

**Simulink® Real-Time™**

Reference



**MATLAB® & SIMULINK®**

R2022b



## How to Contact MathWorks



Latest news: [www.mathworks.com](http://www.mathworks.com)  
Sales and services: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
User community: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Technical support: [www.mathworks.com/support/contact\\_us](http://www.mathworks.com/support/contact_us)



Phone: 508-647-7000



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

*Simulink® Real-Time™ Reference*

© COPYRIGHT 2002–2022 by The MathWorks, Inc.

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

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

### Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://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](http://www.mathworks.com/patents) for more information.

## Revision History

March 2007	Online only	New for Version 3.2 (Release 2007a)
September 2007	Online only	Updated for Version 3.3 (Release 2007b)
March 2008	Online only	Updated for Version 3.4 (Release 2008a)
October 2008	Online only	Updated for Version 4.0 (Release 2008b)
March 2009	Online only	Updated for Version 4.1 (Release 2009a)
September 2009	Online only	Updated for Version 4.2 (Release 2009b)
March 2010	Online only	Updated for Version 4.3 (Release 2010a)
April 2011	Online only	Updated for Version 5.0 (Release 2011a)
September 2011	Online only	Updated for Version 5.1 (Release 2011b)
March 2012	Online only	Revised for Version 5.2 (Release 2012a)
September 2012	Online only	Revised for Version 5.3 (Release 2012b)
March 2013	Online only	Revised for Version 5.4 (Release 2013a)
September 2013	Online only	Revised for Version 5.5 (Release 2013b)
March 2014	Online only	Revised for Version 6.0 (Release 2014a)
October 2014	Online only	Revised for Version 6.1 (Release 2014b)
March 2015	Online only	Revised for Version 6.2 (Release 2015a)
September 2015	Online only	Revised for Version 6.3 (Release 2015b)
March 2016	Online only	Revised for Version 6.4 (Release 2016a)
September 2016	Online only	Revised for Version 6.5 (Release 2016b)
March 2017	Online only	Revised for Version 6.6 (Release 2017a)
September 2017	Online only	Revised for Version 6.7 (Release 2017b)
March 2018	Online only	Revised for Version 6.8 (Release 2018a)
September 2018	Online only	Revised for Version 6.9 (Release 2018b)
March 2019	Online only	Revised for Version 6.10 (Release 2019a)
September 2019	Online only	Revised for Version 6.11 (Release 2019b)
March 2020	Online only	Revised for Version 6.12 (Release 2020a)
September 2020	Online only	Revised for Version 7.0 (Release 2020b)
March 2021	Online only	Revised for Version 7.1 (Release 2021a)
September 2021	Online only	Revised for Version 7.2 (Release 2021b)
March 2022	Online only	Revised for Version 8.0 (Release 2022a)
September 2022	Online only	Revised for Version 8.1 (Release 2022b)



<b>1</b>	<b>Configuration Parameters</b>	
	<b>Simulink Real-Time Options Pane</b> .....	<b>1-2</b>
	Configuration .....	<b>1-2</b>
	Tips .....	<b>1-2</b>
	To get help on an option .....	<b>1-2</b>
<b>2</b>	<b>TLC Options Parameters</b>	
	<b>TLC Command-Line Options</b> .....	<b>2-2</b>
<b>3</b>	<b>Apps</b>	
<b>4</b>	<b>Target Computer Status Monitor</b>	
	<b>Target Computer Status Monitor</b> .....	<b>4-2</b>
	Display Status Monitor .....	<b>4-2</b>
	Display Status Monitor by Using PuTTY .....	<b>4-2</b>
<b>5</b>	<b>Target Computer Command-Line Interface Reference</b>	
	<b>Target Computer Command-Line Interface</b> .....	<b>5-2</b>
	Target Object Commands .....	<b>5-2</b>
	Target Computer RTOS System Commands .....	<b>5-3</b>



# Configuration Parameters

---

## Simulink Real-Time Options Pane

Parameter	Description
<b>Log level</b>	Selects filtering level that limits RTOS system messages that appear in the system log.
<b>Force polling mode</b>	Enables polling mode —instead of interrupt-driven mode— for clocking the real-time application.
<b>Max file log runs</b>	Selects the number of file log run to retain when logs are stored on the target computer.
<b>Compile with GCC -ffast-math</b>	Enables the GCC compiler -ffast-math option when compiling real-time application code.

Control the code created by Simulink Coder™ code generation software for a Simulink Real-Time application. Set up general information about building real-time applications, including target, execution, data logging, and other options.

### Configuration

The **Simulink Real-Time Options** node in the Configuration Parameters dialog box allows you to specify how the software generates the real-time application. To reveal the **Simulink Real-Time Options** node, do the following:

- 1 In the **Code Generation** pane, in the **System target file** list, select `slrealtime.tlc`. This setting generates system target code for Simulink Real-Time.

---

**Note** If you open a model that was originally saved with **System target file** set to `xpctarget.tlc`, the software updates the setting to `slrealtime.tlc`. To retain the updated setting, save the updated model.

---

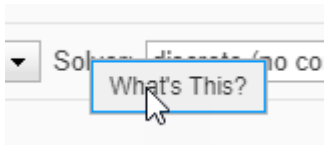
### Tips

- The default values work for the generation of most real-time applications. If you want to customize the build of your real-time application, set the option parameters to suit your specifications.
- To access configuration parameters from the MATLAB® command line, use:
  - `gcs` — To access the current model.
  - `set_param` — To set the parameter value.
  - `get_param` — To get the current value of the parameter.

### To get help on an option

- 1 Right-click the option text label.
- 2 From the context menu, select **What's This**.





## Log level

Selects filtering level for system log messages

**Model Configuration Pane:** Code Generation / Simulink Real-Time Options

### Description

Selects filtering level that limits Simulink Real-Time target computer system messages that appear in the system log. To open the viewer tab in Simulink Real-Time Explorer and view the system log from the target computer `tg`, in the MATLAB Command Window, type:

```
slrtExplorer
```

For more information, see `slrtExplorer`.

### Dependencies

The `SLRTLogLevel` configuration parameter sets the initial value for the `logLevel` option when you build the real-time application.

For more information, see `Application` object.

**Option:** `loglevel`

### Settings

`info` | `trace` | `debug` | `warning` | `error` | `fatal`

**Default:** `info`

`info`

Select to include all system messages in the system log.

`trace`

Select to include all system memory trace messages in the system log.

`debug`

Select to include all system debug messages in the system log.

`warning`

Select to include all system warning messages in the system log.

`error`

Select to include all system error messages in the system log.

`fatal`

Select to include all system fatal messages in the system log.

### Examples

## Get Log Level

- To get the `SLRTLogLevel` option value for real-time application object `my_app` that is loaded on target computer, type in the MATLAB Command Window:

```
my_app = slrealtime.Application('slrt_ex_osc')
app_object.Options.get.loglevel
```

## Recommended Settings

Application	Setting
Debugging	debug
Traceability	trace
Efficiency	info
Safety precaution	info

## Programmatic Use

**Parameter:** `SLRTLogLevel`

**Type:** character vector

**Value:** 'info' | 'trace' | 'debug' | 'warning' | 'error' | 'fatal'

**Default:** 'info'

## Version History

Introduced in R2020b

## See Also

**Force polling mode** | **Max file log runs** | **Compile with GCC -ffast-math** | “Simulink Real-Time Options Pane” on page 1-2 | Application

## Force polling mode

Enables polling mode — instead of interrupt-driven mode — for clocking the real-time application

**Model Configuration Pane:** Code Generation / Simulink Real-Time Options

### Description

Enables polling mode — instead of interrupt-driven mode — for clocking the real-time application. Polling mode can be useful for reducing sample time jitter. But, enabling this option causes the real-time application to consume a CPU core completely to clock and execute the base rate.

### Dependencies

The `SLRTForcePollingMode` configuration parameter sets the initial value for the `pollingThreshold` option when you build the real-time application. Enabling `SLRTForcePollingMode` sets the `pollingThreshold` to a value above the base sample rate. This setting forces clocking the real-time application in polling mode.

**Option:** `pollingThreshold`

### Settings

off | on

**Default:** off

off

When `Force polling mode` is disabled, the real-time application is clocked by a timer interrupt, unless the base sample rate is equal to or below the polling threshold (100  $\mu$ s). If the base sample rate is less than or equal to the threshold, the real-time application is clocked in polling mode.

on

When `Force polling mode` is enabled, the real-time application is always clocked in polling mode.

### Examples

#### Get Force polling mode

- To get the `SLRTForcePollingMode` option value for real-time application object `my_app` that is loaded on target computer, type in the MATLAB Command Window:

```
my_app = slrealtime.Application('slrt_ex_osc')
app_object.Options.get.SLRTForcePollingMode
```

## Recommended Settings

Application	Setting
Debugging	off
Traceability	off
Efficiency	off
Safety precaution	off

## Programmatic Use

**Parameter:** SLRTForcePollingMode

**Type:** character vector

**Value:** 'off' | 'on'

**Default:** 'off'

## Version History

Introduced in R2020b

## See Also

**Log level** | **Max file log runs** | **Compile with GCC -ffast-math** | “Simulink Real-Time Options Pane” on page 1-2 | Application

## Max file log runs

Selects the number of file log runs to retain for the real-time application

**Model Configuration Pane:** Code Generation / Simulink Real-Time Options

### Description

Selects the number of file log runs to retain for the real-time application when logs are stored on the target computer instead of uploaded to the development computer after each simulation run. The logs are stored if auto-import is disabled, or the target is not connected to the host at stop time.

### Dependencies

The `SLRTFileLogMaxRuns` configuration parameter sets the initial value for the `fileLogMaxRuns` option when you build the real-time application.

**Option:** `fileLogMaxRuns`

### Settings

1 | int

**Default:** 1

int

Select the number of file log runs to retain for the real-time application when logs are stored on the target computer instead of uploaded to the development computer after each simulation run.

### Examples

#### Get Max file log runs

- To get the `SLRTFileLogMaxRuns` option value for real-time application object `my_app` that is loaded on target computer, type in the MATLAB Command Window:

```
my_app = slrealtime.Application('slrt_ex_osc')
app_object.Options.get.SLRTFileLogMaxRuns
```

### Recommended Settings

Application	Setting
Debugging	1
Traceability	1
Efficiency	1
Safety precaution	1

## **Programmatic Use**

**Parameter:** SLRTFileLogMaxRuns

**Type:** int

**Value:** int

**Default:** 1

## **Version History**

**Introduced in R2020b**

## **See Also**

**Log level** | **Force polling mode** | **Compile with GCC -ffast-math** | “Simulink Real-Time Options Pane” on page 1-2 | Application

## Compile with GCC -ffast-math

Enables the GCC compiler `-ffast-math` option when compiling real-time application code

**Model Configuration Pane:** Code Generation / Simulink Real-Time Options

### Description

Enables the GCC compiler `-ffast-math` option when compiling real-time application code. This option is disabled by default for Simulink Real-Time models.

By enabling the **Compile with GCC -ffast-math** option, you provide the compiler with more flexibility to optimize floating-point math at the expense of deviating from the IEEE-754 floating-point standard.

For more information about the `-ffast-math` option, see the Semantics of Floating-Point Math in GCC.

- [gcc.gnu.org/wiki/FloatingPointMath/](http://gcc.gnu.org/wiki/FloatingPointMath/)

### Dependencies

None

### Settings

off | on

**Default:** off

off

When UseGCCFastMath is disabled, Simulink Real-Time compiles real-time application code without the compiler `-ffast-math` option.

on

When UseGCCFastMath is enabled, Simulink Real-Time compiles real-time application code with the compiler `-ffast-math` option.

### Examples

#### Get Compile with GCC -ffast-math

- To get the UseGCCFastMath option value for real-time application object `my_app` that is loaded on target computer, type in the MATLAB Command Window:

```
my_app = slrealtime.Application('slrt_ex_osc')
app_object.Options.get.UseGCCFastMath
```



## Recommended Settings

Application	Setting
Debugging	on
Traceability	on
Efficiency	on
Safety precaution	off

## Programmatic Use

**Parameter:** UseGCCFastMath

**Type:** character vector

**Value:** 'off' | 'on'

**Default:** 'off'

## Version History

Introduced in R2020b

## See Also

**Log level** | **Force polling mode** | **Max file log runs** | “Simulink Real-Time Options Pane” on page 1-2 | Application



# TLC Options Parameters

---

## TLC Command-Line Options

TLC command-line options are model options set before code generation to configure the real-time application and the real-time RTOS.

To set these options from the **Code Generation** pane in the Configuration Parameters dialog box, select **Advanced Parameters**. Type the option in the **TLC command line options** text box in this form:

```
-option_name1=option_value1 -option_nameN=option_valueN
```

Prefix each option name with `-a`. Do not leave spaces around the equal sign. Do not place a comma between consecutive value assignments.

To set these options from the Command Window, use the syntax:

```
set_param(model_name, ...  
          'TLCOptions', ...  
          '-option_name1=option_value1 -option_nameN=option_valueN')
```

To read these options from the Command Window, use the syntax:

```
get_param(model_name, 'TLCOptions');
```

To remove these options, use the syntax:

```
set_param(model_name, 'TLCOptions', '')
```

---

**Note** At this time, no TLC options for Simulink Real-Time are supported.

---

# Apps

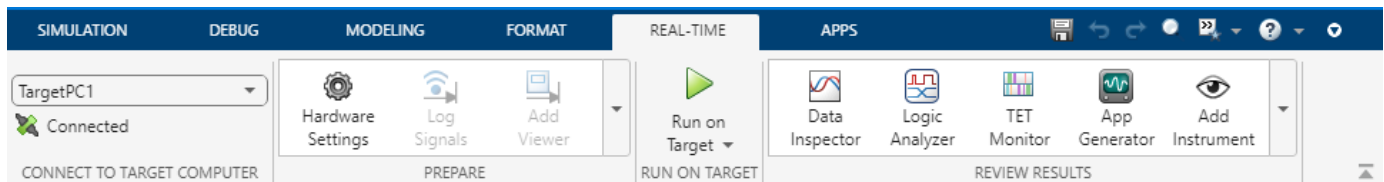
---

## Simulink Real-Time

Generate real-time applications for simulations that run on a target computer and interface with I/O devices in the target computer

### Description

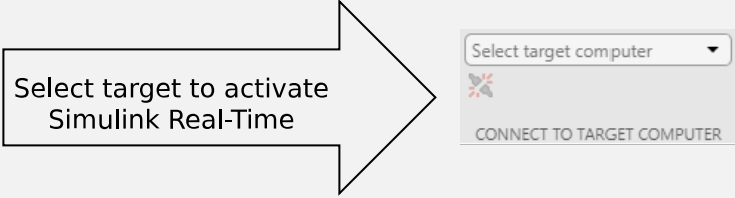
Use the **Simulink Real-Time** app to configure a model to build and run real-time applications on a target computer. The app configures the model to use the Simulink Real-Time code generation target and other configuration parameters for code generation. When you open the app, a **Real-Time** tab is added to the toolstrip. The **Real-Time** tab represents groups of tasks in the Simulink Real-Time workflow.



After you use the app to configure the model for Simulink Real-Time, you can perform these and more tasks from the **Real-Time** tab in the Simulink Editor.







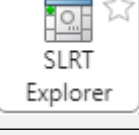



Use the actions in the **Connect to Target Computer** section to select and connect to a target computer.




#### Connect to Target Computer Actions

UI Control	Description
	<p>From the target computers list on the <b>Real-Time</b> computer to which you want to connect. For more information, see the <b>connect</b> function.</p>

Use the actions in the **Prepare** section to configure the model and tune parameters.

## Prepare Actions









UI Control	Description
 <p>Hardware Settings</p>	Configure model to run on target computer. The Configuration Parameters dialog box opens.
 <p>Log Signals</p>	Send signal to the Simulation Data Inspector and workspace. Select one or more signals before using this button.
 <p>Add Viewer</p>	Add a viewer to the selected signals. Select one or more signals before using this button.
 <p>Test Point</p>	Allocate memory and make signals observable when using a Floating Point Test Point. Select one or more signals before using this button.
 <p>Signal Table</p>	Show table to manage signal logging and viewing. The <b>Signal Table</b> tab of the Configuration Parameters dialog box opens in the Simulink Editor.
 <p>Configure Logging</p>	Configure logging. The <b>Data Import/Export</b> tab of the Configuration Parameters dialog box opens.
 <p>SLRT Explorer</p>	Start Simulink Real-Time Explorer. For more information, see <b>Simulink Real-Time Explorer</b> .
 <p>Library Browser</p>	Open block library. The block library browser opens. See the Simulink Real-Time Speedgoat I/O Blockset.
 <p>Control Panel</p>	Launch external mode control panel. The control panel opens.
 <p>Connect Inputs</p>	Link sets of signals from files and workspace to root Inport blocks. The <b>Connect Inputs</b> dialog box opens.

UI Control	Description
 The button features a square icon with binary code (101, 010) and a star, a stop sign, and the text "Hold Updates".	Communicate changes of multiple parameters at once. For more information, see "Using Hold Updates and Update All Parameters".
 The button features a square icon with binary code (101, 010) and a star, a green downward arrow, and the text "Update All Parameters".	Update all parameters on target computer. For more information, see "Using Hold Updates and Update All Parameters".
 The button features a square icon with a green chip and a red eraser, the text "Remove Hardware Configuration", and a sub-description: "Remove hardware configuration from this model".	Remove hardware configuration from this model. The code generation

Use the actions in the **Run on Target** section build the model, deploy the real-time application to the target computer, and run the real-time application.















### Run on Target Actions

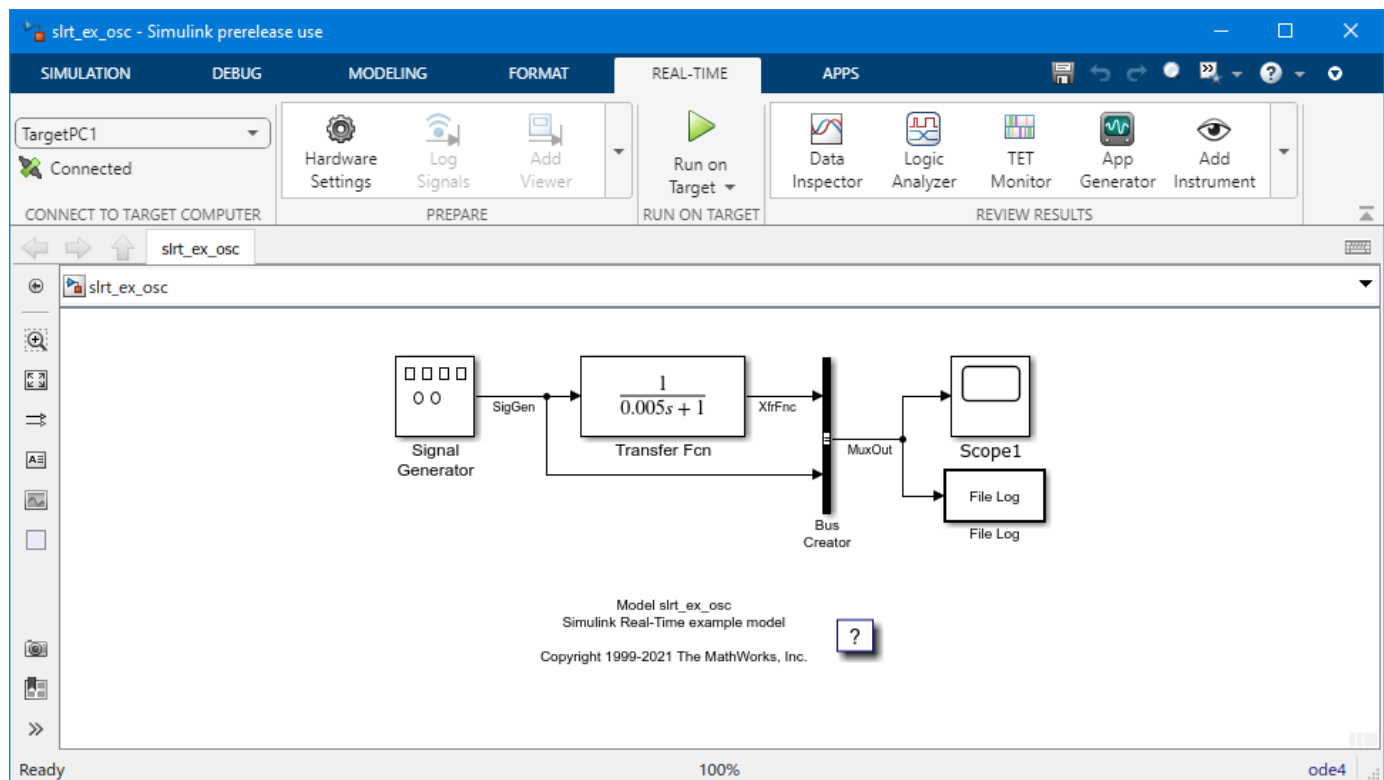
UI Control	Description
	<p>Run application on target computer, observe outputs, and tune parameters. This action also deploys real-time application when model changes are found. For more information, see "Execute Real-Time Application in Simulink External Mode by Using Run on Target".</p>
	<p>Generate a real-time application from Simulink mode. For more information, see "Generate Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>
	<p>Deploy real-time application to target computer. For more information, see "Deploy Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>
	<p>Connect Simulink model to real-time application on target computer. For more information, see "Execute Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>
	<p>Run real-time application. Observe output and tune parameters by using the Run on Target button. For more information, see "Execute Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>
	<p>Restart real-time application on target computer. For more information, see "Execute Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>
	<p>Stop real-time application on target computer. For more information, see "Execute Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>
	<p>Disconnect Simulink model from real-time application on target computer. For more information, see "Execute Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>
<input checked="" type="checkbox"/> <b>AutoImportFileLog</b> Import the file log data on application stop	<p>Import the file log data on application start. For more information, see "Execute Real-Time Application in Simulink External Mode by Using Step-by-Step Commands".</p>

Use the actions in the **Review Results** section instrument the model and observe outputs.

**Review Results Actions**

UI Control	Description
 <p>Data Inspector</p>	View logged data in the Simulation Data Inspector.
 <p>Start Recording</p>	Start signal logging and streaming. For more information, see <code>startRecording</code> . “Signal Logging and Streaming Basics”.
 <p>Stop Recording</p>	Stop signal logging and streaming. For more information, see <code>stopRecording</code> . “Signal Logging and Streaming Basics”.
 <p>Logic Analyzer</p>	Visualize, measure, and analyze transitions and states over time in the Simulation Data Inspector.
 <p>TET Monitor</p>	View real-time application task execution time. For more information, see <b>TET Monitor</b> .
 <p>App Generator</p>	Launch App Generator to create an App Designer instrument panel. For more information, see <b>Simulink Real-Time App Generator</b> .
 <p>Add Instrument</p>  <p>Configure Instrument</p>	Use the Simulink model to select signals to view in the Simulation Data Inspector. When you click the button and add an instrument to the model, the button label changes from <b>Add Instrument</b> to <b>Configure Instrument</b> . For more information, see “Add Instruments to Real-Time Application from Simulink Model”.
 <p>Remove Instrument</p>	Stop recording selected signal values in the Simulation Data Inspector. For more information, see “Add Instruments to Real-Time Application from Simulink Model”.
 <p>Highlight Instrument</p>	Highlight selected signals. For more information, see “Add Instruments to Real-Time Application from Simulink Model”.

UI Control	Description
	Import instrument from file. For more information, see “Add Instrument Application from Simulink Model”.
	Export instrument to file. For more information, see “Add Instruments from Simulink Model”.



## Open the Simulink Real-Time App

In the **Apps** gallery, under **Real-Time Simulation and Testing**, click **Simulink Real-Time**. The **Real-Time** tab opens.

## Examples

- “Tune Parameters by Using Hold Updates and Update All Parameters”
- “Build and Download Real-Time Application by Using Run on Target”
- “Execute Real-Time Application in Simulink External Mode by Using Step-by-Step Commands”
- “Add Instruments to Real-Time Application from Simulink Model”

## Version History

Introduced in R2020b

### Added App Generator

The **Simulink Real-Time App Generator** creates App Designer instrument panels from the Simulink model or from generated application (MLDATX) files.

### AutoImportFileLog Available from Run on Target

The **AutoImportFileLog** check box is available from the **Run on Target** button on the **Real-Time** tab in the Simulink Editor. This checkbox corresponds to the `AutoImportFileLog` option of the `start` function.

### Bind mode for signals to instruments in Simulink Editor

You can bind signals to instruments (also referred to as instrumenting a signal) by using the **Add Instrument** button to enter bind mode in the Simulink Editor. You can select signals in the model and stream signal data for those signals from the real-time application to the Simulation Data Inspector.

### Changed Batch Mode Button Label to Hold Updates

On the **Real-Time** tab, the **Batch Mode** button label changed to **Hold Updates**.

## See Also

### Functions

`connect` | `start`

### Apps

**Simulink Real-Time Explorer** | **Simulink Real-Time TET Monitor** | **Simulink Real-Time App Generator**

### Topics

“Tune Parameters by Using Hold Updates and Update All Parameters”

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

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

“Add Instruments to Real-Time Application from Simulink Model”

# Simulink Real-Time Explorer

Interact with target computer and real-time application running on target computer

## Description

Simulink Real-Time Explorer provides a single point of contact for viewing connection status and interacting with a real-time application. You can monitor and trace signals, tune parameters, and stream data to the Simulation Data Inspector.

---

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

---

Use Simulink Real-Time Explorer for these tasks:

- Connect the development computer and target computer.
- Load, start, and stop a real-time application on target computer.
- View real-time application parameters and signal hierarchy.
- Select real-time application signals for streaming to the Simulation Data Inspector.
- Set real-time application stop time.
- View task execution time (TET).

For examples, click the links in the **More Information** column.

### Target Computer Configuration

Capability	More Information
Configure target computer configuration settings. View target computer disk usage.	"Target Computer Settings"

### Real-Time Application Access and Control

Capability	More Information
<ul style="list-style-type: none"> <li>• Connect target computers to a development computer, and then disconnect them.</li> <li>• Load a prebuilt real-time application into a target computer.</li> <li>• Start and then stop running a real-time application that you downloaded to the target computer.</li> <li>• Display execution time, task execution time, and other properties of the real-time application.</li> <li>• Change stop time without regenerating code.</li> <li>• Start and stop signal logging and streaming by using <b>Start Recording</b> and <b>Stop Recording</b> buttons.</li> <li>• Import file logs from the target computer by using the <b>Import File Log</b> button.</li> </ul>	<ul style="list-style-type: none"> <li>• “Real-Time Application and Target Computer Modes”</li> <li>• “Configure and Control Real-Time Application by Using Simulink Real-Time Explorer”</li> <li>• “Signal Logging and Streaming Basics”</li> </ul>

### Real-Time Application Management

Capability	More Information
<p>By using <b>Configuration</b> tab or right-click menu in <b>Targets</b> tree:</p> <ul style="list-style-type: none"> <li>• View real-time applications available on target computer, including detailed properties.</li> <li>• Delete real-time applications from target computer.</li> <li>• Select a real-time application as startup application.</li> </ul>	<p>“Connect, Load Application, and Start” on page 3-13</p>

### Signal Access

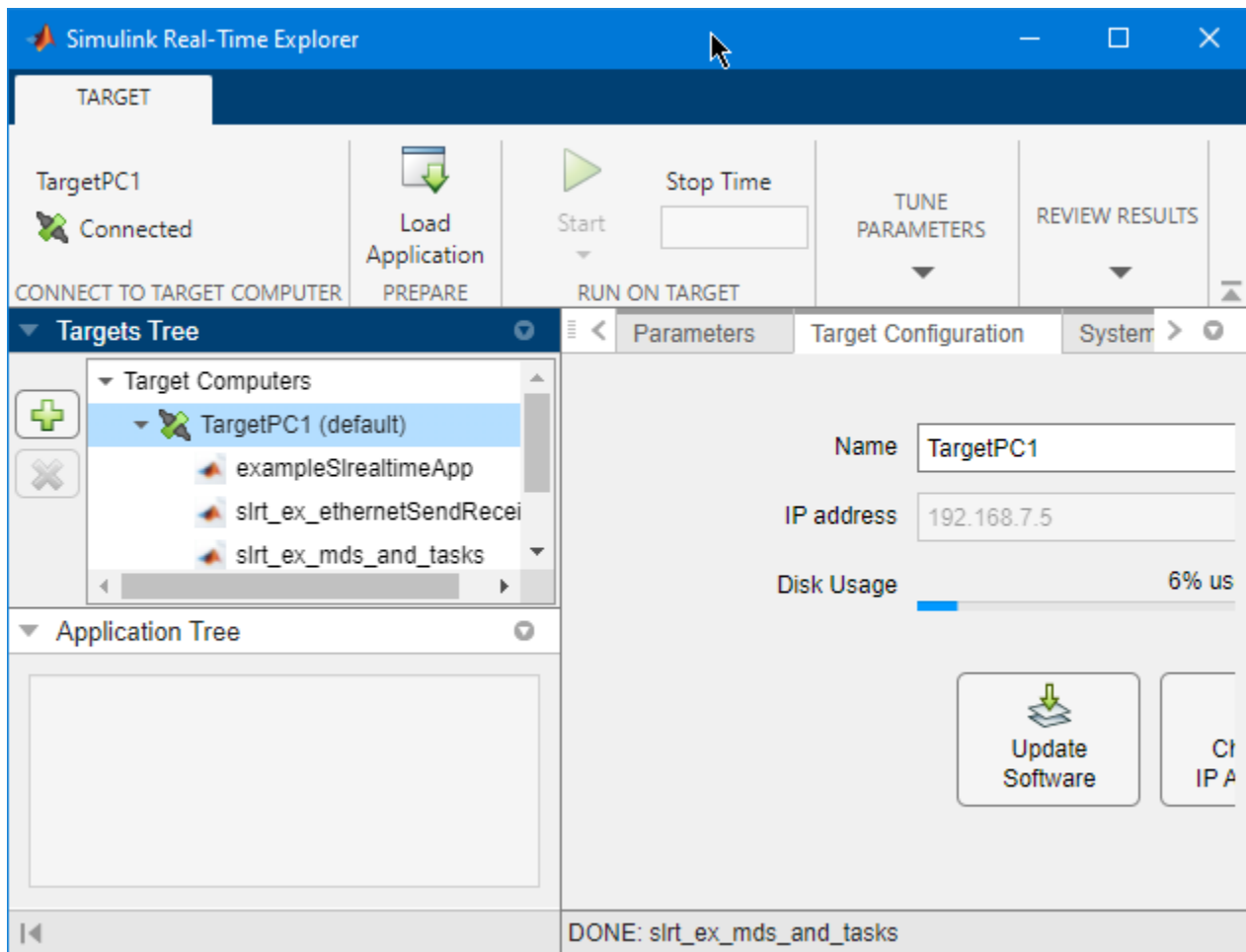
Capability	More Information
Filter and group hierarchical signals	“Display and Filter Hierarchical Signals and Parameters”
Monitor signals.	<ul style="list-style-type: none"> <li>• “Monitor Signals by Using Simulink Real-Time Explorer”</li> <li>• “Instrument a Stateflow Subsystem”</li> </ul>
Create, save, and load signal groups.	“Export and Import Signals in Instrument by Using Simulink Real-Time Explorer”

### Parameter Tuning

Capability	More Information
Filter and group hierarchical parameters	"Display and Filter Hierarchical Signals and Parameters"
Display and tune parameter values while the real-time application is running.	"Tune Parameters by Using Simulink Real-Time Explorer"
Enable parameter table operation by correcting ECU and XCP page selection mismatch by using the <b>Enable Parameter Table</b> button.	<ul style="list-style-type: none"> <li>• copyPage</li> <li>• getECUPage</li> <li>• getNumPages</li> <li>• getXCPPage</li> <li>• setECUAndXCPPage</li> <li>• setECUPage</li> <li>• setXCPPage</li> </ul>
Refresh cached parameter table values by clicking the <b>Refresh Values</b> button.	Use the <b>Refresh Values</b> button for instances in which the parameter table data becomes disabled (for example when page switching occurs),
Use the <b>Hold Updates</b> button and <b>Update All Parameters</b> button to change multiple parameter values simultaneously. These buttons in Explorer operate in the same way as these buttons on the <b>Real-Time</b> tab in the Simulink Editor.	"Tune Parameters by Using Hold Updates and Update All Parameters"

### Monitor Task Execution Time and Target Computer Status

Capability	More Information
Open the <b>TET Monitor</b> tab and monitor task execution time.	Simulink Real-Time TET Monitor
Open the <b>System Log Viewer</b> tab and monitor the target computer system messages.	<p>slrtLogViewer</p> <p>"Target Computer Status Monitor" on page 4-2</p>



## Open the Simulink Real-Time Explorer

From the Simulink Editor, in the **Real-Time** tab, select **Prepare** > **SLRT Explorer**. Or, from the MATLAB Command Window, type:

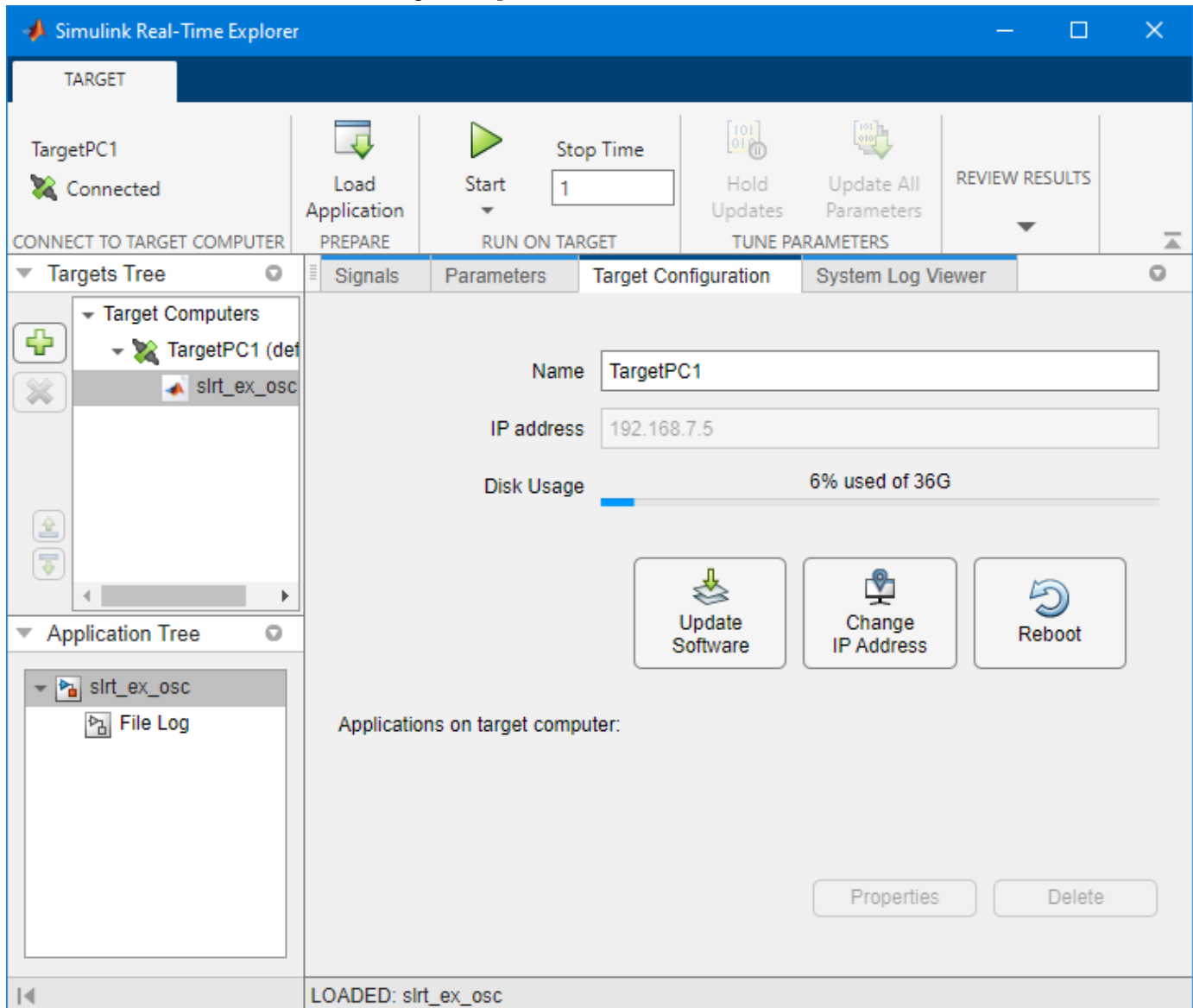
```
slrtExplorer
```

## Examples



## Configure Target, Update, and Reboot

This example shows how to change the IP address of the target computer, update the target computer software, and reboot the target computer.



Open the Simulink Real-Time Explorer.

Select the target computer in the **Targets Tree** panel.

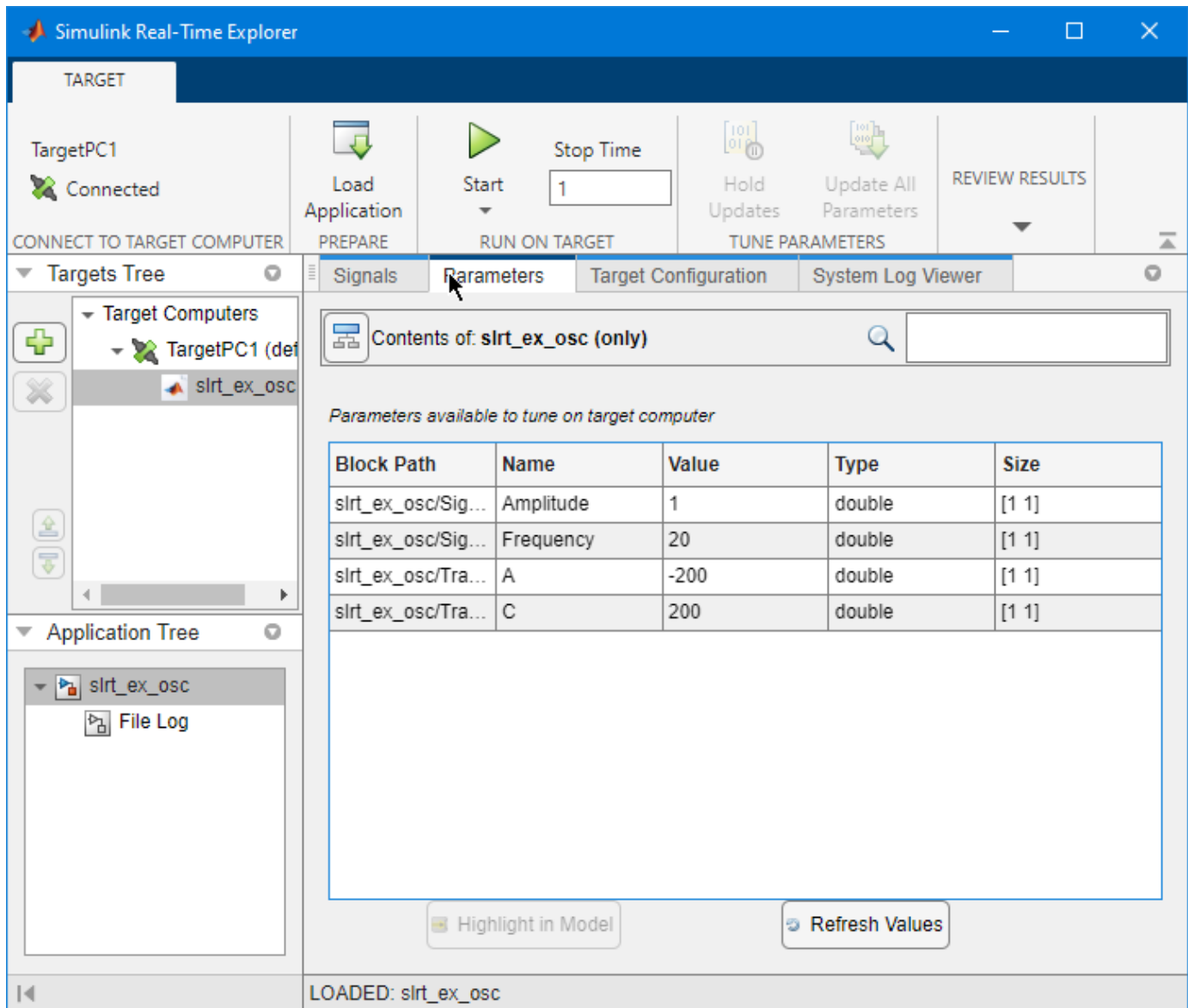
To change the IP address of the target computer, click the **Change IP Address** button.

To update the target computer software, click the **Update Software** button.

To reboot the target computer, click the **Reboot** button.

### Connect, Load Application, and Start

This example shows how to connect to the target computer, load the real-time application, set the stop time, and start the real-time application.



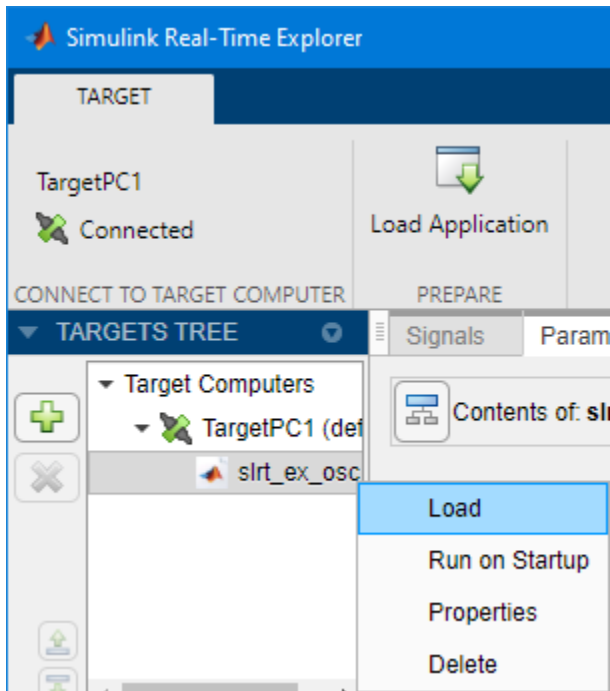
Open the Simulink Real-Time Explorer.

Select the target computer in the **Targets Tree** panel.

To connect to the target computer if not already connected, click **Disconnected** toggling it to **Connected**.

To select and load a real-time application, click **Load Application** and select the MLDATX file.

**Note** You can select and load a real-time application by using the context menu. Right-click on the application and select **Load**.



To select the application stop time, type a value (in seconds) in the **Stop time** field.

To start the application, click the **Start** button.

In Explorer, clicking the **Start** button is equivalent to executing this command for target object `tg`:

```
start(tg, 'ReloadOnStop', true, 'AutoImportFileLog', true)
```

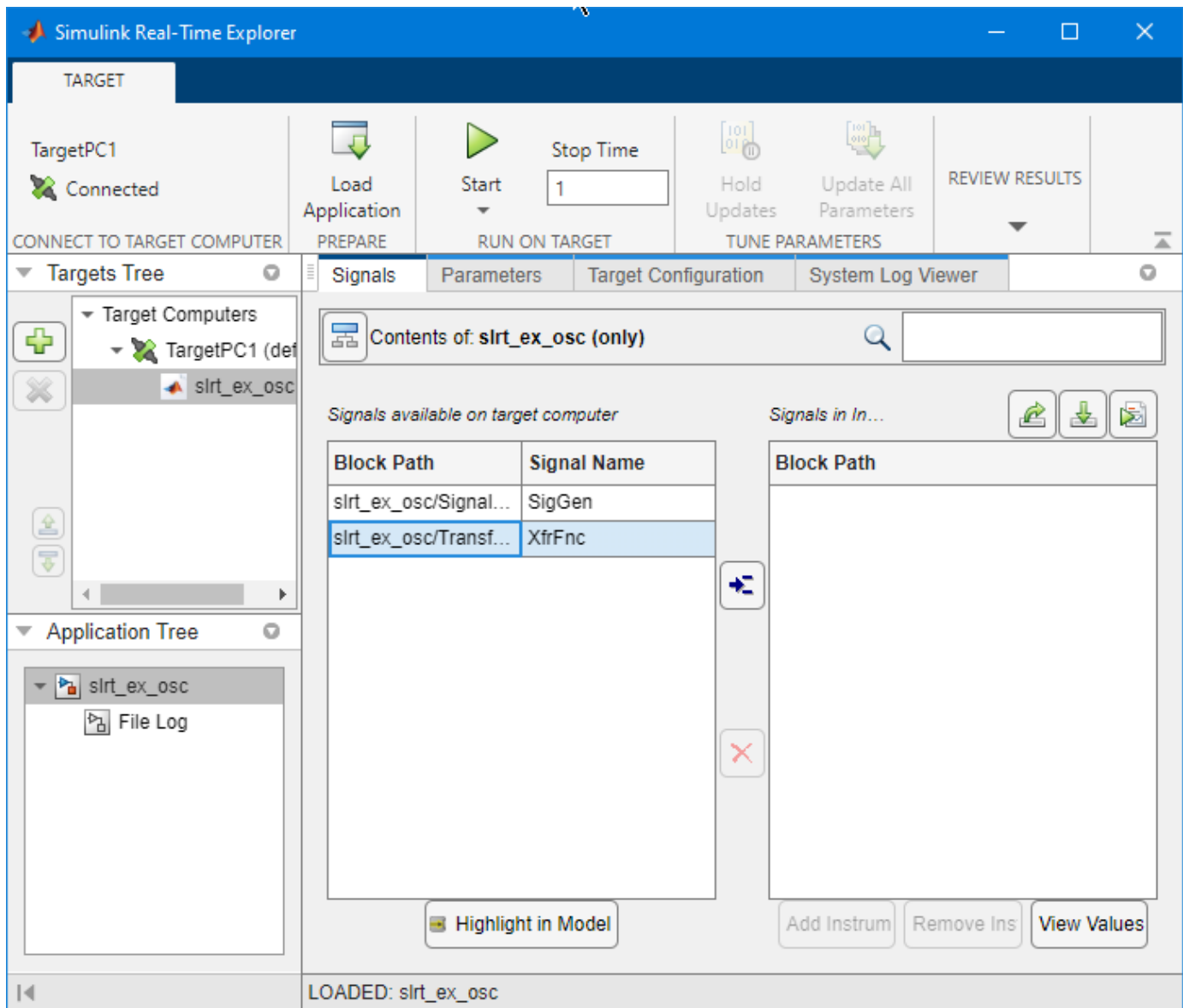
**Note** To change the `ReloadOnStop` and `AutoImportFileLog` operation of the **Start** button, you can:

- Select **Start > ReloadOnStop**
- Select **Start > AutoImportFileLog**.

To stop the real-time application, click the **Stop** button.

### Select Signals and Stream Data to the Simulation Data Inspector

This example shows how to connect to the target computer, load the real-time application, select signals for a signal list, start the real-time application, and view the streaming data in the Simulation Data Inspector.



Open the Simulink Real-Time Explorer.

To connect to the target computer if not already connected, click **Disconnected** toggling it to **Connected**.

To select and load a real-time application, click **Load Application** and select the MLDATX file.

To select signals for streaming, click the application name, select signals from the **Signals** tab, and click the **Add selected signals** button.

To run the application and generate data for streaming, click the **Start** button.

To stream the signal data, select the signals in the **Signals in instrument** list and click the **Add Instrument** button.

To view the streaming signals, click the **Data Inspector** button.

To stop streaming and logging signal data, click the **Stop Recording** button. This button also stops signal logging.

After viewing the data, to stop the real-time application, click the **Stop** button.

## Programmatic Use

`slrtExplorer` opens the Simulink Real-Time Explorer. Operations in the Simulink Real-Time Explorer UI correspond to Simulink Real-Time commands. For example, the explorer **Start** button corresponds to the `start` function.

## Version History

### Introduced in R2020b

#### Added Recording Buttons and Import File Log Button

Added **Start Recording** button and **Stop Recording** button to the **Real-Time** tab in the Simulink Editor and in the Simulink Real-Time Explorer. The **Start Recording** button and **Stop Recording** button have the same functionality as the `startRecording` function and `stopRecording` function.

#### Parameter Table Caching and Parameter Tuning

The **Parameters** tab supports caching parameter table data. By caching the data, updates to parameter data in the table is improved. This improvement is noticeable for real-time applications that have a substantial number of parameters.

The **Parameters** tab supports a **Hold Updates** button and **Update All Parameters** button to change multiple parameter values simultaneously.

#### Disk Usage Display

The target computer disk usage appears on the **Target Configuration** tab and appears on the target computer status monitor.

#### Added Right-Click Menu for Application Options and Add Options to Start Button

The Explorer provides access to real-time application actions and properties through an application context menu and access to real-time application start options through the Start button. When you right-click an application name in the **Targets Tree**, the Explorer displays a menu of actions for the application. The application **Start** button provides `ReloadOnStop` and `AutoImportFileLog` options.

## See Also

`slrtExplorer` | `slrtLogViewer` | `slrtTETMonitor`

**Topics**

“Target Computer Settings”

“Real-Time Application and Target Computer Modes”

“Configure and Control Real-Time Application by Using Simulink Real-Time Explorer”

“Display and Filter Hierarchical Signals and Parameters”

“Monitor Signals by Using Simulink Real-Time Explorer”

“Export and Import Signals in Instrument by Using Simulink Real-Time Explorer”

“Tune Parameters by Using Simulink Real-Time Explorer”

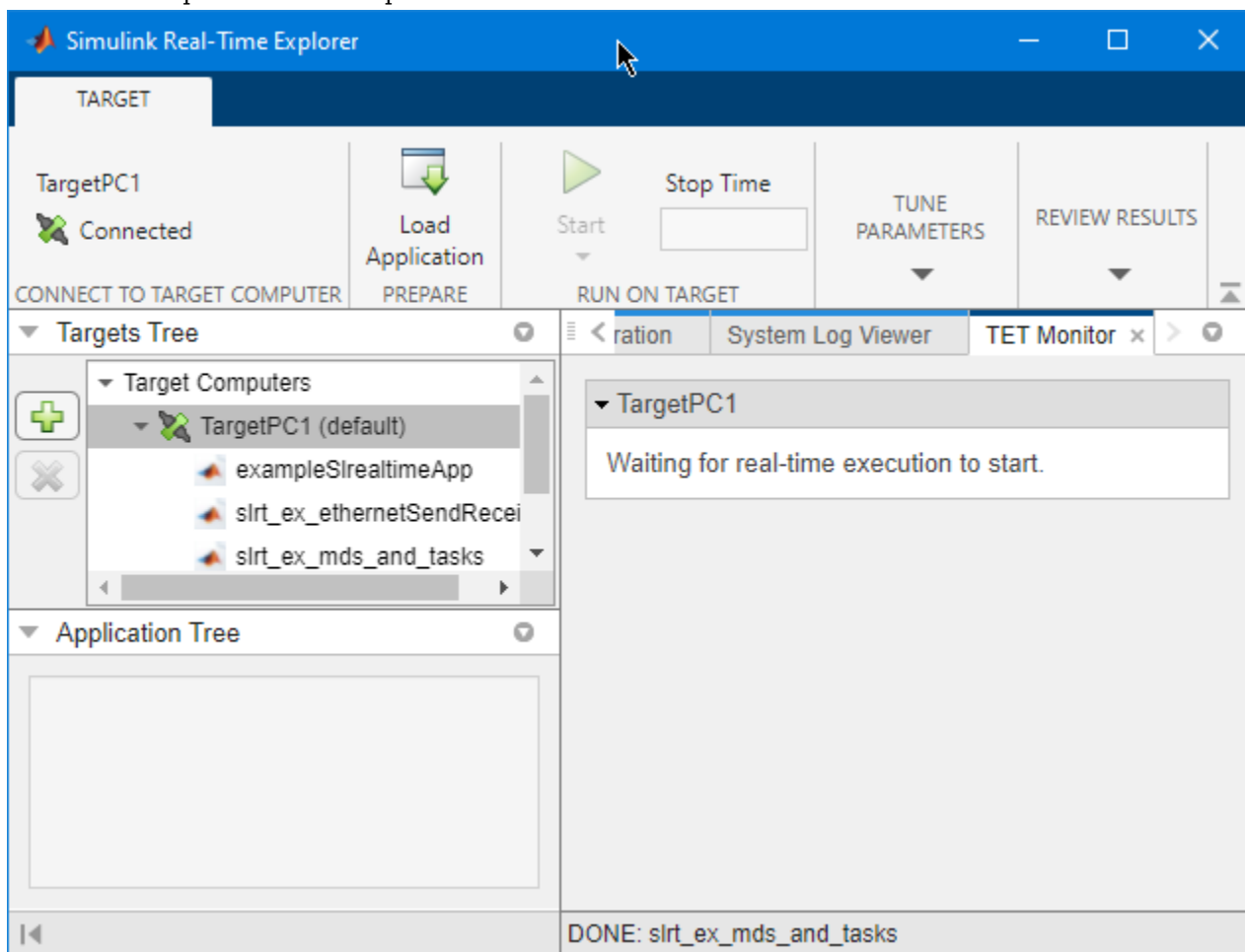
# Simulink Real-Time TET Monitor

Monitor task execution time for the real-time application running on target computer

## Description

Simulink Real-Time Task Execution Time (TET) Monitor lets you view the task execution time for the real-time application running on target computer.

You can open the TET monitor at any time. Depending on the current state of connected target computers, the monitor displays TET data for each real-time application task. Changes to the target computer state are updated in the TET monitor.



## Open the Simulink Real-Time TET Monitor

From the Simulink Editor, in the **Real-Time** tab, select **TET Monitor**. Or, from the MATLAB Command Window, type:

```
slrtTETMonitor
```

## Examples

### Open TET Monitor and View Status

In the “Data Logging with Simulation Data Inspector (SDI)” example, use these additional steps to display the TET monitor.

Open the `slrt_ex_osc` model.

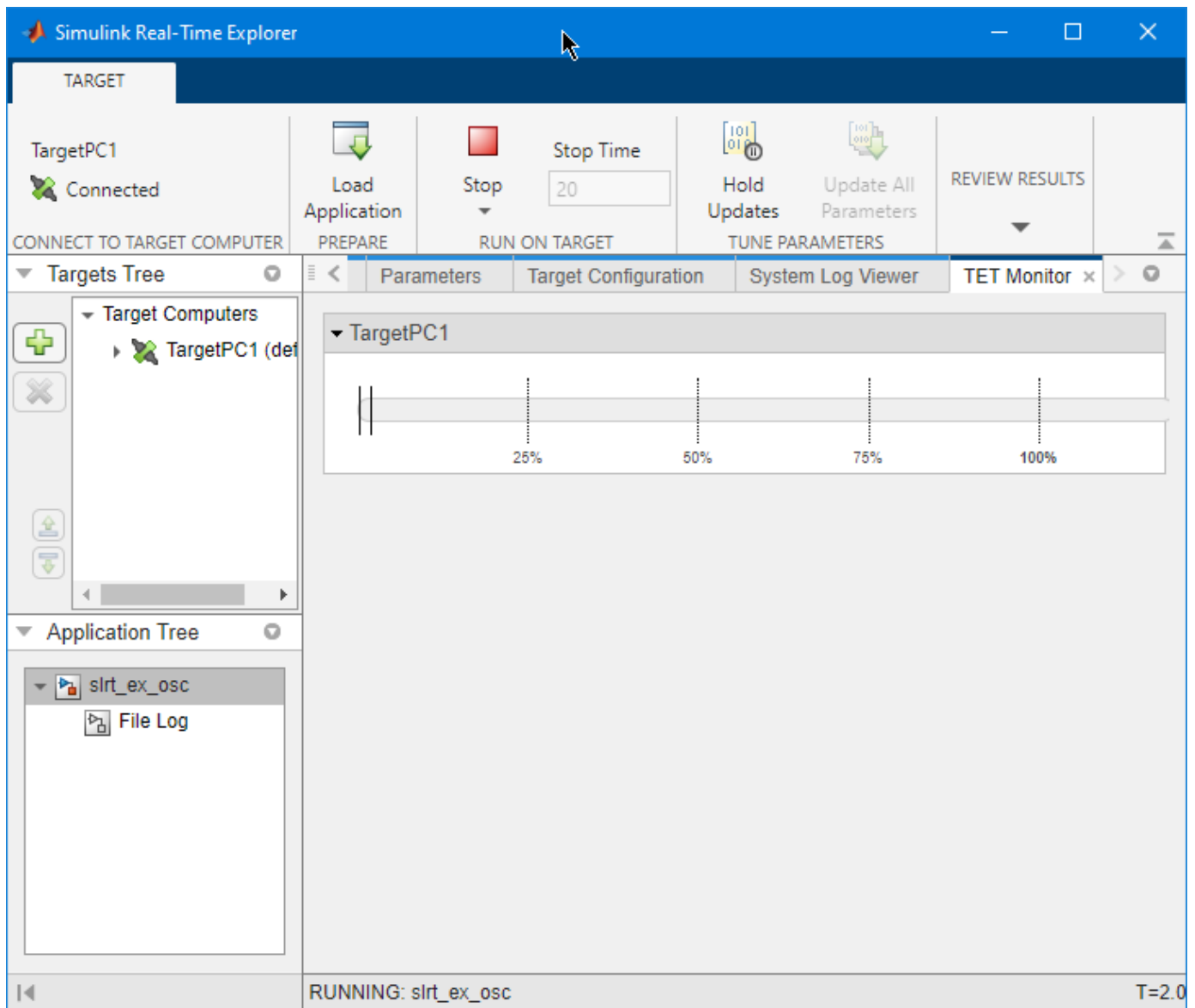
Build the real-time application, load it on the target computer, and start the application. In Simulink Editor **Real-Time** tab, click **Run on Target**.

Open the TET monitor. In the **Real-Time** tab, click **TET Monitor**. Or, in the Command Window, enter:

```
slrtTETMonitor
```

When you run the real-time application, the TET monitor displays status.





## Programmatic Use

`slrtTETMonitor` opens the Simulink Real-Time TET Monitor.

## Version History

Introduced in R2020b

**Added TET Monitor to Explorer**

The TET monitor appears as a tab in Simulink Real-Time Explorer instead of operating as a separate tool. You can open this tab in Simulink Real-Time explorer by using the **TET Monitor** button or by using the `slrtTETMonitor` function to open this tab.

## **See Also**

`slrtTETMonitor` | SLRT Overload Options | `slrtExplorer`

## **Topics**

“Data Logging with Simulation Data Inspector (SDI)”

“Parameter Tuning and Data Logging”

“Real-Time Application and Target Computer Modes”

“Configure and Control Real-Time Application by Using Simulink Real-Time Explorer”

“Execution Profiling for Real-Time Applications”

# Simulink Real-Time App Generator

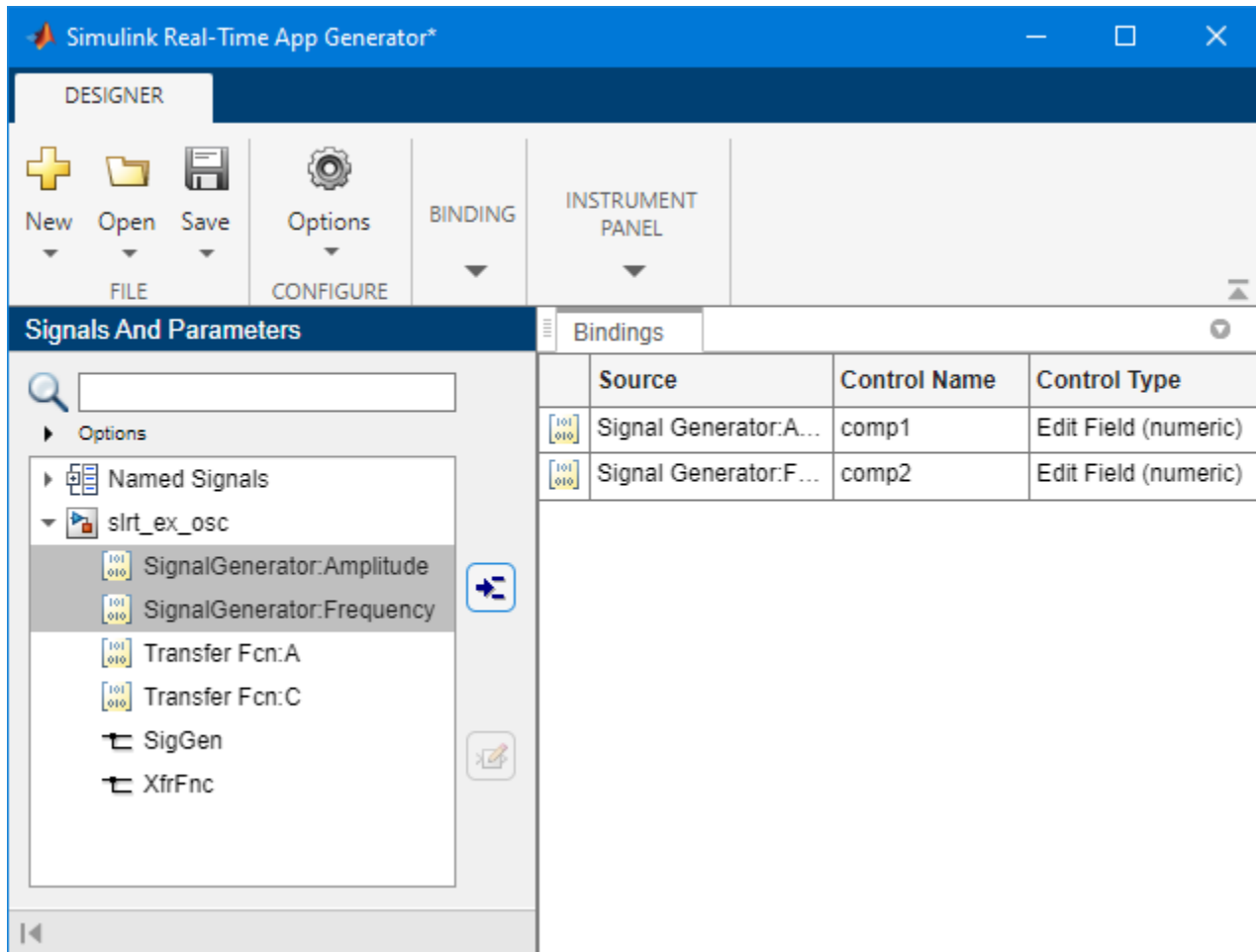
Generate instrument panel app to interact with target computer and real-time application running on target computer

## Description

Simulink Real-Time App Generator helps you generate an instrument panel app that interacts with the target computer and real-time application running on the target computer. You can select signals and parameters in your model to represent as instrument panel controls and configure the controls before generating the app.

To use the Simulink Real-Time App Generator, open the App Generator from the **Real-Time** tab in the Simulink Editor and use the App Generator for these tasks:

- Open a model file SLX or real-time application file MLDATX, and create an instrument panel app.
- Select signals and parameters to add to an instrument panel app.
- Configure controls for instrument panel app.
- Create instrument panel app.
- Save an App Generator session file MAT, and open it in a future App Generator session



## Open the Simulink Real-Time App Generator

From the Simulink Editor, in the **Real-Time** tab, select **Review Results > App Generator**. Or, from the MATLAB Command Window, type:

```
slrtAppGenerator
```

If you open the App Generator from a model, the App Generator populates the **Signals and Parameters** pane with information from the model.

## Examples

### Configure Instrument Panel Controls and Create App

This example shows how to configure instrument panel controls for signals and parameter, then create an instrument panel app by using the App Generator. This example uses example model `slrt_ex_osc`.

Open example model `slrt_ex_osc`. In the MATLAB Command Window, type:

```
open_system(fullfile(matlabroot, 'toolbox', ...  
    'slrealtime', 'examples', 'slrt_ex_osc'))
```

Build the model, creating a real-time application file MLDATX.

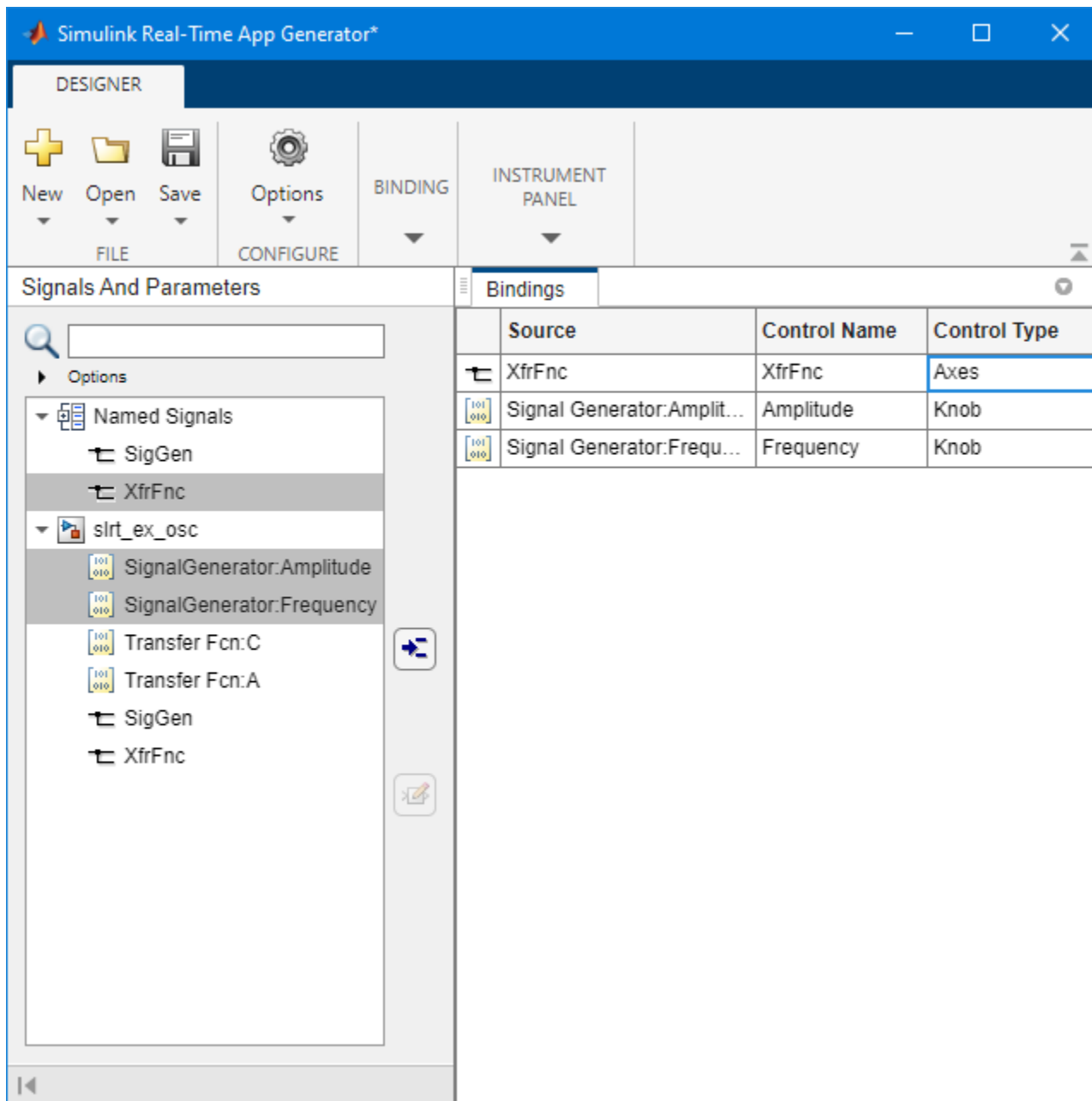
Open the Simulink Real-Time App Generator. In the **Real-Time** tab, select **Review Results > App Generator**.

To create an instrument panel from the real-time application file MLDATX, select **New > New**, click **No** to remove the current session, and select the MLDATX file. For information about the difference between developing an instrument panel from a model SLX file or a real-time application MLDATX file, see the “Tip About MLDATX and SLX Files.” on page 3-0 .

Add signals and parameters to the instrument panel app. From the **Signals and Parameters** pane, select the **Amplitude** parameter, the **Frequency** parameter, and the **XfrFnc** signal. Click the **Add to panel** button.



Configure each control by clicking on its **Control Type** entry and editing the selections for the control. This figure shows a possible configuration for this instrument panel.



To create the instrument panel app, click the **Generate App** button.

After creating the app, you can open it in App Designer to further customize the instrument panel.

The App Generator adds controls to your instrument panel that let the panel interface with the real-time application. These controls include the target computer selector, connect button, load application button, start/stop button, stop time field, and system log. Any instrumented signals from the model are added in an axis component. For more information, see “Create App Designer Instrument Panels by Using Simulink Real-Time Components”.

**Tip About MLDATX and SLX Files.** You can develop an instrument panel app in the App Generator from a model SLX file (if you start the App Generator from the Real-Time tab in the

Simulink Editor) or from a real-time application MLDATX file. It is recommended that you develop the instrument panel based on the MLDATX file, because—when developing from the MLDATX file—the App Generator only lists the signals and parameters that are present in the generated code. If you develop the instrument panel based on the SLX file, the App Generator can list more signals than are present in the generated code. These signals include virtual signals and signals to Scope blocks.

## Open Real-Time Application and Create App

This example shows how to open a real-time application in the App Generator, add signals and parameters to an instrument panel app from the real-time application, and add signals and parameters to the instrument panel app from the model that corresponds to the real-time application..

Open the App Generator. In the MATLAB Command Window, type:

```
slrtAppGenerator
```

To create a new instrument panel app, click the **New** button and select the real-time application file `slrt_ex_osc.mldatx`. You created this file in “Configure Instrument Panel Controls and Create App” on page 3-24.

Add signals and parameters to the instrument panel app. From the **Signals and Parameters** pane, select the Amplitude parameter, the Frequency parameter, and the `XfrFnc` signal. Click the **Add to panel** button.



To add signals and parameters from the model that corresponds to the real-time application, click the **Add From Model** button.

The App Generator opens the model and puts the model in bind mode for signal and parameter selection. For more information about bind mode, see “Add Instruments to Real-Time Application from Simulink Model”.

To return to the App Generator, close bind mode in the model.

To create the instrument panel app, click the **Generate App** button.

- “Create App Designer Instrument Panels by Using App Generator”
- “Create Instrument Panel for Testing Highway Lane Following Controller”

## Programmatic Use

`slrtAppGenerator` opens the Simulink Real-Time App Generator. Operations that the Simulink Real-Time App Generator UI adds to an instrument panel app correspond App Designer controls that are customized for your real-time application.

## Version History

Introduced in R2022a

## **See Also**

`slrtAppGenerator` | `slrtExplorer` | `slrtLogViewer` | `slrtTETMonitor`

## **Topics**

“Create App Designer Instrument Panels by Using App Generator”

“Create Instrument Panel for Testing Highway Lane Following Controller”

“Target Computer Settings”

“Real-Time Application and Target Computer Modes”

“Configure and Control Real-Time Application by Using Simulink Real-Time Explorer”

“Display and Filter Hierarchical Signals and Parameters”

“Monitor Signals by Using Simulink Real-Time Explorer”

“Export and Import Signals in Instrument by Using Simulink Real-Time Explorer”

“Tune Parameters by Using Simulink Real-Time Explorer”



# Simulation Data Inspector

Inspect and compare data and simulation results to validate and iterate model designs

## Description

The Simulation Data Inspector visualizes and compares multiple kinds of data.

Using the Simulation Data Inspector, you can inspect and compare time series data at multiple stages of your workflow. This example workflow shows how the Simulation Data Inspector supports all stages of the design cycle:

**1** “View Data in the Simulation Data Inspector”

Run a simulation in a model configured to log data to the Simulation Data Inspector, or import data from the workspace or a MAT-file. You can view and verify model input data or inspect logged simulation data while iteratively modifying your model diagram, parameter values, or model configuration.

**2** “Inspect Simulation Data”

Plot signals on multiple subplots, zoom in and out on specified plot axes, and use data cursors to understand and evaluate the data. “Create Plots Using the Simulation Data Inspector” to tell your story.

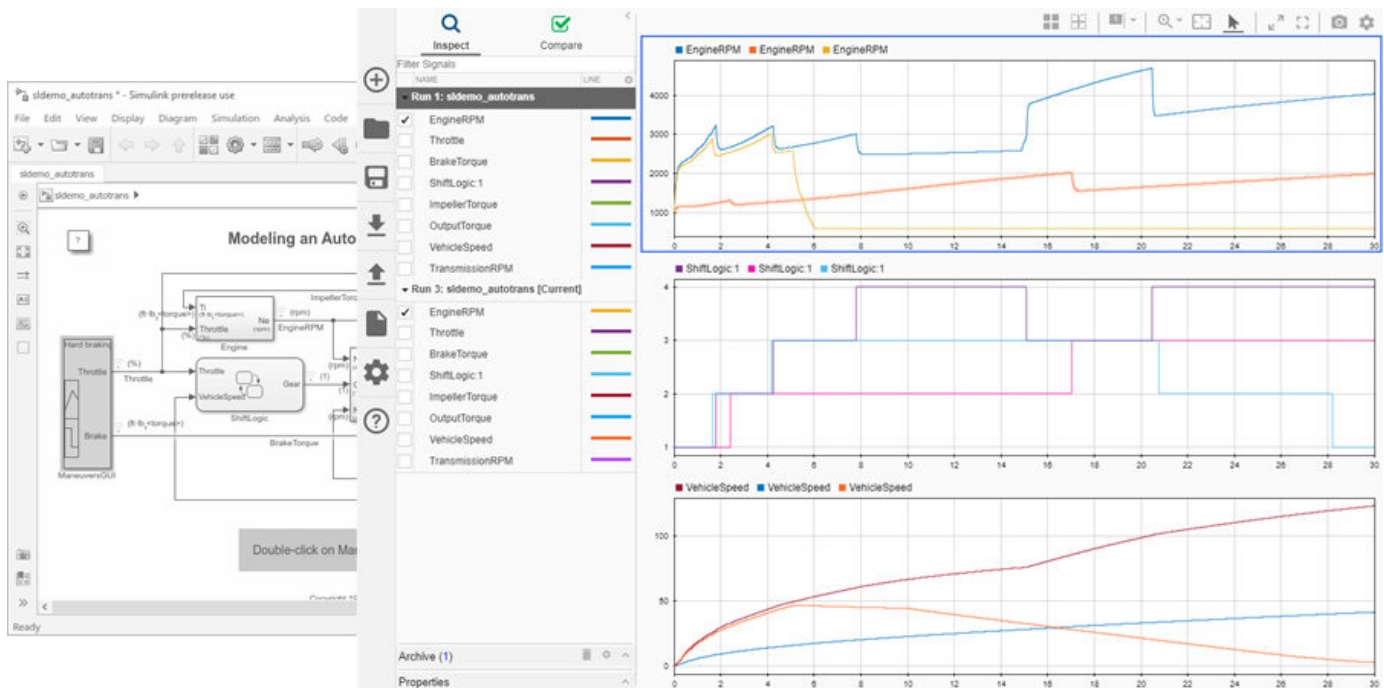
**3** “Compare Simulation Data”

Compare individual signals or simulation runs and analyze your comparison results with relative, absolute, and time tolerances. The compare tools in the Simulation Data Inspector facilitate iterative design and allow you to highlight signals that do not meet your tolerance requirements. For more information about the comparison operation, see “How the Simulation Data Inspector Compares Data”.

**4** “Save and Share Simulation Data Inspector Data and Views”

Share your findings with others by saving Simulation Data Inspector data and views.

You can also harness the capabilities of the Simulation Data Inspector from the command line. For more information, see “Inspect and Compare Data Programmatically”.



## Open the Simulation Data Inspector

- Simulink Toolstrip: On the **Simulation** tab, under **Review Results**, click **Data Inspector**.
- Click the streaming badge on a signal to open the Simulation Data Inspector and plot the signal.
- MATLAB command prompt: Enter `Simulink.sdi.view`.

## Examples

### Apply a Tolerance to a Signal in Multiple Runs

You can use the Simulation Data Inspector programmatic interface to modify a parameter for the same signal in multiple runs. This example adds an absolute tolerance of  $0.1$  to a signal in all four runs of data.

First, clear the workspace and load the Simulation Data Inspector session with the data. The session includes logged data from four simulations of a Simulink® model of a longitudinal controller for an aircraft.

```
Simulink.sdi.clear
Simulink.sdi.load('AircraftExample.mldatx');
```

Use the `Simulink.sdi.getRunCount` function to get the number of runs in the Simulation Data Inspector. You can use this number as the index for a for loop that operates on each run.

```
count = Simulink.sdi.getRunCount;
```

Then, use a for loop to assign the absolute tolerance of  $0.1$  to the first signal in each run.

```
for a = 1:count
    runID = Simulink.sdi.getRunIDByIndex(a);
    aircraftRun = Simulink.sdi.getRun(runID);
    sig = getSignalByIndex(aircraftRun,1);
    sig.AbsTol = 0.1;
end
```

- “View Data in the Simulation Data Inspector”
- “Inspect Simulation Data”
- “Compare Simulation Data”
- “Iterate Model Design Using the Simulation Data Inspector”

## Programmatic Use

`Simulink.sdi.view` opens the Simulation Data Inspector from the MATLAB command line.

## Version History

Introduced in R2010b

## See Also

### Functions

`Simulink.sdi.clear` | `Simulink.sdi.clearPreferences` | `Simulink.sdi.snapshot`

### Topics

“View Data in the Simulation Data Inspector”

“Inspect Simulation Data”

“Compare Simulation Data”

“Iterate Model Design Using the Simulation Data Inspector”



# Target Computer Status Monitor

---

## Target Computer Status Monitor

The status monitor application on the target computer displays the status of the real-time application, disk usage, and other target computer status information.

The target computer display supports multiple sessions. You can choose to display the status monitor (default, session 1) or display the target computer command-line interface (session 2).

### Display Status Monitor

Start the target computer.

The target computer displays session 1 (default) and the target computer status monitor.

```
State: BUSY -> logTest (LOADED)
Execution Time (Current/Stop): 0.0s / 10.0s
Disk Usage: 43.5% used of 10.0 GB
Overruns (Current/Max): 0/0
Task Execution Time (Rate: Current/Max)
TET_1.000e+00: -1.000e+00s / -1.000e+00s

--LOG-----
09:06:43.188662 [info    ] Loading model logTest
09:06:43.463662 [trace  ] Root Level Input Service disabled
09:06:43.464662 [trace  ] Service initialized
09:06:43.479662 [debug  ] LoggingService::initialize no signals to log
09:06:43.482662 [trace  ] This is a test for trace message
09:06:43.482662 [debug  ] This is a test for debug message
09:06:43.482662 [info   ] This is a test for info message
09:06:43.482662 [warning] This is a test for warning message
09:06:43.482662 [error  ] This is a test for error message
09:06:43.482662 [fatal  ] This is a test for fatal message
09:06:43.483662 [info   ] Let's log a string which is long enough to exceed the
screen width. Is displayed correctly on the target screen.
09:06:43.484662 [info   ] Ready to start
```

### Display Status Monitor by Using PuTTY

To view the status monitor from the development computer, use PuTTY to open an SSH client and start the status monitor application `statusmonitor` on the target computer. Keyboard commands for the status monitor include:

- Q (quit)
- Up arrow (scroll up in the log)
- Down arrow (scroll down in the log)

For more information about PuTTY, see “Execute Target Computer RTOS Commands at Target Computer Command Line”.

To display the target computer command-line interface, switch to display session 2:

- 1 Start the target computer.

The target computer displays session 1 and the target computer status monitor.

- 2** To switch to session 2 and use the target computer command-line interface, on the target computer keyboard (console), press **Ctrl+Alt+2**.
- 3** To switch back to session 1 (status monitor), on the target computer keyboard (console), press **Ctrl+Alt+1**.





# Target Computer Command-Line Interface Reference

---

## Target Computer Command-Line Interface

You can load, run, stop, and check the status of a real-time application by using the target computer command-line interface commands.

By default, the target computer displays the session 1 screen with the target computer status monitor. For information about switching to the session 2 screen with the command-line interface, see “Target Computer Status Monitor” on page 4-2.

To read the target computer console log, view the log in the `slrtLogViewer`.

### Target Object Commands

When you are using the target computer command-line interface, target object functions support loading, starting, stopping, and checking the status of the real-time application.

For a description of how to use these commands, see “Control Real-Time Application at Target Computer Command Line”.

---

**Note** To run user commands, log in as user `slrt` by using password `slrt`. To run the system commands (for example, `date`, `ntdate`, `ntpd`, `rtc`, or setting the time zone), login as user `root` by using password `root`.

---

These commands are Target object commands that you can use through the command-line interface on the target computer. Each command appears with its equivalent MATLAB syntax. In the descriptions, `tg_object` is the target object name, and `app_name` is the real-time application MLDATX file name.

- **Target:** `slrealtime listApplications`

**MATLAB:** `getInstalledApplications(tg_object)`

When run from the development computer in the MATLAB Command Window, the `getInstalledApplications` command returns a list of the real-time application that are installed on the target computer.

- **Target:** `slrealtime load --AppName app_name`

**MATLAB:** `load(tg_object, 'app_name')`

When run from the development computer in the MATLAB Command Window, the `load` command deploys the real-time application to the target computer and loads the application. When run from the target computer command interface, the `load` command loads the application.

- **Target:** `slrealtime start`

**MATLAB:** `start(tg_object)`

The `start` command runs the real-time application that is loaded on the target computer.

- **Target:** `slrealtime stop`

**MATLAB:** `stop(tg_object)`

The stop command stops the real-time application that is running on the target computer.

- **Target:** `slrealtime install --AppName app_name`

**MATLAB:** `install(tg_object, 'app_name')`

The install command installs the real-time application for standalone operation on the target computer.

- **Target:** `slrealtime saveParamSet filename`

**MATLAB:** `saveParamSet(tg_object, filename)`

The saveParamSet command saves the parameter set from the loaded the real-time application on the target computer.

- **Target:** `slrealtime loadParamSet filename`

**MATLAB:** `loadParamSet(tg_object, filename)`

The loadParamSet command loads the parameter set into the real-time application on the target computer.

- **Target:** `shutdown -S reboot`

**MATLAB:** `reboot(tg_object)`

The reboot command reboots the target computer.

If you prefer to safely shutdown the RTOS before turning off power to the target computer, you can use the command: `shutdown -S system`

## Target Computer RTOS System Commands

The target computer uses the QNX Neutrino Real-Time Operating System (RTOS). You can run system commands on the target computer from the development computer by using an SSH utility, such as PuTTY. Or, you can run system commands on the target computer from its keyboard (console). Target computer RTOS system command information is available in the Utilities Reference in the QNX Momentics IDE 7.1 User's Guide. All commands that this reference identifies as **Runs on: QNX Neutrino** are supported on the target computer.

Some RTOS commands are required for configuring the target computer. These commands include:

- `date` — set date and time
- `ntdate` — set the local date and time from NTP server
- `ntpd` — start NTP daemon
- `rtc` — set date from hardware clock

---

**Note** To run user commands, log in as user `slrt` by using password `slrt`. To run the system commands (for example, `date`, `ntdate`, `ntpd`, `rtc`, or setting the time zone), login as user `root` by using password `root`.

---

For a description of how to use these commands, see “Execute Target Computer RTOS Commands at Target Computer Command Line”.

## **See Also**

### **More About**

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

### **External Websites**

- QNX Momentics IDE 7.1 User’s Guide
- QNX Momentics IDE 7.1 User’s Guide, Utilities Reference