

**Simulink® Real-Time™**

Reference



**MATLAB® & SIMULINK®**

R2021a



## 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 2002–2021 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)



1	<b>Configuration Parameters</b>
<b>Simulink Real-Time Options Pane</b> .....	1-2
Configuration .....	1-2
Tips .....	1-2
To get help on an option .....	1-2
<b>Log level</b> .....	1-4
Settings .....	1-4
Command-Line Information .....	1-4
Related Real-Time Application Option .....	1-4
<b>Force polling mode</b> .....	1-5
Settings .....	1-5
Command-Line Information .....	1-5
Related Real-Time Application Option .....	1-5
<b>Max file log runs</b> .....	1-6
Settings .....	1-6
Command-Line Information .....	1-6
Related Real-Time Application Option .....	1-6
<b>Compile with GCC -ffast-math</b> .....	1-7
Settings .....	1-7
Command-Line Information .....	1-7

2	<b>Configuration Parameters (slrt.tlc OBSOLETE)</b>
<b>Simulink Real-Time Options Pane (slrt.tlc OBSOLETE)</b> .....	2-2
Configuration .....	2-2
Tips .....	2-3
To get help on an option .....	2-3
<b>Execution mode (slrt.tlc OBSOLETE)</b> .....	2-4
Settings .....	2-4
Command-Line Information .....	2-4
<b>Real-time interrupt source (slrt.tlc OBSOLETE)</b> .....	2-5
Settings .....	2-5
Tips .....	2-5
Command-Line Information .....	2-5

<b>I/O board generating the interrupt (slrt.tlc OBSOLETE)</b> .....	2-6
Settings .....	2-6
Command-Line Information .....	2-6
<b>PCI slot (-1: autosearch) or ISA base address (slrt.tlc OBSOLETE)</b> .....	2-7
Settings .....	2-7
Tip .....	2-7
Command-Line Information .....	2-7
<b>Monitor Task Execution Time (slrt.tlc OBSOLETE)</b> .....	2-8
Settings .....	2-8
Command-Line Information .....	2-8
<b>Signal logging data buffer size in doubles (slrt.tlc OBSOLETE)</b> .....	2-9
Settings .....	2-9
Tips .....	2-9
Command-Line Information .....	2-9
<b>Double buffer parameter changes (slrt.tlc OBSOLETE)</b> .....	2-10
Settings .....	2-10
Tips .....	2-10
Command-Line Information .....	2-10
<b>Load a parameter set from a file on the designated target file system (slrt.tlc OBSOLETE)</b> .....	2-11
Settings .....	2-11
Dependencies .....	2-11
Command-Line Information .....	2-11
<b>File name (slrt.tlc OBSOLETE)</b> .....	2-12
Settings .....	2-12
Tip .....	2-12
Dependencies .....	2-12
Command-Line Information .....	2-12
<b>Generate INCA/CANape extensions (disables the Simulation Data Inspector and Dashboard blocks) (slrt.tlc OBSOLETE)</b> .....	2-13
Settings .....	2-13
Command-Line Information .....	2-13
<b>Enable Stateflow animation (slrt.tlc OBSOLETE)</b> .....	2-14
Settings .....	2-14
Command-Line Information .....	2-14
<b>Arm when connecting to target (slrt.tlc OBSOLETE)</b> .....	2-15
Settings .....	2-15
Command-Line Information .....	2-15

## TLC Options Parameters

### 3

<b>TLC Command-Line Options</b> .....	3-2
---------------------------------------	-----

**4**

**Target Computer Status Monitor**

**5**

<b>Target Computer Status Monitor</b> .....	<b>5-2</b>
Display Status Monitor .....	<b>5-2</b>
Display Status Monitor by Using PuTTY .....	<b>5-2</b>

**Target Computer Command-Line Interface Reference**

**6**

<b>Target Computer Command-Line Interface</b> .....	<b>6-2</b>
Target Object Commands .....	<b>6-2</b>
Target Computer RTOS System Commands .....	<b>6-3</b>





# Configuration Parameters

---

- “Simulink Real-Time Options Pane” on page 1-2
- “Log level” on page 1-4
- “Force polling mode” on page 1-5
- “Max file log runs” on page 1-6
- “Compile with GCC -ffast-math” on page 1-7

## Simulink Real-Time Options Pane

Parameter	Description
<b>Log level</b> on page 1-4	Selects filtering level that limits RTOS system messages that appear in the system log.
<b>Force polling mode</b> on page 1-5	Enables polling mode —instead of interrupt-driven mode— for clocking the real-time application.
<b>Max file log runs</b> on page 1-6	Selects the number of file log run to retain when logs are stored on the target computer.
<b>Compile with GCC -ffast-math</b> on page 1-7	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 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`.

### Settings

**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.

...

### Command-Line Information

**Parameter:** `SLRTLogLevel`

**Type:** character vector

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

**Default:** `'info'`

### Related Real-Time Application Option

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`

## Force polling mode

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.

### Settings

**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.

### Command-Line Information

**Parameter:** SLRTForcePollingMode

**Type:** character vector

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

**Default:** 'off'

### Related Real-Time Application Option

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

## Max file log runs

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.

### Settings

**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.

### Command-Line Information

**Parameter:** SLRTFileLogMaxRuns

**Type:** int

**Value:** int

**Default:** 1

### Related Real-Time Application Option

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

**Option:** fileLogMaxRuns

## Compile with GCC -ffast-math

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/)

### Settings

**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.

### Command-Line Information

**Type:** character vector

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

**Default:** 'off'





# Configuration Parameters (slrt.tlc OBSOLETE)

---

- “Simulink Real-Time Options Pane (slrt.tlc OBSOLETE)” on page 2-2
- “Execution mode (slrt.tlc OBSOLETE)” on page 2-4
- “Real-time interrupt source (slrt.tlc OBSOLETE)” on page 2-5
- “I/O board generating the interrupt (slrt.tlc OBSOLETE)” on page 2-6
- “PCI slot (-1: autosearch) or ISA base address (slrt.tlc OBSOLETE)” on page 2-7
- “Monitor Task Execution Time (slrt.tlc OBSOLETE)” on page 2-8
- “Signal logging data buffer size in doubles (slrt.tlc OBSOLETE)” on page 2-9
- “Double buffer parameter changes (slrt.tlc OBSOLETE)” on page 2-10
- “Load a parameter set from a file on the designated target file system (slrt.tlc OBSOLETE)” on page 2-11
- “File name (slrt.tlc OBSOLETE)” on page 2-12
- “Generate INCA/CANape extensions (disables the Simulation Data Inspector and Dashboard blocks) (slrt.tlc OBSOLETE)” on page 2-13
- “Enable Stateflow animation (slrt.tlc OBSOLETE)” on page 2-14
- “Arm when connecting to target (slrt.tlc OBSOLETE)” on page 2-15

## Simulink Real-Time Options Pane (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

Parameter	Description
<b>Execution mode</b>	Specify execution mode of downloaded code.
<b>Real-time interrupt source</b>	Select a real-time interrupt source from the I/O board.
<b>I/O board generating the interrupt</b>	Specify the board interrupt source.
<b>PCI slot (-1: autosearch) or ISA base address</b>	Enter the slot number or base address for the I/O board generating the interrupt.
<b>Monitor Task Execution Time</b>	Monitor task execution times and append the results to the target object property <code>tg.TETlog</code> .
<b>Signal logging data buffer size in doubles</b>	Enter the maximum number of sample points that the software stores before wrapping.
<b>Double buffer parameter changes</b>	Use a double buffer for parameter tuning. This setting enables parameter tuning so that the process of changing parameters in the real-time application uses a double buffer.
<b>Load a parameter set from a file on the designated target file system</b>	Automatically load a parameter set from a file on the designated target computer file system.
<b>File name</b>	Specify the target computer file name from which to load the parameter set.
<b>Generate INCA/CANape extensions</b>	Enable real-time applications to generate data, such as A2L data, for Vector CANape® and ETAS® Inca.
<b>Enable Stateflow animation</b>	Enables visualization of Stateflow® chart animation.

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 `slrt.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 `slrt.tlc`. To retain the updated setting, save the updated model.

---

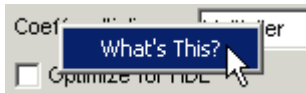
- 2 Select C for the **Language** parameter on the code generation pane.

## 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**.



## Execution mode (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Specify execution mode of downloaded code.

### Settings

**Default:** Real-Time

Real-Time

Executes downloaded code as a real-time application.

Freerun

Executes downloaded code as fast as possible.

Multirate models cannot be executed in Freerun execution mode. On the **Solver** pane in the Configuration Parameters dialog box, clear the check box for **Treat each discrete rate as a separate task**.

### Command-Line Information

**Parameter:** RL32ModeModifier

**Type:** character vector

**Value:** 'Real-Time' | 'Freerun'

**Default:** 'Real-Time'

## Real-time interrupt source (slrt.tlc OBSOLETE)

---

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Select a real-time interrupt source from the I/O board.

### Settings

**Default:** Timer

Timer

Specifies that the board interrupt source is a timer.

Auto (PCI only)

Enables the Simulink Real-Time software to automatically determine the IRQ that the BIOS assigned to the board and use it.

3 to 15

Specifies that the board interrupt source is an IRQ number on the board.

### Tips

- The Auto (PCI only) option is available only for PCI boards. If you have an ISA board (PC/104 or onboard parallel port), set the IRQ manually.
- The Simulink Real-Time software treats PCI parallel port plugin boards like ISA boards. For PCI parallel port plugin boards, set the IRQ manually.
- Multiple boards can share an interrupt number.

### Command-Line Information

**Parameter:** RL32IRQSourceModifier

**Type:** character vector

**Value:** 'Timer' | Auto (PCI only) | '3' | '4' | '5' | '6' | '7' | '8' | '9' | '10' | '11' | '12' | '13' | '14' | '15'

**Default:** 'Timer'

## I/O board generating the interrupt (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Specify the board interrupt source.

### Settings

**Default:** None/Other

Bitflow NEON

Specifies that the interrupt source is the BitFlow™ NEON video board.

GE\_Fanuc(VMIC)\_PCI-5565

Specifies that the interrupt source is the GE® Fanuc VMIC PCI-5565 board.

General Standards 24DSI12

Specifies that the interrupt source is the General Standards 24DSI12 board.

Parallel\_Port

Specifies that the interrupt source is the parallel port of the target computer.

Speedgoat\_IO321

Specifies that the interrupt source is the Speedgoat IO321 FPGA board.

Speedgoat\_IO331

Specifies that the interrupt source is the Speedgoat IO331 FPGA board.

Speedgoat\_IO333

Specifies that the interrupt source is the Speedgoat IO333 FPGA board.

None/Other

Specifies that the I/O board has no interrupt source.

### Command-Line Information

**Parameter:** xPCIRQSourceBoard

**Type:** character vector

**Value:** 'Bitflow NEON' |  
'GE\_Fanuc(VMIC)\_PCI-5565' |  
'General Standards 24DSI12' |  
'Parallel\_Port' |  
'Speedgoat\_IO321' |  
'Speedgoat\_IO331' |  
'Speedgoat\_IO333' |  
'None/Other'

**Default:** 'None/Other'

## PCI slot (-1: autosearch) or ISA base address (slrt.tlc OBSOLETE)

---

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Enter the slot number or base address for the I/O board generating the interrupt.

### Settings

**Default:** -1

The PCI slot can be either -1 (let the Simulink Real-Time software determine the slot number) or of the form `[bus, slot]`.

The base address is a hexadecimal number of the form `0x300`.

### Tip

To determine the bus and PCI slot number of the boards in the target computer, in the Command Window, type:

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

### Command-Line Information

**Parameter:** `xPCIOIRQSlot`

**Type:** character vector

**Value:** '-1' | hexadecimal value

**Default:** '-1'

## Monitor Task Execution Time (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Monitor task execution times and append the results to the target object property `tg.TETlog`.

Task execution time (TET) measures how long it takes the RTOS to run for one base-rate time step. For a multirate model, use the profiler to find out what the execution time is for each rate.

### Settings

**Default:** on

On

Monitors task execution times and appends the results to the target object property `tg.TETlog`.

Off

Does not monitor task execution times.

### Command-Line Information

**Parameter:** `RL32LogTETModifier`

**Type:** character vector

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

**Default:** 'on'



## Signal logging data buffer size in doubles (slrt.tlc OBSOLETE)

---

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Enter the maximum number of sample points that the software stores before wrapping.

### Settings

**Default:** 100000

The maximum value for this option cannot exceed the available target computer memory, which the Simulink Real-Time software also uses to hold other items.

### Tips

- Real-time applications use this buffer to store the time, states, outputs, and task execution time (TET) logs as defined in the Simulink model.
- The maximum value for this option derives from available target computer memory, which the Simulink Real-Time software also uses to hold other items. For example, in addition to signal logging data, the software also uses the target computer memory for the Simulink Real-Time RTOS, real-time application, and scopes.

For example, assume that your model has six data items (time, two states, two outputs, and task execution time). If you enter a buffer size of 100000, the target object property `tg.MaxLogSamples` is calculated as  $\text{floor}(100000 / 6) = 16666$ . After the buffer saves 16666 sample points, it wraps and further samples overwrite the older ones.

- Suppose that you enter a logging buffer size larger than the available RAM on the target computer. When you download and initialize the real-time application, the target computer displays a message, `ERROR: allocation of logging memory failed`. To avoid this error, either install more RAM or reduce the buffer size for logging, and then restart the target computer. To calculate the maximum buffer size available for your real-time application logs, divide the amount of available RAM on your target computer by `sizeof(double)`, or 8. Enter that value for the **Signal logging data buffer size in doubles** value.

### Command-Line Information

**Parameter:** `RL32LogBufSizeModifier`

**Type:** character vector

**Value:** `'100000'` | A valid memory size

**Default:** `'100000'`

# Double buffer parameter changes (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Use a double buffer for parameter tuning. This setting enables parameter tuning so that the process of changing parameters in the real-time application uses a double buffer.

## Settings

**Default:** off



On

Changes parameter tuning to use a double buffer.



Off

Suppresses double buffering of parameter changes in the real-time application.

## Tips

- When a parameter change request is received, the new value is compared to the old one. If the new value is identical to the old one, it is discarded, and if different, it is queued.
- At the start of execution of the next sample of the real-time task, the queued parameters are updated. This operation increases the task execution time (TET) and can cause a CPU overload error.
- Double buffering leads to a more robust parameter tuning interface, but it increases task execution time and the higher probability of overloads. Under typical conditions, keep double buffering off (default).
- If the real-time application contains MATLAB variables as the value of block parameters, the software ignores this double buffering setting. Normal parameter tuning occurs.

## Command-Line Information

**Parameter:** `xpcDbfBuff`

**Type:** character vector

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

**Default:** 'off'

## Load a parameter set from a file on the designated target file system (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Automatically load a parameter set from a file on the designated target computer file system.

### Settings

**Default:** off

On

Enable the automatic loading of a parameter set from the file specified by **File name** on the designated target computer file system.

Off

Suppress the automatic loading of a parameter set from a file on the designated target computer file system.

### Dependencies

This parameter enables **File name**.

### Command-Line Information

**Parameter:** `xPCLoadParamSetFile`

**Type:** character vector

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

**Default:** `'off'`

## File name (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Specify the target computer file name from which to load the parameter set.

### Settings

''

### Tip

If the named file does not exist, the software loads the parameter set built with the model.

### Dependencies

To enable this parameter, set **Load a parameter set from a file on the designated target file system**.

### Command-Line Information

**Parameter:** `xPC0nTgtParamSetFileName`

**Type:** character vector

**Value:** A valid file name

**Default:** ''

## Generate INCA/CANape extensions (disables the Simulation Data Inspector and Dashboard blocks) (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Enable real-time applications to generate data, such as A2L data, for Vector CANape and ETAS Inca.

### Settings

**Default:** off

On

Enables real-time applications to generate data, such as that for A2L, for Vector CANape and ETAS Inca.

Off

Does not enable real-time applications to generate data, such as that for A2L, for Vector CANape and ETAS Inca.

### Command-Line Information

**Parameter:** GenerateASAP2

**Type:** character vector

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

**Default:** 'off'

## Enable Stateflow animation (slrt.tlc OBSOLETE)

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Enables visualization of Stateflow chart animation.

### Settings

**Default:** off



On

Enables visualization of Stateflow chart animation.



Off

Disables visualization of Stateflow chart animation.

### Command-Line Information

**Parameter:** `xPCEnableSFAnimation`

**Type:** character vector

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

**Default:** 'off'

## Arm when connecting to target (slrt.tlc OBSOLETE)

---

**Note** The system target file `slrt.tlc` is obsolete for R2020b. The configuration parameters for that code generation target are not supported.

---

Whether a button or a signal triggers data uploading (as defined by **Source**), the trigger must be armed to allow data uploading to begin.

If the trigger **Source** is signal, monitoring of the trigger signal begins immediately. Data uploading begins when the trigger signal satisfies trigger conditions (as defined in the Trigger signal section).

If you clear **Arm when connecting to target**, manually arm the trigger by clicking the Arm **Trigger** button on the External Mode Control Panel.

### Settings

**Default:** off



On

Connecting to the target arms the trigger.



Off

Data uploading begins immediately after clicking the **Arm Trigger** button on the External Mode Control Panel.

### Command-Line Information

**Parameter:** ExtModeArmWhenConnect

**Type:** character vector

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

**Default:** 'off'





# 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.

---

# Simulink Real-Time Tools and Applications

---

## 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.

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

### Target Computer Configuration

Capability	More Information
Configure target computer configuration settings.	"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> </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> </ul>

### 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”

## Open the Simulink Real-Time Explorer

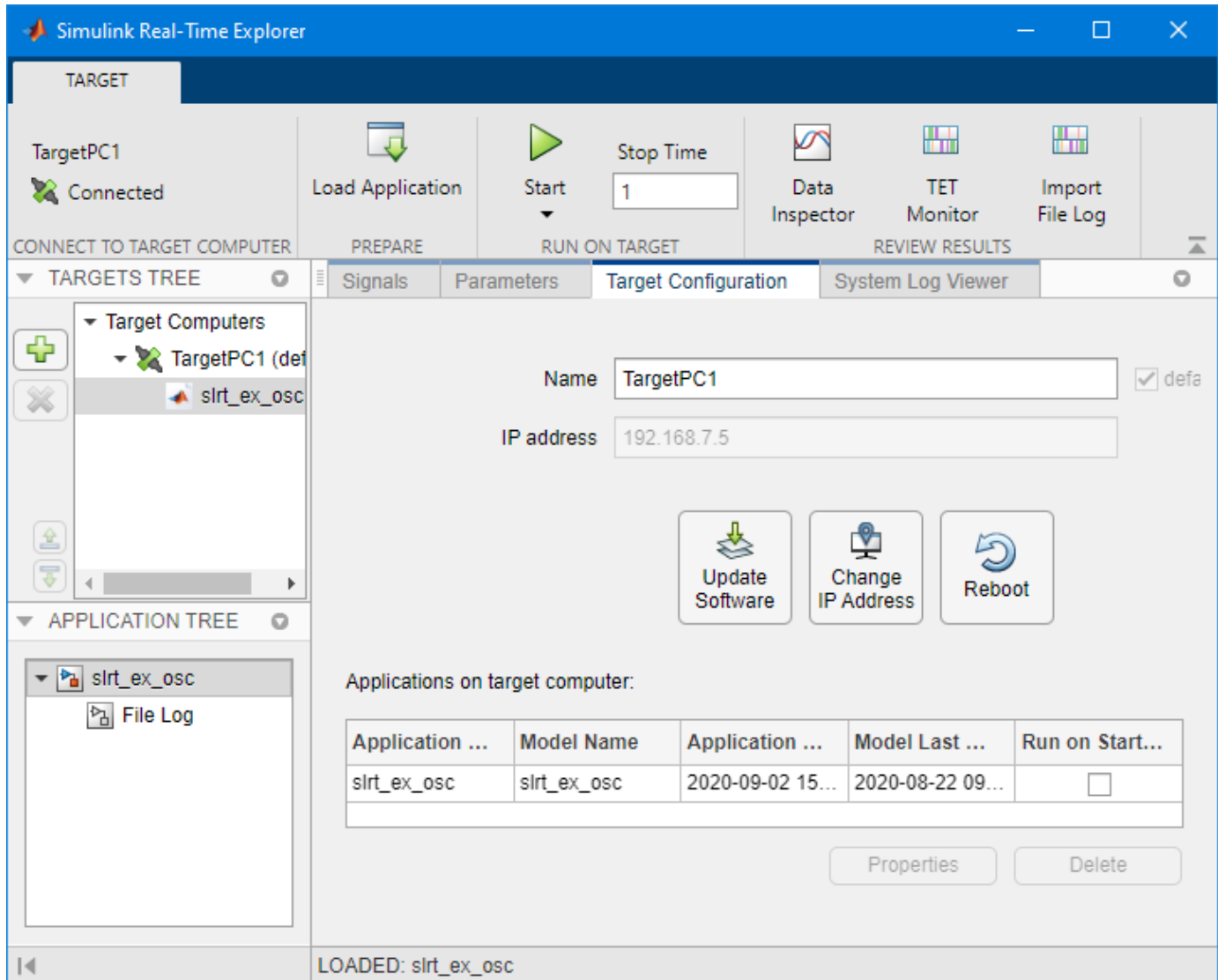
From the Simulink Editor, in the **Real-Time** tab, select **Prepare > SLRT Explorer**. Or, from the MATLAB Command Window, type:

```
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.

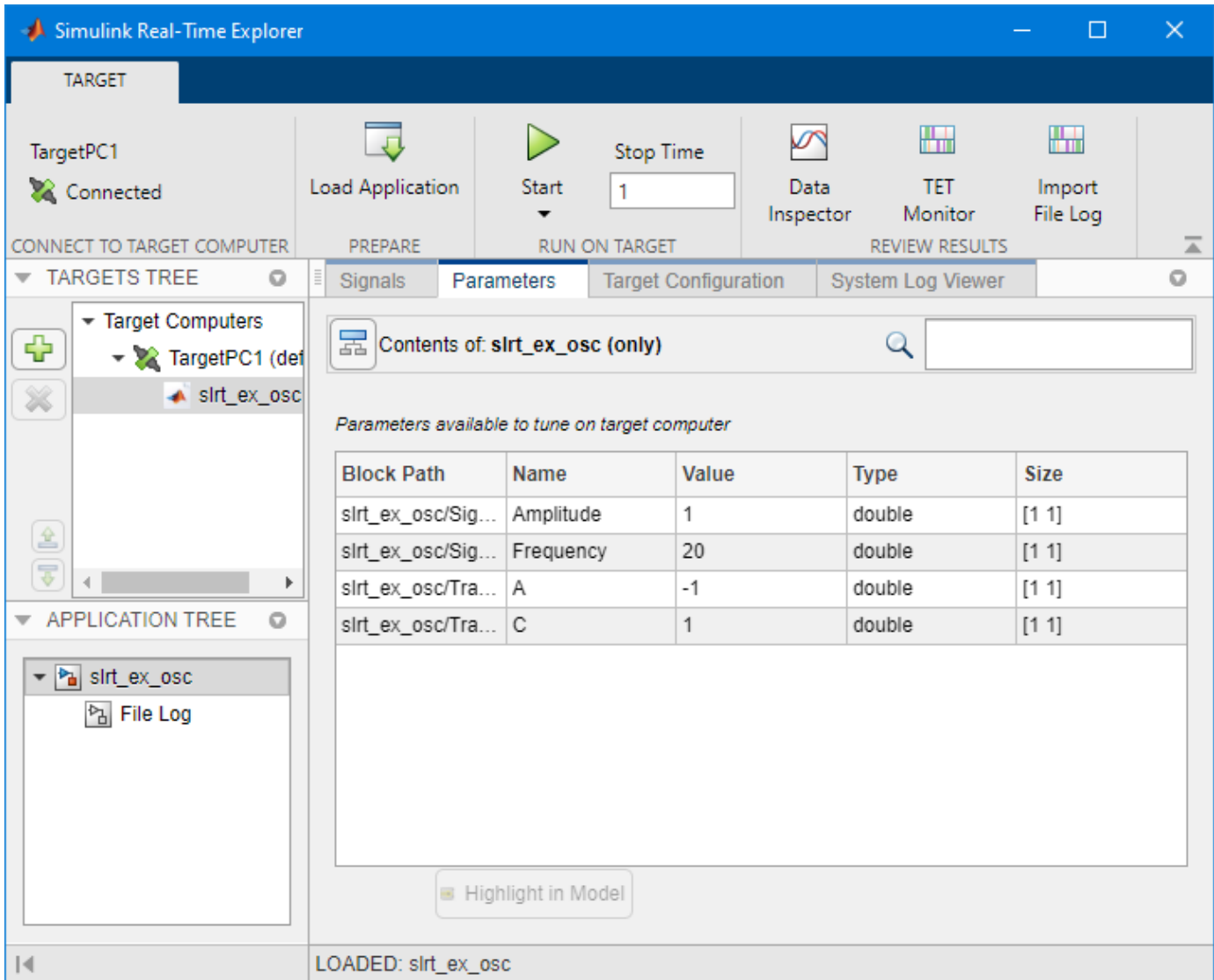
To change the IP address of the target computer, click the **Change IP Address** button.

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.



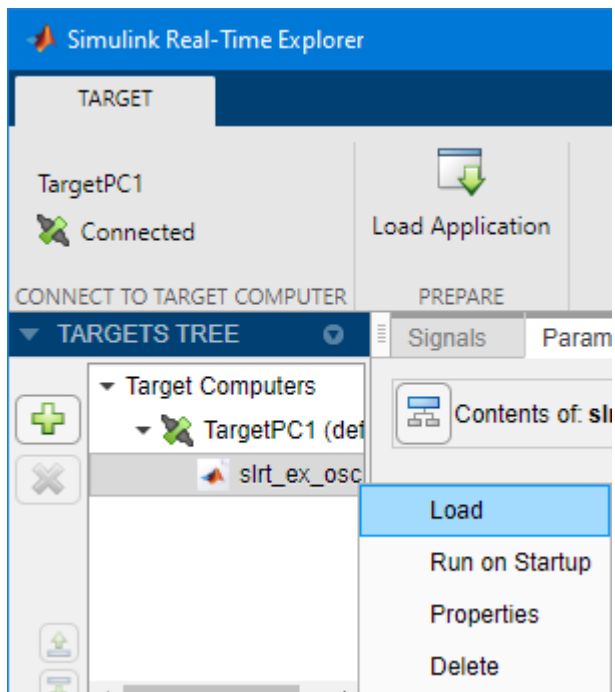
Open the Simulink Real-Time Explorer.

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.

- To 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)
```

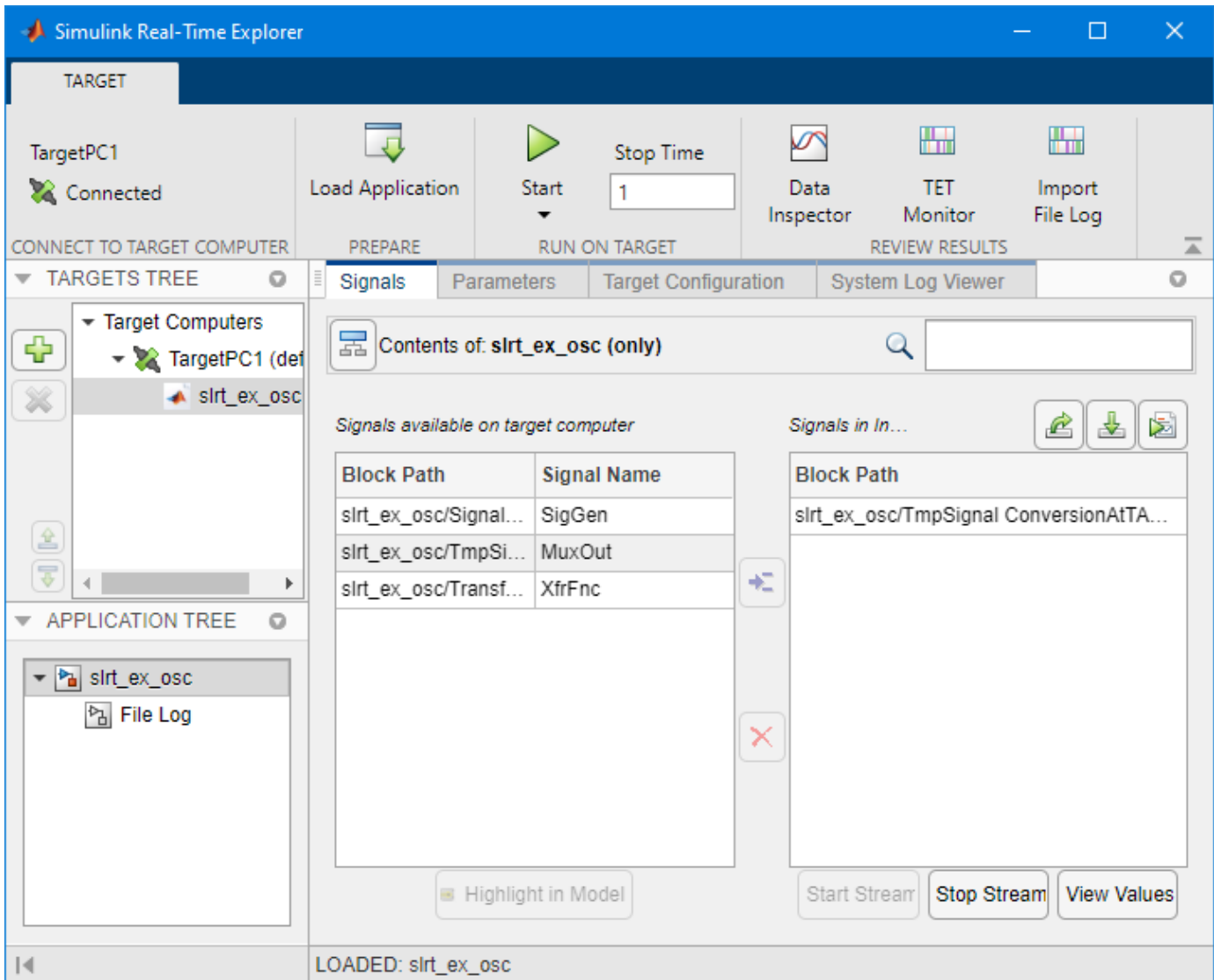
- To change the `ReloadOnStop` and `AutoImportFileLog` operation of the **Start** button, select **Start > ReloadOnStop** or **Start > AutoImportFileLog**.

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

### Select Signals and Stream Data to 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 **Run** button.

To stream the signal data, select the signals in the **Group signals to stream for SDI** list and click the **Stream Signal Group to SDI** button.

To view the streaming signals, click the **Open in SDI** button.

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.

### See Also

`slrtExplorer`

#### 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”](#)

#### Introduced in R2020b

# 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.

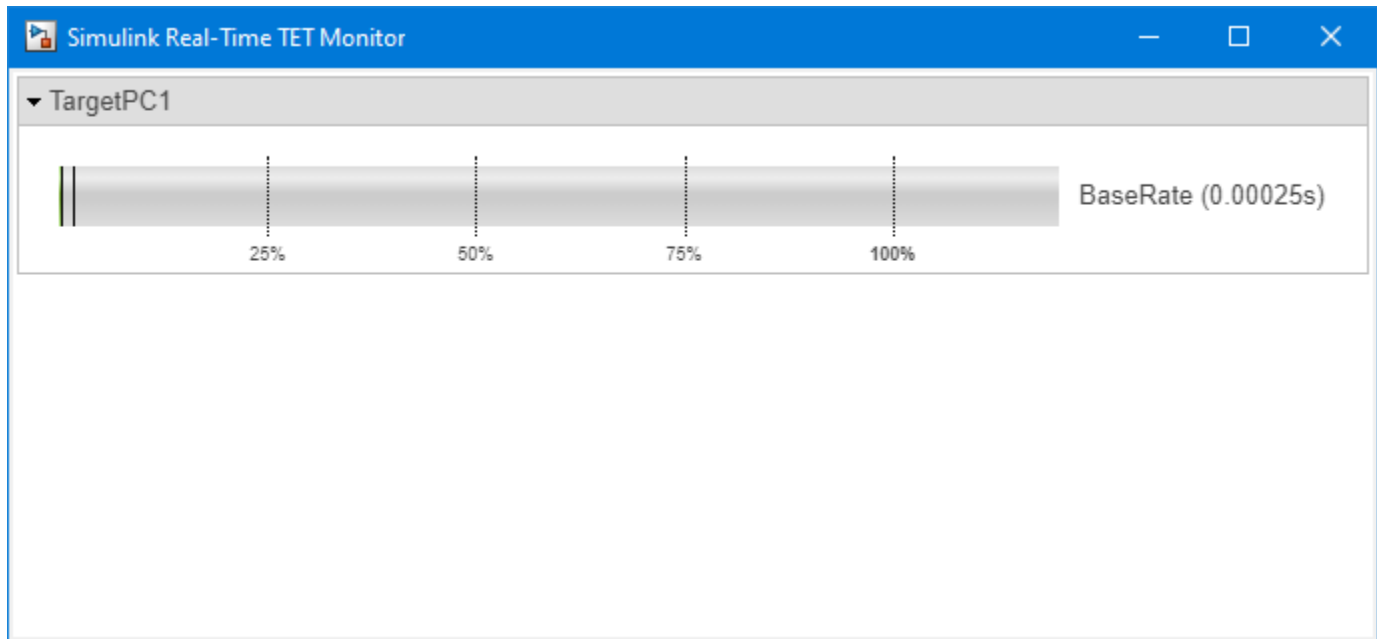
Open the `slrt_ex_osc` model.

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.

### See Also

SLRT Overload Options | `slrtTETMonitor`

### 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”

### Introduced in R2020b

# 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.

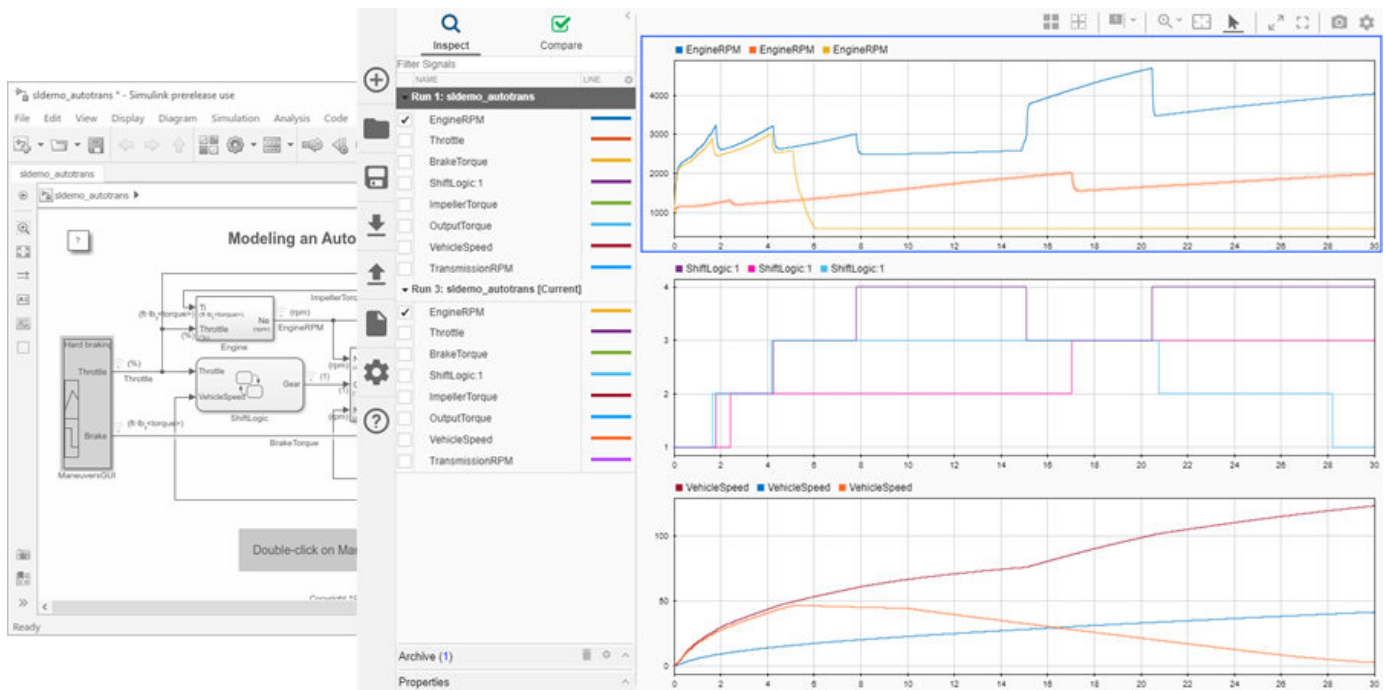
**3** “Compare Simulation Data”

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.

## 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”

### Introduced in R2010b





# 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: IDLE -> slrt_ex_osc_rt (DONE)
EXECUTION TIME: 1.0s / 1.0s
DISK USAGE: 42.3% used of 10.0 GB
TET_2.500e-01: 6.001e-05s / 6.045e-05s

--LOG--
15:13:30.778539 [info    ] Preparing to run model
15:13:30.780539 [info    ] stoptime = 1
15:13:30.780539 [info    ] loglevel = info
15:13:30.780539 [info    ] pollingThreshold = 0.0001
15:13:30.780539 [info    ] fileLogMaxRuns = 1
15:13:30.780539 [info    ] overrideBaseRatePeriod = 0
15:13:30.780539 [info    ] Loading model slrt_ex_osc_rt
15:13:31.310533 [info    ] Waiting for start command
15:13:41.101405 [info    ] Starting model slrt_ex_osc_rt
15:13:41.111405 [info    ] Logging started at 0s
15:13:42.336782 [info    ] TET 0 avg: 6.0007e-05 min: 2.6551e-05 max: 6.0451e-05
15:13:42.339782 [info    ] Logging stopped at 1s
15:13:42.341782 [info    ] Stopping model slrt_ex_osc_rt at 1s
```

### Display Status Monitor by Using PuTTY

To view the status monitor from the development computer, you can 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 5-2.

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”.

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

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

