Simulink® Real-Time™

Getting Started Guide

R2014**b**
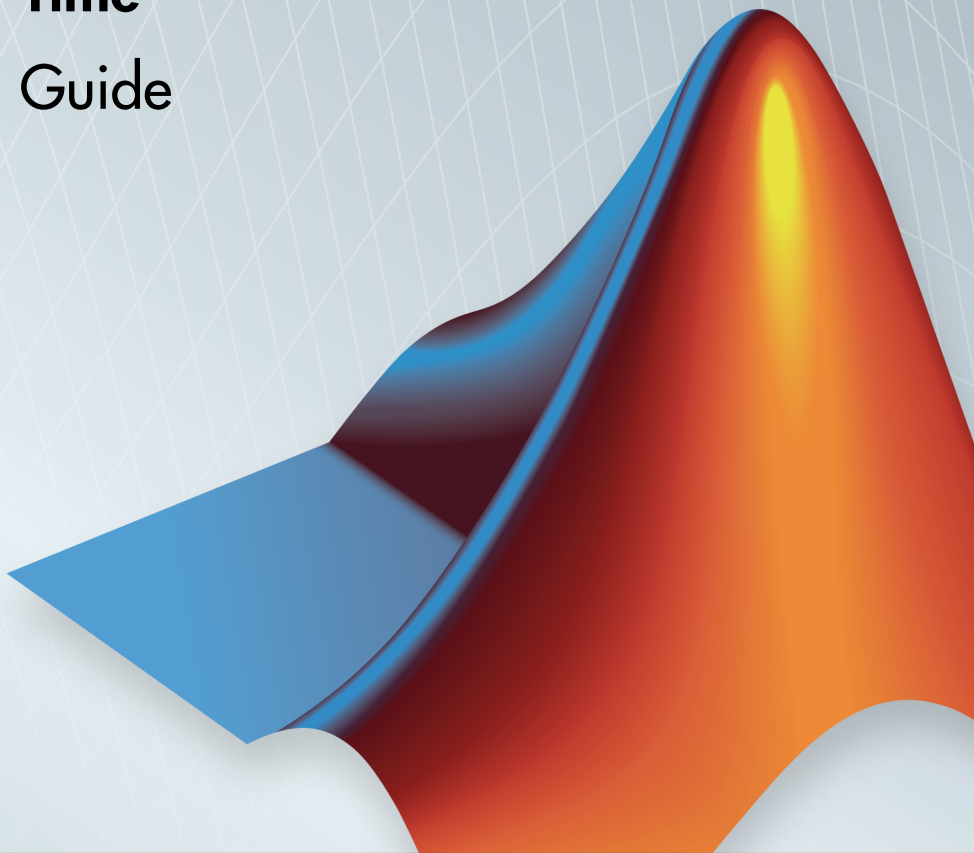
MATLAB®&SIMULINK®
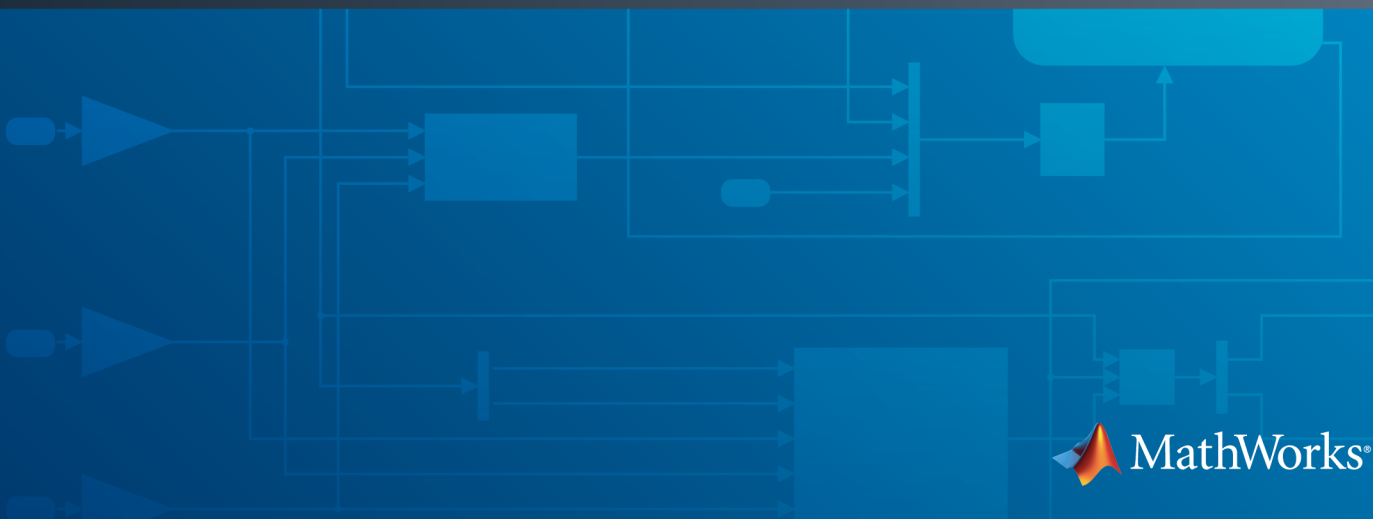
MathWorks®

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.
3 Apple Hill Drive
Natick, MA 01760-2098

Simulink® Real-Time™ Getting Started Guide

© COPYRIGHT 2000–2014 by 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)

Contents

Installation and Configuration

2

Basic Workflows

3

Glossary

1

Introduction

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 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. Simulink Real-Time can also be used with custom target computer and I/O hardware.

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)

Using the Product

The Simulink Real-Time environment uses a target computer, separate from a development computer, for running real-time applications. In this environment you use your desktop computer as a development computer with MATLAB, Simulink, and Stateflow® (optional) software, to create a model using Simulink blocks and Stateflow charts. After creating your model, you can run non-real-time simulations within the Simulink environment.

Use Simulink Real-Time software with Simulink Coder™, Embedded Coder® (optional), and a C/C++ compiler to create real-time executable code that represents your models. The executable code is downloaded from the development computer to the target computer running the Simulink Real-Time real-time kernel. After downloading the executable code, you can run and test your real-time application. Additionally, Simulink Real-Time software lets you add I/O blocks to your model to connect and communicate with your hardware under test

- Hardware requirements — The Simulink Real-Time software requires a development computer, target computer, and, for I/O, the target computer must also have I/O boards supported by the Simulink Real-Time product. The target computer can be a desktop computer, industrial computer, PC/104, PC/104+, or CompactPCI computer.

- Software requirements — The Simulink Real-Time software requires a Microsoft® Visual C++® compiler. In addition, it requires MATLAB, Simulink, and Simulink Coder software.

- Simulink Real-Time standalone mode requirements — Simulink Real-Time standalone mode requires DOS installed on the target computer and a DOS license available at start time. For more information, see "Standalone Boot Method".

- Documentation and help — The Simulink Real-Time software ships with the Simulink Real-Time documentation, which are available online through the MATLAB Help browser window or as PDF files that you can view online or print.

For more information, see MATLAB Central File Exchange for Simulink Real-Time Product (http://www.mathworks.com/matlabcentral/fileexchange/?term=Simulink+Real-Time) and the MathWorks® Support Simulink Real-Time Web site (http://www.mathworks.com/support/search_results.html?q=product:"Simulink+Real-Time").

Required Knowledge

Users who read this documentation should be familiar with

- Using the Simulink and Stateflow products to create models as block diagrams, and simulating those models in Simulink
- The concepts and use of Simulink Coder software to generate executable code

When using the Simulink Coder and Simulink Real-Time products, you do not need to program in C or other programming languages to create, test, and deploy real-time systems.

If you are a new user — Begin with "Getting Started with Simulink Real-Time". This topic gives you an overview of the Simulink Real-Time features and Simulink Real-Time environment. Next, read and try the examples in "Application and Driver Scripts".

If you are an experienced user — After you are familiar with using the Simulink Real-Time software, review the following topics:

- "Configuration Parameters"
- "Setup and Configuration"
- "Command-Line Setup"

Product Features

In this section...

Real-Time Kernel

The Simulink Real-Time software does not require Microsoft DOS, Microsoft Windows®, Linux®, or any other operating system on the target computer. Instead, you boot the target computer with boot media that includes the Simulink Real-Time kernel.

However, Simulink Real-Time standalone mode requires DOS installed on the target computer and a DOS license at start time. For more information, see "Standalone Boot Method".

Target Boot Options

You boot and run the target computer with one of the following boot options. These boot options eliminate the need to install software, modify existing software configurations, or access the hard disk on the target computer. This arrangement allows you to use the target computer for testing real-time applications. When you are finished with your tests, you can use the target computer again as a desktop computer. Software is not permanently installed on the target computer unless you deliberately install a standalone application on the hard disk or flash memory.

- Removable boot devices

  CD, DVD, USB and SD (compact) flash drive, removable hard drive, 3.5-inch floppy disk

- Fixed boot devices

  Hard drives (IDE or serial ATA (SATA)) or flash disks
- Network boot

  Dedicated network

Target Computer BIOS

At the beginning of the target computer boot process, the BIOS is loaded. Among other tasks, the BIOS searches for a bootable image (executable). This bootable image includes 16-bit and 32-bit tasks. The 16-bit task runs first because the CPU is in real mode by default. It prepares the descriptor tables and switches the CPU to protected mode. Next, the 32-bit task runs. It prepares the target computer environment for running the kernel and finally starts the real-time kernel.

You might need to enter the BIOS to customize settings for optimal real-time behavior of the system. For example, you can suppress checking for a keyboard or switch off power save features. Enabled power features can generate system management interrupts (SMIs). These features and support can also degrade real-time performance.

After loading the kernel, the target computer does not make calls to the BIOS or DOS functions. The resources on the CPU motherboard (for example, interrupt controller, UART, and counters) are addressed entirely through I/O addresses.

Real-Time Kernel

After the kernel starts running, it displays a welcome message with information confirming the connection between the development and target computers. The kernel activates the application loader and waits to download a real-time application from the development computer. Upon download, the loader receives the code, copies the different code sections to their designated addresses, and sets the real-time application ready to start. You can now use Simulink Real-Time functions and other utilities to communicate with the real-time application.

It is important to note that after the CPU switches to protected mode (32-bit), Simulink Real-Time does not switch the CPU back to real mode (16-bit).

The generated real-time application and the real-time kernel are compiled with a flat memory model. This provides full 32-bit power without time-consuming 16-bit segment switching and Microsoft DOS extenders.

Real-Time Application

The Simulink Coder, Embedded Coder, MATLAB Coder, and Simulink Real-Time products, and a C compiler, create a real-time application from a Simulink and Stateflow model. Real-time applications created with the Simulink Coder and Simulink Real-Time software run on a standard computer using a Simulink Real-Time real-time kernel.

The real-time application has the following characteristics:

- Memory model — The real-time application is compiled as an application with a flat memory model. This executable is then converted to an image suitable for the Simulink Real-Time software. It provides full 32-bit power without time-consuming 16-bit segment switching and DOS extenders. It also does not rely on DOS or any other Microsoft operating system.

- Task execution time — The real-time application is capable of high-speed, real-time task execution. A small block diagram can run with a sample time as fast as 20 μs (50 kHz). Model size and complexity interact with target computer capabilities to determine the maximum speed (minimal sample time) of execution.

For more information on creating a real-time application, see "Create and Run a Real-Time Application".

Signal Acquisition

The Simulink Real-Time real-time kernel stores signal data from the real-time application in RAM on the target computer. Alternatively, you can have the Simulink Real-Time real-time kernel store signal data in a file on the target computer. In either case, you can use this signal data to analyze and visualize signals. The Simulink Real-Time product supports the following types of signal acquisition:

- Signal monitoring — This is the process of acquiring signal data without time information. In this mode, you can get the current values of one or more signals. The data is not acquired in the real-time task but in the background task. The advantage of this process is that collecting data does not add computational load to the real-time application.

  For example, if you have a LED gauge in a Simulink model on the development computer, you could use signal monitoring to display the status of the signal.

- Signal logging — This is the process of acquiring signal data while a real-time application is running, and then visualizing the collected data after the real-time application stops running. The data is collected in the real-time task and acquired

samples are associated with a time stamp. After the run finishes or you manually stop the run, the development computer makes a request to upload data from the target computer. You can then visualize signals by plotting data on the development computer, or you can save data to a disk.

- Signal tracing — This is the process of acquiring and visualizing signal data while a real-time application is running. The data is collected in the real-time task and acquired samples are associated with a time stamp. It allows you to acquire signal data and visualize it on the target computer or to upload the signal data and visualize it on the development computer while the real-time application is running. The flexibility of this acquisition type is very similar to the behavior of a digital oscilloscope.

For information on acquiring signal data with the Simulink Real-Time software, see:

- "Alternative Configuration and Control Methods" on page 1-19
- "Monitor Signals Using MATLAB Language"
- "Signal Logging Basics"
- "Signal Tracing Basics"

Parameter Tuning

Most Simulink blocks have parameters (such as the amplitude and frequency of a sine wave) that you can change before or while your real-time application is running:

- Interactive — The Simulink Real-Time software supports real-time tuning of parameters while the real-time application is running.
- Scripts and batch procedures — The Simulink Real-Time software also includes commands to change parameters during a run or between runs. By writing a script that incrementally changes a parameter and monitors a signal output and running it on the development computer, you can optimize the value of that parameter.

For information on tuning parameters with the Simulink Real-Time software, see:

- "Alternative Configuration and Control Methods" on page 1-19
- "Parameter Tuning Basics"

Fixed-Point Support

The Simulink Real-Time software supports Simulink fixed-point data. This enables you to

- Monitor and log signals of fixed-point data types
- Tune parameters of fixed-point data types

MATLAB Compiler Support

The Simulink Real-Time software supports the MATLAB Compiler. With this capability, you can use the MATLAB Compiler to take MATLAB files as input and generate redistributable, standalone applications that include Simulink Real-Time functionality.

Standalone applications that include Simulink Real-Time functionality have the following limitations:

- No MATLAB Compiler support, which results in no access to the Simulink Real-Time library (`slrtlib`).
- No Simulink Real-Time Explorer, or other Simulink Real-Time graphical user interface support.
- No code generation functionality.

To use these features, create a file that uses the MATLAB Compiler command-line interface for the Simulink Real-Time software, then use the MATLAB Compiler.

BLAS Library Support

The Simulink Real-Time software supports the Basic Linear Algebra Subprograms (BLAS) library. This library speeds up large matrix (up to `16 x 16`) operations in real-time applications.

If you set up your model to Simulink Real-Time Embedded Coder (`slrtert.tlc`), you can create a custom Code Replacement Library (CRL) based upon the Simulink Real-Time BLAS (XPC_BLAS). For more on CRLs, see:

- "Code Replacement Libraries"
- "Code You Can Replace From Simulink Models"
- "Choose a Code Replacement Library"

Note: Your model accesses the XPC_BLAS library when you build your model with a Microsoft Visual C++ compiler.

Hardware Description

The hardware environment consists of a development computer, target computer, I/O boards in the target computer, and a serial or Ethernet link between the development and target computers. Knowing the different types of computers and I/O supported by the Simulink Real-Time software will help you to set up a real-time testing environment that meets your needs.

For a complete real-time testing solution, see Speedgoat turnkey systems. Speedgoat turnkey systems combine the Simulink Real-Time software with a variety of preconfigured real-time target computers.

In this section...
"Development Computer" on page 1-10
"Target Computer" on page 1-10
"Links Between Development and Target Computers" on page 1-12
"I/O Driver Support" on page 1-13

Development Computer

You can use a computer that runs a Windows operating system supported by MathWorks as the development computer. It must also support an available serial port or Ethernet adapter. In addition, to provide a means to boot the target computer, the development computer must have at least:

- CD or DVD drive
- Dedicated network access
- Removable drive, such as USB and SD (compact) flash drive, removable hard drive, or 3.5-inch floppy disk

For more details on the development computer requirements, see "Development Computer Hardware".

Target Computer

The Simulink Real-Time software supports concurrent access to up to 64 target computers with one host. A target computer can connect to only one development

computer at a time. A target computer cannot connect to multiple development computers.

A target computer can be a Windows compatible system with a 32-bit Intel® or AMD® processor (386 compatible or higher). It must also support a free serial port or an Ethernet adapter compatible with the Simulink Real-Time software. In addition, the target computer must contain a removable drive, CD or DVD drive, or have the ability to belong to a dedicated network. Using Simulink Real-Time standalone mode, you can transfer files from the removable drive or CD to a hard disk or flash memory.

A target computer can be one of the following:

- Desktop Computer — This computer is booted from a special target boot kernel created by the Simulink Real-Time software and stored on a target boot disk, removable boot drive, or network boot image.

  When you boot the target computer from the target boot kernel, the Simulink Real-Time software uses the resources on the target computer (CPU, RAM, and serial port or Ethernet adapter) without changing the files already stored on the hard drive.

  After you are done using your desktop computer as a target computer, you can reboot your computer without the target boot kernel and resume normal use of your desktop computer.
- Industrial Computer — This computer is booted from a special target boot kernel stored on a target boot disk, removable boot drive, hard disk, flash memory, or network boot image.

  When using an industrial target computer, you can select PC/104, PC/104+, CompactPCI, or single-board computer (SBC) hardware.

Note: Simulink Real-Time is designed to work with desktop target computers only. Do not use a laptop as a target computer.

For more details on the requirements of the target computer, see "Computer Hardware".

Links Between Development and Target Computers

The Simulink Real-Time product supports two connection types and communication protocols between the development computer and the target computer: serial and Ethernet.

Serial — The development and target computers are connected directly with a serial cable using their RS-232 ports. This cable is wired as a null modem link that can be up to 5 meters long and with a transfer rate between 1200 and 115200 baud.
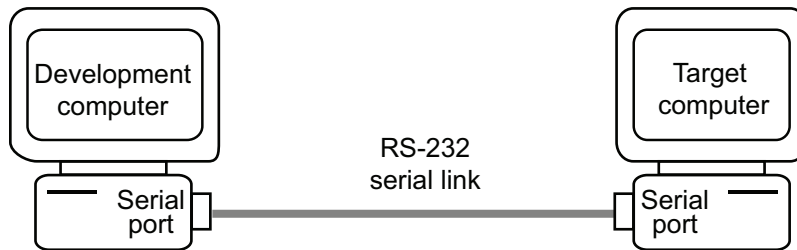
For detailed information on setting up the hardware and software for a serial link, see "Serial Link Setup".

Note: RS-232 communication type will be removed in a future release. Use TCP/IP instead.

Ethernet — The development and target computers are connected through an Ethernet network. The network can be a LAN, the Internet, or a direct connection using a crossover Ethernet cable. Both the development and target computers are connected to the network via Ethernet adapters using the TCP/IP protocol for communication.

When using a network connection, the target computer requires a supported Ethernet adapter card. The data transfer rate can be 10 megabits/second, 100 megabits/second, or 1 gigabit/second. For a list of supported cards, see `https://www.mathworks.com/products/simulink-real-time/supported/xpc-target-supported-ethernet-chipsets.pdf`.
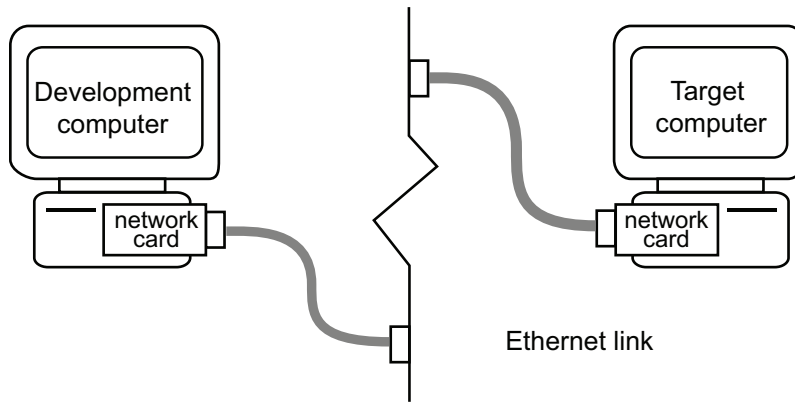
Ethernet link

For detailed information on setting up the hardware and software for Ethernet links, see "Ethernet Link Setup".

Advantages of Ethernet Links

A connection between the development and target computers using Ethernet has advantages over serial links:

- Higher data throughput — Network communication using Ethernet can transfer data up to 100 Mbit/second instead of the maximum data transfer rate of 115 kBaud with a serial link.
- Longer distances between the development and target computers — By using repeaters and gateways you do not restrict the distance between your development and target computers to the length of a serial cable. Communication over the Internet is also possible.

This manual does not include information for installing network cards or the TCP/IP protocol on your development computer. To install and configure your network cards and TCP/IP protocol, contact your system administrator.

I/O Driver Support

The Simulink Real-Time product supports a wide range of third-party I/O boards. The list of supported I/O boards includes ISA, PCI, PCIe, PMC, PC/104, PC/104+, and CompactPCI hardware. The drivers are represented by Simulink blocks. Your interaction with the I/O boards is through these Simulink blocks and the parameter dialog boxes. MathWorks does not manufacture the boards.

Note: You are responsible for taking all required precautions and implementing safeguards when interfacing hardware with the Simulink Real-Time product. You are also solely responsible for the content of your models that controls such hardware.

I/O board library — The I/O board library contains Simulink blocks for the Simulink Real-Time product. You drag and drop blocks from the I/O library and connect I/O drivers to your model the same way you would connect a standard Simulink block.

I/O support — The I/O device library supports approximately 300 standard boards. I/O boards plug into the target computer expansion bus, PC/104 stack, or industrial computer chassis. There is also support for modules that plug into IP or PMC carrier boards. The Simulink Real-Time block library supports the following I/O block categories:

- Analog input (A/D) and analog output (D/A) — Connect sensors and actuators to your real-time application.
- Digital input and output — Connect to switches, on/off devices, and communicate information in parallel.
- RS-232/422/485 support — Use the COM1 or COM2 ports for serial communication with external devices. You can also access multiple RS-232, RS-422, and RS-485 serial ports using Quatech® and Commtech devices.
- CAN support — You can use CAN-AC2, CAN-AC2-PCI, and CAN-AC2-104 boards from Softing® GmbH AG with Simulink Real-Time CAN blocks to interface with a CAN field bus network. This interface provides communication through a CAN network between real-time applications and remote sensors and actuators.

  The Simulink Real-Time CAN blocks are compatible with CAN specifications 2.0A and 2.0B and use both dynamic object mode and FIFO mode.
- GPIB support — Special RS-232 drivers support communication with a GPIB control module from National Instruments® to external devices with a GPIB connector.
- UDP support — Communicate with another system using the standard UDP/IP network protocol.
- Counter-Timers — Use the counter-timer blocks for measuring pulse and frequency with modulation applications.
- Watchdog — Monitor an interrupt or memory location, and reset the computer if an application does not respond.
- Incremental encoder — Change motion into numerical information for determining position, direction of rotation, and velocity.

- Shared memory — Use shared memory blocks with multiprocessing applications.

- LVDT — Use the North Atlantic Industries, Inc. 73LD3, 76CL1, 76LD1, and 76CL1 boards with Simulink Real-Time LVDT blocks to work with LVDT applications.

- ARINC-429 — Use the Condor Engineering® CEI-X20 boards with Simulink Real-Time ARINC-429 blocks to interface with the ARINC 429 data bus.

- MIL-STD-1553 — Use the Condor Engineering PCI-1553 and QPCI-1553 series boards with Simulink Real-Time MIL-STD-1553 blocks to interface with the MIL-STD-1553 data bus.

- Audio — Use the audio blocks to work with audio applications.

- Thermocouple — Use the Measurement Computing™ PCI-DAS-TC board with Simulink Real-Time thermocouple blocks to work with thermocouple applications.

Real-Time Test Environment

In this section...

Software Setup

The real-time test environment is a place to design, build, and test a real-time application as a free-running application and as a real-time application. It also includes communication between the development and target computers.

You create a free-running test environment by creating an initial model in regular Simulink.

You create a real-time test environment for a Simulink model by connecting a development computer, target computer, and the hardware you want to test. You run the following software on the development computer:

- Simulink Real-Time
- Simulink
- Simulink Coder
- MATLAB Coder
- A C compiler

After connecting the development computer to the target computer via a single Ethernet or RS-232 connection, you:

1 Connect the target computer to the hardware you want to test.

2 Download code generated by Simulink Coder from a Simulink model to the target computer via the communications connection.

Once you make the connections, you can:

- Access and interactively control the target computer and real-time application.

- Tune parameters before, during, and after real-time execution.
- Acquire, monitor, and log signal data.

Link Between Development and Target Computers

Whether using a serial connection (RS-232) or a Ethernet connection (TCP/IP), information is exchanged between the development and target computers. This information includes

- Real-time application — Download a real-time application from the host to the target computer.
- Control — Change properties and control the real-time application. This includes starting and stopping the real-time application, changing sample and stop times, and getting task execution time information about the real-time application and CPU.
- Signal data — Upload signal data from the target computer for analysis after the real-time application is finished running, or view signal data during the run.
- Parameter values — Download parameter values to the target computer between runs or during a run.

Note: RS-232 communication link will be removed in a future release. Use TCP/IP instead.

Simulink Real-Time Stand Alone Mode

Simulink Real-Time standalone mode enables applications generated with Simulink Real-Time to run on a target computer without being connected to a development computer. You can run your applications on a standalone target computer for data acquisition, calibration, testing, and small-batch production scenarios. You can distribute the applications royalty-free to multiple target computers.

When you have completed developing and testing, you can use the real-time application as a real-time system that runs on a dedicated target computer without needing to connect to the development computer.

When you start the target computer in Simulink Real-Time standalone mode, the target computer runs Microsoft DOS, starts the DOS program `xpcboot.com` from `autoexec.bat`, and then starts the Simulink Real-Time kernel from `xpcboot.com`:

When using Boot Floppy or CD Boot, you do not need DOS environment to load and run the Simulink Real-Time kernel. `DOSLoader` mode, like `StandAlone` mode, starts the target computer with DOS, starts the DOS program `xpcboot.com` from `autoexec.bat`, and then starts the kernel from `xpcboot.com`.

With standalone mode, you can deploy an unlimited number of real-time applications for standalone operation. For more information on standalone mode, see "Standalone Boot Method".

Alternative 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, and you can use other development environments to create 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.

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

Interface	Environment Properties	Control	Signal Acquisition	Parameter Tuning
"Simulink Real-Time Explorer" on page 1-20	X	X	X	X
"MATLAB Command-Line Interface" on page 1-22	X	X	X	X
"Simulink External Mode Interface" on page 1-24		X	X	X
"Simulink with Simulink Real-Time Blocks" on page 1-25			X	
"Target Computer Command-Line Interface" on page 1-26		X	X	X
"Custom GUI with Simulink Real-Time API for Microsoft .NET Framework" on page 1-27		X	X	X
"Custom GUI with Simulink Real-Time C API" on page 1-28		X	X	X
"Web Browser Interface" on page 1-29		X	X	X

Simulink Real-Time Explorer

The Simulink Real-Time software offers a graphical user interface (GUI) for configuring the development and target computers and interacting with a real-time application. To open the Simulink Real-Time GUI, in the MATLAB Command Window, type `slrtexplr`.

Simulink Real-Time Explorer includes the following capabilities:

- Environment configuration — Use the **Target Properties** pane to configure the Simulink Real-Time environment properties and create a Simulink Real-Time bootable image.

  Use node **File system** under the **MATLAB Session** tree to browse the target computer file system.
- Control — Use the **Targets** and **Applications** panes to load, unload, and run real-time applications. You can change stop time and sample times without regenerating code. You can get task execution time information during or after the last run.
- Signal acquisition — Use the **Scopes** pane and the **Model Hierarchy** node in the **Applications** pane to interactively monitor signals, add a host, target, or file scope, add or remove signals, and save and load signal groups.
- Parameter tuning — Use the **Model Hierarchy** node in the **Applications** pane to change tunable parameters in your real-time application and save and load parameter groups.
- Instrumentation — Use the **Palette** and **Panels** panes to create a graphical instrument panel for acquiring signals and tuning parameters.
- Window configuration — Use the tab and the icon to make multiple workspaces visible simultaneously.

  Use **File** > **Save Layout** and **Load Layout** to save and restore the Simulink Real-Time Explorer window layout.

For more information, see:

- "Ethernet Link Setup"
- "Serial Link Setup"
- "Target Computer Settings"
- "Target Boot Methods"

- "Execute Real-Time Application Using Simulink Real-Time Explorer"
- "Monitor Signals Using Simulink Real-Time Explorer"
- "Create Target Scopes Using Simulink Real-Time Explorer"
- "Create Host Scopes Using Simulink Real-Time Explorer"
- "Create File Scopes Using Simulink Real-Time Explorer"
- "Tune Parameters Using Simulink Real-Time Explorer"

MATLAB Command-Line Interface

You can interact with the Simulink Real-Time environment through the MATLAB command-line interface. Enter Simulink Real-Time functions in the MATLAB window on the development computer. 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, at the most basic level, what you can do with the Simulink Real-Time environment.

The GUIs provided with the Simulink Real-Time product are for completing the most common tasks. They use the Simulink Real-Time functions but do not extend their functionality. The command-line interface provides an interactive environment that you can extend.

The MATLAB command-line interface includes the following functions:

- Environment — Create a target boot kernel and directly change the environment properties without using a graphical interface.
- Control — Reboot the target computer, download a real-time application, start and stop real-time applications, and change start and sample times without regenerating code. Get task execution time information during or after the last run. Add and remove scopes, add/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 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 parameters while the real-time application is running, and use Simulink Real-Time functions to change parameters in between runs.

For more information, see:

- "Command-Line Ethernet Link Setup"
- "Command-Line Serial Link Setup"
- "Command-Line Target Computer Settings"
- "Command-Line Target Boot Methods"

- "Execute Real-Time Application Using MATLAB Language".
- "Monitor Signals Using MATLAB Language"
- "Configure Target Scopes Using MATLAB Language"
- "Create Host Scopes Using Simulink Real-Time Explorer"
- "Create File Scopes Using Simulink Real-Time Explorer"
- "Log Signal Data Using Outport Block and MATLAB Language".
- "Tune Parameters Using MATLAB Language".

Simulink External Mode Interface

Use Simulink in external mode to connect your Simulink block diagram to your real-time application. The block diagram becomes a graphical 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 the following functions:

- Control — Control is limited to connecting the Simulink block diagram to the real-time application, and starting and stopping the real-time application.

- Signal acquisition — You can use Simulink external mode to establish a communication channel between your Simulink block diagram and your real-time application. The block diagram becomes a graphical user interface to your real-time application and Simulink scopes can acquire signal data from the real-time application.

- Parameter tuning — Select external mode, and change parameters in the real-time application by changing parameters in the Block Parameters dialog boxes. Once you change a value and click **OK**, the new value is downloaded to the target computer and replaces the existing parameter while the real-time application continues to run.

For more information, see:

- "Execute Real-Time Application Using Simulink External Mode"
- "Trace Signals Using Simulink External Mode".
- "Tune Parameters Using Simulink External Mode".

Simulink with Simulink Real-Time Blocks

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 user interface model.

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.

For information on acquiring signal data with `Scope` blocks, see:

- "Add Simulink Real-Time Scope Block"
- "Set Target Scope Block Parameters"
- "Signal Tracing Basics"

For information on using Simulink Real-Time `To` and `From` blocks to transfer data to and from a Simulink user interface model, see "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. Enter commands in the command line on the target computer. This interface is useful with standalone applications that are not connected to the development computer.

The target computer command-line interface includes the following functions:

- 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.

For more information, see "Control Application at Target Computer Command Line".

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

Use the .NET API Simulink Real-Time framework to develop solutions (applications, human-machine interface (HMI) software, batch runs) that use the Simulink Real-Time software. 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 allow you to manipulate and interact with it. This document presents this reference using the C# language.

For more information, see "Using the Simulink Real-Time API for Microsoft .NET Framework".

Custom GUI with Simulink Real-Time C API

Use the C API to create a GUI application interface to a real-time application using a development environment that can link in a DLL.

Use the GUI application to control the application, tune parameters, and acquire signal data from a real-time application. The custom GUI runs on the development computer and communicates with the real-time application on the target computer using RS-232 or TCP/IP communication. A GUI application can be a console or Windows application using ActiveX® components.

For more information, see "Using the C API".

Web Browser Interface

If the target computer is connected to a network (TCP/IP), you can use a Web browser to interact with the real-time application from a computer connected to the network. If the target computer is connected to the development computer with an RS-232 cable, and is using the TCP/IP to RS-232 gateway, you can use a Web browser on the development computer.

The Web browser interface includes the following functions:

- Control — Start and stop the real-time application, and change the stop time and sample time.
- Signal acquisition — Signal tracing is limited to viewing a snapshot of a screen captured from the target computer screen. Add target scopes, add or remove signals, and set triggering modes. You can also monitor signal values.
- Parameter tuning — Change parameters in an HTML form, and then submit that form to make the changes in your real-time application.

For more information, see:

- "Web Browser Interface"
- "Trace Signals Using a Web Browser"
- "Log Signals Using a Web Browser"
- "Tune Parameters Using a Web Browser".

2

Installation and Configuration

Setup and Configuration

The initial hardware and software requirements for installing Simulink Real-Time are:

- "Development Computer Hardware"
- "Development Computer Software Installation"

If you are not using a Speedgoat target computer, see "Custom Target Computer Configuration".

The key setup and configuration steps using MATLAB language and Simulink Real-Time Explorer are:

1 "Command-Line C Compiler Configuration"
2 "Ethernet Link Setup"

   If your system does not support Ethernet communication, see "Serial Link Setup".
3 "Target Computer Settings"
4 "Target Boot Methods"

The key checkout steps using MATLAB language are:

- "Run Confidence Test on Configuration"
- "Run Benchmark Test on Configuration"

Development Computer Hardware

In the Simulink Real-Time software environment, the development computer is usually your desktop computer. Here you install the Simulink Real-Time product, along with MATLAB, Simulink, and other required and optional software. 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 one or more of:

- 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

Communications

Select one of the following methods for the development computer to communicate with the target computer:

- One available Ethernet interface (PCI, ISA, or USB) connected to a network (see "Ethernet Link Setup" on page 2-18 for details)
- One available serial port (COM1 or COM2) with a 9-pin or 25-pin D-sub connector (see "Serial Link Setup" on page 2-34 for details)

Computer Hardware

The target computer must be a 32- or 64-bit PC-compatible system. For example, you can use a second desktop computer or an industrial system like a PC/104 or CompactPCI as the target computer.

At a minimum, the Simulink Real-Time product requires the following target computer hardware.

Hardware	Description
CPU	Intel 386/486/Pentium or AMD K5/K6/Athlon with or without a floating-point coprocessor
Chip set	PC compatible with UART, programmable interrupt controller, keyboard controller, and counter
RAM	The Simulink Real-Time requires 8 MB or more of dynamic RAM

- To support Simulink Real-Time, 64-bit target computers must run in 32-bit mode.
- Simulink Real-Time is designed to work with desktop target computers only. Do not use a laptop as a target computer.
- The Simulink Real-Time kernel can use only 2 GB of memory. You can acquire several megabytes of data during a run, depending on how much memory you install in the target computer.

Peripherals

To start the Simulink Real-Time kernel, the target computer requires one or more of:

- 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
- PXE-compatible Ethernet adapter

A hard drive is not required unless you want to access the target computer file system (for file scopes).

- You can copy files to a hard drive or flash memory and start the Simulink Real-Time kernel from that device.
- If you want to access the target computer file system on a hard drive, see "File Systems".
- The hard drive must be a parallel ATA (PATA)/Integrated Device Electronics (IDE) or serial ATA (SATA) drive. For better performance, consider configuring this drive as a primary master.
- Verify that the hard drive is not cable-selected.
- The Simulink Real-Time product supports file systems of type FAT-12, FAT-16, or FAT-32.

A keyboard is required to control the target computer when you create standalone applications. If a keyboard is not connected, the BIOS might display an error message (keyboard failure). With a current BIOS, you can use the BIOS setup to skip the keyboard test.

A monitor is required to display results on the target computer. However, you can access target results using Simulink Real-Time functions on the development computer.

Communications

For the target computer to communicate with the development computer, select one of the following methods:

- One free supported Ethernet adapter (PCI, ISA, or USB) connected to a network (for supported Ethernet adapters, see "Ethernet Link Setup").

  If you want to start the target computer from the network, the Ethernet adapter must be compatible with the **Preboot eXecution Environment (PXE)** specification.

- One free serial port (COM1 or COM2) with a 9-pin or 25-pin D-sub connector (for details, see "Serial Link Setup"). Use a serial null modem cable to connect the target computer to the development computer.

Form Factors

Simulink Real-Time supports the following target computer hardware form factors:

- ISA
- PCI
- PMC
- PC/104 and PC/104+
- PCIe
- CompactPCI

I/O Boards

You can install I/O boards in the PCI or ISA slots of the target computer. These boards provide a direct interface to the sensors, actuators, or other devices for real-time control or signal processing applications. The Simulink Real-Time software supports I/O functionality via the blocks in `slrtlib`. You can also write a custom driver.

- For information on Speedgoat I/O modules, see "Speedgoat I/O Configuration".
- For information on Simulink Real-Time I/O blocks, see "Simulink Real-Time I/O Configuration".
- For information on writing custom drivers, see "Custom I/O Drivers".
- For information on installing and connecting I/O hardware, see the hardware manufacturer's documentation.

BIOS Settings

The Simulink Real-Time kernel does not require that an operating system be installed on the target computer. However, you must make the following BIOS settings for Simulink Real-Time to work properly.

- **Disable Plug-and-Play (PnP)** — Disable the PnP operating system feature so the PCI BIOS can set up the plugged-in PCI cards. The Simulink Real-Time kernel is not a PnP operating system. If this feature is enabled, PCI devices do not work with the Simulink Real-Time product.

- **Disable power saving mode**— Disable all power saving modes.

- **Disable PCI class `0xff` board detection** — Do not detect PCI boards with class code `0xff` in the target computer BIOS. Turn this option `Off` to enable the BIOS to detect and configure PCI boards.

- **Disable hyper-threading** — If your target computer supports hyper-threading capabilities, disable these capabilities. Enabling hyper-threading can degrade the performance of the target computer.

Depending upon your configuration, you can also make the following target computer BIOS settings:

- **Configure boot order** — Set the boot order for the target computer BIOS. You can start the target computer using the following methods:

  - Boot floppy disk
  - CD/DVD bootable ROM
  - Flash drive
  - Removable hard drive
  - Dedicated network boot
  - Bootable hard drive

  Configure your target computer BIOS to use your preferred boot order.

- **Enable multicore processor support** — If your target computer includes a multicore processor (see "Multicore Processors"), you can configure the Simulink Real-Time software to take advantage of the individual cores. See "Multicore Processor Configuration" and "How Simulink Solves Parallel and Multicore Processing Problems".

To take advantage of a multicore processor, you must disable hyper-threading in the target computer BIOS.

If the target computer has only a single-core processor, you cannot use the multicore capabilities of the Simulink Real-Time software.

- **Enable USB communication** — If you are using USB communications, enable USB ports for the target computer.
- **Enable RS-232 communication** — If you are using RS-232 communications, enable COM ports for both development and target computers. Through the BIOS, verify that COM1 has a base address of `3F8` and an IRQ of `4`. Verify that COM2 has a base address of `2F8` and an IRQ of `3`. These values are the default base address values. Do not change these values.

Note: RS-232 communication type will be removed in a future release. Use Ethernet instead.

Multicore Processors

The Simulink Real-Time software can run on target computer hardware equipped with a 32-bit or 64-bit x86 compatible CPU (386 or higher). It can allocate and manage up to 32 CPUs, including single core and multicore processors. For example, Simulink Real-Time multicore is supported on the following platforms:

- Intel Core™ 2 Duo processor
- Intel Core 2 Quad processor
- Intel Core i5 processor
- Intel Core i7 processor

You can check for multicore CPU support in the target computer BIOS.

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 runtime. The Simulink Real-Time software is distributed on a DVD or as a file you download from the Web.

Verify that you have met the following requirements:

- 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".
- You must have Microsoft .NET Framework 4.0 installed on the development computer. The installation software will print an error message if the framework is not installed. See "Installing the Microsoft .NET Framework".
- You must have a Microsoft C compiler installed on the development computer. See "Command-Line C Compiler Configuration".

For more information about the file structure after installation, including the location of examples, see "Files on the Development Computer".

The next task is one of "Ethernet Link Setup" or "Serial Link Setup".

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 Web site.

Simulink Real-Time includes standalone mode. With standalone mode, you can start the target computer from a device other than a floppy disk or CD/DVD and deploy standalone real-time applications separate from the development computer.

Files on the Development Computer

Files are located in the:

- MATLAB working folder — Simulink models (`model`), Simulink Real-Time applications (`model.dlm`)

  Select a working folder outside the MATLAB root. See "Setting an Initial Working Folder".
- Simulink Coder Build folder — The Simulink Coder C code files (`model.c`, `model.h`) are in a subfolder called `modelname_xpc_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

Setting an Initial Working Folder

Set your initial MATLAB working folder outside the MATLAB root folder. The default MATLAB root folder is `c:\matlab`. If your MATLAB working folder is below or inside the MATLAB root, files created by Simulink and Simulink Coder are mixed with the MATLAB folders.

You can use the `cd` command to temporarily set your working folder, or set your initial working folder in the MATLAB shortcut:

1 Right-click the MATLAB desktop icon.
2 Click **Properties**. In the **Start in** text box, enter the folder path that you want MATLAB to use initially. Make sure you choose a folder outside the MATLAB root folder.
3 Click **OK**, and then start MATLAB.

To check your working folder, in the MATLAB Command Window, type`pwd`.

Installing the Microsoft .NET Framework

To install Microsoft .NET Framework from the Simulink Real-Time installation folder:

1 In the Command Window, type:

   ```
   cd([matlabroot '\toolbox\rtw\targets\xpc\xpc\bin\dotnetfx'])
   ```
2 In the Current Folder tab, click dotNetFx40_Full_x86_x64.exe.

Ethernet Link Setup

This topic describes using Ethernet links to connect the development computer and target computer. For a serial link, see "Serial Link Setup".

Ethernet link
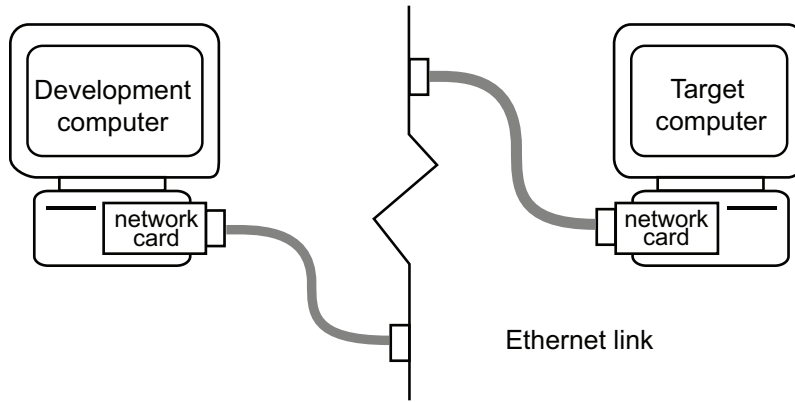
Use one of these procedures as required:

- "PCI Bus Ethernet Setup"
- "USB-to-Ethernet Setup"

The next task is "Target Computer Settings".

PCI Bus Ethernet Setup

If your target computer has a PCI bus, use an Ethernet card for the PCI bus. The PCI bus has a faster data transfer rate than the other bus types.

Follow these procedures:

1 "PCI Bus Ethernet Hardware"
2 "PCI Bus Ethernet Settings"

The next task is "Target Computer Settings".

PCI Bus Ethernet Hardware

To install the PCI bus Ethernet card:

1 Acquire a supported PCI bus Ethernet card.

   For the most current Ethernet link requirements, see `https://www.mathworks.com/products/simulink-real-time/supported/xpc-target-supported-ethernet-chipsets.pdf`.

   If you want to start the target computer from the network, the Ethernet adapter must be compatible with the **Preboot eXecution Environment (PXE)** specification.

2 Turn off your target computer.

3 If the target computer already has an unsupported Ethernet card, remove the card.

4 Plug the supported Ethernet card into a free PCI bus slot.

5 Assign a static IP address to the target computer Ethernet card.

   Although the target computer Ethernet card must have a static IP address, the development computer network adapter card can have a Dynamic Host Configuration Protocol (DHCP) address and can be accessed from the network. When using the product with TCP/IP, you must configure the DHCP server to reserve static IP addresses to prevent these addresses from being assigned to other systems.

6 Connect your target computer Ethernet card to your LAN using an unshielded twisted-pair (UTP) cable.

   You can directly connect your computers using a crossover UTP cable with RJ45 connectors. Both computers must have static IP addresses. If the development computer has a second network adapter card, that card can have a DHCP address.

The next task is "PCI Bus Ethernet Settings".

PCI Bus Ethernet Settings

After you install the PCI bus Ethernet card, before you can build and download a real-time application, you must specify the environment properties for the development and target computers.

1   Ask your system administrator for the following information for your target computer:

   - IP address
   - Subnet mask address
   - Port number (optional)
   - Gateway (optional)

2   In the MATLAB Command Window, type `slrtexplr`.

3   In the **Targets** pane, expand the target computer node.

4   On the toolbar, click the Target Properties icon .

   - To add a node representing another target computer, in the **Targets** pane, click the Add Target icon .
   - To remove a node representing a target computer, right-click the node and select **Remove**.

5   In the **Target Properties** pane, click **Host-to-Target communication**.

6   Select **Communication type** `TCP/IP`.

7   Set **IP address** to the IP address for your target computer (for example, `10.10.10.15`).

8   Set **Subnet mask** to the subnet mask address of your LAN (for example, `255.255.255.0`).

9   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).

10 Set **Gateway** (optional) to the gateway required to access the target computer, if a gateway is required. 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, you might not need to change this setting. If you communicate from a development computer located in a LAN different from your target computer (especially via the Internet), you must define a gateway and enter its IP address in this box.

11 Select **Bus type** PCI.

12 For **Target driver** Auto (default). If you need to select a particular driver, see `http://www.mathworks.com/products/xpctarget/supported-hardware/xPC_Target_Supported_Ethernet_Chipsets.pdf` and select the required driver from the list.

For **Target driver** Auto, the software determines the target computer TCP/IP driver from the card installed on the target computer. If a supported Ethernet card is not installed in your target computer, the software returns an error.

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

14 Press **Enter**, then click the Save icon on the toolbar.

The Simulink Real-Time Explorer window looks like this figure.
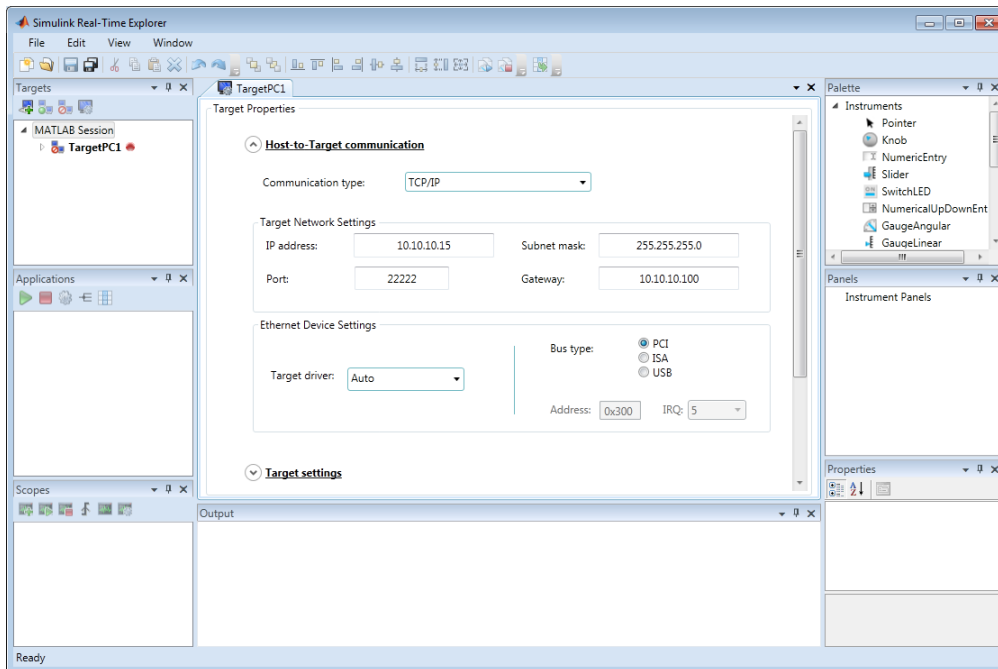
Repeat this procedure as required for each target computer.

The next task is "Target Computer Settings".

USB-to-Ethernet Setup

If the target computer has a USB 2.0 port but does not have a PCI or ISA Ethernet card, use a USB-to-Ethernet adapter.

Follow these procedures:

1 "USB-to-Ethernet Hardware"
2 "USB-to-Ethernet Settings"

The next task is "Target Computer Settings".

USB-to-Ethernet Hardware

To install the USB-to-Ethernet adapter:

1 Acquire a supported USB-to-Ethernet adapter.

For the most current Ethernet link requirements, see `https://www.mathworks.com/products/simulink-real-time/supported/xpc-target-supported-ethernet-chipsets.pdf`.

If you want to start the target computer from the network, the Ethernet adapter must be compatible with the **Preboot eXecution Environment (PXE)** specification.

2 Turn off your target computer.

3 Plug an Ethernet-to-USB adapter into the USB port on the target.

4 Connect the Ethernet-to-USB adapter to your LAN using an unshielded twisted-pair (UTP) cable.

5 Assign a static IP address to the target computer USB-to-Ethernet adapter.

Although the target computer Ethernet adapter must have a static IP address, the development computer network adapter can have a Dynamic Host Configuration Protocol (DHCP) address and can be accessed from the network. When using the product with TCP/IP, you must configure the DHCP server to reserve static IP addresses to prevent these addresses from being assigned to other systems.

6 Connect your target computer Ethernet adapter to your LAN using an unshielded twisted-pair (UTP) cable.

You can directly connect your computers using a crossover UTP cable with RJ45 connectors. Both computers must have static IP addresses. If the development computer has a second network adapter, that adapter can have a DHCP address.

Do not connect the development computer USB port to the target computer USB port using a USB cable. A USB-to-Ethernet adapter plugged into the target computer USB port behaves like an Ethernet card installed on the target computer.

The next task is "USB-to-Ethernet Settings".

USB-to-Ethernet Settings

After you have installed the USB-to-Ethernet adapter, before you can build and download a real-time application, you must specify the environment properties for the development and target computers.

1. Ask your system administrator for the following information for your target computer:

   - IP address
   - Subnet mask address
   - Port number (optional)
   - Gateway (optional)

2. In the MATLAB Command Window, type `slrtexplr`.

3. In the **Targets** pane, expand the target computer node.

4. On the toolbar, click the Target Properties icon ![icon].

   - To add a node representing another target computer, in the **Targets** pane, click the Add Target icon ![icon].
   - To remove a node representing a target computer, right-click the node and select **Remove**.

5. In the **Target Properties** pane, click **Host-to-Target communication**.

6. Select **Communication type** `TCP/IP`.

7. Set **IP address** to the IP address for your target computer (for example, `10.10.10.15`).

8. Set **Subnet mask** to the subnet mask address of your LAN (for example, `255.255.255.0`).

9. 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).

10. Set **Gateway** (optional) to the gateway required to access the target computer, if a gateway is required. 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, you might not need to change this setting. If you communicate from a development computer located in a LAN different from your target computer (especially via the Internet), you must define a gateway and enter its IP address in this box.

11 Select **Bus type** USB.

12 For **Target driver**, select one of USBAX772, USBAX172, or Auto.

If **Target driver** is Auto, the software sets the driver to USBAX772, the driver most commonly used.

13 Press **Enter**, then click the Save icon on the toolbar.

The Simulink Real-Time Explorer window looks like this figure.
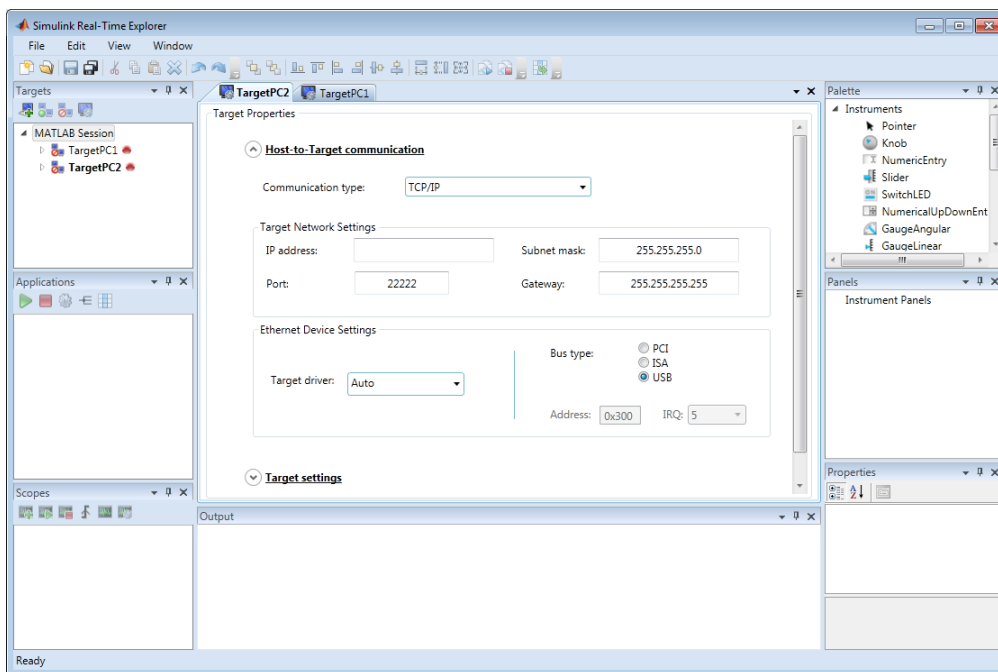
Repeat this procedure as required for each target computer.

The next task is "Target Computer Settings".

ISA Bus Ethernet Setup

Your target computer might not have an available PCI bus slot or USB 2.0 port. In these cases, use an Ethernet card for an ISA bus.

Note: Links between development and target computers using ISA bus Ethernet adapters will be removed in a future release. Use PCI or USB bus adapters instead.

---

1 "ISA Bus Ethernet Hardware"
2 "ISA Bus Ethernet Settings"

The next task is "Target Computer Settings".

ISA Bus Ethernet Hardware

To install an ISA bus Ethernet card:

1   Acquire a supported ISA bus Ethernet card.

   For the most current Ethernet links requirements, see `https://www.mathworks.com/products/simulink-real-time/supported/xpc-target-supported-ethernet-chipsets.pdf`.

   If you want to start the target computer from the network, the Ethernet adapter must be compatible with the **Preboot eXecution Environment (PXE)** specification.

2   Turn off your target computer.

3   On your ISA bus card, assign an IRQ and I/O-port base address by moving the jumpers or switches on the card. Write down these settings, because you must enter them in Simulink Real-Time Explorer.

   Set the IRQ line to `5` and the I/O-port base address to around `0x300`. If one of these hardware settings leads to a conflict in your target computer, select another IRQ or I/O-port base address.

   If your ISA bus card does not contain jumpers to set the IRQ line and the base address, after installation use the utility on the installation disk supplied with your card to manually assign the IRQ line and base address.

   If you use an Ethernet card for an ISA bus within a target computer that has a PCI bus, after installation you must reserve the chosen IRQ line number for the Ethernet card in the PCI BIOS. To set up the PCI BIOS, refer to your BIOS setup documentation.

   Do not configure the card as a PnP-ISA device.

4   If the target computer already has an unsupported Ethernet card, remove the card. Plug the compatible network card into a free ISA bus slot.

5   Assign a static IP address to the target computer Ethernet card.

   Although the target computer Ethernet card must have a static IP address, the development computer network adapter card can have a Dynamic Host Configuration Protocol (DHCP) address and can be accessed from the network. When

using the product with TCP/IP, you must configure the DHCP server to reserve static IP addresses to prevent these addresses from being assigned to other systems.

6 Connect your target computer Ethernet card to your LAN using an unshielded twisted-pair (UTP) cable.

You can directly connect your computers using a crossover UTP cable with RJ45 connectors. Both computers must have static IP addresses. If the development computer has a second network adapter card, that card can have a DHCP address.

The next task is "ISA Bus Ethernet Settings".

ISA Bus Ethernet Settings

After you have installed the ISA bus Ethernet card, before you can build and download a real-time application, you must specify the environment properties for the development and target computers.

1. Ask your system administrator for the following information for your target computer:

   - IP address
   - Subnet mask address
   - Port number (optional)
   - Gateway (optional)

2. In the MATLAB Command Window, type `slrtexplr`.

3. In the **Targets** pane, expand the target computer node.

4. On the toolbar, click the Target Properties icon .

   - To add a node representing another target computer, in the **Targets** pane, click the Add Target icon .
   - To remove a node representing a target computer, right-click the node and select **Remove**.

5. In the **Target Properties** pane, click **Host-to-Target communication**.

6. Select **Communication type** `TCP/IP`.

7. Set **IP address** to the IP address for your target computer (for example, `10.10.10.15`).

8. Set **Subnet mask** to the subnet mask address of your LAN (for example, `255.255.255.0`).

9. 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).

10. Set **Gateway** (optional) to the gateway required to access the target computer, if a gateway is required. 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, you might not need to change this setting. If you communicate from a development computer located in a LAN different from your target computer (especially via the Internet), you must define a gateway and enter its IP address in this box.

11 Select **Bus type** ISA.

12 For **Target driver**, select one of NE2000 or SMC91C9X.

   Target driver Auto is not supported for **Bus type** ISA.

13 Set **Address** and **IRQ** to values that correspond with the jumper settings or ROM settings on your ISA bus Ethernet card.

14
   Press **Enter**, then click the Save icon 💾 on the toolbar.

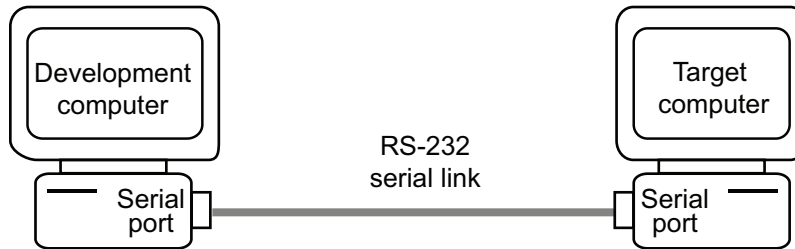The Simulink Real-Time Explorer window looks like this figure.
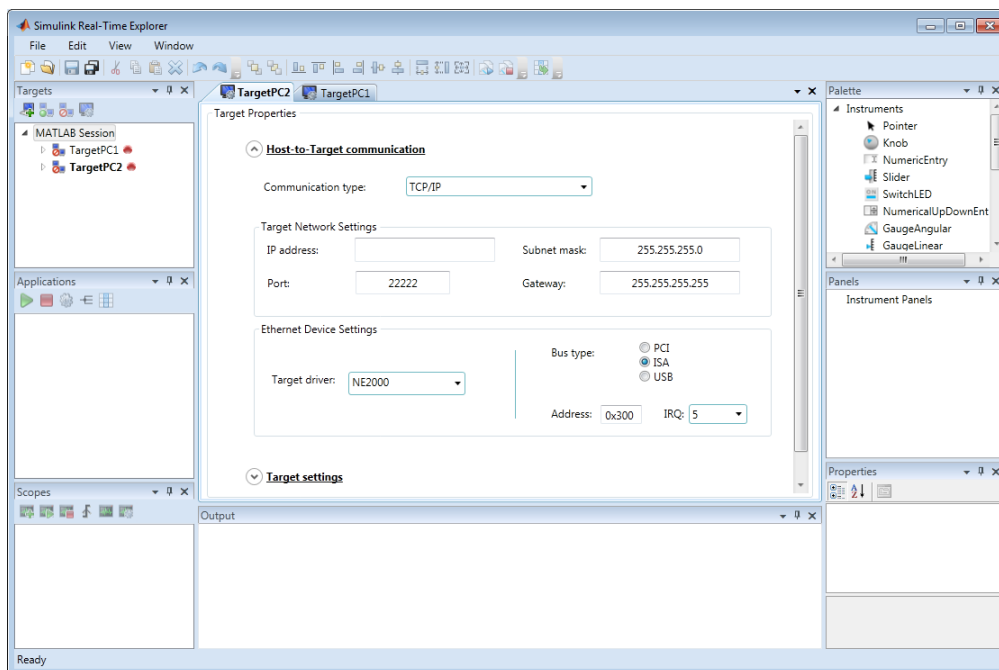
Repeat this procedure as required for each target computer.

The next task is "Target Computer Settings".

Serial Link Setup

This topic describes using serial links between the development and target computers. For Ethernet links, see "Ethernet Link Setup".

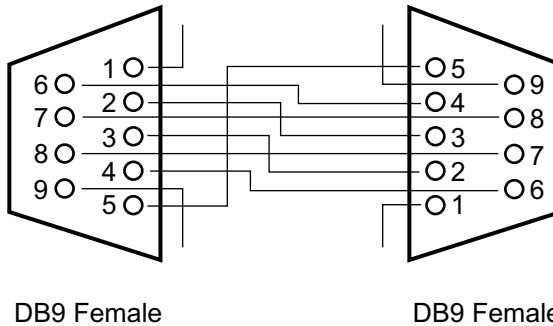**Note:** RS-232 communication link will be removed in a future release. Use Ethernet instead.

- "Serial Link Hardware"
- "Serial Link Settings"

The next task is "Target Computer Settings".

Serial Link Hardware

Before you can use a serial link, you must install the following RS-232 hardware:

1 Acquire a null modem cable:

DB9 Female   DB9 Female

2 Connect the development and target computers with the null modem cable, using either the COM1 or COM2 port.

Note which port is in use on the development computer. You need to set the development computer port in the environment property settings.

The next task is "Serial Link Settings".

Serial Link Settings

After you have installed the serial link hardware, before you can build and download a real-time application, you must specify the environment properties for the development and target computers.

Note:

- Do not use host scopes and a scope viewer on the development computer to acquire and display large blocks of data. The slowness of the RS-232 connection causes large delays for large blocks of data.

- **Boot mode** type Network is not supported when a serial link is used.

---

1 In the MATLAB Command Window, type slrtexplr.

2 In the **Targets** pane, expand the target computer node.

3 On the toolbar , click the Target Properties icon 🖳.

   - To add a node representing another target computer, in the **Targets** pane, click the Add Target icon 🖥️.
   - To remove a node representing a target computer, right-click the node and select **Remove**.

4 In the **Target Properties** pane, click **Host-to-Target communication**.

5 Select **Communication type** RS-232.

   The pane changes to display only parameters pertinent to a serial link.

6 For **Host port**, select one of COM1 or COM2. The default is COM1. Simulink Real-Time selects the target computer port automatically.

7 Select a **Baud rate** as high as possible. The default is 115200.

   A baud rate less than 38400 can cause communication failures.

8 Press **Enter**, then click the Save icon 💾 on the toolbar.

The Simulink Real-Time Explorer window looks like this figure.
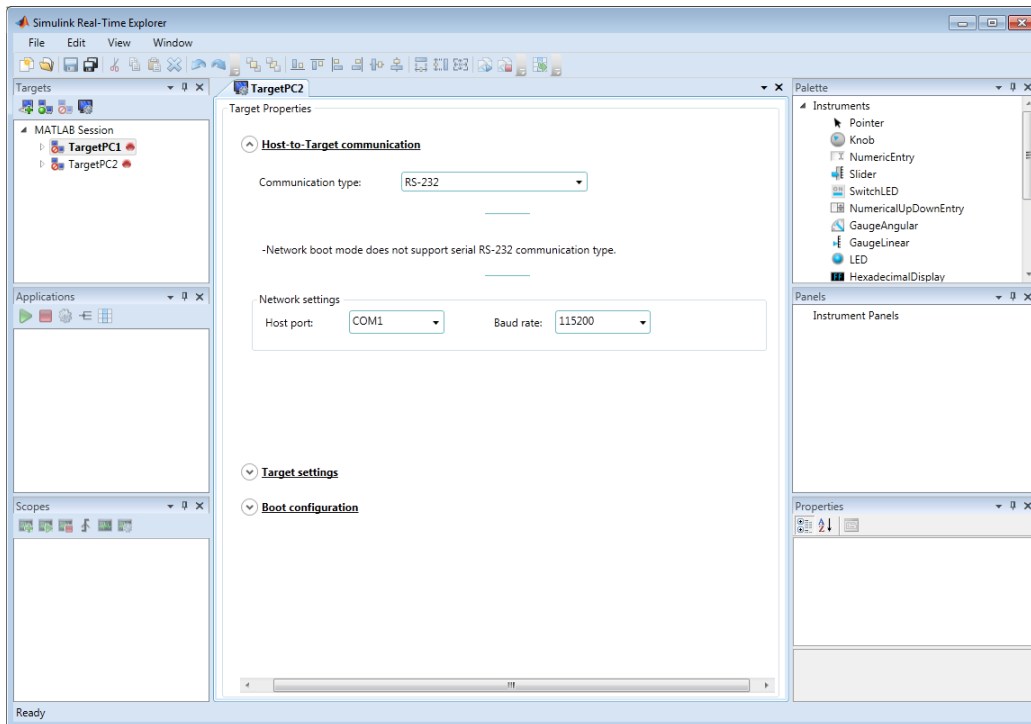
Repeat this procedure as required for each target computer.

The next task is "Target Computer Settings".

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 In the MATLAB Command Window, type `slrtexplr`.

2 In the **Targets** pane, expand the target computer node.

3 On the toolbar, click the Properties icon .

4 In the **Target Properties** workspace, click **Target settings**.

5 Execute the following target computer settings as required:

- **Graphics mode** — Leave this check box selected if you want to display information, such as a target scope, in graphic format. If you want to display information as text, clear it.

  To use the full features of a target scope, install a keyboard on the target computer.

- **USB Support** — Leave this check box selected if you want to use a USB port on the target computer; for example, to connect a USB mouse. Otherwise, clear it.

- **Secondary IDE** — Select this check box only if you want to use the disks connected to a secondary IDE controller.

- **Multicore CPU** — Select this check box only if your target computer has multicore processors that the kernel can use for background tasks.

- **Target is a 386/486** — Select this check box if your target computer has a 386 or 486 compatible processor.

  If your target computer has a Pentium or higher compatible processor, selecting this check box slows the performance of your target computer.

- **RAM size** — Leave **Auto** selected if you want the real-time application to read the target computer BIOS and determine the amount of memory up to a maximum of 2 GB. Select **Manual** if the real-time application cannot read the BIOS. You must enter the amount of memory, in megabytes, into the **Size(MB)** text box up to a maximum of 2 GB.

  The **RAM size** setting defines the total amount of installed RAM in the target computer used for the kernel, real-time application, data logging, and other functions that use the heap.

The **Size(MB)** text box is enabled for **RAM size Manual** only.

The Simulink Real-Time kernel can use only 2 GB of memory.

- **Model size** — Select a value from the list that reserves the required amount of memory on the target computer for the real-time application. The remaining memory is used by the kernel and by the heap for data logging.

  Selecting too high a value leaves less memory for data logging. Selecting too low a value does not reserve enough memory for the real-time application and creates an error. You can approximate the size of the real-time application by the size of the DLM file produced by the build process.
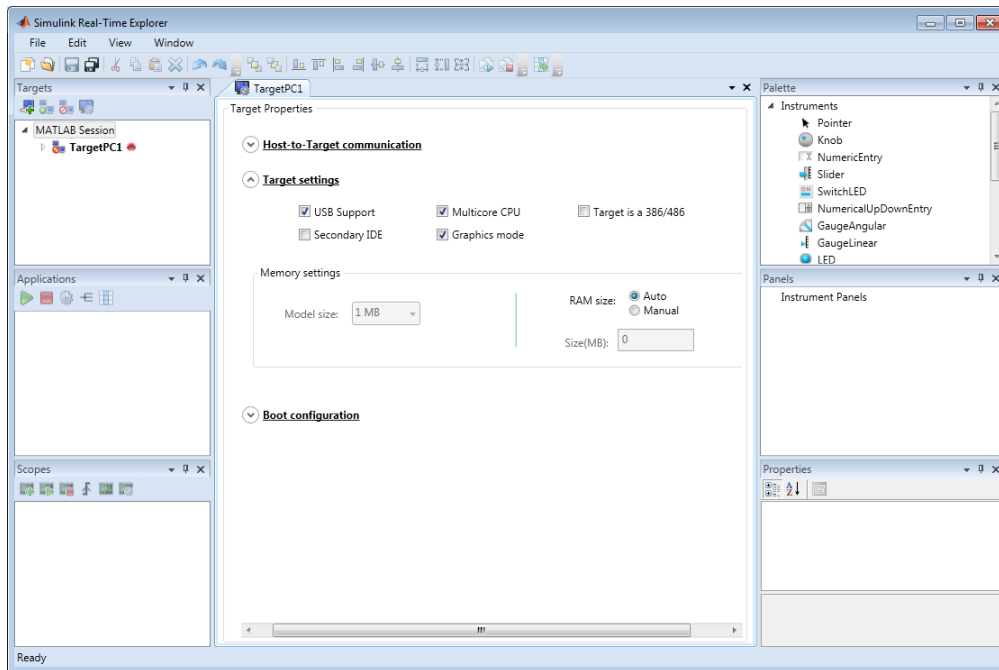
  Setting **Model size** is enabled for **Boot mode** `Stand Alone` only.

  You can use only **Model size** values `1 MB` and `4 MB`.

6 Press **Enter**, then click the Save icon ![save icon] on the toolbar.

The Simulink Real-Time Explorer window looks like this figure.

Repeat this procedure as required for each target computer.

The next task is "Target Boot Methods".

Target Boot Methods

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

Speedgoat systems come with `DOS Loader` software preinstalled. You can set up the `DOS Loader` boot method on your host or configure another boot method. See your Speedgoat system documentation or follow the link from "Speedgoat Real-Time Target Machines" for further information.

1 Before creating a boot kernel, perform "Kernel Creation Prechecks".

2 Select one of the following methods:

- "Network Boot Method"
- "CD/DVD Boot Method"
- "DOS Loader Boot Method"
- "Removable Disk Boot Method"
- "Standalone Boot Method"

3 For boot methods other than `Stand Alone`, perform "Run Confidence Test on Configuration".

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

Kernel Creation Prechecks

Configure your Simulink Real-Time system before creating the target boot kernel. At a minimum, do the following:

1   Check the physical connections between the development and target computers. If you are using TCP/IP, these are Ethernet connections that may pass through a LAN.

2   Check your target computer BIOS settings (see "BIOS Settings").

3   Check that you have write permission for your current working folder.

4   In the MATLAB Command Window, type `slrtexplr`.

5   In the **Targets** pane, expand the target computer node.

6   On the toolbar , click the Target Properties icon .

7   In the **Target Properties** pane, click **Host-to-Target communication**.

8   Check the link settings. As required, see:

   · "Ethernet Link Setup"
   · "Serial Link Setup"

9   In the **Target Properties** pane, click **Boot configuration**.

10 Check that **Boot mode** is set to the required value.

11 Press **Enter**, then click the Save icon on the toolbar.

Repeat this procedure as required for each target computer.

CD/DVD Boot Method

After you have configured the target computer environment parameters, you can use Simulink Real-Time Explorer to create a boot CD or DVD that loads and runs the Simulink Real-Time kernel. Your development computer must run under one of the following Windows systems:

- Microsoft Windows 7
- Microsoft Windows Vista™
- Microsoft Windows XP Service Pack 2 or 3 with Image Mastering API v2.0 (IMAPIv2.0), available at http://support.microsoft.com/kb/KB932716.

To create a bootable CD/DVD using MATLAB language, see "Command-Line CD/DVD Boot Method".

To create a boot CD or DVD:

1 In the MATLAB Command Window, type `slrtexplr`.

2 In the **Targets** pane, expand the target computer node.

3 On the toolbar , click the Target Properties icon .

4 Select **Boot configuration** and set **Boot mode** to `CD`.

5 Click **Create boot disk**.

6 In the **Select CD-ROM Drive** list, select the CD/DVD read/write drive.

7 When prompted, insert an empty CD or DVD in the development computer CD/DVD read/write drive.

8 Click **Burn Disk**. A progress bar is displayed.

9 When the write operation has finished, click **OK** in the status dialog box.

10 Remove the CD or DVD from the drive.

11 Insert the bootable CD/DVD into your target computer CD/DVD drive and restart the target computer.

12 Press **Enter**, then click the Save icon on the toolbar.

Repeat this procedure as required for each target computer.

The next task is "Run Confidence Test on Configuration".

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. You do not need a boot CD or a removable boot drive.

There are the following limitations:

- Do not use the network boot method on a corporate or nondedicated network. Doing so might interfere with dynamic host configuration protocol (DHCP) servers and cause problems with the network.
- Your Ethernet card must be compatible with the Preboot eXecution Environment (PXE) specification.
- If the target and development computers communicate by a serial link (RS-232), you cannot start the target computer across the network.
- If **Boot mode** `Stand Alone` is enabled, you cannot start the target computer across the network.

Before you start, establish an Ethernet connection between the development and target computers, using the procedure in "Ethernet Link Setup".

1 In the MATLAB Command Window, type `slrtexplr`.

2 In the **Targets** pane, expand the target computer node.

3 On the toolbar , click the Target Properties icon .

4 Select **Boot configuration** and set **Boot mode** to `Network`.

5 If required, click **Reset**.

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

6 To enter the MAC address of the target computer manually, click **Reset**. 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.

8   Turn on the target computer.

9   Enter the target computer BIOS and set up the target computer for a LAN or network boot.

If the BIOS allows you to change the boot order, consider setting the boot order so that the floppy disk, CD, or USB drive come before the LAN option. Doing so allows you to start the target computer from a network kernel even if the target computer does not have a kernel boot disk or removable drive.

10   Restart the target computer.

11   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.
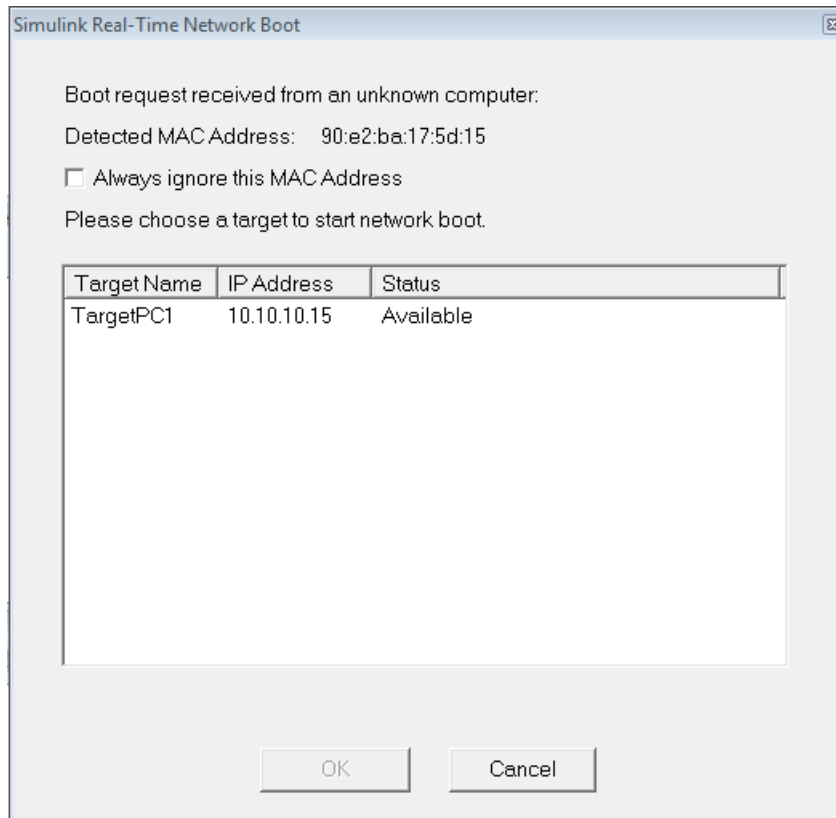
a Select the target computer name for the physical target computer.

b Click **OK**.

The development computer network boot server displays a pop-up from the system tray, indicating that the boot server is being downloaded to the target computer.

If you click **Cancel** instead of selecting a target computer name, the kernel ignores the start request for 90 seconds the next time you try to start the target computer across the network.

12
Press **Enter**, then click the Save icon on the toolbar.

The Simulink Real-Time Explorer window looks like this figure.
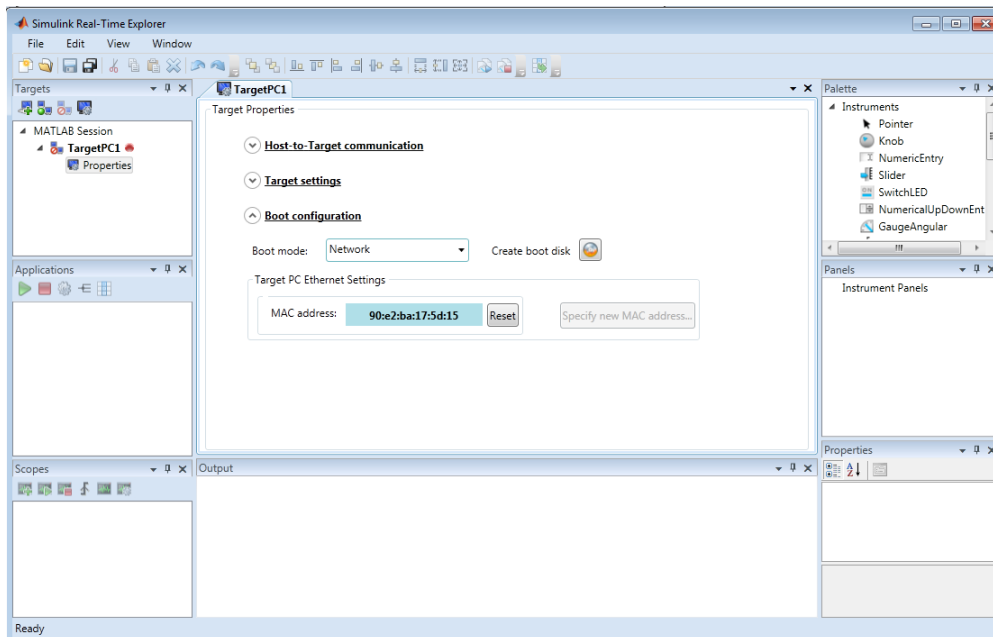
Repeat this procedure as required for each target computer.

Tip To configure network boot using MATLAB language, see "Command-Line Network Boot Method".

The next task is "Run Confidence Test on Configuration".

Removable Disk Boot Method

After you have configured the target computer environment parameters, you can use a target boot floppy disk, removable drive, or USB flash drive to load and run the Simulink Real-Time kernel. This topic describes using Simulink Real-Time Explorer to create a removable boot disk.

If you are creating a removable boot disk from a USB flash drive, before performing this procedure, you must create a bootable partition on the drive. See "Create a Bootable Partition".

To create a removable boot disk using MATLAB language, see "Command-Line Removable Disk Boot Method".

To create a removable boot disk:

1 In the MATLAB Command Window, type slrtexplr.

2 In the **Targets** pane, expand the target computer node.

3 On the toolbar , click the Target Properties icon .

4 Select **Boot configuration** and set **Boot mode** to Removable Disk.

5 If you are creating a removable boot disk from a USB drive, insert the USB drive in the development computer USB port. Wait for it to be recognized.

6 Click **Create boot disk**.

7 In the **Removable Disk Selector** list, select the drive.

8 If required, insert an empty removable disk in the development computer drive and click **OK**. A progress bar is displayed.

9 When the write operation has finished, remove the removable disk from the drive or USB port.

10 Insert the removable boot disk into your target computer drive or USB port. Restart the target computer.

11 Press **Enter**, then click the Save icon on the toolbar.

Repeat this procedure as required for each target computer.

The next task is "Run Confidence Test on Configuration".

Create a Bootable Partition

Before you create a removable boot drive from a USB flash drive, you must create a bootable partition on the drive:

1 On the development computer, open a DOS command window.

2 In the DOS command window, type:

```
diskpart
```

3 At the diskpart prompt, type:

```
list disk
```

Note the disk numbers of the existing development computer disks.

4 Insert the flash drive. Wait for the development computer to recognize the drive. Note the drive device name.

5 At the diskpart prompt, type:

```
list disk
```

Note the new disk number *N*. This number is the disk number of the drive.

6 Type:

```
select disk N
```

7 **Caution** The `clean` command deletes all of the data from disk *N*. Select the correct disk, or you might delete all of the data from your development computer.

Type:

```
clean
create partition primary
select partition 1
active
format fs=fat32 quick
exit
```

8 In the development computer task bar, click **Safely remove hardware and eject media** and select the flash drive name.

9 Remove the drive from the development computer.

DOS Loader Boot Method

`DOS Loader` mode allows you to start the Simulink Real-Time kernel on a target computer from a fixed or removable device with DOS boot capability, such as a hard disk or flash memory. After starting the target computer, you can download your application from the development computer over a serial or Ethernet link between the development and target computers.

Speedgoat systems come with `DOS Loader` software preinstalled. You can set up the `DOS Loader` boot method on your host or configure another boot method. See your Speedgoat system documentation or follow the link from "Speedgoat Real-Time Target Machines" for further information.

To run in `DOS Loader` mode, the target computer boot device must provide a minimal DOS environment complying with certain restrictions. For details, see "Create a DOS System Disk" and "DOS Loader Mode Restrictions".

To create a DOS Loader boot disk using MATLAB language, see "Command-Line DOS Loader Boot Method".

To create and use DOS Loader files:

1 In the MATLAB Command Window, type `slrtexplr`.
2 In the **Targets** pane, expand the target computer node.
3 On the toolbar , click the Target Properties icon ![icon].
4 Select **Boot configuration** and set **Boot mode** to `DOS Loader`.
5 In the **Location** field, enter or browse to the folder where you want to create the DOS Loader boot files. This location can be a local folder on the development computer or a removable storage device that you use to start the target computer. By default, the folder is the current working folder.
6 Click **OK**. A progress bar appears to indicate execution.

   This operation creates the following boot files in the specified location:
       autoexec.bat
       xpcboot.com
       *.rtb
7 If you create boot files on a local hard disk, copy these files to a floppy disk, CD/DVD, or other removable storage media.

8 Transfer the boot files to your target computer or insert the removable media containing the boot files into the target computer drive or USB port.

9 Verify that `autoexec.bat` file is on the DOS boot path (typically the root folder).

10 In the BIOS of the target computer, select the required boot device.

11 Start the target computer.

When the target computer starts, it loads DOS, which executes the `autoexec.bat` file. This file starts the Simulink Real-Time kernel (*.rtb). The target computer then awaits commands from the development computer.

12 Press **Enter**, then click the Save icon on the toolbar.

Repeat this procedure as required for each target computer.

The next task is "Run Confidence Test on Configuration".

Create a DOS System Disk

To use the DOS-based modes such as DOS Loader and Stand Alone, you need a minimal DOS system on the target computer boot device.

Simulink Real-Time software does not include a DOS license. You must obtain a valid DOS license for your target computer.

To create a DOS system disk:

1 Acquire a copy of a DOS system, such as FreeDOS, MS-DOS®, PC DOS, or Caldera OpenDOS.

2 Use the following DOS command to copy the DOS system files and command interpreter from *drive1* to the boot device, *drive2*:

   sys *drive1 drive2*

3 Copy additional DOS utilities to the boot disk, including:

   • DOS editor — to edit text files on the target computer.
   • format program — to format a hard disk or flash memory.
   • fdisk program — to create partitions.
   • sys program — to transfer a DOS system onto another drive, such as the hard disk drive.

   A config.sys file is not required.

4 Transfer additional files to the disk as required.

DOS Loader Mode Restrictions

To use DOS Loader mode, the target computer DOS environment must comply with the following restrictions:

- The CPU must execute in real mode.
- While loaded in memory, the DOS partition must not overlap the address range of a real-time application.

To satisfy these restrictions:

- Avoid additional memory managers, such as `emm386` or `qemm`.
- Avoid utilities that attempt to load in high memory (for example, `himem.sys`).
- Avoid including memory manager entries in the `autoexec.bat` file.
- Avoid using a `config.sys` file.

Standalone Boot Method

You can configure the Simulink Real-Time software to run as a standalone application. For more information on **Boot mode** `Stand Alone`, see "Standalone Mode".

To run in `Stand Alone` mode, the target computer and its DOS environment must meet specific requirements:

- "Standalone Target Computer Prechecks"
- "Standalone Mode Restrictions"

To set up your target computer for `Stand Alone` mode:

1 "Standalone Settings"

2 "Standalone Real-Time Application Build"

3 "Standalone Real-Time Application Transfer"

4 "Standalone Real-Time Application Boot Configuration"

The next task is restarting the target computer and testing your application in `Stand Alone` mode. The confidence test is not intended for stand alone execution. You must create and execute your own confidence test for `Stand Alone` mode.

Standalone Mode

In **Boot mode** `Stand Alone` you can deploy control systems, DSP applications, and other systems on target computer hardware for use in production applications using target computer hardware. Typically these production applications are found in systems where production quantities are low to moderate.

In `Stand Alone` mode, you can bundle the kernel and real-time application into one entity on the target computer, independent of the development computer. You can configure a target computer to automatically start executing your embedded application for continuous operation each time the system is started. If the target computer has a keyboard, you can control the real-time application using the target computer command-line interface. You can also control it from the host using custom GUIs or the Web browser interface. You can deploy host-side GUIs developed with the Simulink Real-Time C and .NET APIs on Microsoft Windows development computers, without installing MATLAB software.

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

Use this mode to load the target computer with both the Simulink Real-Time kernel and a real-time application. 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 automatically starts the real-time application that you loaded with the kernel. `Stand Alone` mode eliminates the need for a development computer and allows you to deploy real-time applications on target computers.

Regardless of the mode, you initially start your target computer with DOS from a boot device. DOS starts the Simulink Real-Time kernel. After the Simulink Real-Time kernel starts, DOS is no longer available on the target computer. To make DOS available, you must restart the target computer with DOS without starting the Simulink Real-Time kernel.

Simulink Real-Time standalone mode requires a boot device with DOS installed. Otherwise, it does not have specific requirements as to the type of boot device. DOS software and license are not included with Simulink Real-Time. If DOS is not installed, you must first start the target computer from a Simulink Real-Time boot disk or network boot image, and only then download and run the real-time application.

In the following instances, you might want to use standalone mode. If you deploy the target computer in a small or rugged environment, you might have one of these situations:

- The target computer does not have a removable drive.
- After setting up the target system, you must remove the target computer removable drive.
- 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 Simulink 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. `Stand Alone` applications are automatically set to continue running for an infinite time or until the target computer is turned off.

Standalone Mode Restrictions

To use `Stand Alone` mode, the target computer DOS environment must comply with the following restrictions:

- The CPU must execute in real mode.
- While loaded in memory, the DOS partition must not overlap the address range of a real-time application.

To satisfy these restrictions:

- Avoid additional memory managers, such as `emm386` or `qemm`.
- Avoid utilities that attempt to load in high memory (for example, `himem.sys`).
- Avoid including memory manager entries in the `autoexec.bat` file.
- Avoid using a `config.sys` file.

There are additional restrictions on target and scope settings:

- In the Simulink Real-Time Explorer **Target settings** pane, you can use only **Model size** values `1 MB` and `4 MB`. Setting **Model size** is enabled for **Boot mode** `Stand Alone` only.
- If you want to use real-time Scope blocks to display or record output:
  - You must use only `Target` or `File` type blocks.
  - You must select the **Start scope when application starts** check box in the block parameters dialog box.

Standalone Target Computer Prechecks

In `Stand Alone` mode, the real-time application is combined with the kernel on a hard drive or flash memory. The kernel and real-time application start together on the target computer. The development computer does not need to be connected to the target computer, but it must be able to start from DOS.

For each target computer, verify that the system meets the following requirements:

- The target computer hard drive must be a serial ATA (SATA) or parallel ATA (PATA)/ Integrated Device Electronics (IDE) drive.
- The target computer must not cable select the hard drive.
- The target computer BIOS must be able to detect the hard drive.
- The hard drive must be formatted with one of these file systems:

    - FAT-12
    - FAT-16
    - FAT-32

- You must install a supported version of DOS on the target computer hard drive. If required, create a standard DOS boot device from a CD ROM, 3.5–inch floppy drive, flash drive, or hard drive. See "Create a DOS System Disk".
- You must configure the Ethernet link. This communication transfers the kernel/real-time application files built on the development computer to the target computer.

Standalone Settings

Use Simulink Real-Time Explorer to set the kernel environment properties. When you are done, you can create a standalone kernel/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   In the MATLAB Command Window, type slrtexplr.
2   In the **Targets** pane, expand the target computer node.
3   On the toolbar , click the Target Properties icon .
4   Select **Boot configuration** and set **Boot mode** to Stand Alone.
5   Press **Enter**, then click the Save icon on the toolbar.

Repeat this procedure as required for each target computer.

The next task is "Standalone Real-Time Application Build".

Standalone Real-Time Application Build

After you set 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 combined kernel and real-time application with utility files.

1   In the MATLAB window, type the name of your Simulink model, for example, `xpc_osc3`. A Simulink window opens displaying the model.

2   From the **Code** menu, click **C/C++ Code** > **Build Model**.

The Simulink Coder and Simulink Real-Time software create a folder, `xpc_osc3_xpc_emb`, containing the following files:

- `autoexec.bat` — Contains Simulink Real-Time-specific code that calls the `xpcboot.com` executable to start the Simulink Real-Time kernel (the `*.rtb` file).

- `xpcboot.com` — Loads and executes the `*.rtb` file. This static file is part of the Simulink Real-Time software.

- `xpc_osc3.rtb` — Contains the Simulink Real-Time kernel and the real-time application. This image also contains applicable options, such as serial or Ethernet communications and the COM port or IP address of the target computer.

You do not need a `config.sys` file to start the kernel using `autoexec.bat` and `xpcboot.com`.

The **Model size** parameter gives an upper bound on the size of the compiled real-time application (DLM). It is only enabled in `Stand Alone` mode. You can use only **Model size** values `1 MB` and `4 MB`. If the DLM is larger than **Model size**, the build software generates an error.

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

The next task is "Standalone Real-Time Application Transfer".

Standalone Real-Time Application Transfer

After building the kernel/real-time application on a development computer, transfer the files to a target computer using the Simulink Real-Time FTP functions:

1 Start the target computer from a target computer boot disk.
2 In the MATLAB Command Window, change to the folder that contains the kernel/real-time application files.
3 In the MATLAB Command Window, create a Simulink Real-Time FTP object:

```
fsys = SimulinkRealTime.fileSystem
```

4 Use the FTP object to create the folder C:\xpcfiles on the target computer:

```
mkdir(fsys, 'xpcfiles')
```

5 Move to the xpcfiles folder:

```
cd(fsys, 'xpcfiles')
```

6 Copy these files to the xpcfiles folder:

```
copyFileToTarget(fsys, 'autoexec.bat')
copyFileToTarget(fsys, 'xpcboot.com')
copyFileToTarget(fsys, 'xpc_osc3.rtb')
```

7 Remove the boot disk from the target computer.

Repeat this procedure as required for each target computer.

The next task is "Standalone Real-Time Application Boot Configuration".

Standalone Real-Time Application Boot Configuration

After transferring the files to a target computer, configure the target computer to run the real-time application automatically upon start.

1 Restart the target computer and bring up the DOS prompt.

   If the target computer that you want to start in `Stand Alone` mode was previously started from the network boot image, select the `Stand Alone` check box to disable the network boot capability.

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 Edit the target computer file `C:\autoexec.bat` to include the following lines:

```
cd C:\xpcfiles
autoexec
```

   Adding these commands to `C:\autoexec.bat` directs the system to execute the `autoexec.bat` file located in `C:\xpcfiles`.

4 Restart the target computer.

   The target computer executes the following sequence of calls:

   a `C:\autoexec.bat`
   b `C:\xpcfiles\autoexec.bat`
   c `C:\xpcfiles\xpcboot.com`
   d `C:\xpcfiles\<application>.rtb`

Repeat this procedure as required for each target computer.

Note: Do not confuse `C:\xpcfiles\autoexec.bat` with `C:\autoexec.bat`. `C:\xpcfiles\autoexec.bat` includes the command `xpcboot.com`, which starts the Simulink Real-Time kernel and the standalone application. `C:\autoexec.bat` includes the files you want the system to execute when the system starts.

Continue by restarting the target computer and testing your application in `Stand Alone` mode. The confidence test is not intended for standalone execution. You must create and execute your own confidence test for `Stand Alone` mode.

Run Confidence Test on Configuration

This topic describes how to use a script to 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 TCP/IP for linking the development and target computers and for starting the target computer.

1 Create a network boot image and restart the computer. See "Network Boot Method".

2 In the MATLAB Current Folder window, select a folder outside the MATLAB root folder.

   You cannot save build files within the MATLAB root folder because Simulink Coder does not allow this action. If you select a current folder within the MATLAB tree, the Simulink Real-Time test procedure fails when trying to build a real-time application.

3 In the MATLAB Command Window, type `slrttest`.

   MATLAB runs the test script for the default target computer and displays messages indicating whether the test passed or failed. If you use RS-232 communication, the first test is skipped.

```
### Simulink Real-Time Test Suite
### Host-Target interface is: TCP/IP (Ethernet)
### 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 xpcosc 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 one of the following:

- "Run Benchmark Test on Configuration".
- "Application and Driver Scripts".

Run Benchmark Test on Configuration

After running the confidence test, `slrttest`, you can use `slrtbench` to determine how your target computer performs relative to comparable computers.

You must be using TCP/IP for linking the development and target computers and for starting the target computer.

1   Create a network boot image and restart the computer. See "Network Boot Method".

2   In the MATLAB Current Folder window, select a folder outside the MATLAB root folder.

    Simulink Coder does not allow build files to be saved within the MATLAB root folder. If you select a current folder within the MATLAB tree, the Simulink Real-Time test procedure fails when trying to build a real-time application.

3   In the MATLAB Command Window, type `slrtbench this`.

    `slrtbench this` displays two plots containing benchmark results for the five benchmarks run on your target computer, compared with the representative benchmark results for other target computers.

    • For each target computer tested, the first plot lists the minimal achievable sample time for the benchmarks, in microseconds.

    • The second plot contains a bar graph of representative computers, ranked by relative performance.

    The results are labeled by CPU type, CPU clock rate, and the compiler used to compile the application. The entry for your target computer is highlighted.

    For more benchmarking options, see `slrtbench`.

4   Evaluate the results.

    • If your target computer has performance comparable to other target computers, you are ready to build and download a model to the target computer.

    • If your target computer has performance significantly lower than other target computers, consider some of the steps outlined in "How Can I Improve Run-Time Performance?".

Continue by trying some examples in "Application and Driver Scripts".

3

Basic Workflows

Rapid Control Prototyping

Use rapid control prototyping to test a design with a minimal hardware plant model.

1 **Create a Simulink or Stateflow model.**

   Create block diagrams in Simulink using simple drag-and-drop operations. Set model and block parameters to discretized signal and fixed-step solver, and select a sample time compatible with the fixed-step solver.

2 **Simulate the model as a free-running 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 hardware 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.

3 **Configure the development and target computers.**

   Configure the communication method between the development and target computers.

   Configure the development and target computers using:

   · "Simulink Real-Time Explorer" on page 1-20

   · "MATLAB Command-Line Interface" on page 1-22

4 **Prepare the model for real-time execution.**

   Set the model parameters to values compatible with real-time execution:

   · Discretized signal

   · Fixed-step solver

   · Sample time compatible with the fixed-step solver

   Add Simulink Real-Time I/O blocks representing your I/O boards. If you have a custom I/O board, create a custom driver block representing the board. See "Add I/O Blocks to Simulink Model".

Add real-time Scope blocks to visualize the simulation results. See "Add Simulink Real-Time Scope Block".

5   Configure the build environment.

Configure the build environment, including Simulink Coder options, Simulink Real-Time build options, and C compiler options, to create a real-time application that runs on the target computer.

At this point, you may configure the real-time application to run using Simulink Real-Time standalone mode.

6   Connect to external hardware.

Install I/O boards in the target computer and connect the I/O boards to the hardware against which you want to execute the real-time application.

7   Restart the target computer.

Restart the target computer with the Simulink Real-Time real-time kernel using:

· Target computer hardware boot control.

· "MATLAB Command-Line Interface" on page 1-22

8   Build and download the real-time application.

Build and download the real-time application using:

· "Simulink Real-Time Explorer" on page 1-20 (load and unload only)

· "MATLAB Command-Line Interface" on page 1-22

· "Simulink External Mode Interface" on page 1-24

· "Target Computer Command-Line Interface" on page 1-26

· "Web Browser Interface" on page 1-29

9   Execute the real-time application.

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

Execute using:

- "Simulink Real-Time Explorer" on page 1-20
- "MATLAB Command-Line Interface" on page 1-22
- "Simulink External Mode Interface" on page 1-24
- "Target Computer Command-Line Interface" on page 1-26
- "Web Browser Interface" on page 1-29

10 Visualize signals.

Create Simulink Real-Time scopes and use them to acquire and display signal data from the real-time application.

Scopes created by Simulink 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.

Visualize signals using:

- "Simulink Real-Time Explorer" on page 1-20
- "MATLAB Command-Line Interface" on page 1-22
- "Simulink External Mode Interface" on page 1-24
- "Simulink with Simulink Real-Time Blocks" on page 1-25
- "Target Computer Command-Line Interface" on page 1-26
- "Web Browser Interface" on page 1-29

11 Tune parameters.

Tune observable model parameters such as time delays, input and output amplitudes, and input and output frequencies.

Note: Simulink Real-Time does not support tuning parameters of complex or multiword data types.

Tune parameters using:

- "Simulink Real-Time Explorer" on page 1-20
- "MATLAB Command-Line Interface" on page 1-22
- "Simulink External Mode Interface" on page 1-24
- "Target Computer Command-Line Interface" on page 1-26
- "Web Browser Interface" on page 1-29

12 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 for later use.

Hardware-in-the-Loop Simulation

Hardware-in-the-loop (HIL) simulation builds on the test harness and simulation results of rapid control prototyping to verify a physical prototype of the product. HIL simulation is especially valuable to:

- Substitute for unavailable parts of the system.
- Test the system for safety and performance.
- Minimize expensive downtime for the rest of the system.
- Test operation and failure conditions that are difficult to replicate.

With HIL simulation, you perform one or more of the following tasks:

- **Model the plant.** — Use Simulink and Simulink Real-Time to model the plant for testing the physical prototype.
- **Execute a graphical model.** — Add blocks to a Simulink user interface model with Simulink Real-Time To blocks. See "Simulink Real-Time Interface Blocks to Simulink Models".
- **Write a regression test harness.** — Extend the MATLAB regression tests written for rapid control prototyping to cover functionality over the full parameter range.
- **Write a graphical test harness.** — Use one of the APIs (.NET, C) to write a graphical test harness suitable for probing real-time application behavior from the development computer. As part of this effort, you can use MATLAB Coder to translate MATLAB regression test scripts into C for integration into the test environment.
- **Connect to a physical prototype.**—Modify the system model to replace the Simulink Real-Time design with an I/O board connected to the physical prototype.
- **Program a physical prototype.** — Use HDL Coder to generate FPGA code to program the physical prototype.
- **Configure an embedded application.** — Configure the real-time application to run using Simulink Real-Time standalone mode.
- **Execute the application and display the results.** — Use the test harness to execute the embedded real-time application, display and log the results, and tune model parameters.

As a test engineer, you can build on this workflow to create repeatable product tests to support volume manufacturing. For example, you can extend and categorize the MATLAB regression tests into smoke, go/no-go, and diagnostic tests, and use MATLAB

Coder to translate MATLAB regression test scripts into C for integration into a production test environment.

4

Tutorial and Examples

This topic describes Simulink Real-Time functionality with a simple Simulink model without I/O blocks. You can try these procedures whether or not you have I/O hardware 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

You must have the following configuration:

- Single PCI bus target computer
- TCP/IP communication type
- Network target boot method

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

1 "Configure Link Between Development and Target Computers"
2 "Configure Target Settings"
3 "Configure Boot Configuration"
4 "Run the Confidence Test"

The next task is "Create and Run a Real-Time Application".

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.

Tip The **Target Network Settings** values are representative only. Consult your network administrator for actual values. For more on network configuration, see "Ethernet Link Setup".

1　In the MATLAB Command Window, type slrtexplr.

2　In the **Targets** pane, expand the target computer node.

3　On the toolbar, click the Properties icon .

4　In the **Target Properties** workspace, click **Host-to-Target communication**.

5　Select **Communication type** TCP/IP.

6　Under **Target Network Settings**, set values such as the following.

- **IP address**: 10.10.10.15
- **Port**: 22222
- **Subnet mask**: 255.255.255.0
- **Gateway**: 10.10.10.10

7　Under **Ethernet Device Settings**, set the following values:

- **Target driver**: Auto
- **Bus type**: **PCI**

8　Press **Enter**, then click the Save icon on the toolbar.

The dialog box looks like this figure.
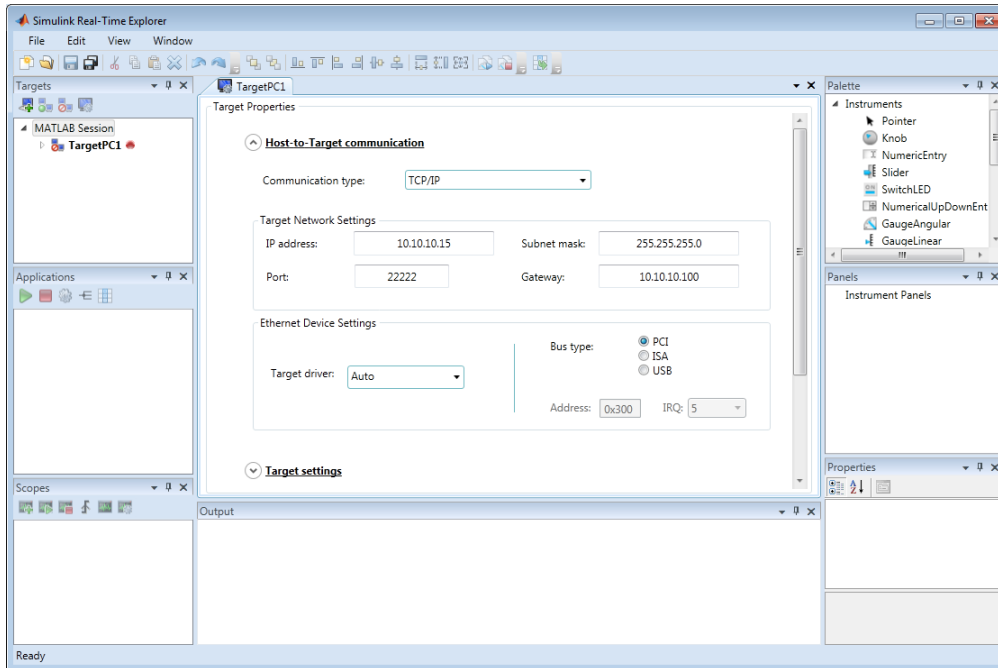
**Note:** Settings **Address** and **IRQ** are enabled for **Bus type ISA** only.

The next task is "Configure Target Settings".

Configure Target Settings

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

1 In the MATLAB Command Window, type `slrtexplr`.

2 In the **Targets** pane, expand the target computer node.

3 On the toolbar, click the Properties icon [icon].

4 In the **Target Properties** workspace, click **Target settings**.

5 Select **USB Support**, **Multicore CPU**, and **Graphics mode**.

6 Clear **Secondary IDE** and **Target is a 386/486**.

7 Under **Memory settings**, select **Ram size Auto**.

8 Press **Enter**, then click the Save icon [icon] on the toolbar.

The dialog box looks like this figure.
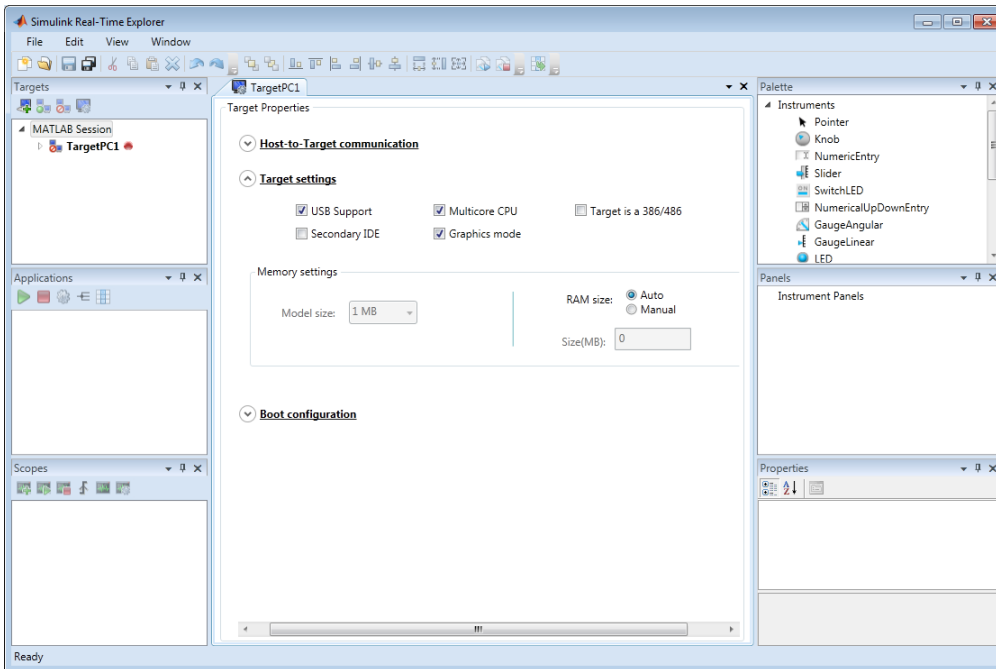
Note:

- Setting **Size(MB)** is enabled for **RAM size Manual** only.

- Setting **Model size** is enabled for **Boot mode** Stand Alone only.

The next task is "Configure Boot Configuration".

Configure Boot Configuration

To run a Simulink Real-Time model on a target computer, you must configure the target boot method. For more on boot methods, see "Target Boot Methods".

1. In the MATLAB Command Window, type `slrtexplr`.

2. In the **Targets** pane, expand the target computer node.

3. On the toolbar, click the Properties icon .

4. In the **Target Properties** workspace, click **Boot Configuration**.

5. Select **Boot mode** `Network`.

6. Click **Create boot disk**.

7. Press **Enter**, then click the Save icon on the toolbar.

The dialog box looks like this figure.
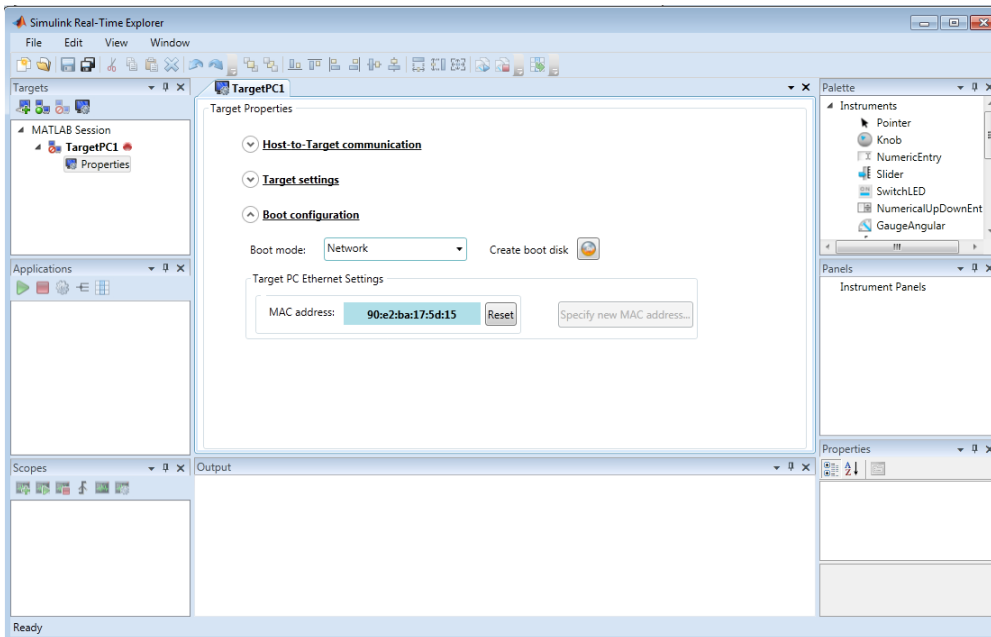
The next task is "Run the Confidence Test".

Run the Confidence Test

Validate the setup and configuration by running the confidence test.

1 Create a network boot image and restart the computer. See "Network Boot Method".

2 In the MATLAB Current Folder window, select a folder outside the MATLAB root folder.

You cannot save build files within the MATLAB root folder because Simulink Coder does not allow this action. If you select a current folder within the MATLAB tree, the Simulink Real-Time test procedure fails when trying to build a real-time application.

3 In the MATLAB Command Window, type `slrttest`.

MATLAB runs the test script for the default target computer and displays messages indicating whether the test passed or failed. If you use RS-232 communication, the first test is skipped.

```
### Simulink Real-Time Test Suite
### Host-Target interface is: TCP/IP (Ethernet)
### 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 xpcosc 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".

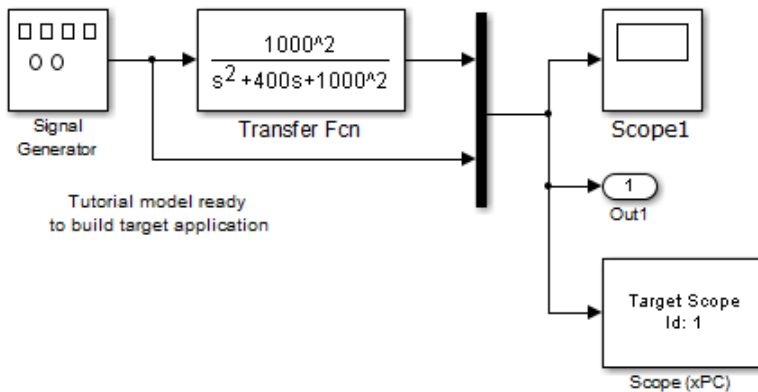The next task is "Create and Run a Real-Time Application".

Create and Run a Real-Time Application

Simulink model `xpc_osc1` is a non-real-time model of a damped oscillator. This tutorial describes how to transform `xpc_osc1` into a Simulink Real-Time model configured to build as a real-time application.

To access the example folder, type:

```
addpath(fullfile(matlabroot,'help','toolbox','xpc',
       'examples'));
```

The final model, `xpc_osc3`, looks like this figure.

The tutorial shows how to use the Simulink Real-Time software, the Simulink Coder environment, and a third-party compiler. It describes how to create a real-time application from `xpc_osc3`, build and download it, and execute it on the target computer.

The following links guide you to the four parts of the tutorial.

1 "Transform Simulink Model to Real-Time Application"
2 "Start Target Hardware"
3 "Build and Download Real-Time Application"
4 "Execute Real-Time Application Using Simulink External Mode"

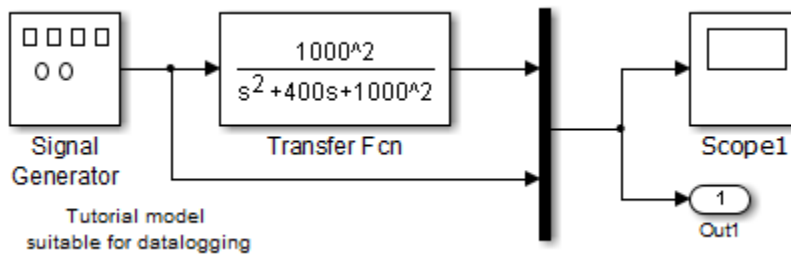The next task is "Configure and Control a Real-Time Application".

Create and Run Simulink Model

The example model, xpc_osc1, is a non-real-time model of a damped oscillator. It includes a Transfer Fcn block and a Signal Generator block. To visualize and log signals while simulating, it also includes standard Simulink Scope and Out1 (Outport) blocks.

To access the example folder, type:

```
addpath(fullfile(matlabroot,'help','toolbox','xpc',
        'examples'));
```

The final model looks like this figure.

The following links guide you to the six parts of the tutorial.

1 "Create Simulink Model"
2 "Configure Signal Generator"
3 "Configure Transfer Function"
4 "Configure Scope Block"
5 "Configure Simulation Parameters"
6 "Simulate Using Simulink"

The next task is "Transform Simulink Model to Real-Time Application".
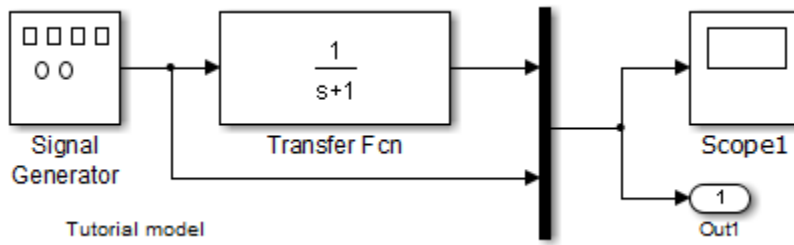
Create Simulink Model

The example model, xpc_osc, is a non-real-time model of a damped oscillator.

1 In the MATLAB Command Window, type simulink.

2 In the Simulink library window, from the **File** menu, point to **New**, and then click **Model**.

3 In the left pane, double-click **Simulink**, and then click **Continuous**.

4 Click and drag the Transfer Fcn block to the Simulink model window.

5 Click **Sources**, and add a Signal Generator block.

6 Click **Sinks**, and add a Scope block and an Out1 (Outport) block.

7 Click **Signal Routing**, and add a Mux block.

8 Make the following block-to-block connections:

- Signal Generator output to Transfer Fcn input.
- Signal Generator output to Mux input.
- Transfer Fcn output to Mux input.
- Mux output to Scope input.
- Mux output to Out1 input.

9 As required, double-click signals and enter names.

If you provide a name for a signal in the Signal name property of the Signal Properties dialog box, that name appears in the target computer GUI scope graph after you build and download the model to the target computer. By default, if you do not enter a name for the signal in this dialog box, the scope graph displays the signal identifier rather than a name.

10 From the **File** menu, click **Save As** and enter a file name. For example, enter xpc_osc and then click **OK**.

The model, xpc_osc, looks like this figure (unconfigured).

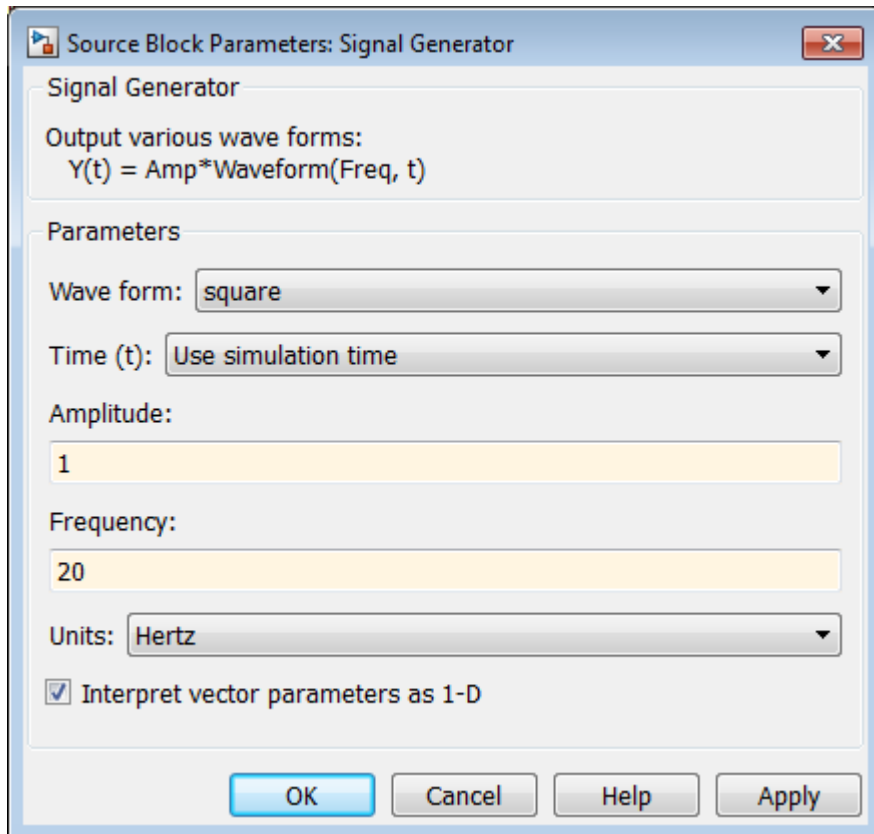Tutorial model

The next task is "Configure Signal Generator".

Configure Signal Generator

The example model, xpc_osc, is a non-real-time model of a damped oscillator.

1   In the MATLAB Command Window, type xpc_osc.

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

2   In the Simulink window, double-click the Signal Generator block.

3   In the Block Parameters dialog box, from the **Wave form** list, select square.

4   In the **Amplitude** text box, enter 1.

5   In the **Frequency** text box, enter 20.

6   From the **Units** list, select Hertz.

7 Do one of the following:

- Click **Apply** to apply the changes to the model and leave the dialog box open.
- Click **OK** to apply the changes to the model and close the dialog box.

8 From the **File** menu, click **Save As** and enter a file name. For example, enter xpc_osc1, and then click **OK**.
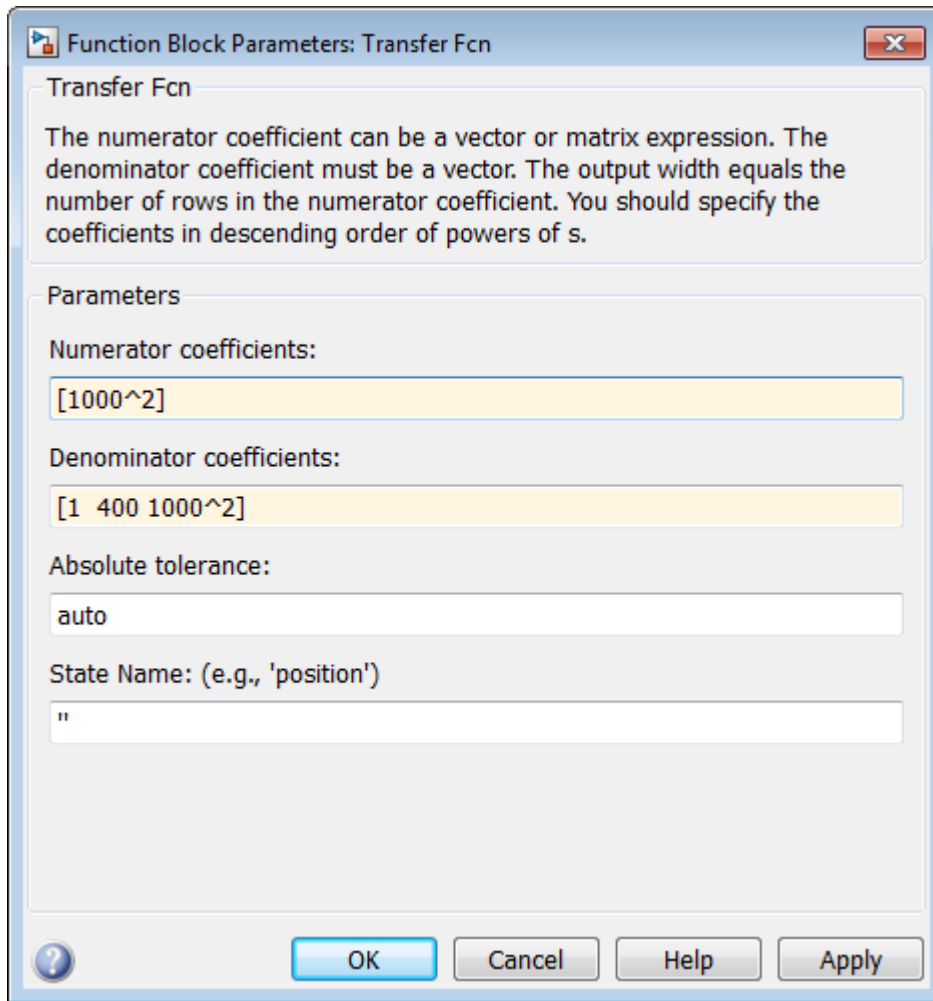
The next task is "Configure Transfer Function".

Configure Transfer Function

The example model, xpc_osc1, is a non-real-time model of a damped oscillator.

1   In the MATLAB Command Window, type xpc_osc1.

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

2   In the Simulink window, double-click the Transfer Fcn block. The Block Parameters dialog box opens.

3   In the **Numerator** text box, enter [1000^2].

4   In the **Denominator** text box, enter[1 400 1000^2].

Function Block Parameters: Transfer Fcn

Transfer Fcn

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.

Parameters

Numerator coefficients:

[1000^2]

Denominator coefficients:

[1 400 1000^2]

Absolute tolerance:

auto

State Name: (e.g., 'position')

''

OK   Cancel   Help   Apply

5 Do one of the following:

- Click **Apply** to apply the changes to the model and leave the dialog box open.
- Click **OK** to apply the changes to the model and close the dialog box.

6 From the **File** menu, click **Save**.

The next task is "Configure Scope Block".

Configure Scope Block

The example model, xpc_osc1, is a non-real-time model of a damped oscillator. You enter or change scope parameters to specify the *x*-axis and *y*-axis in a Scope window. Other properties include the number of graphs in one Scope window and the sample time for models with discrete blocks.

For more information, see "Trace Signals Using Simulink External Mode".

1 In the MATLAB Command Window, type xpc_osc1.

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

2 In the Simulink window, double-click the Scope block.

3
   On the toolbar, click the **Parameters** icon

4 In the Scope window, click the **General** tab. In the **Number of axes** box, enter the number of graphs that you want in one Scope window. For example, enter 1 for a single graph. Do not select the **floating scope** check box.

5 In the **Time range** box, enter the upper value for the time range. For example, enter 0.2 second.

6 From the **Tick labels** list, choose all.

7 From the **Sampling** list, choose Sample time and enter 0 in the text box. Entering 0 indicates that Simulink evaluates this block as a continuous-time block. If the model has discrete blocks, enter the **Fixed step size** you entered in the Configuration Parameters dialog box.

   Your Scope parameters dialog box looks like this figure.
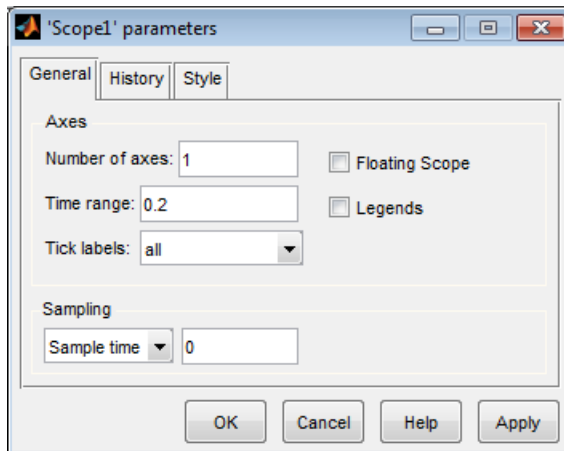
8   Do one of the following:

   · Click **Apply** to apply the changes to the model and leave the dialog box open.

   · Click **OK** to apply the changes to the model and close the dialog box.

9   In the Scope window, point to the *y*-axis and right-click it.

10 From the context menu, click **Axes Properties**. The Scope properties: axis 1 dialog box opens.

11 In the **Y-min** and **Y-max** text boxes, enter the range for the *y*-axis in the Scope window. For example, enter -3 and 3.

   Your Scope axis dialog box looks like this figure.
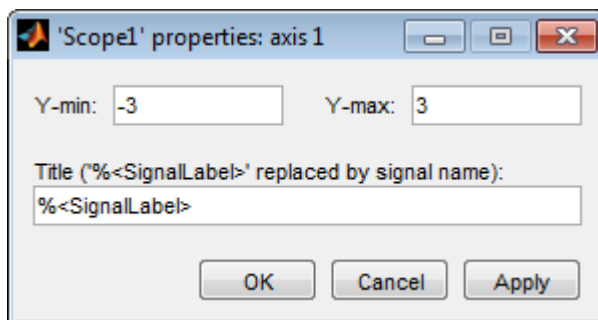
12 Do one of the following:

- Click **Apply** to apply the changes to the model and leave the dialog box open.
- Click **OK** to apply the changes to the model and close the dialog box.

13 From the **File** menu, click **Save**.

The next task is "Configure Simulation Parameters".

Configure Simulation Parameters

The example model, xpc_osc1, is a non-real-time model of a damped oscillator. During a simulation, Simulink saves signal data to MATLAB variables using Outport blocks. The default MATLAB variables are tout, xout, and yout. While running a real-time application, the Simulink Real-Time interface uses these same variables to pass signal data to target object parameters. A target object is a structure in the MATLAB workspace that the Simulink Real-Time software uses to interact with a real-time application. The default target object is tg. The default parameters are Time, tg.States, and tg.Output.

1   In the MATLAB Command Window, type xpc_osc1.

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

2   In the Simulink window, click **Simulation** > **Model Configuration Parameters**.

3   In the Configuration Parameters dialog box, click the **Solver** node.

    Simulink displays the **Solver** pane. The **Simulation section** of this pane defines the initial stop and sample time for the real-time application.

4   In the **Solver options** section, enter 0 seconds in the **Start time** box. In the **Stop time** box, enter an initial stop time. For example, enter 1 second. To change this time after creating the real-time application, change the target object property tg.Stoptime to the new time using the MATLAB command-line interface. To specify an infinite stop time, enter inf.

5   From the **Type** list, select Fixed-step. Simulink Coder does not support variable-step solvers.

6   From the **Solver** list, select a solver. For example, select the general-purpose solver ode4 (Runge-Kutta).

7   In the **Fixed step size** box, enter the sample time for the real-time application. For example, enter 0.00025 seconds (250 microseconds). You can change this value after creating the real-time application.

    If you find that 0.000250 seconds results in overloading the CPU on the target computer, try a larger **Fixed step size**, such as 0.0002 seconds.

    If the model contains both continuous and discrete states, the sample times of the discrete states can only be multiples of **Fixed step size**. If the model contains no continuous states, enter 'auto' in **Fixed step size**. The sample time is taken from the model.

The **Solver** pane looks like this figure.
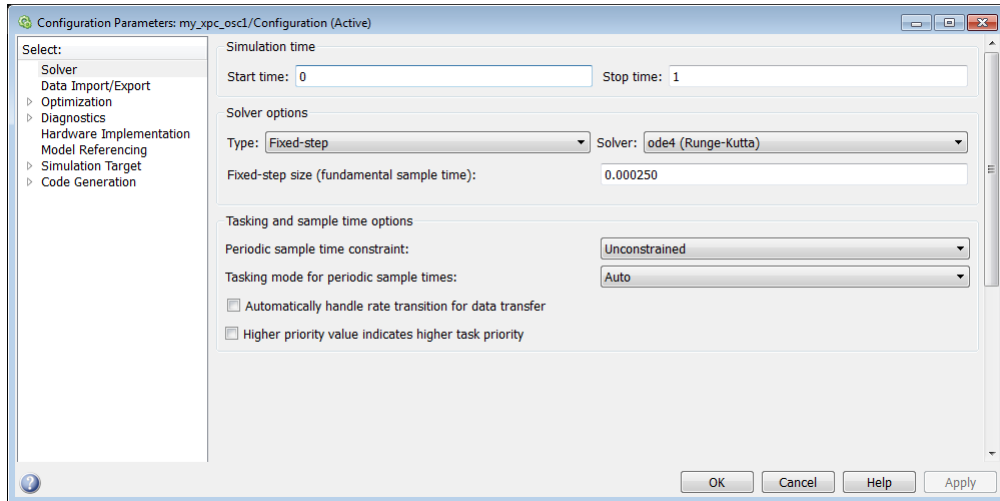

8   Click the **Data Import/Export** node.

The **Data Import/Export** pane opens. This pane defines the model signals logged during a simulation of the model or while running the real-time application.

9   In the **Save to workspace** section of this pane, select the **Time**, **States**, and **Output** check boxes.

When the real-time application is running in `Freerun` mode, data is saved to the variables `tout` and `yout`. When the real-time application is running in `Real-Time` mode, data is saved to the target object properties `TimeLog`, `StateLog`, and `OutLog`. However, you must still select the **Time**, **States**, and **Output** check boxes for data to be logged to the target object properties.

You can improve runtime performance by turning one or more **Save to workspace** check boxes off. See "How Can I Improve Run-Time Performance?".

The **Data Import/Export** pane looks like this figure.
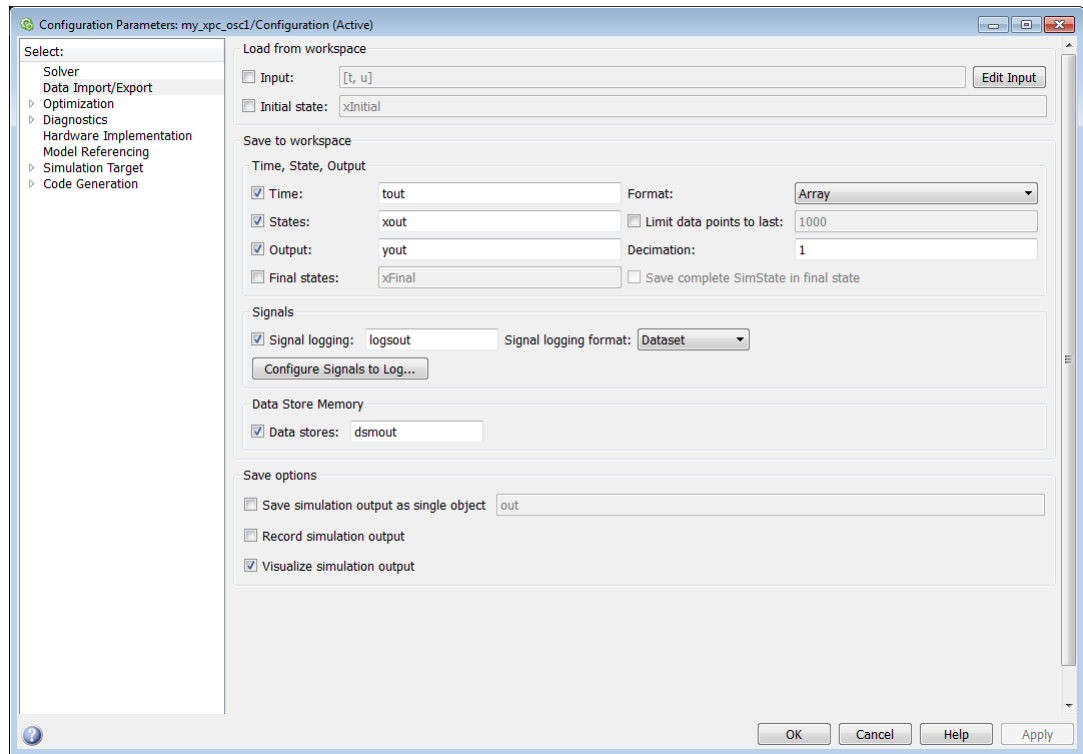
10 Do one of the following:

  - Click **Apply** to apply the changes to the model and leave the dialog box open.
  - Click **OK** to apply the changes to the model and close the dialog box.

11 From the **File** menu, click **Save**.

The next task is "Simulate Using Simulink".

Simulate Using Simulink

The example model, xpc_osc1, is a non-real-time model of a damped oscillator. You use Simulink in normal mode to observe the non-real-time behavior of the model. After you load the Simulink model, you can run a simulation.

**1** In the MATLAB Command Window, type xpc_osc1.

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

2 In the Simulink window, double-click the Scope block.

3 In the Scope window, select **Simulation** > **Mode** > **Normal**.

4

On the toolbar, click the Run icon ▶ to begin simulation.

The Simulink software runs the simulation and plots the signal data in the Scope window.

During the simulation, the Scope window displays the samples for one time range, increases the time offset, and then displays the samples for the next time range.
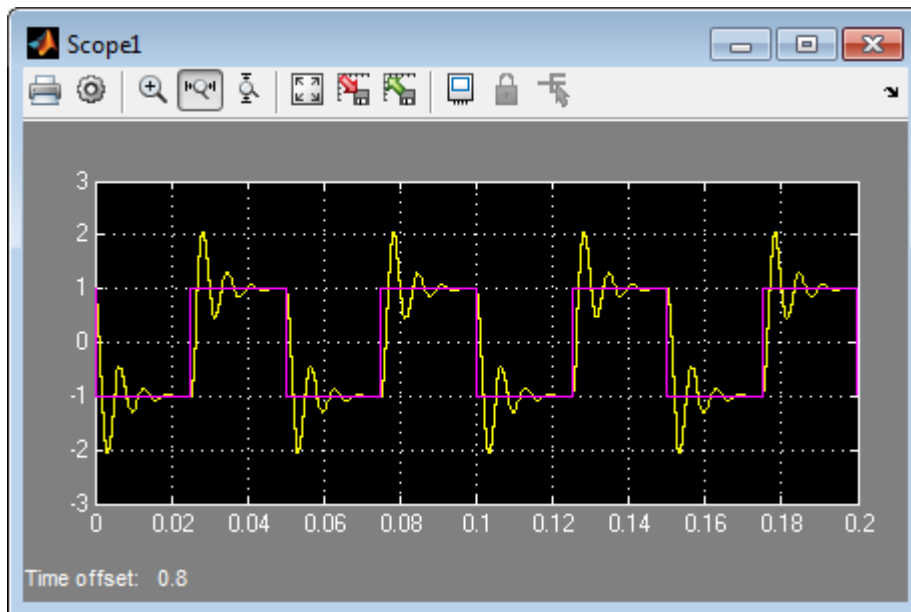
5   You can either let the simulation run to its stop time, or stop the simulation manually. To stop the simulation manually, on the toolbar, click the Stop icon ■.

The next task is "Transform Simulink Model to Real-Time Application".

Transform Simulink Model to Real-Time Application

To run a Simulink model as a real-time application under Simulink Real-Time, you must add and configure the Simulink Real-Time Scope block and set Configuration Parameters for code generation and target execution.

1 "Add Simulink Real-Time Scope Block"

2 "Set Target Scope Block Parameters"

3 "Set Configuration Parameters"

The next task is "Start Target Hardware".

Add Simulink Real-Time Scope Block

The example model, xpc_osc1, is a non-real-time model of a damped oscillator. 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 window, type xpc_osc1.

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

2 In the Simulink window, from the **View** menu, click **Library Browser**.

The Simulink Library Browser window opens.

**3** In the left pane, browse to and double-click node **Simulink Real-Time**.

A list of I/O block categories opens.

4 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.

The model xpc_osc1 looks like this figure.
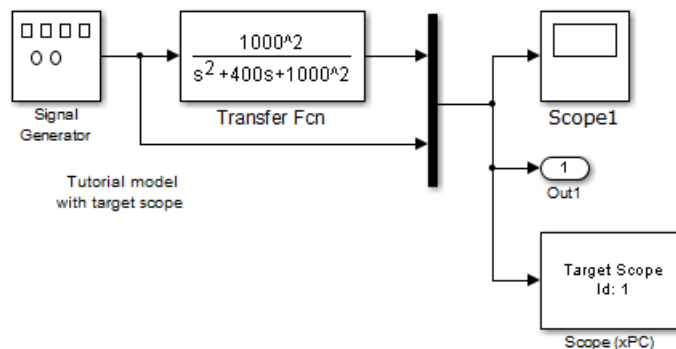
7 From the **File** menu, click **Save As**. Enter a file name. For example, enter xpc_osc2 and then click **OK**.

The next task is "Set Target Scope Block Parameters".

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 Scope parameters, see "Configure Real-Time Host Scope Blocks".

1   In the MATLAB window, type `xpc_osc2`.

    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   Type `[0,0]` in the **Y-axis limits** text box. This value means that display scaling is `auto`.

8   Type `1000` in the **Number of samples** text box. For a **Scope mode** of `Graphical redraw`, this value means that 1000 samples are acquired before the graph is redrawn.

9   Type `0` in the **Number of pre/post samples** text box. This value means that samples are not saved before a trigger or skipped after a trigger.

10 Type `1` in the **Decimation** text box. 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.
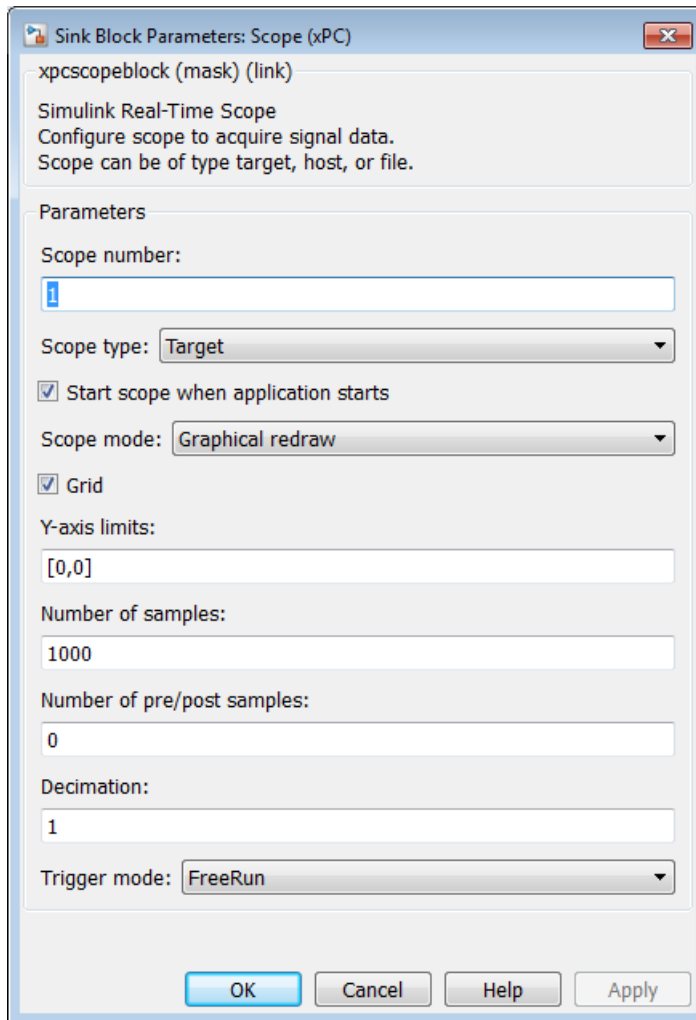
12 Do one of the following:

- Click **Apply** to apply the changes to the model and leave the dialog box open.
- Click **OK** to apply the changes to the model and close the dialog box.

13 From the **File** menu, click **Save**.

The next task is "Set Configuration Parameters".

Set Configuration Parameters

The example model, xpc_osc2, 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 give 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 window, type xpc_osc2.

   MATLAB loads the oscillator model and displays the Simulink block diagram.
2 In the Simulink window, click **Simulation** > **Model Configuration Parameters**.
3 In the Configuration Parameters dialog box, click the **Code Generation** node. The code generation pane opens.
4 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 xpc_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. The code generation pane looks like this figure.
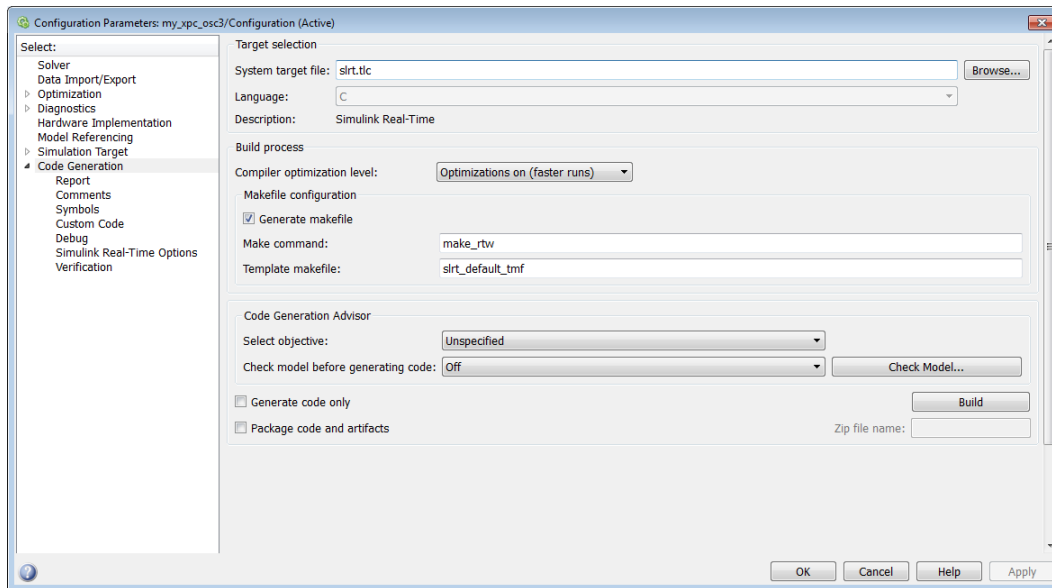
If you have Embedded Coder software, you can build an ERT real-time application. To build an ERT real-time application, in the **Target selection** section, click **Browse** at the **System target file** list. Click `slrtert.tlc`, and then click **OK**.

If you select `slrtert.tlc` without Embedded Coder installed, the build fails.

5 In the left pane, click the **Simulink Real-Time Options** node.

The associated pane is displayed. These are model-level configuration parameters that you can set for the model. For a description of the options on this node, see "Configuration Parameters".

6 From the **Execution mode** list, select either `Real-Time` or `Freerun`.

`Freerun` mode execution is similar to a simulation, but uses the generated code. The target computer runs the real-time application as fast as possible, which may be faster than real-time execution. However, `Freerun` mode execution does not support variable-step solvers.

In addition, multirate models cannot be executed in `Freerun` execution mode. On the **Solver** pane in the Configuration Parameters dialog box, set **Tasking mode for periodic sample times** to `SingleTasking`.

7   From the **Real-time interrupt source** list, select a source. The default value is `Timer`.

8   Select the **Log Task Execution Time** check box to log task execution times to the target object property `tg.TETlog`.

The task execution time is the time in seconds to complete calculations for the model equations and post outputs during each sample interval. If you do not select this box, the average TET value appears as Not a Number (`NaN`).

9   In the **Signal logging buffer size in doubles** box, enter the maximum number of sample points to save before wrapping, for example, 100000. This buffer includes the time, states, outputs, and task execution time logs.

10   In the **Name of Simulink Real-Time object created by build process** box, enter the name of the target object created by the build process. The default target object name is `tg`.

The **Simulink Real-Time Options** pane should now look like the figure shown.
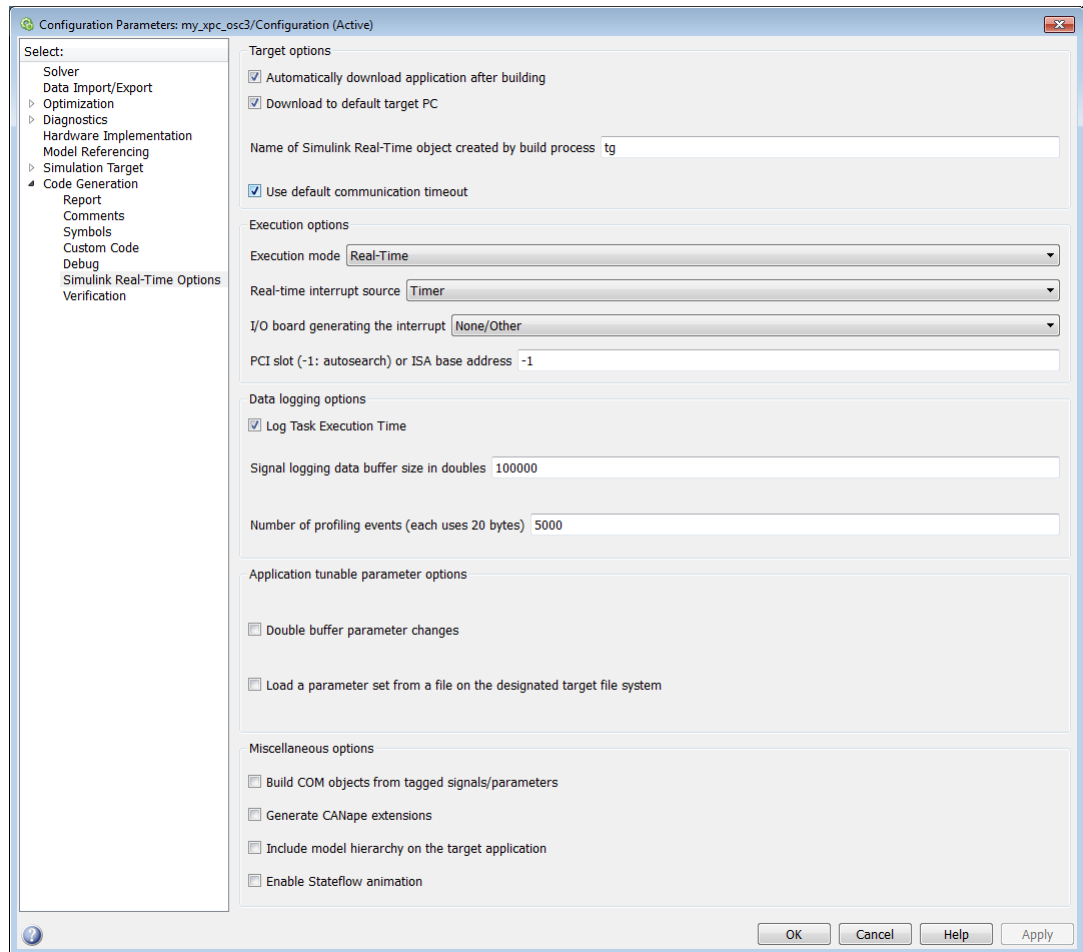
11 Click **OK**.

12 From the **File** menu, click **Save as**. Enter a file name. For example, enter `xpc_osc3` and then click **Save**.

The next task is "Start Target Hardware".

Start Target Hardware

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 using the Simulink Real-Time Explorer and created a target boot disk for that setup, you can start the target computer. Before building the real-time application, you must start the target computer because the build process automatically downloads the real-time application to the target computer. Be sure that you have followed the instructions from "Real-Time System Configuration".

1 Insert the target boot disk into the target computer disk drive.
2 Turn on the target computer or press the **Reset** button.

The target computer displays a screen like this screen.
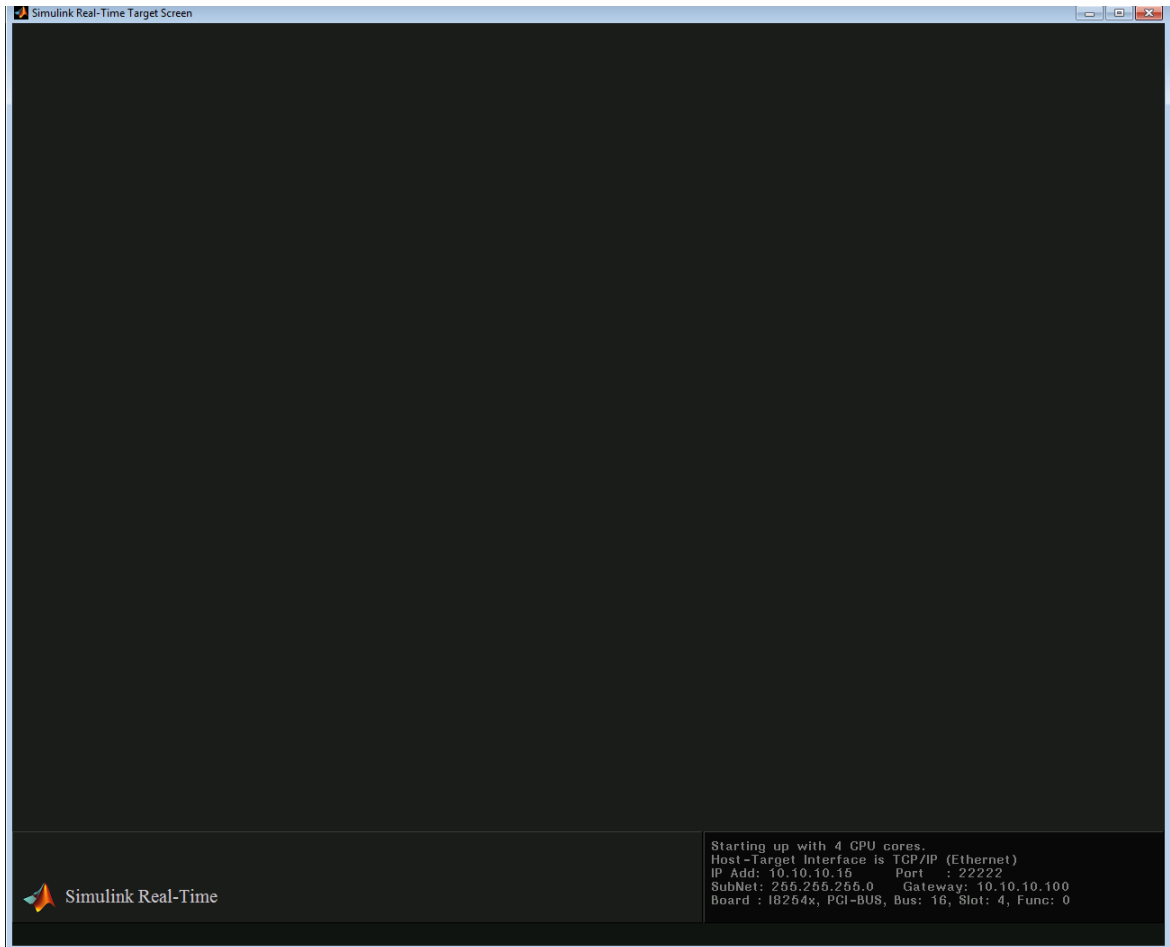
If you have a keyboard attached to the target computer, you can activate that keyboard by typing **C**. Press the **Page Up** and **Page Down** keys to page up and down in the target computer monitor.

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: The kernel can use only 2 GB of memory. Of the available memory, the kernel uses 3 MB for itself and reserves 1 MB for the real-time application.

The next task is "Build and Download Real-Time Application".

Build and Download Real-Time Application

The example model, xpc_osc3, is a real-time model of a damped oscillator. You use the Simulink Real-Time build process to generate C code, compile, link, and download the real-time application to the target computer.

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 Configuration Parameters".
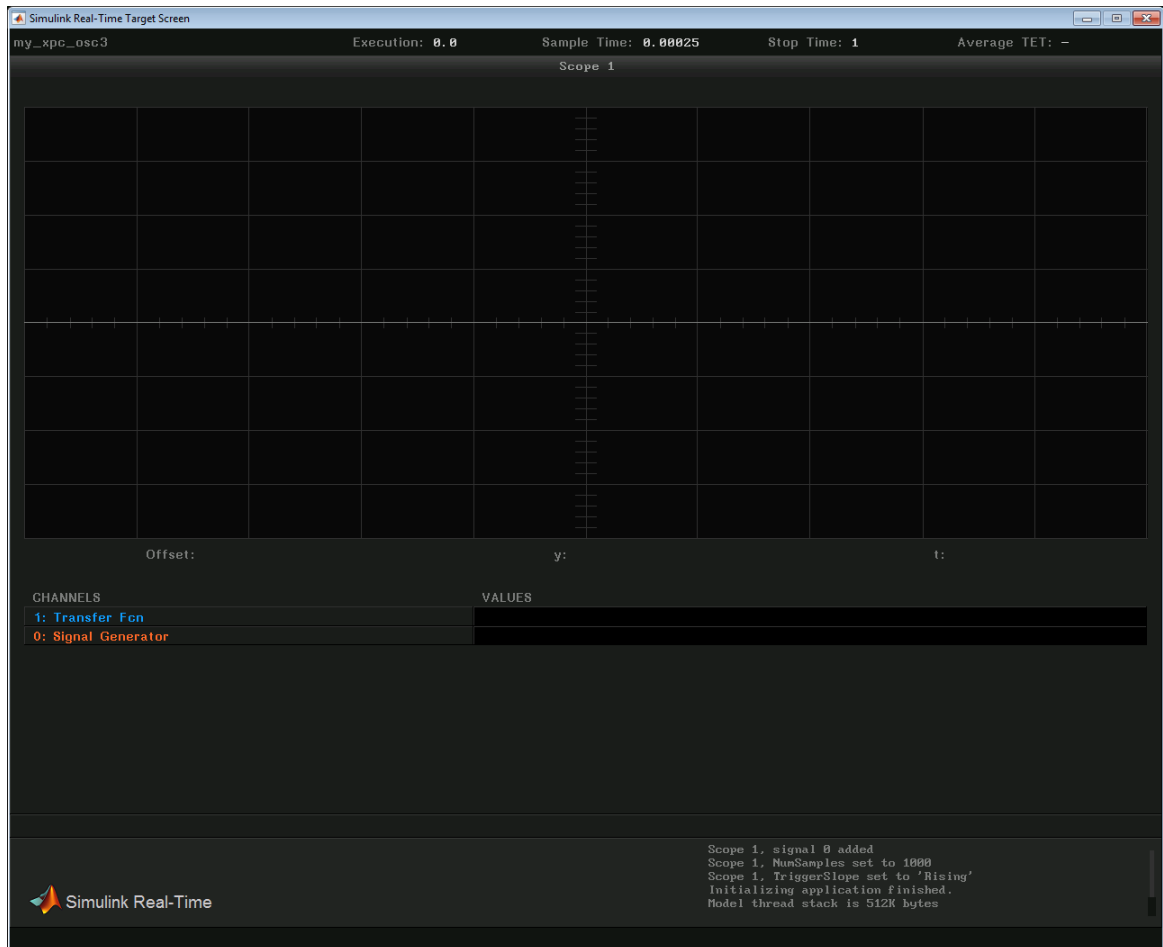
1 In the MATLAB window, type xpc_osc3.

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

2 In the Simulink window, on the toolbar, click the Build Model icon 🔳 on the toolbar.

   On the development computer, after completing a build without detecting an error, MATLAB displays lines like the following:

```
### Starting Simulink Real-Time build procedure for model:
xpc_osc3
. . .
### Successful completion of Simulink Real-Time build procedure for
model: xpc_osc3
```

   After compiling, linking, and downloading the real-time application, Simulink Real-Time creates a target object in the MATLAB workspace. The default name of the target object is tg. For more information about the target object, see "Target Driver Objects".

   If you have a monitor connected to your target computer, the monitor screen looks like this screen.

3   In the MATLAB window, type:

tg

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

If the software detects a error during build and download, see "Troubleshooting in Simulink Real-Time".

If you accidentally download a real-time application built with a different version of the Simulink Real-Time product than the one on the target computer, the following error message appears on the target computer monitor. The download fails.

```
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. The target object is defined by a set of properties and associated methods. You control the real-time application and computer by setting the target object properties.

The next task is "Execute Real-Time Application Using Simulink External Mode" on page 4-42.

Execute Real-Time Application Using Simulink External Mode

Control of the Simulink 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. In Simulink external mode, the model can connect only to the default target computer.

Note: Do not use Simulink external mode while Simulink Real-Time Explorer is running. Use only one interface or the other.

After you create and download a real-time application to the target computer, you can run the real-time application. This procedure uses the Simulink model xpc_osc3 as an example (see "Build and Download Real-Time Application" on page 4-39). You must already have specified the required target computer environment on the **Simulink Real-Time Options** node of the Simulink Real-Time parameters dialog box. In particular, you must specify the target computer to which you want to connect. See the **Use the default target PC** check box description in "Simulink Real-Time Options Configuration Parameters".

1   Select **Simulation** > **Mode** > **External**.

A check mark appears next to the menu item **External**. Simulink external mode is activated. Simulink external mode connects the Simulink model to the real-time application as a simple graphical user interface.

2

On the toolbar, click the Run icon .

The real-time application begins running. If you have a monitor connected to your target computer, you can view output that looks like this output.
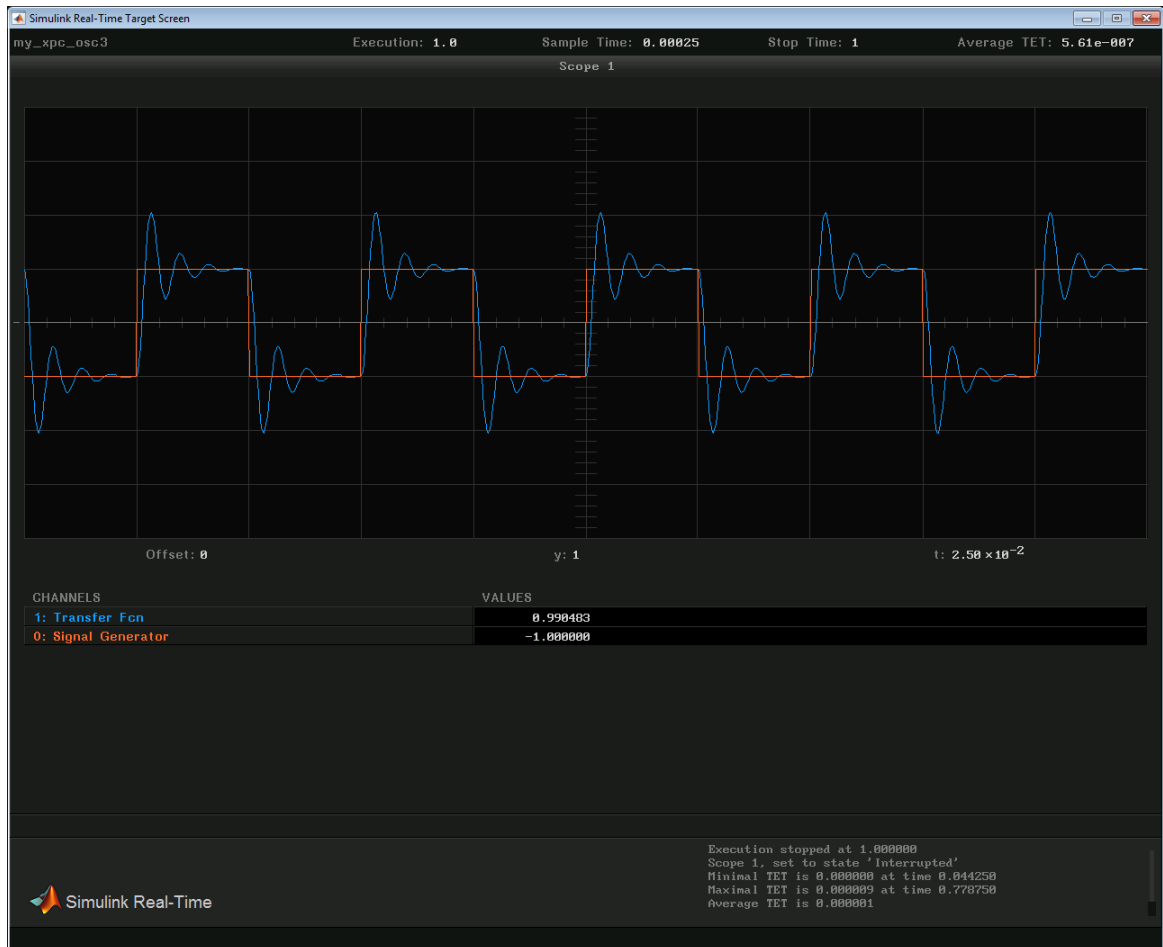
3 To stop execution, on the toolbar, click the Stop icon ⏹.

The next task is "Configure and Control a Real-Time Application".

Configure and Control a Real-Time Application

You can configure and control a real-time application using the Simulink Real-Time Explorer as well as Simulink External Mode. This tutorial focuses on interaction using Simulink Real-Time Explorer. The model is xpc_osc3, a real-time model of a damped oscillator.

To access the example folder, type:

```
addpath(fullfile(matlabroot,'help','toolbox','xpc',
        'examples'));
```

- "Execute Real-Time Application Using Simulink Real-Time Explorer"
- "Change Stop Time and Sample Time"

# Execute Real-Time Application Using 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 Hardware". This procedure begins with a real-time application already downloaded to the target computer (see "Build and Download Real-Time Application").

1 In the MATLAB window, type `slrtexplr`.

2 In the **Targets** pane, click the target computer icon for which you have downloaded the application and click the Connect icon on the toolbar.

   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 and click the Start icon on the toolbar.

   The application starts running. The dialog box looks like this figure.
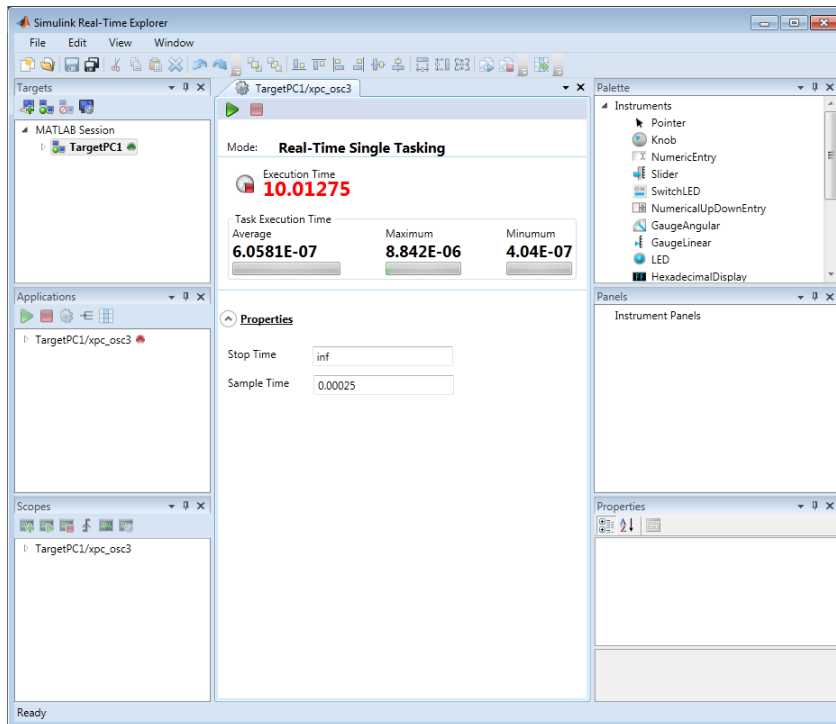
You can view the output of the real-time Scope block by connecting a monitor to your target computer.

5 To stop execution, click the real-time application and click the Stop icon ⬛ on the toolbar.

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
```

See also "Create Target Scopes Using Simulink Real-Time Explorer" and "Log Signal Data Using Outport Block and Simulink Real-Time Explorer".

Change Stop Time and Sample Time

This procedure is for changing the stop time and sample time in the real-time application configuration. You must have already downloaded the real-time application to a target computer.

1 In Simulink Real-Time Explorer, expand the node of the loaded real-time application in the **Applications** pane.

2 On the toolbar, click the Properties icon .

3 In the **Application Configuration** workspace, under Execution Parameters, enter a new value for **Stop time**. For example, enter inf and press **Enter**.

   The real-time application now runs until you stop it.

4 Enter a new value for **Sample Time**. For example, enter 0.00010 and press **Enter**.

5 To start execution, click the real-time application and click the Start icon on the toolbar.

6 To stop execution, click the real-time application and click the Stop icon on the toolbar.

   The dialog box looks like this figure.

Some blocks cannot handle sample time changes at run-time (see "Blocks Whose Outputs Depend on Inherited Sample Time"). There will be a warning at application build time that changing the sample time might cause incorrect results upon execution. If you see such a warning, change **Fixed step size** in the **Solver** pane of the Configuration Parameters dialog box and rebuild the application.

If you choose 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.

Create an Application Showing Model Hierarchy

To create a real-time application that shows model hierarchy in Simulink Real-Time Explorer:

1   Open the Simulink window.

2   Click **Simulation** > **Model Configuration Parameters**.

3   Under the **Code Generation** node, select the **Simulink Real-Time Options** node.

4   Select the **Include model hierarchy on the target application** check box.

5   From the **Code** menu, click **C/C++ Code** > **Build Model**.

Create a Prebuilt Real-Time Application

To build a real-time application without downloading it:

1 In the Simulink window, click **Simulation** > **Model Configuration Parameters**.
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 **Code** menu, click **C/C++ Code** > **Build Model**.

Download a Prebuilt Real-Time Application

To download a prebuilt application from Simulink Real-Time Explorer:

1 In the MATLAB window, type `slrtexplr`.
2 In the **Targets** pane, click the required target computer icon and click the Connect icon on the toolbar.

   The target computer icon turns Connected .
3 Do one of the following:

   • From the MATLAB window or from Microsoft Windows, drag the prebuilt application DLM file to the target computer icon.

   • Right-click the required target computer icon and click **Load Application**. The **DLM Application Selector** dialog box opens. Enter or browse to the prebuilt application DLM file.

Simulate Simulink Model Using MATLAB Language

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

After you load the Simulink model into the MATLAB workspace, you can run a simulation. This procedure uses the Simulink model xpc_osc2 as an example. You must have loaded this model and entered the MATLAB variables tout and yout in the **Data I/O** pane in the Configuration Parameters dialog box.

1 In the MATLAB window, type:

```
output=sim('xpc_osc2','SimulationMode','normal');
```

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

Simulink runs a simulation in normal mode through to completion. You cannot manually stop the simulation. For further information on the sim command, see the Simulink documentation.

2 After Simulink finishes the simulation, type:

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

MATLAB opens a plot window and displays the output response as shown in this figure.
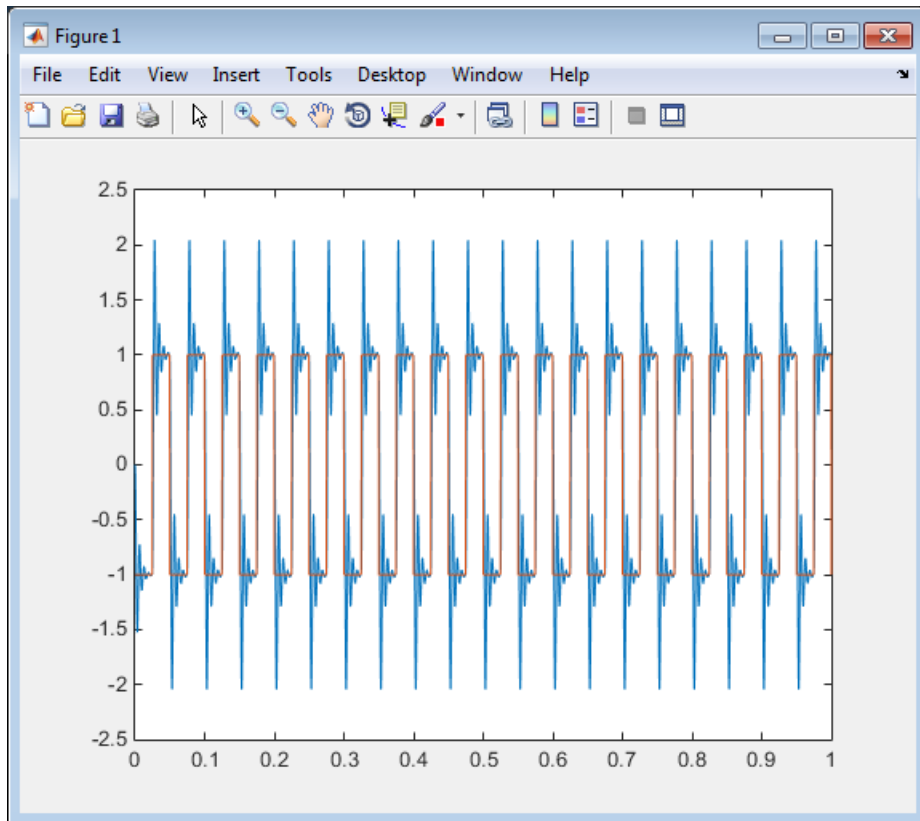
When the real-time application is running in `Real-Time` mode, data is not saved to the variables `tout` and `yout`. Instead, data is saved in the target computer memory and can be retrieved through the target object properties `tg.TimeLog`, `tg.StateLog`, and `tg.OutLog`. However, in the Configuration Parameters dialog box, you must still select the **Time**, **States**, and **Output** check boxes for data to be logged to the target object properties.

Prepare Real-Time Application Using MATLAB Language

You can configure your real-time application using MATLAB language and build and download it to the target computer. This example uses the xpc_osc3 model and assumes the default target computer is configured for network boot mode and communication has been established.

1 Check that the connection to the target computer is functioning.

```
slrtpingtarget
```

2 Load the model.

```
model_name = 'xpc_osc3';
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 and download the model.

```
rtwbuild(model_name);
pause(30);
```

5 Assign tg to the target computer.

```
tg = slrt;
```

The next step is to execute the real-time application. See "Execute Real-Time Application Using MATLAB Language" on page 4-55.

Execute Real-Time Application Using 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 xpc_osc3 as an example. You must have already carried out the steps in "Prepare Real-Time Application Using MATLAB Language" on page 4-54.

1 In the MATLAB window, type:

```
start(tg)
```

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

```
Target: TargetPC1
  Connected   = Yes
  Application   = xpc_osc3
  Mode    = Real-Time Single-Tasking
  Status    = running
```

2 On the target computer screen, the **Execution** line changes from stopped to running. The **AverageTET** line is periodically updated with a new value.
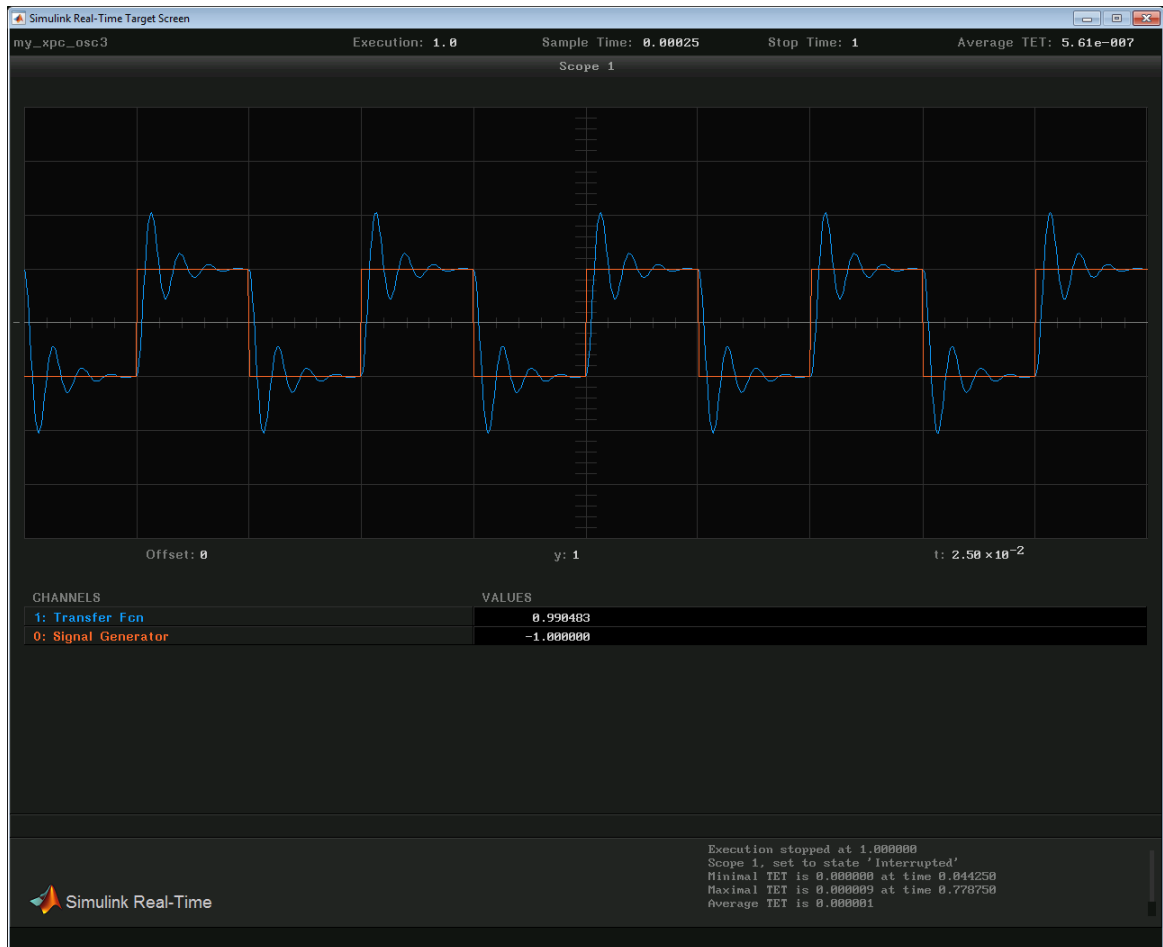
3 In the MATLAB window, type:

```
stop(tg)
```

The real-time application stops running.

The Simulink Real-Time software allows you to change many properties and parameters without rebuilding the real-time application. Two of these properties are StopTime and SampleTime.

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

```
tg.StopTime = 1000
```

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

```
tg.SampleTime = 0.01
```

Some blocks cannot handle sample time changes at run-time (see "Blocks Whose Outputs Depend on Inherited Sample Time"). There will be a warning at application build time that changing the sample time might cause incorrect results upon execution. If you see such a warning, change **Fixed step size** in the **Solver** pane of the Configuration Parameters dialog box and rebuild the application.

If you choose 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.

Application and Driver Scripts

The Simulink Real-Time examples show the features of the Simulink Real-Time product. They are also scripts that you can view to understand how to write your own scripts for creating and testing real-time applications.

Examples fall into two categories, general applications and drivers. The following table lists the general application examples.

Description	File Name
Real-time parameter tuning and data logging	Parameter Tuning and Data Logging
Freerun display mode of a host	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 to a file on the target computer	Data Logging With a File Scope
Frame signal processing **Note:** This example requires DSP System Toolbox™ software.	Frame Signal Processing
Simulink Real-Time software as a real-time spectrum analyzer	Spectrum Analyzer

Description	File Name
Creating a 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

The driver examples category contains examples for a number of driver applications, including, but not limited to:

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

Note: Because these examples illustrate the use of driver blocks in a Simulink Real-Time environment, you might need specific hardware to run them.

You can access Simulink Real-Time general application and driver examples through MATLAB Help. In this window, click **Simulink Real-Time > Demos** to see the available example categories.

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 a MATLAB editing window.

Glossary

application	See *real-time application*.
build process	Process of generating a real-time application from your Simulink model, compiling, linking, and downloading the generated code to create a *real-time application*.
execution	Running the *real-time application* on the target computer in either `Real-Time` or `Freerun` mode.
executable code	See real-time application.
kernel	Real-time software component running on the target computer that manages the downloaded *real-time application*.
model	Simulink and/or Stateflow model.
parameter tuning	Process of changing block parameters and downloading the new values to a *real-time application* while it is running or not running.
sample rate	Rate the *real-time application* is stepped in samples/second. Reciprocal of the *sample time*.
sample time	Interval, in seconds, between the execution of *real-time application* steps.
signal logging	Acquiring and saving signal data created during a real-time execution.
signal monitoring	Getting the values of one or more signals without time information.
signal tracing	Acquiring and displaying packages of signal data during real-time execution.
simulation	Running a non-real-time simulation of the Simulink and Stateflow model on the development computer.
real-time application	Executable code generated from a Simulink and Stateflow model, which can be executed by the Simulink Real-Time kernel on the target computer.