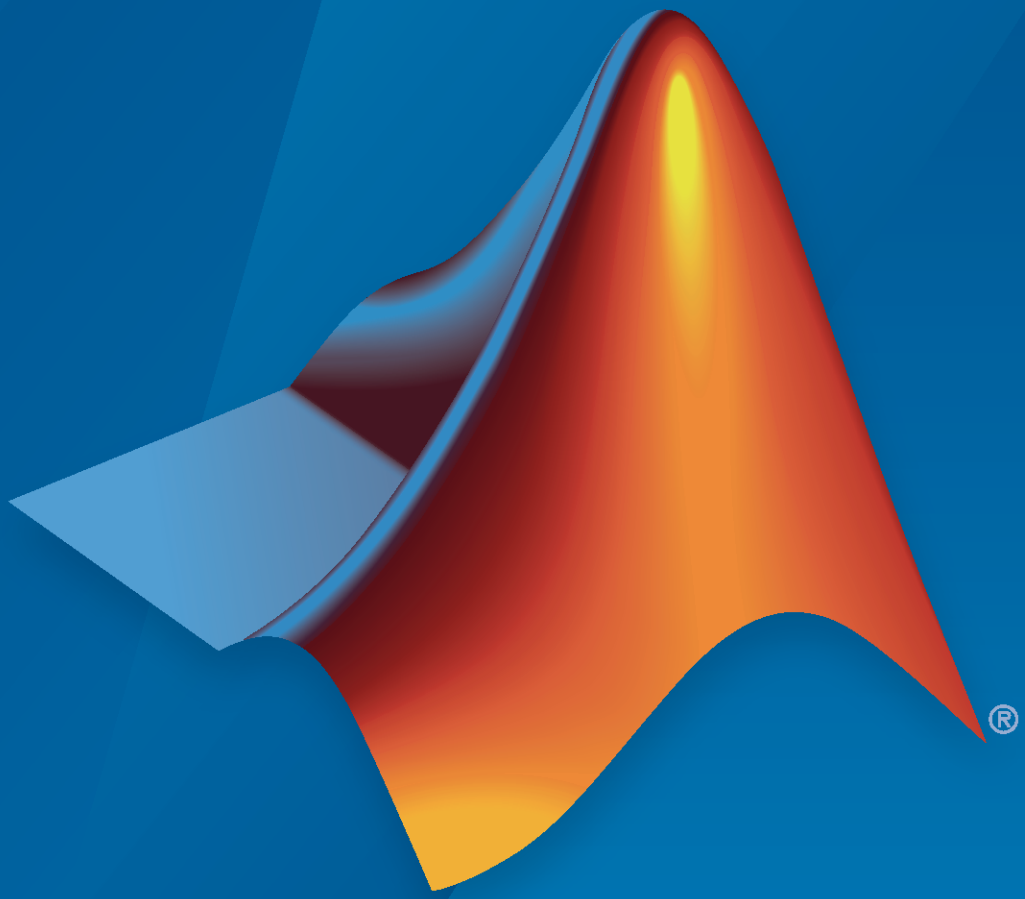
Simulink® Real-Time™

R2020b & Later Upgrade User's Guide



MATLAB® & SIMULINK®

R2024b

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

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

Simulink® Real-Time™ R2020b & Later Upgrade User's Guide

Trademarks

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

Patents

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

Revision History

September 2020	Online only	New 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)
March 2023	Online only	Revised for Version 8.2 (Release 2023a)
September 2023	Online only	Revised for Version 23.2 (R2023b)
March 2024	Online only	Revised for Version 24.1 (R2024a)
September 2024	Online only	Revised for Version 24.2 (R2024b)

Contents

1 Upgrade Systems and Models for Release R2020b

Upgrade Systems and Models for Release R2020b

Check model for upgradable Simulink Real-Time features

Check ID: `mathworks.design.slrealtimeUpgrades`

Check the model for Simulink Real-Time features from a previous release that require an upgrade for compatibility with the current release.

Description

This check identifies upgradable Simulink Real-Time features and upgrades the features where possible and recommends manual upgrades when needed. Block forwarding in the `slrealtimelib` library and the Speedgoat I/O Blockset eases the upgrade process.

Results and Recommended Actions

Condition	Recommended Action
System Target File upgrade	Change the System Target File to `slrealtime.tlc`.
Simulink Real-Time block upgrades	Upgrades blocks when possible. Identifies blocks for manual upgrade when needed.
Signal Upgrades	Upgrades signals for Instrument object compatibility.
Simulink Real-Time configuration parameter upgrades	Upgrades configuration parameters for toolchain build and real-time application options.

Capabilities and Limitations

You can:

- Run this check on your Simulink Real-Time models from release R2020a and previous releases.

See Also

- “Troubleshoot Model Upgrade for R2020b” on page 1-3
- “Troubleshoot System Upgrade for R2020b” on page 1-5

Troubleshoot Model Upgrade for R2020b

When building a pre-R2020b Simulink Real-Time models in R2020b, build errors occur. Some of the blocks in the model have the label **Obsolete Simulink Real-Time Block**.

What This Issue Means

For R2020b, you must upgrade the Simulink Real-Time model. A model Upgrade Advisor check is available to support this upgrade.

Try This Workaround

To upgrade your model for Simulink Real-Time R2020b features, use this software upgrade process:

1. Install Simulink Real-Time and other required products.
2. Install the Simulink Real-Time Target Support Package by using the MATLAB **Add-Ons** menu.
3. Install the Speedgoat I/O Blockset. Go to the Speedgoat website.
4. Upgrade the target machine software. See “Troubleshoot System Upgrade for R2020b” on page 1-5.
5. Open a R2020b or earlier Simulink Real-Time model.
6. Run the **Upgrade Advisor**. On the **Modeling** tab, select **Model Advisor > Upgrade Advisor**.
7. In the Upgrade Advisor dialog box, select **Check model for upgradable Simulink Real-Time features**. Click the **Run This Check** button. The advisor lists the upgrades that it can apply and the upgrades that require manual changes.
8. After the check, click the **Upgrade** button. The advisor upgrades the model by applying all nonmanual changes.

Apply Manual Changes

When the Upgrade Advisor encounters model issues that it cannot resolve by using automated changes, the Upgrade Advisor reports these as **Warning: Some of the features require manual changes**. It is a good practice to copy the text of these warnings and save it for reference as you apply changes to your model.

Apply Speedgoat Utilities

In R2020b, Speedgoat software provides support for configuring interrupts and Ethernet communications.

To configure model interrupts, use a single block from the Speedgoat I/O Blockset. You can use this block for asynchronous subsystems or for triggering the base rate of the model. The list of available interrupt sources reflects the Speedgoat I/O Blockset blocks configured in the model. For more information see the block documentation in `speedgoatlib_utilities`.

To configure additional Ethernet ports on the target machine, the Speedgoat I/O Blockset provides a tool to configure these Ethernet ports. The configuration is specific to the target machine and reflects the labeling on the front plates of the target machine. For more information, in the MATLAB Command Window, type:

```
speedgoat.configureEthernet
```

See Also

“Troubleshoot System Upgrade for R2020b” on page 1-5

Troubleshoot System Upgrade for R2020b

After upgrading Simulink Real-Time software to R2020b and installing the Simulink Real-Time Target Support Package, the system cannot download real-time applications to the target computer.

What This Issue Means

The upgrade for release R2020b requires software upgrades to the Simulink Real-Time software and the Speedgoat Target Machine software.

Try This Workaround

In R2020b, the change to a 64-bit POSIX compliant real-time operating system on the target computer requires a software upgrade for your Speedgoat real-time target machine. For this upgrade, the Speedgoat I/O Blockset provides an interactive tool. To retain compatibility with previous versions of MATLAB, you can choose to upgrade the target machine to a dual-boot system where you can select the operating system at startup. Or, you can upgrade the target machine to use only the new operating system. For the upgrade, you need a USB drive and the target machine needs a keyboard and a monitor.

To upgrade your software:

1. Install Simulink Real-Time and other required products.
2. Install the Simulink Real-Time Target Support Package by using the MATLAB **Add-Ons** menu.
3. Install the Speedgoat I/O Blockset. Go to the Speedgoat customer portal.
4. To start the target machine upgrade, in the MATLAB Command Window, type:

   ```
   speedgoat.migrateTarget
   ```

For more information, go to the Speedgoat website.

See Also

More About

- “Troubleshoot Model Upgrade for R2020b” on page 1-3

Troubleshoot MATLAB API Call Upgrade for R2020b

In R2020b, many objects and functions in the Simulink Real-Time API changed.

What This Issue Means

MATLAB® m-scripts written by using the Simulink Real-Time API require updates to run in R2020b and later releases.

Try This Workaround

These tables identify Simulink Real-Time API changes in R2020 and identify new workflows. Use these tables to guide your updates to pre-R2020b m-scripts. For information about Simulink Real-Time in previous releases, see:

`https://www.mathworks.com/help/doc-archives.html`

Target Computer Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`slrealtime.Application` object, `ApplicationName` property `slrealtime.target` object, `Application` property	Get name of real-time application.	Changed name	Use `Target` object, `Application` property.
`slrealtime.target` object, `Connected` property value is `'Yes'`.	Get communication status between development computer and target computers.	Changed name	Use `Target` object, `Connected` event. Use `connect` function to make connection.
`close` function	Close connection between development and target computers.	Changed name	Use `disconnect` function to close connection.
`getDiskSpace` function	Get free space and total space on the drive, in bytes.	Moved to Speedgoat API	See the `speedgoat.getDiskSpace` function at www.speedgoat.com.

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`load` function	Download real-time application to target computer.	Unchanged	Use `load` function after creating `Target` object. The real-time application is not automatically loaded as part of the build process.
`reboot` function	Restart target computer.	Unchanged	Use `reboot` function to restart target computer in standalone mode. Other, previously supported boot modes are not available.
`saveparamset`	Save a set of parameters to a real-time application	Changed name. Changed workflow.	Use `saveParamSet` to save the parameter set to a file.
`loadparamset`	Load a set of saved parameters to a real-time application	Changed name. Changed workflow.	Use `importParamSet` to Import the parameter set into a `ParameterSet` object on the development computer.

Real-Time Execution Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`slrealtime.target` object, `CPUoverload` property	Get CPU overload status.	Changed name	Use `Target` object, `TargetStatus` property or `ModelStatus` property.
`slrealtime.target` object, `AvgTET` property	Get average task execution time.	Changed name	Use `Target` object, `ModelStatus.TETInfo` property.
`slrealtime.target` object, `ExecTime` property	Get real-time application execution time.	Changed name	Use `Target` object, `ModelStatus.TETInfo` property.
`slrealtime.target` object, `MaxTET` property	Get maximum task execution time.	Changed name	Use `Target` object, `ModelStatus.TETInfo` property.
`slrealtime.target` object, `MinTET` property	Get minimum task execution time.	Changed name	Use `Target` object, `ModelStatus.TETInfo` property.
`slrealtime.target` object, `SampleTime` property	Get time between samples (step size).	Changed name	Use `Target` object, `ModelStatus.TETInfo.Rate` property.

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`start` function	Start execution of real-time application on target computer.	Changed workflow	Use `start` function to start real-time application and set real-time application options.
`slrealtime.target` object, `Status` property	Get execution status of real-time application.	Changed workflow	Use `Target` object, `ModelStatus` property or `status` function.
`stop` function	Stop execution of real-time application on target computer.	Unchanged	Use `stop` function.
`slrealtime.target` object, `StopTime` property	Get or set time when real-time application stops running.	Changed workflow	Get by using `Target` object, `ModelStatus.StopTime` property and set by using `setStopTime` function.
`slrealtime.target` object, `TETLog` property	Access storage in the MATLAB workspace for task execution time.	Changed workflow	Use `slrtTETMonitor` function.

Execution Profiler Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`getProfilerData` function	Retrieve profile data object.	Unchanged	Use `getProfilerData` function. Start the execution profiler before you start the real-time application.
`slrealtime.target` object, `ProfilerStatus` property	Get state of profiler.	Changed workflow	Use `Target` object, `Application` property. There are added property values.
`resetProfiler` function	Reset profiling service state to Ready.	Unchanged	Use `resetProfiler` function. The profiler resets itself when you start the real-time application.
`startProfiler` function	Start profiling service on target computer.	Unchanged	Use `startProfiler` function. Start the execution profiler before you start the real-time application.

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`stopProfiler` function	Stop profiling service on target computer.	Changed limitation	Use `stopProfiler` function. The profiler auto stop at 1GB is eliminated.

Parameter Tuning Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`setparam` function	Change value of tunable parameter in real-time application.	Changed syntax	Use `setparam` function.
`getparam` function	Read value of observable parameter in real-time application.	Changed syntax	Use `getparam` function.
`slrealtime.target` object, `NumParameters` property	Get number of tunable parameters.	Changed workflow	Use `Application` object and `getParameters` function.
`slrealtime.target` object, `Parameters` property	Get list of tunable parameters.	Changed workflow	Use `Application` object and `getParameters` function.
`slrealtime.target` object, `ShowParameters` property	Set flag to display the list of parameters.	Changed workflow	Use `Application` object and `getParameters` function.

Signal Tracing Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`getsignal` function	Get single numerical value of a signal.	Changed workflow	Use `Instrument` object and `connectScalar` function.
`slrealtime.target` object, `NumSignals` property	Get number of observable signals.	Changed workflow	Use `Application` object and `getSignals` function.

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`slrealtime.target` object, `OutputLog` property	Access storage in MATLAB workspace for output or Y-vector.	Changed workflow	Use Simulink model Signal logging parameter: `set_param(bdroot,... 'SignalLogging','on')`
`slrealtime.target` object, `ShowSignals` property	Set flag to display the list of signals.	Changed workflow	Use `Application` object and `getSignals` function.
`slrealtime.target` object, `Signals` property	Get list of observable signals.	Changed workflow	Use `Application` object and `getSignals` function.

Signal Logging Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`importLogData` function	Import buffered logging data to the active session of the Simulation Data Inspector.	Replaced	Use `list` function and `import` function. For more information, see the `Target.FileLog` object.

Instrumentation Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`addscope` function	Create a scope of specified type.	Replaced	Use the Simulation Data Inspector. Use File Log blocks instead File Scope blocks. Use `Instrument` objects instead of Host Scope blocks.
`getscope` function	Return scope identified by scope number.	Replaced	Use the Simulation Data Inspector. Use File Log blocks instead File Scope blocks. Use `Instrument`objects instead of Host Scope blocks.

Ethernet and EtherCAT Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
Target Computer Settings provided information about Ethernet setup on target computer.	Configure Ethernet ports on target computer	Moved to Speedgoat API	See the `speedgoat.configureEthernet` function at www.speedgoat.com.
`slrealtime.etherCAT.filterNotifications` function	Display EtherCAT notifications in human-readable format	Changed name	Use `slrealtime.EtherCAT.filterNotifications` function

Target Computer Settings Operations

R2020a and Previous Command	Command Description	R2020b and Later Command Status	New Workflow
`slrealtime.getTargetSettings` function	Get target computer settings.	Changed name	Use `getTargetSettings` function.

See Also

“Check model for upgradable Simulink Real-Time features” on page 1-2

More About

- “Troubleshoot Model Upgrade for R2020b” on page 1-3
- “Troubleshoot System Upgrade for R2020b” on page 1-5

Troubleshoot S-Function Build Upgrade for R2020b

A pre-R2020b model uses S-Functions. When migrating this model to the current release, this S-Function related message appears in the build log:

```
undefined reference to 'C function in S-Function'
```

What This Issue Means

In R2020b, the compiler for model builds changed from a Windows® compatible C compiler to the QNX® Neutrino® C++ compiler. To accommodate this change, update S-Function code for C++ compatibility.

Tip Fortran S-Functions are not supported in R2020b and later releases of Simulink Real-Time.

Try This Workaround

To update S-Function code for C++ compatibility, modify the declaration of the S-Function in the header file.

For example, update this C S-Function declaration:

```
void getAllDataMyFunction(short);
void myfunction_initialize(short, unsigned char*, short*);
void myfunction_terminate(short, int);
```

Updated the code to this C++ S-Function declaration:

```
#ifdef __cplusplus
extern "C" {
#endif

void getAllDataMyFunction(short);
void myfunction_initialize(short, unsigned char*, short*);
void myfunction_terminate(short, int);

#ifdef __cplusplus
}
#endif
```

Before building the updated code, remember to remove all artifacts from previous model builds.

See Also

More About

- “External Code Integration”

External Websites

- How to mix C and C++