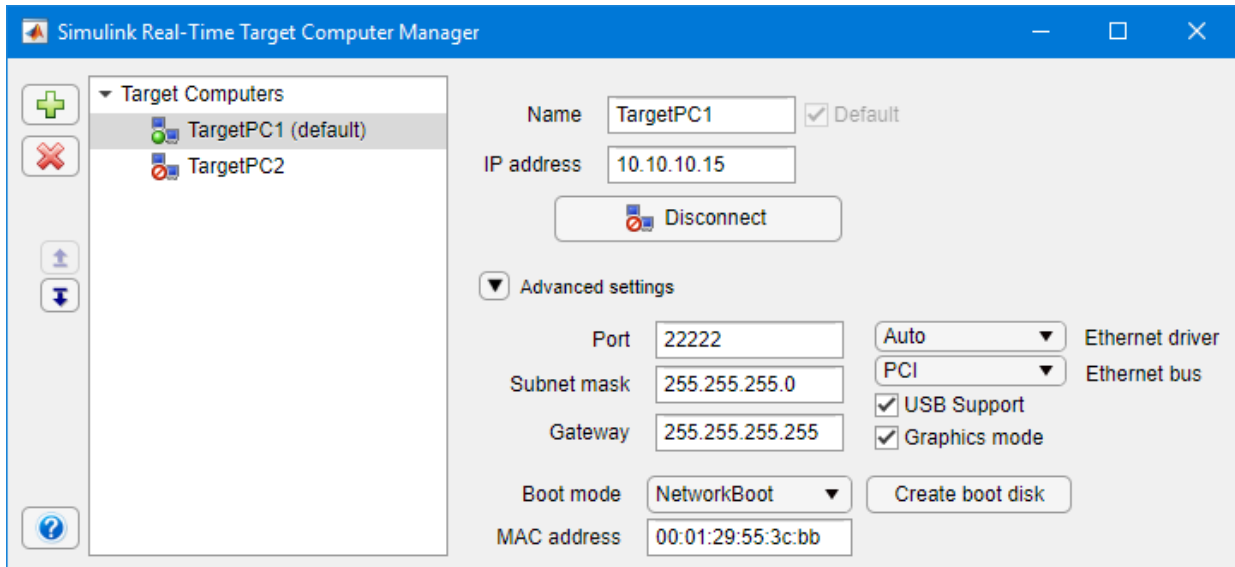
- 6 Set **Subnet mask** to the subnet mask address of your LAN (for example, 255 . 255 . 255 . 0).
- 7 Set **Port** (optional) to a value greater than 20000 and less than 65536. This property is set by default to 22222, a value higher than the reserved area (telnet, ftp, and so on).
- 8 If a gateway is required, set **Gateway** (optional) to the gateway to access the target computer. This property is set by default to 255 . 255 . 255 . 255, which means that you do not use a gateway to connect to your target computer. If you connect your computers with a crossover cable, leave this property as 255 . 255 . 255 . 255.

If you communicate with the target computer from within your LAN, do not change the default setting. If you communicate from a development computer within a LAN different from your target computer, define a gateway and enter its IP address here. In particular, create a gateway if you access the target computer via the Internet.

- 9 Select **Ethernet bus** PCI. All Speedgoat target machines use bus type PCI.
- 10 Select **Ethernet driver** Auto (default). This setting depends on type of Speedgoat target machine. The information is provided in the Speedgoat documentation and is available in the label attached to the target machine.

If the target computer has multiple Ethernet cards, see the Speedgoat documentation and "Ethernet Card Selection by Index".

The Target Computer Manager window looks like this figure.



Repeat this procedure as required for each target computer.

See Also

getPCIInfo

More About

- “Ethernet Card Selection by Index”
- “Command-Line Ethernet Card Selection by Index”

Target Computer Settings

To run a Simulink Real-Time model on a target computer, you must configure the target settings to match the capabilities of the target computer.

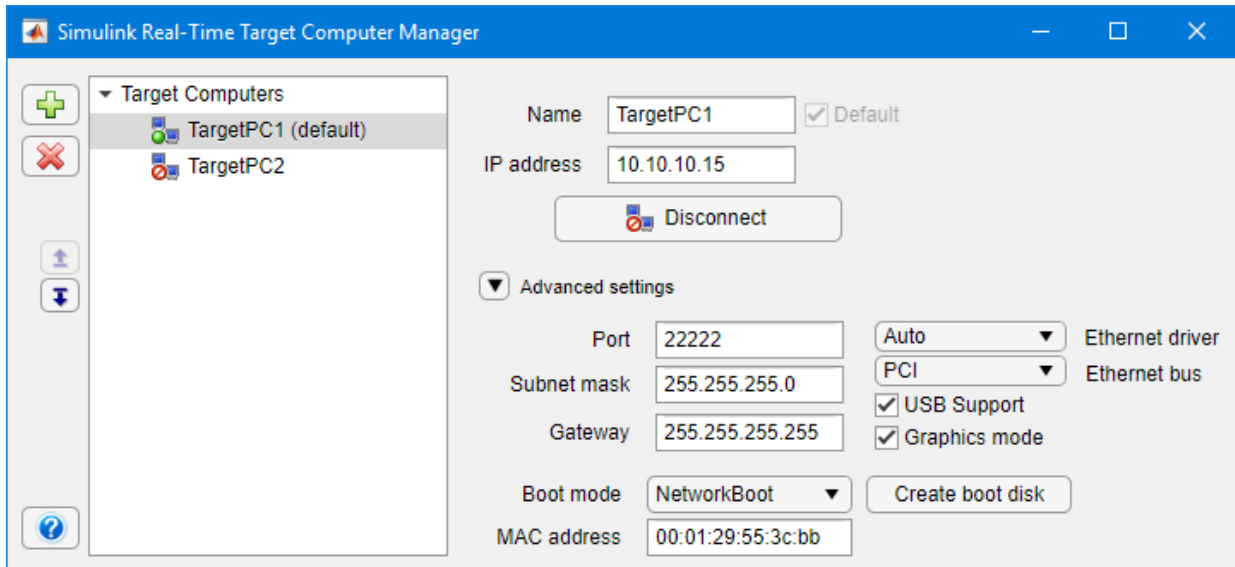
- 1 Open a Simulink Real-Time model by creating a model from the Simulink start page. In the MATLAB Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page and create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.

- 2 In the Simulink Editor, open the Target Computer Manager. In the **Real-Time** tab, from the targets list, click **Target Computer Manager**.
- 3 From the **Target Computers** list, select the target computer node.
 - To add a node representing another target computer, in the **Targets** pane, click the **Add target** button.
 - To remove a node representing a target computer, select the node and click the **Remove target** button.
- 4 From the **Advanced settings** list, execute the following target computer settings as required:
 - **USB Support** — If you want to use a USB port on the target computer, for example, to connect a USB keyboard, leave this check box selected. Otherwise, clear it.
 - **Graphics mode** — If you want to display information, such as a target scope, in graphic format, leave this check box selected. If you want to display information as text, clear it.

The Target Computer Manager window looks like this figure.



Repeat this procedure as required for each target computer.

See Also

More About

- “PCI Bus Ethernet Setup” on page 2-9
- “Target Computer Boot Methods” on page 2-14

Target Computer Boot Methods

You can start your target computer with the Simulink Real-Time kernel by using one of several methods.

Speedgoat systems come with **DOS Loader** software installed. You can set up the **DOS Loader** boot method on your development computer or configure another boot method. For more information, see your Speedgoat system documentation or follow the link from “Speedgoat Target Computers and Support”.

- 1 Select one of these methods:
 - “Network Boot Method” on page 2-15
 - “DOS Loader Boot Method” on page 2-19
 - “Standalone Boot Method” on page 2-20
- 2 For boot methods other than **Stand Alone**, perform “Run Confidence Test on Configuration” on page 2-26.

For boot method **Stand Alone**, create a model-specific confidence test, restart the target computer, and run that confidence test. The default confidence test is not intended for standalone execution.

See Also

More About

- “PCI Bus Ethernet Setup” on page 2-9
- “Target Computer Settings” on page 2-12

Network Boot Method

After you have configured the target computer environment parameters, you can use a dedicated Ethernet network to load and run the Simulink Real-Time kernel.

Network boot method limitations:

- Do not use the network boot method on a corporate or nondedicated network. Doing so can interfere with dynamic host configuration protocol (DHCP) servers and cause network issues.
- Your Speedgoat target machine must be compatible with the Preboot eXecution Environment (PXE) specification. See the Speedgoat documentation.
- If **Boot mode Stand Alone** is enabled, you cannot start the target computer across the network.

Before you start working with Simulink Real-Time, establish an Ethernet connection between the development and target computers by using the procedures in “PCI Bus Ethernet Setup” on page 2-9.

- 1 Open a Simulink Real-Time model by creating a model from the Simulink start page. In the MATLAB Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page and create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.

- 2 In the Simulink Editor, open the Target Computer Manager. In the **Real-Time** tab, from the targets list, click **Target Computer Manager**.
- 3 In the **Target Computers** list, select the target computer node.
 - To add a node representing another target computer, in the **Targets** pane, click the **Add target** button.
 - To remove a node representing a target computer, select the node and click the **Remove target** button.
- 4 To clear the MAC address, from the **Advanced settings** list, delete the value in the **MAC address** box.


If you clear the MAC address, the next time that the target computer starts, by default, the software obtains the MAC addresses of accessible target computers. The

software displays them for confirmation in the Simulink Real-Time Network Boot dialog box.

- 5 To enter the MAC address of the target computer manually, in the **MAC address** box, enter the address in the format `xx:xx:xx:xx:xx:xx`.

The next time that the target computer starts, the software selects and starts the target computer that matches this MAC address. The Simulink Real-Time Network Boot dialog box does not open.

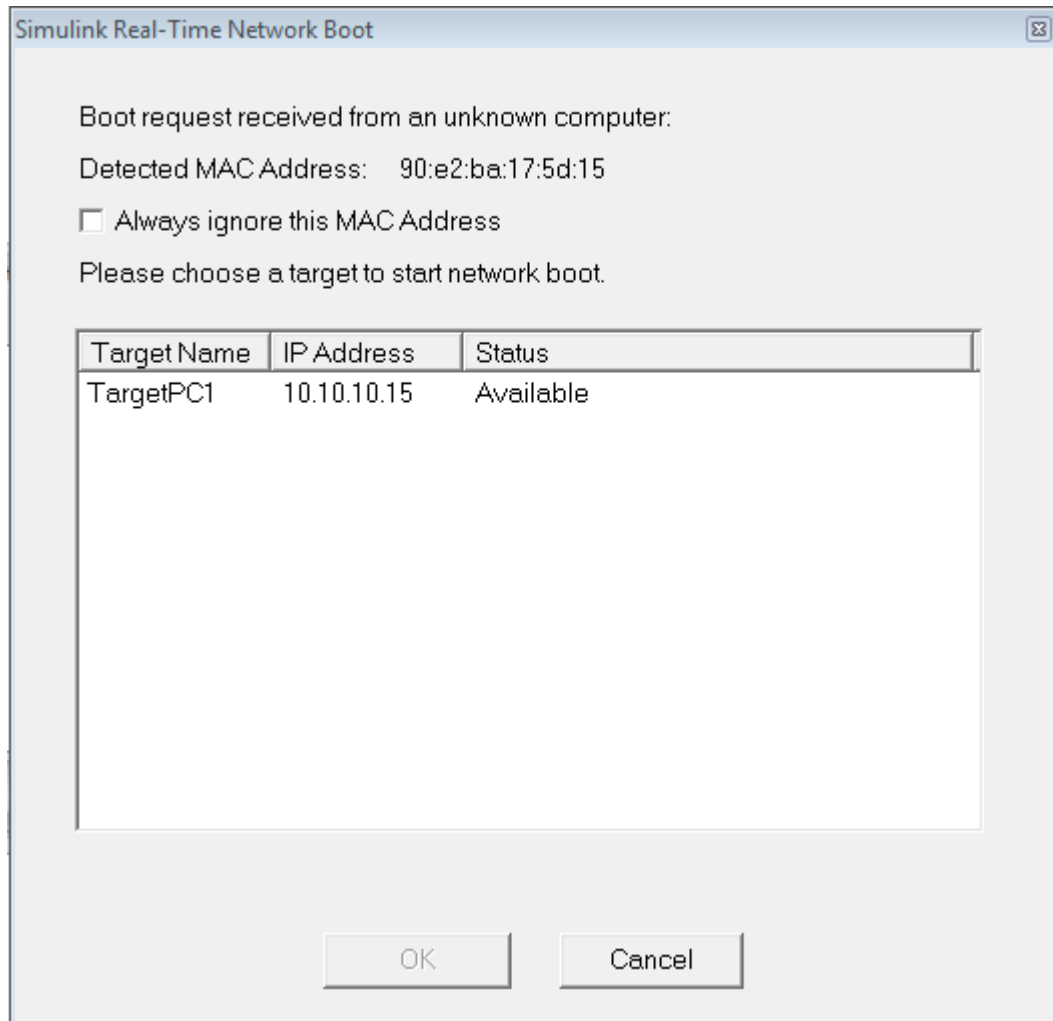
- 6 Click **Create boot disk**.

The software creates and starts a network boot server process on the development computer. You see a minimized icon () in the bottom right system tray on the development computer.

- 7 Turn on the target computer.

The first time the network boot server process detects a target computer, it displays the Simulink Real-Time Network Boot dialog box. This dialog box contains physical target computer names and the corresponding IP addresses.

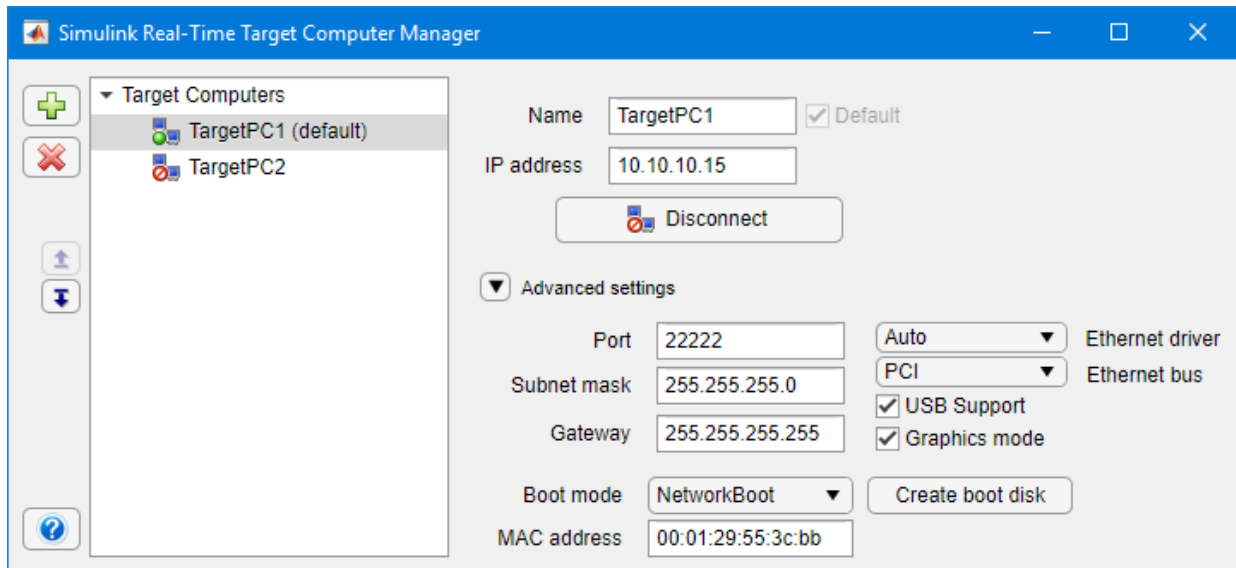
- 8 Select the target computer name for the physical target computer.
- 9 Click **OK**.



The development computer network boot server displays a message indicating that the boot server is being downloaded to the target computer.

If you click **Cancel**, the kernel waits for 90 seconds before responding the next time that you try to start the target computer across the network.

The Target Computer Manager window looks like this figure.



Repeat this procedure as required for each target computer.

Tip To configure the network boot by using MATLAB language, see “Command-Line Network Boot Method”.

After you complete network boot for a new target computer, test the configuration. For more information, see “Run Confidence Test on Configuration” on page 2-26.

See Also

More About

- “DOS Loader Boot Method” on page 2-19
- “Standalone Boot Method” on page 2-20

DOS Loader Boot Method

In DOS Loader mode, you start the Simulink Real-Time kernel on a Speedgoat target machine from the installed solid state drive or hybrid hard drive with DOS boot capability. After starting the target computer, you can download your real-time application from the development computer over a connection between the development and target computers.

Speedgoat systems come with DOS Loader software installed. You can set up the DOS Loader boot method on your development computer or configure another boot method. For more information, see your Speedgoat system documentation or follow the link from “Speedgoat Target Computers and Support”.

See Also

More About

- “Network Boot Method” on page 2-15
- “Standalone Boot Method” on page 2-20

Standalone Boot Method

You can configure the Simulink Real-Time software to run as a standalone real-time application on your Speedgoat target machine. For more information on **Boot mode Stand Alone**, see “Standalone Mode” on page 2-24.

Standalone Settings

Speedgoat target machines come with DOS Loader software installed. You can set up the DOS Loader boot method on your development computer or configure another boot method. For more information about booting and kernel transfer, see your Speedgoat target machine documentation or follow the link from “Speedgoat Target Computers and Support”.

To set the kernel environment properties, use Simulink Real-Time Explorer . When you are done, you can create a standalone kernel and real-time application.

For **Boot mode Stand Alone**, you do not create a Simulink Real-Time boot disk or network boot image. Instead, you copy files created from the build process to the target computer hard drive.

- 1 Open a Simulink Real-Time model by creating a model from the Simulink start page. In the MATLAB Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page and create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.

- 2 In the Simulink Editor, open the Target Computer Manager. In the **Real-Time** tab, from the targets list, click **Target Computer Manager**.
- 3 From the **Target Computers** list, select the target computer node.
 - To add a node representing another target computer, in the **Targets** pane, click the **Add target** button.
 - To remove a node representing a target computer, select the node and click the **Remove target** button.
- 4 From the **Advanced settings** list, set **Boot mode** to **Stand Alone**. Check the host-target communications settings, which correspond to the Speedgoat target machine. See “PCI Bus Ethernet Setup” on page 2-9.

Repeat this procedure as required for each target computer.

Real-Time Application Build

After you set the Simulink Real-Time boot mode to Stand Alone, you can use Simulink Real-Time, Simulink Coder, and a C/C++ compiler in Stand Alone mode to build a standalone kernel and real-time application with utility files.

- 1 In the Command Window, open your Simulink model, for example, `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`).
- 2 Build the model. In the Simulink Editor, on the **Real-Time** tab, click **Run on Target > Build Application**.

The Simulink Coder and Simulink Real-Time software create a folder, `ex_slrt_rt_osc_slrt_emb`, containing the files that boot the target computer and run the real-time application.

Repeat this procedure as required for each real-time application.

Real-Time Application Transfer and Boot Configuration

After building the kernel and real-time application on a development computer, transfer the files to a target computer by using the `SimulinkRealTime.fileSystem` object functions. Configure the target computer to run the real-time application upon startup.

For this procedure, your target computer must support network boot mode. If it does not support network boot mode, see “Application Transfer and Boot Configuration with USB Flash Drive” on page 2-22.

- 1 Restart the target computer in DOS mode and open the DOS prompt.

If the target computer was previously started from the network boot image, to disable the network boot capability, kill the boot server from Windows Task Manager.
- 2 Start the target computer by using network boot mode.
- 3 In MATLAB, change to the folder that contains the kernel and real-time application files.
- 4 Copy these files to the root folder of the target computer C:\ drive:

```
tg = slrt;  
SimulinkRealTime.copyFileToTarget(tg, 'xpckrnl.rtb')  
SimulinkRealTime.copyFileToTarget(tg, 'ex_slrt_rt_osc.mldatx')
```

- 5 Restart the target computer.

To boot the kernel and start the real-time application, the target computer executes the following sequence of calls:

```
C:\autoexec.bat  
C:\rttboot.com  
C:\xpckrnl.rtb  
C:\<application>.mldatx
```

Repeat this procedure for each target computer that you start in Stand Alone mode.

Continue by testing a real-time application in Stand Alone mode.

Application Transfer and Boot Configuration with USB Flash Drive

Speedgoat target machines come with DOS Loader software installed. You can set up the DOS Loader boot method on your development computer or configure another boot method. For more information about booting and kernel transfer, see your Speedgoat target machine documentation or follow the link from “Speedgoat Target Computers and Support”.

If the target computer does not support network boot mode, use a USB flash drive as a boot device.

- 1 Restart the target computer in DOS mode and open the DOS prompt.
- 2 At the DOS prompt, save a copy of the target computer C:\autoexec.bat file to a backup file, such as C:\autoexec_back.wrk.
- 3 Connect the USB flash drive to the development computer.
- 4 Copy these files to the root folder of the USB flash drive:

```
autoexec.bat  
rttboot.com  
xpckrnl.rtb  
ex_slrt_rt_osc.mldatx
```

- 5 Remove the flash drive from the development computer and connect it to the target computer.

- 6 Restart the target computer.

To boot the kernel and start the real-time application, the target computer executes the following sequence of calls:

```
C:\autoexec.bat  
C:\rttboot.com  
C:\xpckrnl.rtb  
C:\<application>.mldatx
```

Repeat this procedure for each target computer that you start in Stand Alone mode.

Continue by testing a real-time application in Stand Alone mode.

See Also

More About

- “Network Boot Method” on page 2-15
- “DOS Loader Boot Method” on page 2-19
- “Standalone Mode” on page 2-24

Standalone Mode

In Stand Alone mode, you can create control, DSP, and other system models for use in production systems running on target computers.

In Stand Alone mode, you can start the kernel and run the real-time application on the target computer, independent of the development computer. You can configure a target computer to start executing your real-time application automatically for continuous operation each time the system is started. If the target computer has a keyboard, you can control the real-time application by using the target computer command-line interface. You can create custom interfaces with the Simulink Real-Time C and .NET APIs and run them on Microsoft Windows development computers, without installing MATLAB software.

Note Stand Alone mode does not require a connection between the development and target computers.

To load the target computer with the Simulink Real-Time kernel and a real-time application, use Stand Alone mode. This mode of operation can start the kernel on the target computer from a flash disk or hard disk. After starting the kernel on the target computer, Stand Alone mode also starts the real-time application that you loaded with the kernel. Stand Alone mode eliminates the need for a development computer and enables you to run real-time applications on target computers.

Regardless of the mode, you initially start your target computer by using DOS from a boot device. DOS starts the Simulink Real-Time kernel. After the Simulink Real-Time kernel starts, DOS is not available on the target computer. To make DOS available, you must restart the target computer by using DOS without starting the Simulink Real-Time kernel.

Simulink Real-Time standalone mode requires a boot device with DOS installed. Speedgoat target machines include this boot device..

Standalone mode is an option if the target computer does not support network boot from the development computer.

If you want to view signals on the target computer in Stand Alone mode, you must provide a monitor for the target computer. Before building the real-time application, you must also add real-time Scope blocks of types `target` and `file`.

If you do not want to view signals on the target computer, you can run your Simulink Real-Time system as a black box without a monitor or keyboard. Standalone real-time applications are automatically set to continue running for an infinite amount of time or until the target computer is turned off.

See Also

More About

- “Standalone Boot Method” on page 2-20

Run Confidence Test on Configuration

Test the target boot process, the connection between the development and target computers, and the basic functionality of the Simulink Real-Time software. Use this test under the following circumstances:

- To validate your initial product installation.
- As the first step in a troubleshooting procedure.

You must be using network boot mode for starting the target computer.

- 1 Create a network boot image and restart the target computer. See “Network Boot Method” on page 2-15.
- 2 In the current folder window, select a folder outside the MATLAB root folder.
- 3 In the MATLAB Command Window, type `slrttest`.

MATLAB executes the test script for the default target computer and displays messages indicating whether the test passed or failed.

```
### Simulink Real-Time Test Suite
### Host-Target interface is: TcpIp
### Test 1, Ping target computer 'TargetPC1' using system ping: ... OK
### Test 2, Ping target computer 'TargetPC1' using SLRTPINGTARGET: ... OK
### Test 3, Software reboot the target computer 'TargetPC1': ..... OK
### Test 4, Build and download a Simulink Real-Time application using model ...
slrttestmdl to target computer 'TargetPC1': ... OK
### Test 5, Check host-target command communications with 'TargetPC1': ... OK
### Test 6, Download a pre-built Simulink Real-Time application to target ...
computer 'TargetPC1': ... OK
### Test 7, Execute the Simulink Real-Time application for 0.2s: ... OK
### Test 8, Upload logged data and compare with simulation results: ... OK
### Test Suite successfully finished
```

- 4 Evaluate the results.
 - If the tests return OK, you are ready to build and download a real-time application to the target computer.
 - If one or more tests return FAILED, see “Troubleshooting in Simulink Real-Time”.

The next task is “Application and Driver Scripts” on page 4-40.

See Also

More About

- “Troubleshoot with Confidence Test”

Basic Workflows

Real-Time Simulation and Testing

Real-time simulation and testing is used in two ways, differing only in whether the design or the prototype is being modeled in Simulink software.

- **Rapid control prototyping (RCP)** — Model a design by using Simulink Real-Time software and connect it to a physical system by using a target computer. You can work out the design flaws before investing in a physical prototype and uncover requirements for new applications.
- **Hardware-in-the-loop (HIL) simulation** — Connect a Simulink Real-Time model that is a physical prototype of the design to a software plant model that represents the hardware system. You can test the prototype for safety and performance without expensive downtime for the rest of the system. You can test operation and failure conditions that are difficult to replicate and substitute for unavailable parts of the hardware system.

Real-Time Simulation and Testing Workflow

- “Create a Simulink or Stateflow model.” on page 3-3
- “Simulate the model as a non-real-time application.” on page 3-3
- “Configure the development and target computers.” on page 3-3
- “Prepare the model for real-time execution. Configure the build environment.” on page 3-3
- “Connect to the physical hardware. Restart the target computer. Configure the build environment.” on page 3-4
- “Connect to the target computer. Build and download the real-time application.” on page 3-4
- “Execute the real-time application.” on page 3-4
- “Visualize signals.” on page 3-5
- “Tune parameters.” on page 3-5
- “Prepare regression and stress tests.” on page 3-6

A workflow process for real-time simulation and testing leads from model creation to regression and stress testing.

Create a Simulink or Stateflow model.

- 1 In the Simulink Editor, on the **Simulation** tab, from **New**, select **Model**.
- 2 From the Simulink start page, select a model template or click **Blank Model**.
- 3 Create block diagrams in Simulink by dragging blocks to your model. If possible, set model and block parameters to use a fixed-step solver and specify a sample time compatible with the real-time requirements of your model.

Simulate the model as a non-real-time application.

Simulink uses a computed time vector to step the model. After computing the outputs for a given time value, Simulink immediately repeats the computation for the next time value until it reaches the stop time.

Because the computed time vector is not connected to a physical clock, the outputs are calculated as fast as your computer can run. The elapsed time of the simulation can differ significantly from the elapsed time of the real system.

You can log simulation results for later comparison.

Configure the development and target computers.

- 1 Configure the communication method between the development and target computers.
- 2 Configure the development and target computers by using:
 - Simulink Real-Time Target Computer Manager
 - “MATLAB Command-Line Interface” on page 1-6

Prepare the model for real-time execution. Configure the build environment.

- 1 Set the model Configuration Parameters to values compatible with real-time execution:
 - On the **Apps** tab, click **Simulink Real-Time**. This selection sets the **System target file** model parameter to `slrt.tlc` and sets the **Type** model parameter to **Fixed-step**.
 - On the **Real-Time** tab, click **Hardware Settings**. In **Configuration Parameters** > **Solver** > **Fixed-step size** select a step size that is compatible with the real-time requirements of your model.

- On the **Real-Time** tab, click **Hardware Settings**. Select options for **Configuration Parameters > Code Generation > Simulink Real-Time Options**.
- 2 Add Simulink Real-Time I/O blocks representing your I/O boards to the model.
 - 3 To visualize the simulation results, mark signals in the model for data logging with the Simulation Data Inspector. See “View Data with the Simulation Data Inspector” (Simulink).

Connect to the physical hardware. Restart the target computer. Configure the build environment.

- 1 Connect the development computer to the Speedgoat target machine. Connect the IO Modules in the target computer to the physical hardware where applicable.
- 2 Restart the target computer with the Simulink Real-Time real-time kernel by using:
 - Target computer restart button.
 - “MATLAB Command-Line Interface” on page 1-6

Connect to the target computer. Build and download the real-time application.

- 1 On the **Real-Time** tab, toggle the **Toggle target computer connection** to **Connected** and click **Run on Target**. For more information about building and downloading the real-time application, see:
 - Simulink Real-Time Target Computer Manager
 - “MATLAB Command-Line Interface” on page 1-6
 - “Simulink External Mode Interface” on page 1-8
 - “Target Computer Command-Line Interface” on page 1-10

Execute the real-time application.

- 1 Execute the real-time application under command from the development computer or by restarting the target computer in standalone mode.

The Simulink Real-Time software uses real-time resources on the target computer. Based on your sample rate, the Simulink Real-Time software uses interrupts to step the model at the sample rate. With each new interrupt, the real-time application computes the block outputs from your model.

- 2 Execute using:

- **Simulink Real-Time Explorer**
- “MATLAB Command-Line Interface” on page 1-6
- “Simulink External Mode Interface” on page 1-8
- “Target Computer Command-Line Interface” on page 1-10

Visualize signals.

- 1** Create real-time scopes and Simulink Real-Time Explorer instruments. Use them to acquire and display signal data from the real-time application. You can filter and group hierarchical signals in Explorer.
- 2** Scopes created by real-time scope blocks acquire data according to Simulink sample time rules. Scopes can gather data at the top level or in an enabled or triggered subsystem. Scopes created dynamically (from the MATLAB Command Window or the API) sample at the base rate, irrespective of the sample time of their signals.
- 3** To create instrument panels, use Simulink Real-Time Explorer to create instrument panels. You can drag graphical instruments to the instrument panels and drag signals to the instruments to display signal data.

Visualize signals using:

- **Simulink Real-Time Explorer**
- “MATLAB Command-Line Interface” on page 1-6
- “Simulink External Mode Interface” on page 1-8
- “Add Simulink Real-Time Blocks to Simulink Model” on page 1-9
- “Target Computer Command-Line Interface” on page 1-10

Tune parameters.

- 1** Tune observable parameters such as time delays, input and output amplitudes, and input and output frequencies. You can filter and group hierarchical signals in Simulink Real-Time Explorer.

Note Simulink Real-Time does not support parameters of multiword data types.

- 2** Tune parameters using:
 - **Simulink Real-Time Explorer**
 - “MATLAB Command-Line Interface” on page 1-6

- “Simulink External Mode Interface” on page 1-8
- “Target Computer Command-Line Interface” on page 1-10

Prepare regression and stress tests.

Write MATLAB scripts that perform parameter sweep and extreme-value testing in a repeatable manner; accumulating results as known good data.

Special Purpose Tasks

In addition to the typical real-time simulation and testing workflow process steps, some special purpose tasks could support your development process.

- **Visualize the model with Simulink Real-Time Explorer instruments.**

Configure Simulink Real-Time Explorer by using instrument panels configured with real-time scopes and with signal display and parameter tuning graphical instruments.

- **Run the model where the physical device runs.**

Configure the real-time application to run outside MATLAB, for example in a moving vehicle. Export the Simulink Real-Time Explorer configuration as a standalone control program.

- **Prepare regression and product tests with Simulink Test™.**

Extend the regression and stress tests to cover functionality over the full parameter range.

- **Integrate the model into a third-party test environment.**

Use MATLAB Coder to translate MATLAB regression test scripts into C for integration into a third-party test environment.

See Also

More About

- “Target Computer Configuration and Control Methods” on page 1-3

Tutorial and Examples

This tutorial describes Simulink Real-Time functionality with a simple Simulink model without I/O blocks. You can try these examples without I/O modules on the target computer. Once you are familiar with the setup, build, and target execution process, you can try some of the more advanced Simulink Real-Time examples.

- “Set Up and Configure Simulink Real-Time” on page 4-2
- “Create and Run Real-Time Application from Simulink Model” on page 4-9
- “Apply Simulink Real-Time Model Template to Create Real-Time Application” on page 4-23
- “Configure and Control a Real-Time Application” on page 4-26
- “Process a Real-Time Application by Using Simulink Real-Time Explorer” on page 4-31
- “Simulate Simulink Model by Using MATLAB Language” on page 4-33
- “Prepare Real-Time Application by Using MATLAB Language” on page 4-36
- “Execute Real-Time Application with MATLAB Language” on page 4-37
- “Application and Driver Scripts” on page 4-40

Set Up and Configure Simulink Real-Time

For this tutorial, you must have this configuration:

- Single PCI bus target computer
- Network target computer boot method

You must have installed and configured a C compiler as part of your Simulink Real-Time installation. If not, see “Command-Line C Compiler Configuration”.

Configure Link Between Development and Target Computers

To run a Simulink Real-Time model on a target computer, you must connect the development and target computers in a network.

The target network settings values are representative only. Consult your network administrator for actual values. For more on network configuration, see “PCI Bus Ethernet Setup” on page 2-9.

- 1 Open a Simulink Real-Time model by creating a model from the Simulink start page. In the MATLAB Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page and create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.

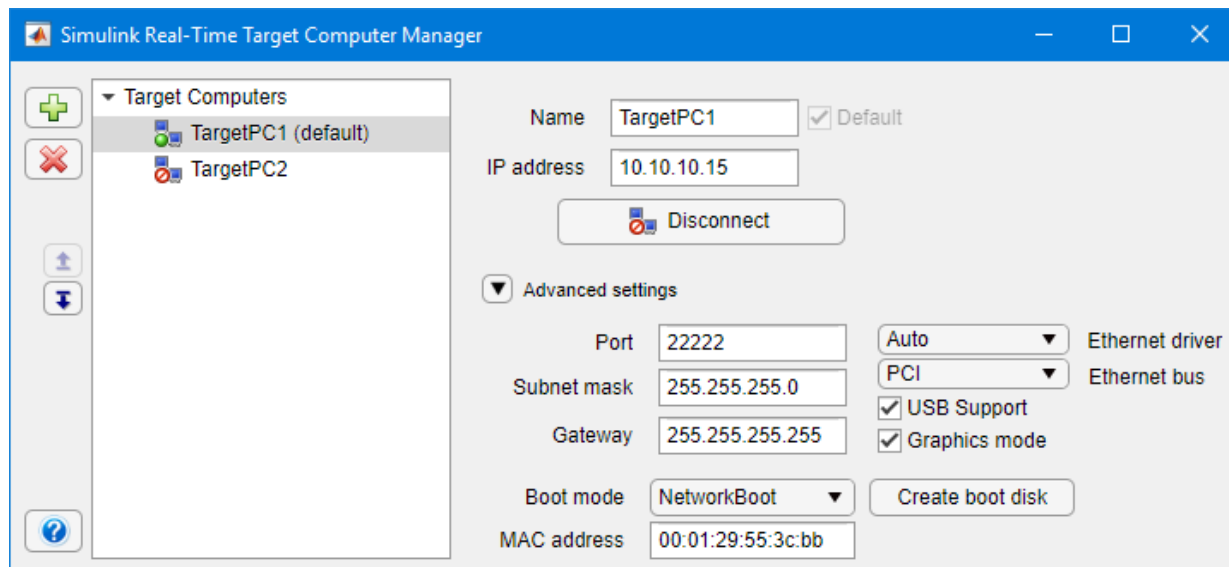
- 2 In the Simulink Editor, open the Target Computer Manager. In the **Real-Time** tab, from the targets list, click **Target Computer Manager**.
- 3 From the **Target Computers** list, select the target computer node.
 - To add a node representing another target computer, in the **Targets** pane, click the **Add target** button.
 - To remove a node representing a target computer, select the node and click the **Remove target** button.
- 4 From the **Advanced settings** list, set **IP address** to the IP address for your target computer (for example, Speedgoat target machines are preconfigured to address `192.168.7.1`).

- 5 Set **Subnet mask** to the subnet mask address of your LAN (for example, 255.255.255.0).
- 6 Set **Port** (optional) to a value greater than 20000 and less than 65536. This property is set by default to 22222, a value higher than the reserved area (telnet, ftp, and so on).
- 7 If a gateway is required, set **Gateway** (optional) to the gateway required to access the target computer. This property is set by default to 255.255.255.255, which means that you do not use a gateway to connect to your target computer. If you connect your computers with a crossover cable, leave this property as 255.255.255.255.

If you communicate with the target computer from within your LAN, do not change the default setting. If you communicate from a development computer within a LAN different from your target computer, define a gateway and enter its IP address here. In particular, create a gateway if you access the target computer via the Internet.

- 8 Select **Ethernet bus** PCI. All Speedgoat target machines use bus type PCI.
- 9 Select **Ethernet driver** Auto (default). This setting depends on type of Speedgoat target machine. The information is provided in the Speedgoat documentation and is available in the label attached to the target machine.

The Target Computer Manager window looks like this figure.



Configure Target Settings

To run a Simulink Real-Time model on a target computer, you must configure the target settings.

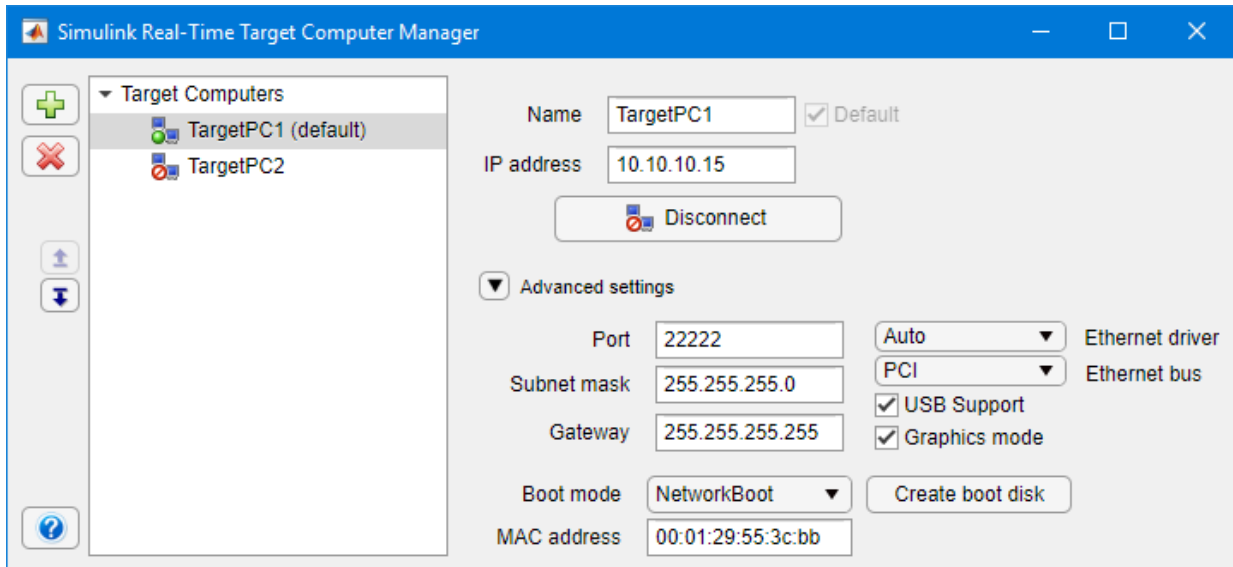
- 1 Open a Simulink Real-Time model by creating a model from the Simulink start page. In the MATLAB Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page and create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.

- 2 In the Simulink Editor, open the Target Computer Manager. In the **Real-Time** tab, from the targets list, click **Target Computer Manager**.
- 3 In the **Target Computers** list, select the target computer node.
 - To add a node representing another target computer, in the **Targets** pane, click the **Add target** button.
 - To remove a node representing a target computer, select the node and click the **Remove target** button.
- 4 From the **Advanced settings** list, execute the following target computer settings as required:
 - **USB Support** — If you want to use a USB port on the target computer, for example, to connect a USB keyboard, leave this check box selected. Otherwise, clear it.
 - **Graphics mode** — If you want to display information, such as a target scope, in graphic format, leave this check box selected. If you want to display information as text, clear it.

The Target Computer Manager window looks like this figure.



Configure Boot Configuration

To run a Simulink Real-Time model on a target computer, you must configure the target computer boot method. For more on boot methods, see “Target Computer Boot Methods” on page 2-14.

- 1 Open a Simulink Real-Time model by creating a model from the Simulink start page. In the MATLAB Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page and create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.

- 2 In the Simulink Editor, open the Target Computer Manager. In the **Real-Time** tab, from the targets list, click **Target Computer Manager**.
- 3 In the **Target Computers** list, select the target computer node.
 - To add a node representing another target computer, in the **Targets** pane, click the **Add target** button.

- To remove a node representing a target computer, select the node and click the **Remove target** button.

4 Expand the list.


5 To clear the MAC address, from the **Advanced settings** list, delete the value in the **MAC address** box.

If you clear the MAC address, the next time the target computer starts, by default, the software automatically obtains the MAC addresses of accessible target computers. The software displays them for confirmation in the Simulink Real-Time Network Boot dialog box.

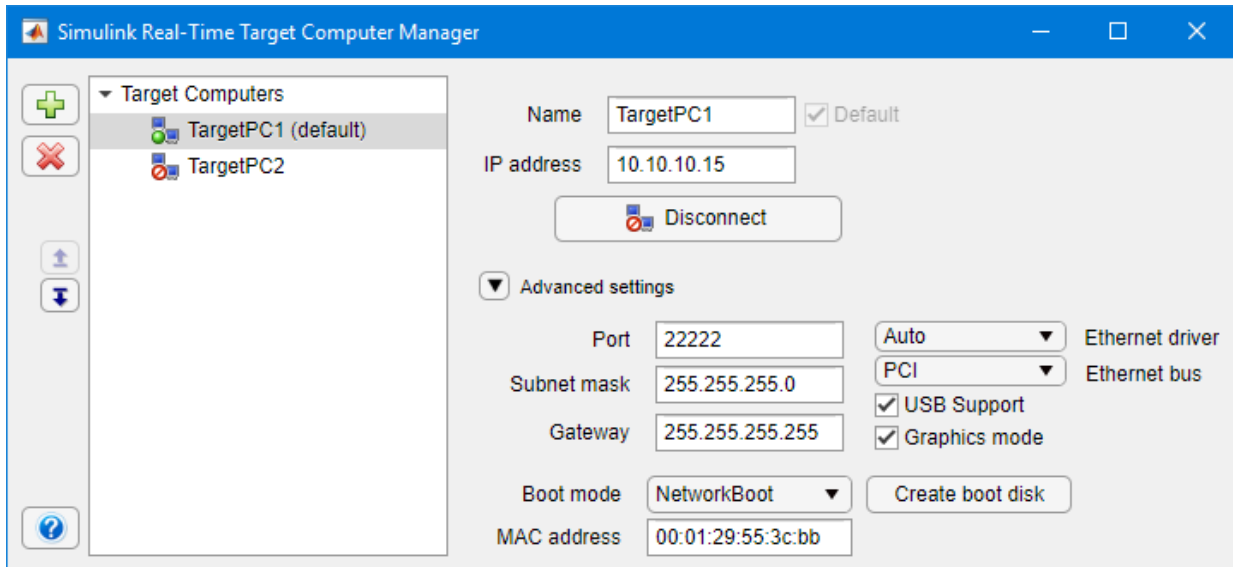
6 To enter the MAC address of the target computer manually, in the **MAC address** box, enter the address in the format `xx:xx:xx:xx:xx:xx`.

The next time the target computer starts, the software selects and starts the target computer that matches this MAC address. The Simulink Real-Time Network Boot dialog box does not open.

7 Click **Create boot disk**.

The software creates and starts a network boot server process on the development computer. You see a minimized icon () in the bottom right system tray on the development computer.

The Target Computer Manager window looks like this figure.



Run the Confidence Test

Validate the setup and configuration by running the confidence test.

- 1 Create a network boot image and restart the target computer. See “Network Boot Method” on page 2-15.
- 2 In the current folder window, select a folder outside the MATLAB root folder.
- 3 In the MATLAB Command Window, type `slrttest`.

MATLAB executes the test script for the default target computer and displays messages indicating whether the test passed or failed.

```
### Simulink Real-Time Test Suite
### Host-Target interface is: TcpIp
### Test 1, Ping target computer 'TargetPC1' using system ping: ... OK
### Test 2, Ping target computer 'TargetPC1' using SLRTPINGTARGET: ... OK
### Test 3, Software reboot the target computer 'TargetPC1': ..... OK
### Test 4, Build and download a Simulink Real-Time application using model ...
slrttestmdl to target computer 'TargetPC1': ... OK
### Test 5, Check host-target command communications with 'TargetPC1': ... OK
### Test 6, Download a pre-built Simulink Real-Time application to target ...
computer 'TargetPC1': ... OK
### Test 7, Execute the Simulink Real-Time application for 0.2s: ... OK
### Test 8, Upload logged data and compare with simulation results: ... OK
### Test Suite successfully finished
```

4 Evaluate the results.

- If the tests return OK, you are ready to build and download a real-time application to the target computer.
- If one or more tests return FAILED, see “Troubleshooting in Simulink Real-Time”.

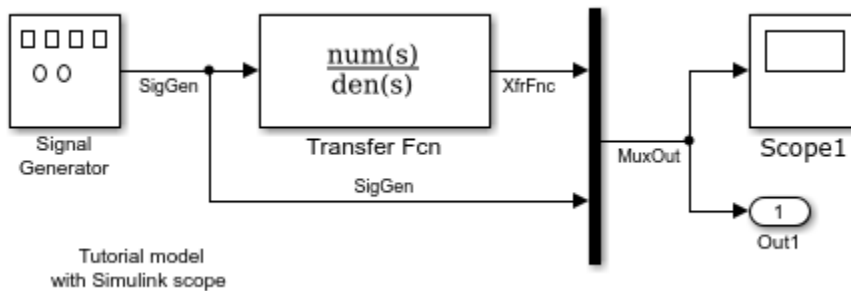
See Also

More About

- “Create and Run Real-Time Application from Simulink Model” on page 4-9

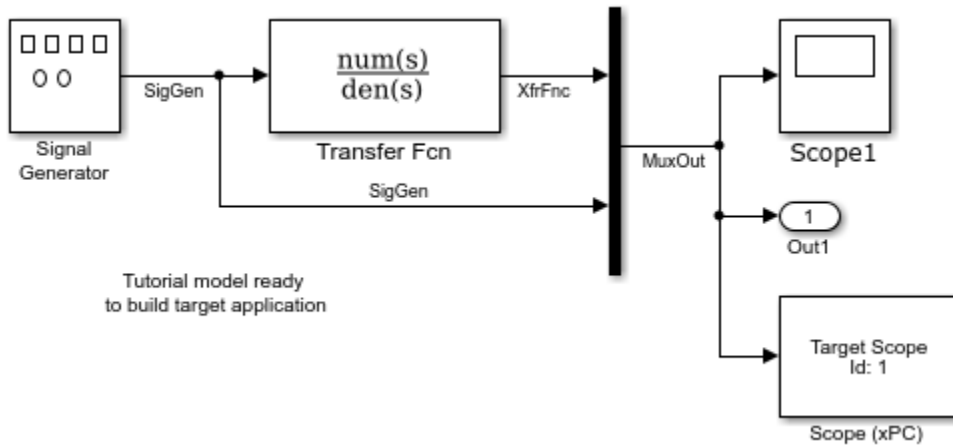
Create and Run Real-Time Application from Simulink Model

This tutorial begins with a non-real-time Simulink model of a damped oscillator, `ex_slrt_nrt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_nrt_osc')))`). To complete the tutorial, you must already know how to create, configure, and simulate such a model.



As you go through the tutorial, you transform `ex_slrt_nrt_osc` into a Simulink Real-Time model configured to build as a real-time application. You then build, download, and execute the real-time application on the target computer.

The final model is `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`):



Transform Simulink Model to Real-Time Application

To run a Simulink model as a real-time application under Simulink Real-Time, add and configure a real-time Scope block and set configuration parameters for code generation and target execution.

Add Simulink Real-Time Scope Block

Simulink Real-Time supports a real-time Scope block. There are three types of Simulink Real-Time scopes: target, host, and file. This tutorial uses the target scope, which displays execution data on the target computer monitor.

- 1 In the MATLAB Command Window, type `ex_slrt_nrt_osc`.

MATLAB loads the oscillator model and displays the Simulink block diagram.

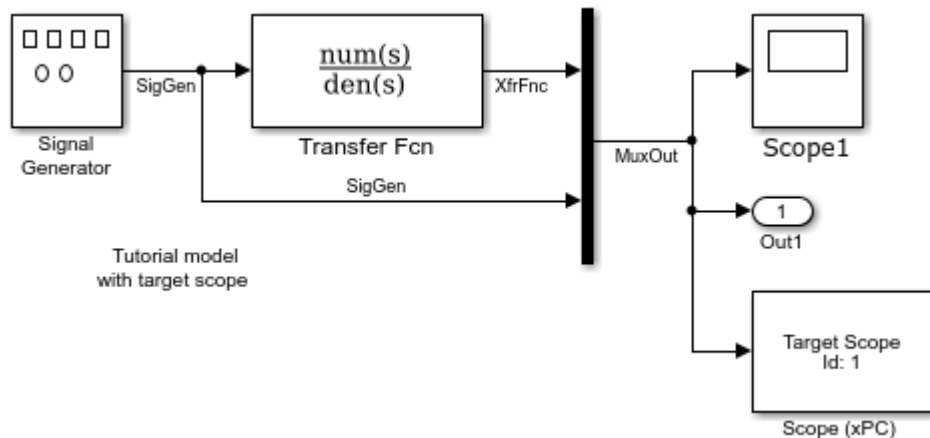
- 2 In Simulink Editor, from the **Real-Time** tab **Prepare** section, click **Library Browser**.
- 3 In the Simulink, in the left pane, browse to and double-click the node **Simulink Real-Time**.

A list of I/O block categories opens.

- 4 From the list of I/O block categories, click node **Displays and Logging**.
- 5 Click and drag block Scope to the Simulink block diagram.

- Simulink adds a new real-time Scope block to the model with a scope identifier of 1.
- 6 Connect the Mux output to the Scope block input.
 - 7 On the **Simulation** tab, from the **Save** button, click **Save As**. Enter a file name. For example, enter `ex_slrt_ucf_osc`, and then click **OK**.

The unconfigured model is `ex_slrt_ucf_osc`
 (open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_ucf_osc')))).



Set Target Scope Block Parameters

Scope block parameters define the signals to trace on the scope and trigger modes. The Simulink Real-Time Scope block dialog box changes depending on which scope type you are configuring: `target`, `host`, or `file`. For this tutorial, configure a target scope.

For more information about real-time target Scope parameters, see “Configure Real-Time Target Scope Blocks”.

- 1 In the MATLAB Command Window, type `ex_slrt_ucf_osc`.
 MATLAB loads the oscillator model and displays the Simulink block diagram.
- 2 Double-click block **Scope**.
- 3 Select **Scope type** `Target`. This value means that the scope display appears on the target computer monitor.

- 4 In the block parameters dialog box, select the **Start scope when application starts** check box.

This setting is mandatory in Stand Alone mode because the development computer is not available to issue a command to start scopes.

- 5 Select **Scope mode** Graphical redraw.
- 6 Select the **Grid** check box.
- 7 In the **Y-axis limits** text box, type $[0, 0]$. This value means that the display scaling is auto.
- 8 In the **Number of samples** text box, type 1000. For a **Scope mode** of Graphical redraw, this value means that 1000 samples are acquired before the graph is redrawn.
- 9 In the **Number of pre/post samples** text box, type 0. This value means that samples are not saved before a trigger or skipped after a trigger.
- 10 In the **Decimation** text box, type 1. This value means that data is collected at each sample interval.
- 11 Select **Trigger mode** FreeRun. This value means that the trigger event is automatic and no external trigger specification is required.

The target scope dialog box looks like this figure.

Block Parameters: Scope (xPC)

xpcscopeblock (mask) (link)

Simulink Real-Time Scope
Configure scope to acquire signal data.
Scope can be of type target, host, or file.

Parameters

Scope number:

Scope type:

Start scope when application starts

Scope mode:

Grid

Y-axis limits:

Number of samples:

Number of pre/post samples:

Decimation:

Trigger mode:

OK Cancel Help Apply

12 Click **OK**, and then from the **Simulation** tab, click **Save**.

Set Configuration Parameters

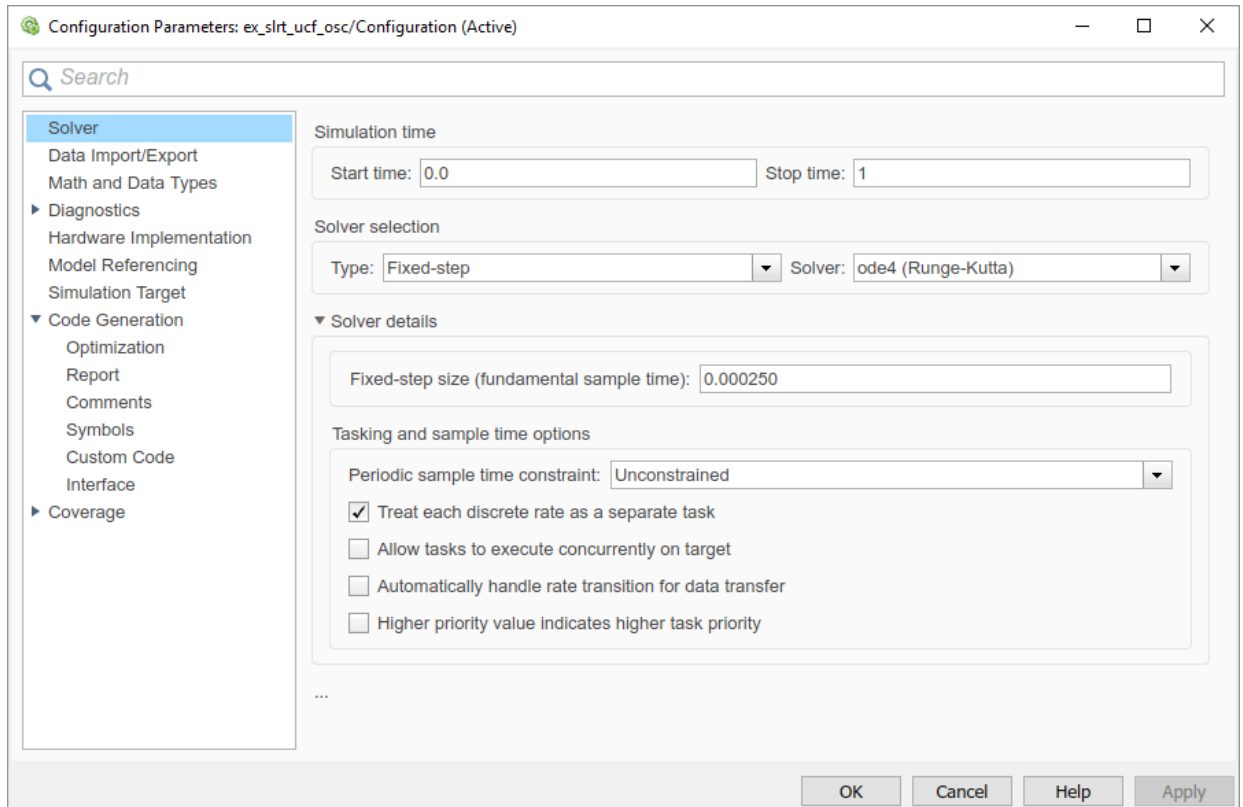
The example model, `ex_slrt_ucf_osc`, is a non-real-time model of a damped oscillator. You enter the simulation and real-time run parameters in the Configuration Parameters dialog box. These parameters provide information to Simulink Coder on how to build the real-time application from the Simulink model.

After you open a Simulink model and start the target computer, you can enter the simulation parameters.

- 1** In the MATLAB Command Window, type `ex_slrt_ucf_osc`.
- 2** In Simulink Editor, from the **Real-Time** tab, click **Hardware Settings**.
- 3** In the Configuration Parameters dialog box, click the **Solver** node.
- 4** In the **Solver selection** section, from the **Type** list, select **Fixed-step**.
- 5** From the **Solver** list, select a solver. For example, select the general-purpose solver `ode4` (Runge-Kutta).
- 6** Under **Solver details**, in the **Fixed-step size (fundamental sample time)** box, enter the sample time for the real-time application. For example, enter `0.00025` seconds (250 microseconds). After creating the real-time application, you can change this value.

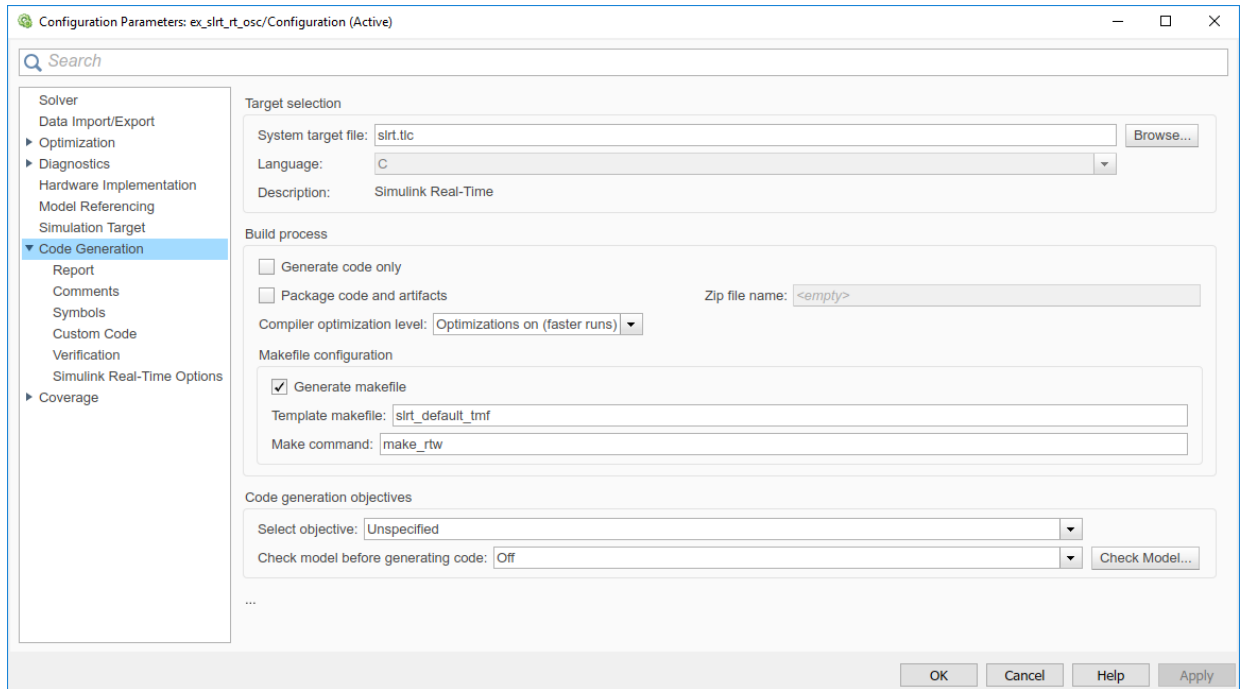
If you find that a value overloads the CPU on the target computer, try a larger **Fixed-step size** value, such as `0.0002` seconds.

The sample times of the model blocks can be multiples only of **Fixed-step size**. If you enter `'auto'` in **Fixed-step size**, the fundamental sample time is calculated from the sample times of the model blocks.



- 7 In the Configuration Parameters dialog box, click the **Code Generation** node.
- 8 To build a basic real-time application, in the **Target selection** section, click **Browse** at the **System target file** list. Click `slrt.tlc`, and then click **OK**.

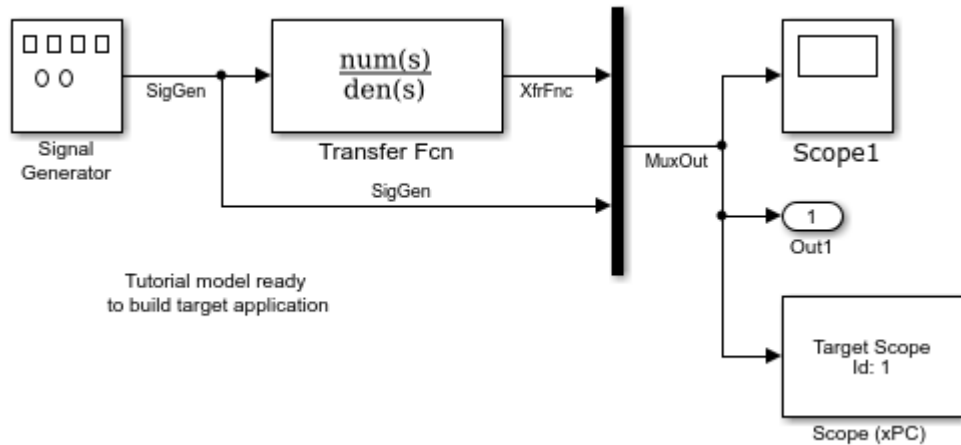
The system target file `slrt.tlc`, the template makefile `slrt_default_tmf`, and the make command `make_rtw` are automatically entered into the page. The **Simulink Real-Time Options** node appears in the left pane.



9 Click **OK**.

10 In the **Simulation** tab, from the **Save** button, click **Save As**.

Enter a file name. For example, enter `ex_slrt_rt_osc`
(`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`).



Start Target Computer

Starting the target computer loads and starts the Simulink Real-Time kernel on the target computer. The loader then waits for the Simulink Real-Time software to download the real-time application from the development computer.

After you have configured the Simulink Real-Time product by using the Simulink Real-Time Explorer, you can start the target computer and establish the development computer connection to the target computer. Before building the real-time application, you start the target computer. After the build process completes, you download the real-time application to the target computer.

- 1 Start the target computer.
- 2 Observe that the target computer displays a Simulink Real-Time screen.
- 3 If you have a keyboard attached to the target computer, you can activate that keyboard or console for target computer commands by typing **C**. Press the **Page Up** and **Page Down** keys to page up and down in the target computer monitor. For more information, see "Target Computer Commands".

The status window shows that the kernel is in loader mode and waiting to load a real-time application. The memory value is the number of bytes of target computer memory available for the heap, for running scopes, and for data acquisition buffers.

Note Target computer memory for the real-time application executable, the kernel, and other uses is limited to a maximum of 4 GB.

Build and Download Real-Time Application by Using Run on Target

The example model is a real-time model of a damped oscillator, `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`). To generate C code, compile, link, and download the real-time application to the target computer, use the Simulink Real-Time build process .

After you enter changes in the Configuration Parameters dialog box, you can build the real-time application. By default, the build procedure downloads the real-time application to the default target computer, as designated in Simulink Real-Time Explorer. For further details on setting the target computer for a real-time application, see “Simulink Real-Time Options Pane”.

- 1 In the MATLAB Command Window, type `ex_slrt_rt_osc`.

MATLAB loads the oscillator model and displays the Simulink block diagram.

- 2 In Simulink Editor, from the **Real-Time** tab, click **Run on Target**.

On the development computer, you can view the build status after completing a build in the **Diagnostics Viewer**. On the **Debug** tab, click **Diagnostics**.

If you have a monitor connected to your target computer, the monitor screen displays a Simulink Real-Time screen with the application name.

- 3 In the MATLAB Command Window, type:

```
tg = slrt('TargetPC1')
```

MATLAB displays a list of properties for the target object `tg`.

If the software detects an error during build and download, see “Troubleshooting in Simulink Real-Time”.

If you download a real-time application built with a Simulink Real-Time version different from the version of the kernel on the target computer, the software prints an error:

```
Mismatch between model and kernel versions
```


To prevent this version mismatch, rebuild real-time applications with each new Simulink Real-Time release.

During the build process, the Simulink Real-Time software creates a target object that represents the real-time application running on the target computer. You control the real-time application and computer by setting the target object properties and calling target object functions. For a list of these functions, such as `load`, see `SimulinkRealTime.target`.



Execute Real-Time Application in Simulink External Mode by Using Step by Step Commands

Control of the real-time application with Simulink is limited to connecting a Simulink model to a real-time application through external mode, and then starting the real-time application. Using Simulink external mode is one method to tune parameters.

The model connects to the target computer that you have selected in the Simulink Editor on the **Real-Time** tab.

Note To avoid target computer communication issues, do not use Simulink external mode while Simulink Real-Time Explorer is running. Use only one interface.

After you build and download a real-time application to the target computer, you can run the real-time application. This procedure uses the Simulink model `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`). See “Build and Download Real-Time Application by Using Run on Target” on page 4-18. You must already have specified the required target computer environment. Specifically, you must select the target computer to which you want to connect. See the **Build for default target computer** check box description in “Simulink Real-Time Options Pane”.

- 1 In the Simulink Editor, connect to the target computer. In the **Real-Time** tab, click **Toggle target computer connection** . The connection status toggles to the connected state .
- 2 In the **Real-Time** tab, from the **Run on Target**, start the **Step by Step Commands** by selecting **Build Application**.



- 3 After the model build completes, in the **Real-Time** tab, from the **Run on Target**, select **Deploy to Target**.



- 4 After the real-time application deploys to the target, in the **Real-Time** tab, from the **Run on Target**, select **Connect Model**.

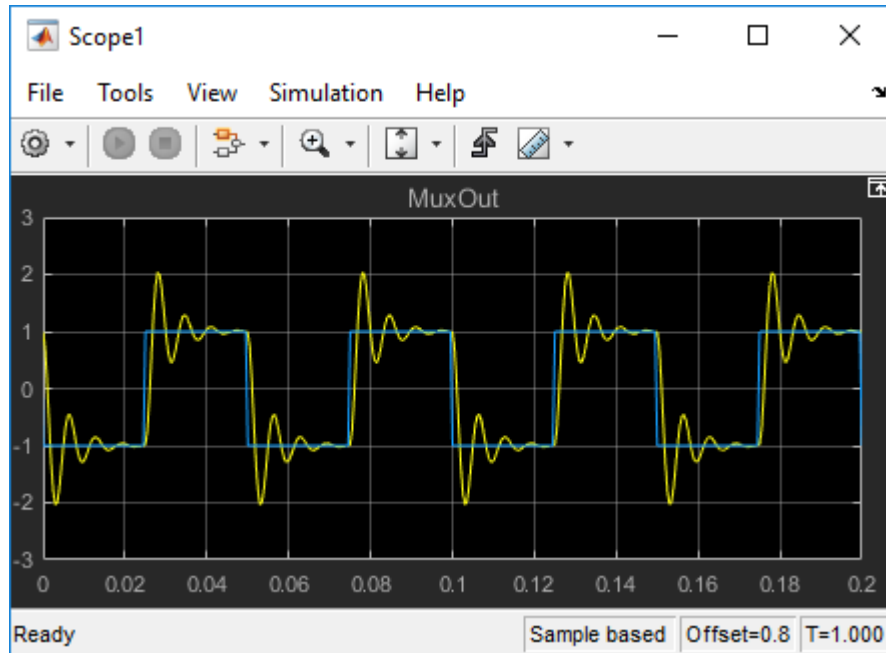


After you connect the model to the real-time application, the status bar displays External to indicate that the model is ready for external mode simulation. In this mode, the model serves as the interface to the application. This connection permits tuning parameters in the real-time application without rebuilding the application. This connection also permits observing signal activity in the real-time application by using, for example, SimulinkScope blocks in the model.

- 5 In the **Real-Time** tab, from the **Start Application (Run on Target)**, select **Start Application**.




The current Simulink model parameters are downloaded from the development computer to the real-time application. The real-time application begins running, and the Simulink Scope block displays output.

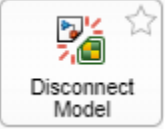




- 6 To stop execution, in the **Real-Time** tab, from the **Stop Application (Run on Target)**, select **Stop Application**.



When the real-time application deployed to the target computer, the Step by Step Commands provide several additional workflow options.

Commands	Step by Step Operations
	<p>In the Step by Step Commands, after you deploy the real-time application to the target computer, you can connect the model to the real-time application. The Connect Model operation lets you use the model as an interface to the real time application for external mode simulation.</p>

Commands	Step by Step Operations
 <p>Disconnect Model</p>	<p>In the Step by Step Commands, after you connect the model to the real-time application on the target computer, you can disconnect the model from the real-time application. The Disconnect Model operation lets you keep the real-time simulation running on the target computer and re-connect to the model at a later time.</p>
 <p>Start Application</p>	<p>In the Step by Step Commands, after you deploy the real-time application to the target computer and connect the model to the real-time application, you can start the real-time application. The Start Application operation lets you start the real time application for external mode simulation.</p>
 <p>Restart Application</p>	<p>When you are connected in external mode, to restart the real-time application from time $t=0$, click the Restart Application button. This button restarts the application and maintains the external mode connection.</p>

See Also

Mux | Scope | Scope | SimulinkRealTime.target | load | viewTargetScreen

More About

- “System Configuration”
- “Set Up and Configure Simulink Real-Time” on page 4-2
- “Apply Simulink Real-Time Model Template to Create Real-Time Application”

Apply Simulink Real-Time Model Template to Create Real-Time Application

This example shows how to use the Simulink Real-Time template to create a Simulink model. Starting from the model template for Simulink Real-Time provides a new model that has configuration parameters set up for building a real-time application.

To see the Simulink Real-Time commands for each operation in this example, view the example code.

Create a Simulink Model from Template

To create this model from the Simulink start page, in the MATLAB Command Window, type:

```
simulink
```

Select the Simulink Real-Time template from the start page and create the `exampleSlrtApp` model. Or, in the Command Window, use the `Simulink.createFromTemplate` command.



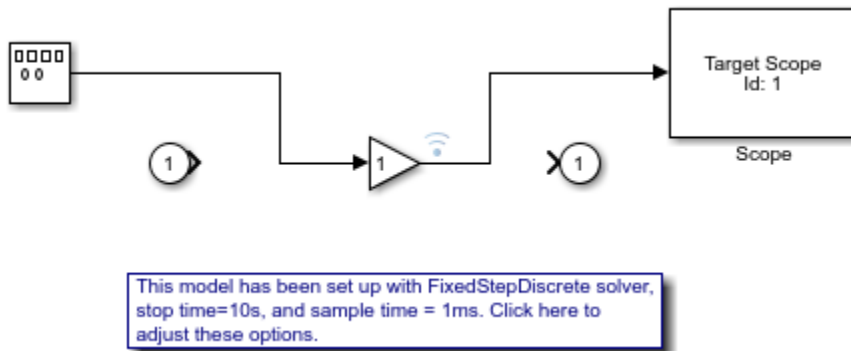
This model has been set up with FixedStepDiscrete solver, stop time=10s, and sample time = 1ms. [Click here to adjust these options.](#)

Add Blocks, Connections, and Data Logging to the Model

The Simulink Real-Time model template contains a Gain block that connects an inport to an output. To make this model produce more interesting data to view on a target scope and to view with the Simulation Data Inspector (SDI), add these blocks and connections. You can use the UI for these steps or use the script commands shown:

- Add a signal generator by using the `add_block` command. Use the `set_param` command to set its Amplitude parameter value to 4 and set its Frequency parameter value to 400.

- Add a target scope by using the `add_block` command.
- Remove the connections between the Gain block, inport, and outport by using the `delete_line` command.
- Connect the signal generator to the gain block input by using the `add_line` command.
- Connect the gain block output to the target scope by using the `add_line` command.
- Mark the Gain block output for data logging by using the `set_param` command.



Build the Real-Time Application and View Logged Data

You are ready to build the real-time application, run it on the target computer, and view the logged data with these steps:

- Make sure that the development computer has a connection to the target computer.
- Build the model and download the real-time application to the target computer. On the **Real-Time** tab, click **Run on Target**. Or, use the `rtwbuild` command and the `load` command.
- Run the real-time application and log data by using the `start` command.
- Open the Simulation Data Inspector by double-clicking the Simulation Data Inspector icon on the Gain block output signal or by using the `Simulink.sdi.view` command.

More Information

- "Create and Run Real-Time Application from Simulink Model" on page 4-9
- "Configure and Control a Real-Time Application" on page 4-26




- Simulation Data Inspector

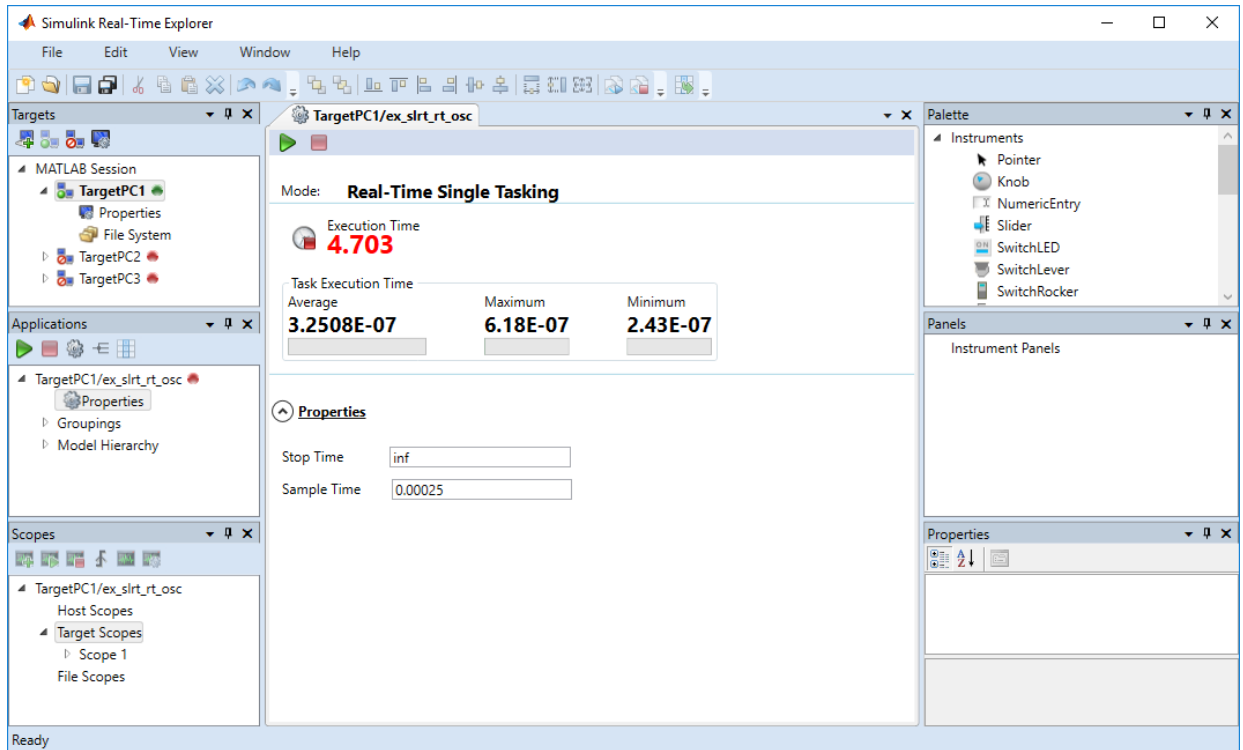
Configure and Control a Real-Time Application

You can configure and control a real-time application by using the Simulink Real-Time Explorer and Simulink external mode. This tutorial uses Simulink Real-Time Explorer. The model is a real-time model of a damped oscillator, `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`)).

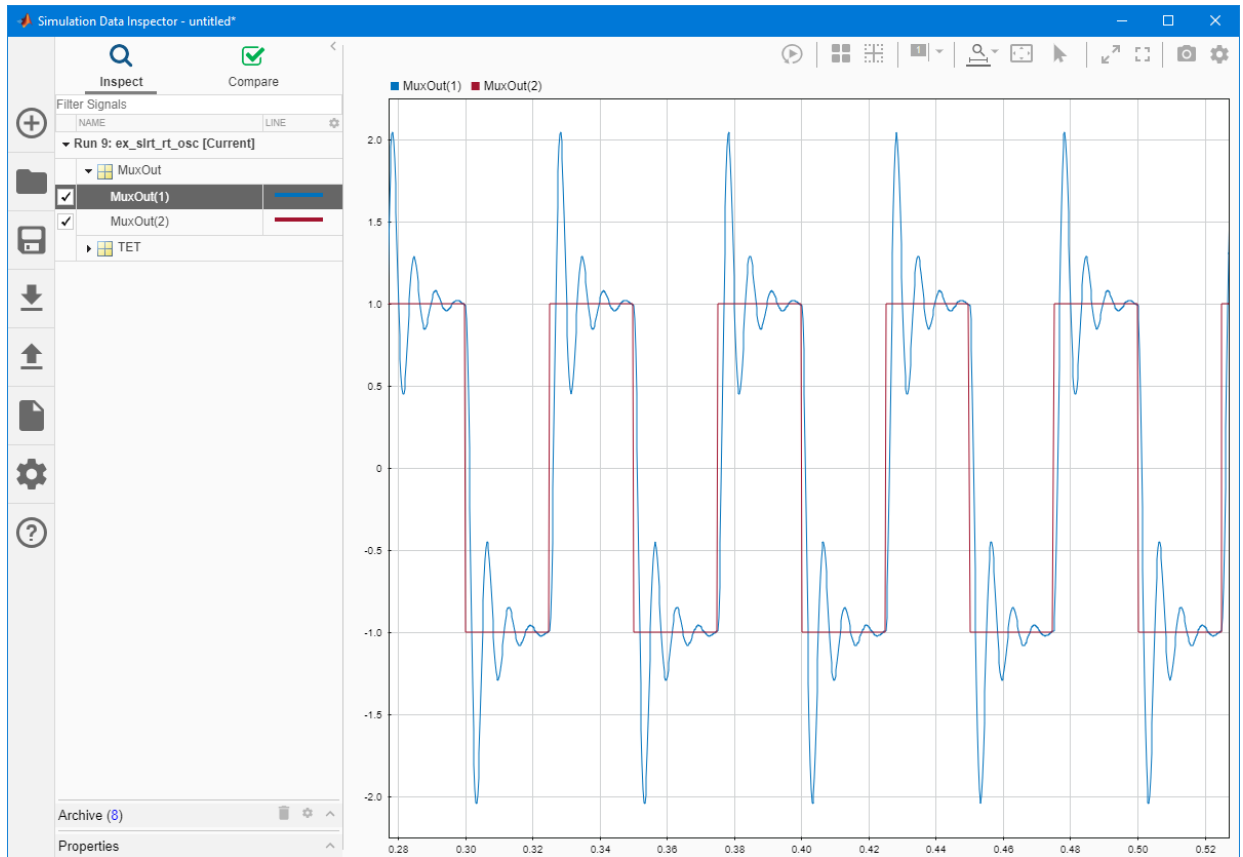
Execute Real-Time Application with Simulink Real-Time Explorer


To perform this procedure, you must have already created a Simulink Real-Time boot disk and started the target computer. See “Start Target Computer” on page 4-17). This procedure begins with the real-time application `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`) already downloaded to the target computer. See “Build and Download Real-Time Application by Using Run on Target” on page 4-18.

- 1 In the Simulink Editor, open Simulink Real-Time Explorer. In the **Real-Time** tab, from the **Prepare** section, click **SLRT Explorer**.
- 2 In the **Targets** pane, click the target computer icon for which you have downloaded the real-time application. On the toolbar, click the **Connect** button . The target computer icon turns to **Connected** .
- 3 In the **Applications** pane, select the real-time application.
- 4 To start execution, click the real-time application. On the toolbar, click the **Start** button . The application starts running.



To view output from the real-time application, in the Simulink Editor, mark the output signal for buffered data logging. Then rebuild, load, and run the real-time application. Open the Simulation Data Inspector and view the output. In the **Real-Time** tab, click **Data Inspector**.



- 5 To stop execution, click the real-time application. In the **Real-Time** tab, click the **Stop** button .


The real-time application on the target computer stops running. The target computer displays messages like these messages.

```
minimal TET: 0.0000006 at time 0.001250
maximal TET: 0.0000013 at time 75.405500
```



Alternatively, you can use functions in the MATLAB Command Window to start and stop the real-time application. For more information about these functions, such as `load`, see `SimulinkRealTime.target`.

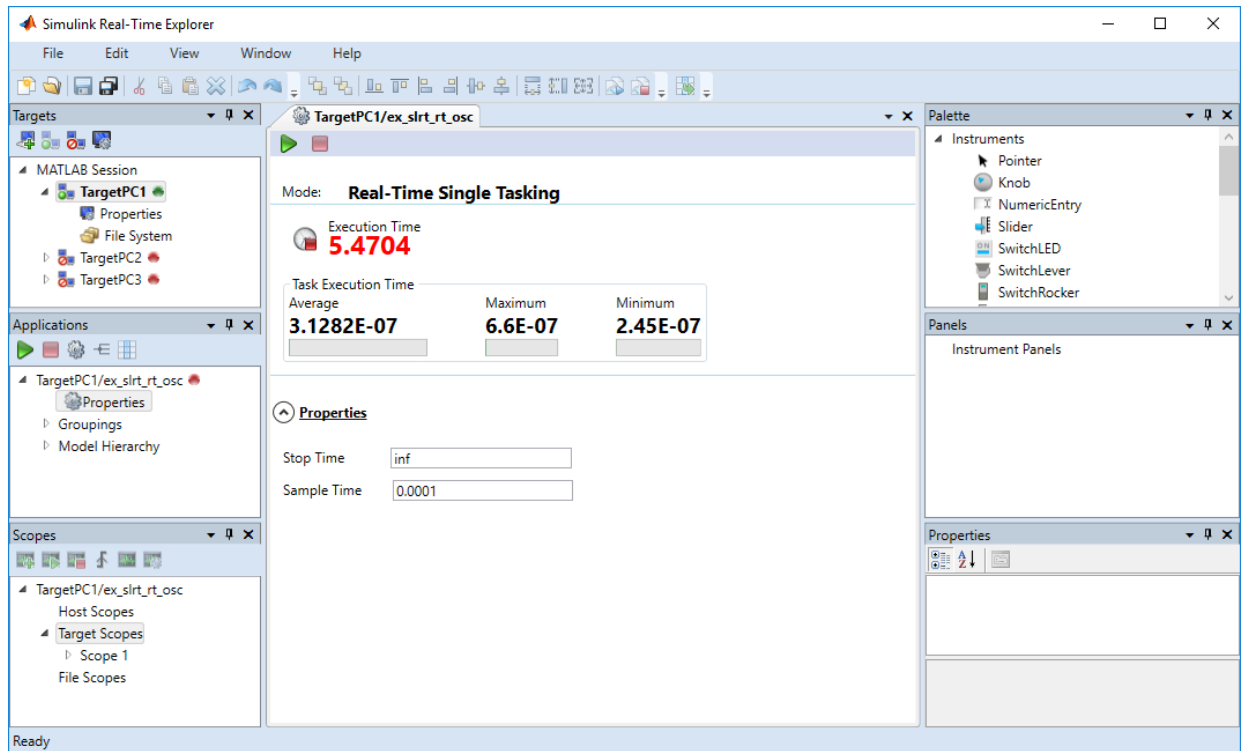
Change Stop Time and Sample Time

To observe operation of the real-time application, you can change the stop time and sample time in the real-time application configuration. You must have already downloaded the real-time application `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`) to a target computer.

- 1 Open Simulink Real-Time Explorer. In the **Real-Time** tab, from the **Prepare** section, click **SLRT Explorer**.
- 2 In Simulink Real-Time Explorer, in the **Applications** pane, expand the node of the loaded real-time application.
- 3 On the toolbar, click the **Properties** button .
- 4 In the **Application Configuration** workspace, under the **Properties** arrow, enter a new value for **Stop time**. For example, enter `inf` and press **Enter**. The real-time application now runs until you stop it.
- 5 Enter a new value for **Sample Time**. For example, enter `0.00010` and press **Enter**.

Note Some blocks produce incorrect results when you change their sample time at run time. If you include such blocks in your model, the software displays a warning message during model build. To avoid incorrect results, change the sample time in the original model, and then rebuild and download the model.

- 6 To start execution, click the real-time application. In Simulink Real-Time Explorer, click the **Start** button .
- 7 To stop execution, click the real-time application. In Simulink Real-Time Explorer, click the **Stop** button .



If you specify a sample time that is too small, a CPU overload can occur. If a CPU overload occurs, the target object property `CPUOverload` changes to `detected`. In that case, change **Fixed step size** to a larger value and rebuild the model.

See Also

Scope | viewTargetScreen

More About

- “Create Target Scopes with Simulink Real-Time Explorer”
- “Log Signal Data with Output Blocks and Simulink Real-Time Explorer”
- “Blocks Whose Outputs Depend on Inherited Sample Time” (Simulink)



Process a Real-Time Application by Using Simulink Real-Time Explorer

You can create a real-time application for later downloading.

- 1 In Simulink Editor, from the **Real-Time** tab, click **Hardware Settings**.
- 2 Under the **Code Generation** node, select the **Simulink Real-Time Options** node.
- 3 Clear the **Automatically download application after building** check box.
- 4 From the **Real-Time** tab, click **Run on Target**.



Load a Preexisting Real-Time Application

After you have built a real-time application, you can load it from the build folder.

- 1 In MATLAB, navigate to the build folder.
- 2 In the Simulink Editor, open Simulink Real-Time Explorer. In the **Real-Time** tab, from the **Prepare** section, click **SLRT Explorer**.
- 3 In the **Targets** pane, click the required target computer icon and then click the **Connect** button . The target computer icon turns **Connected** .
- 4 Do one of the following:
 - From the build folder, drag the prebuilt real-time application DLM file to the target computer icon.
 - Right-click the target computer icon and click **Load**. In the **DLM Application Selector** dialog box, enter or browse to the prebuilt application DLM file.

Unload a Real-Time Application from the Target Computer

To remove a real-time application from the target computer:

- 1 In the Simulink Editor, open Simulink Real-Time Explorer. In the **Real-Time** tab, from the **Prepare** section, click **SLRT Explorer**.
- 2 Right-click the target computer icon and click **Unload**.
- 3 In the **Targets** pane, click the target computer icon and click the **Disconnect** button . The target computer icon turns **Disconnected** .

See Also

More About

- “Simulate Simulink Model by Using MATLAB Language” on page 4-33

Simulate Simulink Model by Using MATLAB Language

Run a simulation of the Simulink model to observe the non-real-time behavior of the model.

This procedure uses the Simulink model `ex_slrt_nrt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_nrt_osc')))`). Before you run the simulation, load this model. From the **Modeling** tab, click **Model Settings**. Enable **Data Import/Export > Time** variable `tout` and enable **Data Import/Export > Output** variable `yout`.

- 1 In the MATLAB Command Window, type:

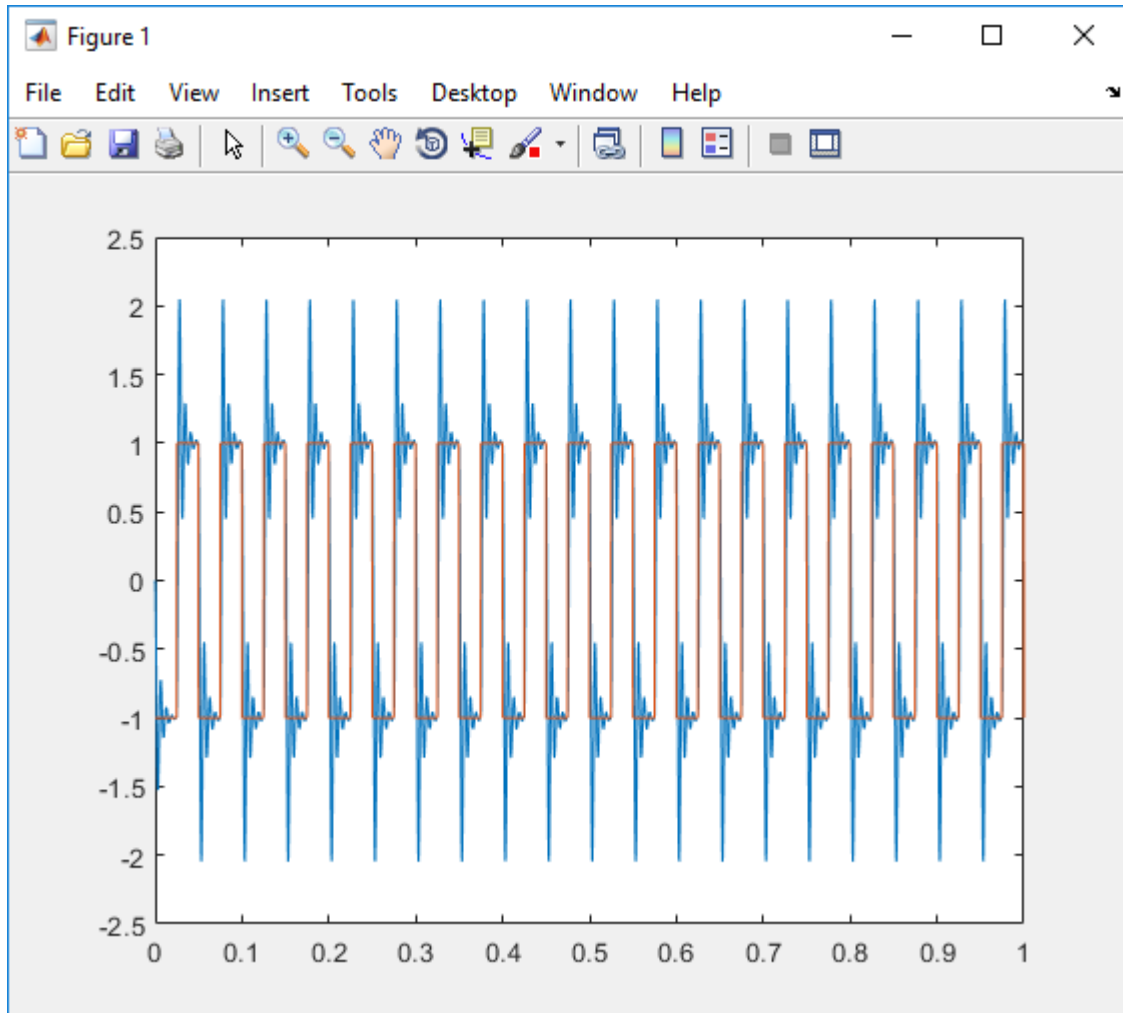
```
output=sim('ex_slrt_nrt_osc','SimulationMode','normal');
```

The signal from the signal generator is logged to memory through the Outport block.

- 2 After Simulink finishes the simulation, type:

```
plot(output.get('tout'), output.get('yout'))
```

MATLAB opens a plot window and displays the output response.



When the real-time application is running in Real-Time mode, data is saved in the target computer memory. You retrieve the data through the target object properties `tg.TimeLog`, `tg.StateLog`, and `tg.OutLog`. In the Configuration Parameters dialog box, you must still select the **Time**, **States**, and **Output** check boxes for data to be logged to the target object properties.

See Also

Outport

More About

- “Process a Real-Time Application by Using Simulink Real-Time Explorer” on page 4-31
- “Prepare Real-Time Application by Using MATLAB Language” on page 4-36

Prepare Real-Time Application by Using MATLAB Language

You can configure your real-time application by using the MATLAB language and build and download it to the target computer. This example uses the `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`). It assumes that the default target computer is configured for network boot mode and that communication has been established.

- 1 Make sure that the connection to the target computer is functioning.

```
slrtpingtarget
```

- 2 Load the model.

```
model_name = 'ex_slrt_rt_osc';  
open_system(model_name);
```

- 3 Set the start and stop time parameters.

```
set_param(model_name, 'StartTime', '0');  
set_param(model_name, 'StopTime', '10');
```

- 4 Build the model, assign `tg` to the target computer, and download the model.

```
rtwbuild(model_name);  
tg = slrt('TargetPC1');  
load(tg, model_name);  
pause(30);
```

See Also

More About

- “Execute Real-Time Application with MATLAB Language” on page 4-37

Execute Real-Time Application with MATLAB Language

Run the real-time application with generated code to observe the real-time behavior of the model. This procedure uses the Simulink model `ex_slrt_rt_osc` (`open_system(docpath(fullfile(docroot, 'toolbox', 'xpc', 'examples', 'ex_slrt_rt_osc')))`). You must have already carried out the steps in “Prepare Real-Time Application by Using MATLAB Language” on page 4-36. In the model, select the output signal and mark the signal for data logging with the Simulation Data Inspector.

This example uses functions in the MATLAB Command Window to start and stop the real-time application. For more information about these functions, such as `load`, see `SimulinkRealTime.target`.

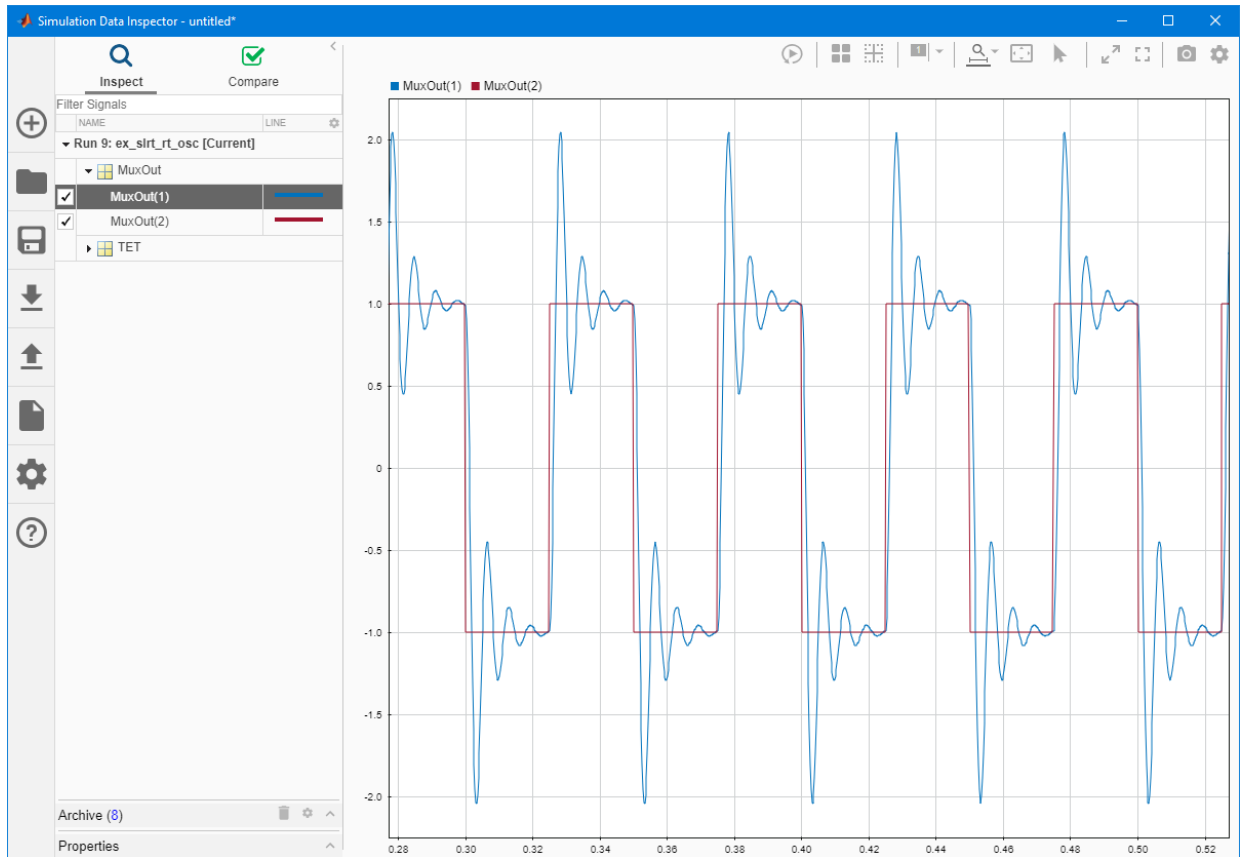
- 1 In the MATLAB Command Window, type:

```
start(tg)
```

The real-time application starts running on the target computer. In the Command Window, the status of the target object changes from stopped to running.

```
Target: TargetPC1
        Connected           = Yes
        Application          = ex_slrt_rt_osc
        Mode                 = Real-Time Single-Tasking
        Status               = running
```

On the target computer screen, the **Execution** line changes from stopped to running. The **AverageTET** line is periodically updated with a new value.



2 In the Command Window, type:

```
stop(tg)
```

The real-time application stops running.

You can change many properties and parameters without rebuilding the real-time application. Two of these properties are `StopTime` and `SampleTime`.

3 Change the stop time. For example, to change the stop time to 1000 seconds, type:

```
tg.StopTime = 1000
```

4 Change the sample time. For example, to change the sample time to 0.01 seconds, type:

```
tg.SampleTime = 0.01
```

Note Some blocks produce incorrect results when you change their sample time at run time. If you include such blocks in your model, the software displays a warning message during model build. To avoid incorrect results, change the sample time in the original model, and then rebuild and download the model.

If you specify a sample time that is too small, a CPU overload can occur. If a CPU overload occurs, the target object property `CPUOverload` changes to `detected`. In that case, change **Fixed step size** to a larger value and rebuild the model.

See Also

`viewTargetScreen`

More About

- “Prepare Real-Time Application by Using MATLAB Language” on page 4-36
- “Blocks Whose Outputs Depend on Inherited Sample Time” (Simulink)

Application and Driver Scripts

These examples show the features of Simulink Real-Time. Use these scripts as a starting point for writing your own scripts for creating and testing real-time applications.

Examples include general applications examples and driver examples. You can access the general application and driver examples through MATLAB Help. In the help window, click **Real-Time Simulation and Testing > Simulink Real-Time > Examples**.

General Examples

The general application examples include models and scripts.

Description	File Name
Real-time parameter tuning and data logging	"Parameter Tuning and Data Logging"
Freerun display mode of a host scope	"Signal Tracing With a Host Scope in Freerun Mode"
A software-triggered host scope	"Signal Tracing Using Software Triggering"
A signal-triggered host scope	"Signal Tracing Using Signal Triggering"
A scope-triggered host scope	"Signal Tracing Using Scope Triggering"
Signal tracing with a target scope	"Signal Tracing With a Target Scope"
Pre- and post-triggering of a host scope	"Pre- and Post-Triggering of a Host Scope"
Time- and value-equidistant data logging	"Time- and Value-Equidistant Data Logging"
Logging signal data with the Simulation Data Inspector to a file on the development computer	"Data Logging With Simulation Data Inspector (SDI)"
Frame signal processing	"Frame Signal Processing"
Simulink Real-Time software as a real-time spectrum analyzer	"Spectrum Analyzer"

Description	File Name
Creating a custom client application to interface with the target computer	"Simple Client Application With the .NET API"
Concurrent execution of a model using the Simulink Real-Time profiling tool	"Concurrent Execution on Simulink® Real-Time™ "

Driver Examples

These driver examples include models and scripts showing the use of driver blocks in a Simulink Real-Time environment. To run some of these examples, you must install specific hardware.

- Analog and digital I/O
- ARINC 429
- Asynchronous events
- Audio
- CAN (CAN_MESSAGE data types)
- Counters, timers, pulse width modulators (PWM)
- Digital signal processing
- Encoders
- EtherCAT
- FPGA
- J1939
- MIL-STD-1553
- Precision Time Protocol (PTP)
- Raw Ethernet
- RS-232
- Shared/reflective memory
- UDP
- Video

Edit Scripts

To locate and edit an example such as “Signal Tracing With a Host Scope in Freerun Mode”:

- 1 Follow the link to the example and determine its MATLAB file name, in this case `scfreerundemo`.

- 2 In the MATLAB Command Window, type:

```
which scfreerundemo
```

MATLAB displays the location of the MATLAB example file.

```
C:\MATLAB\toolbox\rtw\targets\xpc\xpcdemos\scfreerundemo.m
```

- 3 Type:

```
edit scfreerundemo
```

MATLAB opens the file in the MATLAB Editor.