Simulink® Real-Time™ Release Notes

MATLAB®&SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news:	`www.mathworks.com`
Sales and services:	`www.mathworks.com/sales_and_services`
User community:	`www.mathworks.com/matlabcentral`
Technical support:	`www.mathworks.com/support/contact_us`
Phone:	508-647-7000

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

Simulink® Real-Time™ Release Notes

Trademarks

Patents

Contents

R2022b

R2022a

R2021b

R2021a

R2020b

1

R2022b

Communicate with DDS middleware

In R2022b, your real-time application can communicate with DDS middleware by using a DDS protocol.

DDS is Data Distribution Service. This protocol requires the data communication in real-time distributed systems. This protocol requires the DDS Blockset product which provides application and blocks for modeling and simulating software applications that publish or subscribe to DDS middleware and supports both RTI and `eProsima` DDS vendors specific implementation.

These blocks let you communicate the data from a Simulink Real-Time model using a DDS protocol. For more information, see:

- DDS Receive
- DDS Send

Control stimulation for root level inports and Playback blocks

In R2022b, there are more options for controlling stimulation of root level inports and Playback blocks. These added options include:

- The `start(tg)` function `StartStimulation` option lets you control whether stimulation of root inports and Playback blocks starts when the real-time application starts.
- The `start(tg.stimulation)` function lets you control whether stimulation starts for some inports, some Playback blocks, or all inports and Playback blocks.
- The `getStatus(tg.stimulation)` function lets you get stimulation status for some inports, some Playback blocks, or all inports and Playback blocks.
- The `pause(tg.stimulation)` function lets you pause stimulation for some inports, some Playback blocks, or all inports and Playback blocks.
- The `reloadData(tg.stimulation)` function lets you reload signal data for stimulation of selected inports or Playback blocks.

Control signal logging and streaming with recording workflows

In previous releases, the workflows for signal logging and signal streaming had separate controls. In R2022a, the `startRecording` function and `stopRecording` function provided combined programmatic control of signal logging and signal streaming. In R2022b, the **Start Recording** button and **Stop Recording** button are available on the **Real-Time** tab in the Simulink Editor and in the Simulink Real-Time Explorer. These buttons provide combined UI control for signal logging and signal streaming. For more information, see “Signal Logging and Streaming Basics”.

In previous releases, after successful file log data import from the target computer, Simulink Real-Time deleted the imported log from the target computer. In R2022b, auto deletion of imported file logs does not occur. To remove file log data from the target computer, use the `discard` function. Or, use the **File Log Max Runs** option to determine how many run logs are retained on the target computer. For more information, see the `import` function and the `start` function option `FileLogMaxRuns`.

For more information about changes to signal logging and streaming workflows, see “Functionality being removed or changed” on page 1-4.

Apply robust signal-to-instrument binding in generated instrument apps

In R2022b, you can use a signal name instead of a full block path to create robust binding between a signal and an instrument. This support applies to the `SignalTable` component and functions such as `connectLine` or `connectScalar`. For more information, see callback code recommendations in example “Create App Designer Instrument Panels by Using Simulink Real-Time Components”.

Log and stream multiple Simulink messages per sample time

In R2022b, additional instrumentation support is available when logging or streaming Simulink messages. For Simulink Real-Time, the affected blocks are the Ethernet Receive block, Ethernet Send block, UDP Receive block, or UDP Send block. In previous releases, when multiple messages occurred in a sample period, only the last message in the real-time application was logged or streamed. In R2022b, all messages that are marked for logging or streaming and occur during a sample period are logged or streamed to the Simulation Data Inspector.

Additions to Simulink Real-Time App Generator

In R2022b, there are additions to the Simulink Real-Time App Generator. These include added properties for the `Menu` component and the ability to select a name for the output MLAPP application file. For more information, see “Create App Designer Instrument Panels by Using App Generator”.

Additions to Simulink Real-Time App Designer Components

In R2022b, there are additions to the Simulink Real-Time App Designer components. These include the added `RebootButton` component, the added `RecordButton` component, the added properties for the `StartStopButton`, and the added context menus for the `SignalTable` component and `ParameterTable` component. For more information, see the reference pages for these components.

Additions to API for Parameter Sets

In R2022b, you can get the startup parameter set for an `Application` object by using the `startupParameterSet` value in the `Options` property. You can update the parameter set selection by using the `updateStartupParameterSet` function. You can enable auto save of the parameter set by using the `updateAutoSaveParameterSetOnStop` function.

In R2022b, you can delete parameter sets from a real-time application on a `Target` object by using the `deleteParamSet` function.

Additions to XCP protocol support

In R2022b, additional support for the XCP protocol enables the XCP client to attempt to restart communications if an error occurs while the real-time application keeps running. For more information, see “XCP Client Mode”.

Apply CAN-FD protocol by using XCP CAN blocks

In R2022b, the XCP CAN FD blocks support the CAN-FD protocol over XCP. For more information, see:

- XCP CAN FD Configuration block
- XCP CAN FD Data Acquisition block
- XCP CAN FD Data Stimulation block
- XCP CAN FD Transport Layer block

Find unresponsive or nonmatching EtherCAT devices

In R2022b, the EtherCAT Get Scanbus Error Data block lets you check your EtherCAT® network for devices that are not responding. For more information, see the EtherCAT Get Scanbus Error Data block.

Enhanced ASAP2 file generation

In R2022b, you can generate multiple versions (including latest version) of an ASAP2 file according to the ASAM ASAP2 standard. The `Generate Calibration Files` tool enables you to customize the ASAP2 file. For example, you can include or exclude comments, include the name of the ASAP2 file, and include the location where to save the ASAP2 file. For more information, see “Generate ASAP2 and CDF Calibration Files”.

You can use the `coder.asap2.export` to apply more customizations, such as adding new `compumethod` names, grouping the data elements based on their properties. For more information, see “Customize Generated ASAP2 File”.

Additions to Simulink Real-Time ASAM XIL API and support package

In R2022b, support for additional methods have been added to the ASAM XIL API and Simulink Real-Time ASAM XIL Support package version 1.2. These methods are used to support writing XIL MDF files. For more information, see “Classes and Methods of ASAM XIL API”.

In R2022b, the ASAM XIL support package supports ASAM XIL - V2.1.0.

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Removed `enable` file logging function.	An error indicates that this function is not available.	Use the `startRecording` function.	Not applicable
Removed `disable` file logging function.	An error indicates that this function is not available.	Use the `stopRecording` function.	Not applicable

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Removed `isRecording` function. These `Target` object utility functions have been added: `deleteParamSet` and `removeAllApplications`.	An error indicates that this function is not available.	Use the `startRecording` function or `stopRecording` function. These functions generate an error if used when recording is already started or already stopped.	Not applicable
Added compatibility with Simulink dashboard blocks to the `startRecording` function and `stopRecording` function. Also, you can set the start or stop recording state to control logging and streaming independent of the run or stop state of the real-time application.	When a model includes Simulink dashboard blocks, using the `startRecording` function or `stopRecording` function does not generate an error.	Not applicable	Not applicable
Added **Start Recording** button and **Stop Recording** button to the **Real-Time** tab in the Simulink Editor and in the Simulink Real-Time Explorer. Also, these buttons are available as Simulink Real-Time components in App Designer.	The **Start Recording** button and **Stop Recording** button have the same functionality as the `startRecording` function and `stopRecording` function.	Not applicable	Not applicable
Added interaction between Enable File Log block and `AutoImportFileLog` option. If a model includes this block, the option has no effect when `stopRecording` is called..	If a model includes a Enable File Log block, the `start(tg) AutoImportFileLog` option has no effect, and the `startRecording` function and `stopRecording` function only control signal streaming (not File Log logging).	Not applicable	Not applicable

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Auto deletion of file log data no longer occurs as part of the operation of the `import` function. The number of log runs retained on the target computer is only controlled by the `start` function option `FileLogMaxRuns`.	Using the `import` function does not cause removal of file log data from the target computer.	Not applicable	Not applicable
The `Importing` property of the `Target.FileLog` object is removed.	In R2022b, an error appears when you use this property. This property does not exist for a `Target.FileLog` object.	Not applicable	Not applicable
For compatibility with the MATLAB® Compiler™, if no target computer is set in the instrument panel app, the initial value for the `TargetSelector` control is `Enter_IP_Address_Here`.	In R2022b, this initial target computer selection improves compatibility with the MATLAB Compiler for compiling standalone instrument panel apps.	Not applicable	Not applicable
The Sample Time parameter for the Thread Trigger block is removed. The block inherits sample time from the input signal to the block.	Save the model in R2022b to update the block.	Not applicable	Not applicable
To support App Designer instrument panels that use the Python®-MATLAB bridge, the `getBufferedData` function lets apps pull buffered data for a real-time application instrument. For more information, see the `getBufferedData` function.	Added in R2022b.	Not applicable	Not applicable

2

R2022a

Linux platform for development computer

In R2022a, you can use Simulink Real-Time on Windows® and Linux® platforms. To get started, see Enable Development Computer Communication (Linux).

If you develop App Designer applications (such as instrument panels for real-time applications) and use the App Compiler to produce standalone executable applications, remember the standalone application is not cross-platform. The executable type depends on the platform (Windows or Linux) on which the application was generated.

For blocks that support Windows and Linux platforms, use Speedgoat I/O Blockset release 9.4.0 for R2022a. For more information, see Speedgoat documentation.

Note The Simulink Real-Time ASAM XIL support package is supported on the Windows platform only.

Instrument panel app generator

In R2022a, you can generate an App Designer instrument panel MLAPP file from a model or real-time application. You can open the MLAPP file in App Designer for additional changes.

To get started, in the Simulink Editor, on the **Real-Time** tab, click the **App Generator** button. For more information, see **Simulink Real-Time App Generator** and Create App Designer Instrument Panels by Using App Generator.

Real-time permanent magnet synchronous motor (PMSM) example

To aid development of real-time motor applications, the Run Real-Time Simulation of Permanent Magnet Synchronous Motor example shows how to run a real-time simulation of a permanent magnet synchronous motor (PMSM) that is externally controlled at high switching frequency. The real-time application runs on a Speedgoat target computer that has a Speedgoat® IO334 I/O module with the IO334-21 plug-in board installed.

Target computer persistent variables

In R2022a, by using Simulink Real-Time persistent variables, you can write the last value of a signal when a real-time application stops, store the value, and read it when the real-time application starts. An odometer that records mileage cumulatively is a possible application for a persistent variable. Add persistent variables to your model by using the Persistent Variable Write block and Persistent Variable Read block. The API for persistent variables enables you to access the variable values as MATLAB variables. For more information, see:

- Persistent Variable Write
- Persistent Variable Read
- `getPersistentVariables`
- `setPersistentVariables`
- Apply Persistent Variables in Real-Time Applications

File logging decimation and frame-based input

In R2022a, functions enable you to get and set decimation values for File Log blocks in a model.

- `getAllFileLogBlocks`
- `getFileLogDecimation`
- `setFileLogDecimation`

In previous releases, the File Log processed sample-based input, treating elements as channels. In R2022a, you can configure the block to process sample-based input or frame-based input by configuring the **Input Processing** block parameter. For processing frame-based input, the block treats columns as channels. For more information, see the File Log block and Tune Decimation for File Log Data Without Model Rebuild.

Recording controls for file logging and streaming

In R2022a, recording functions enable you to start and stop logging from File Log blocks and start and stop signal streaming from the model. For more information, see these functions:

- `isRecording`
- `startRecording`
- `stopRecording`

For more information about the file logging and streaming workflow, see Signal Logging Basics.

Parameter table caching and parameter tuning in Simulink Real-Time Explorer

In R2022a, the **Parameters** tab in Simulink Real-Time Explorer supports caching parameter table data. By caching the data, updates to parameter data in the table are improved. This improvement is noticeable for real-time applications that have a substantial number of parameters. For instances in which the parameter table data becomes disabled (for example, when page switching occurs), the **Parameters** tab provides a **Refresh Values** button.

In R2022a, the **Parameters** tab in Simulink Real-Time Explorer supports a **Hold Updates** button and **Update All Parameters** button to change multiple parameter values simultaneously. These buttons in Explorer operate in the same way as these buttons operated on the **Real-Time** tab in the Simulink Editor.

For more information, see Simulink Real-Time Explorer.

Ethernet Receive and Send blocks

In R2022a, the Ethernet Receive block and the Ethernet Send block enable you to receive and send custom Ethernet packets. For more information, see:

- Ethernet Receive block
- Ethernet Send block
- Apply 802.1Q VLAN Tag by Using Ethernet Send and Receive Blocks

EtherCAT SDO complete access mode

In R2022a, you can select the access mode for the EtherCAT Async SDO Upload block and EtherCAT Async SDO Download block. When the **Access Mode** is `Complete Access`, the protocol for CoE access to variables provides:

- Access to all subindexes attached to a single index in the CoE dictionary for a single terminal device.
- Read or write all subindexes in the time it takes to read or write a single one of them.
- Simultaneously update all subindexes when a tuning parameter set is being written.
- Capture a simultaneous read of all subindexes.
- Allow use of EtherCAT devices that require complete access for configuration.

For more information, see EtherCAT Async SDO Upload block and EtherCAT Async SDO Download block. For an example, see Update Async SDO Block Variables by Using Complete Access Mode.

API for real-time application list root inports

In R2022a, the `getRootLevelInports` function enables you to get the root-level inports in a real-time application object. For more information, see `getRootLevelInports`.

API for real-time application uninstall

In R2022a, the `removeApplication` function enables you to uninstall real-time applications from the target computer. This function provides a MATLAB command that corresponds to the **Delete** application option in Simulink Real-Time Explorer. For more information, see `removeApplication`.

Functional Mock-up Unit version 2

In R2022a, Functional Mock-up Unit (FMU) support enables you to work with FMU version 1 and FMU version 2 files for co-simulation. This change affects generated code. If you are using FMU files on Simulink Real-Time from previous releases, it is recommended to generate a new FMU binary compatible with Simulink Real-Time. For more information, see Apply Functional Mock-up Units by Using Simulink Real-Time and Compile Source Code for Functional Mock-up Units.

ASAM XIL support package version 1.1

In R2022a, use the Simulink Real-Time XIL API support package version 1.1. This version of the support package adds methods in the MAPort class, SignalFactory class, SignalGeneratoryFactory class, and SignalGenerator class. For more information about supported classes and methods, see Classes and Methods of ASAM XIL API.

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Updates to display of status log messages	Line wrap occurs for long status log messages in the status monitor display.	Not applicable	Not applicable
The UDP Send block and UDP Receive block added support for data as Simulink messages.	The blocks handle data packets as Simulink messages.	Not applicable	Not applicable
Application Compiler supports Linux development computer	Linux development computer can compile applications into standalone executables.	Not applicable	Not applicable
The `TargetSelector` control for instrument panel apps provides a **Simulink Normal Mode** selection. This selection enables you to interface the instrument panel with a normal mode simulation of a model.	To use an instrument panel app to interface with a model for normal mode simulation, select **Simulink Normal Mode** in the target selector control instead of selecting a target computer.	Not applicable	Not applicable
In the App Designer `ParameterTable` component, data for struct parameters update correctly.	Because an App Designer instrument panel monitors operation of a real-time application, data for struct parameters update correctly in a parameter table component on the instrument panel.	Not applicable	Not applicable
For the App Designer `ParameterTuner` component, the `changeComponentValue` function enables you to change parameter values programmatically.	Using this function to change the value of the component pushes the value to the real-time application on the target computer.	Not applicable	Not applicable
The `removeApplication` function enables you to remove a real-time application and its related files from a target computer.	New in R2022a	Not applicable	Not applicable

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
The `getRootLevelInports` function enables you to list the root level inports in an `Application` object.	New in R2022a	Not applicable	Not applicable
The `Instrument` object supports bus signals.	New in R2022a	Not applicable	Not applicable
The `AutoImportFileLog` check box that is available from the **Start** button in Simulink Real-Time Explorer is now also available from the **Run on Target** button on the **Real-Time** tab in the Simulink Editor. This check box corresponds to the `AutoImportFileLog` option of the `start` function.	New in R2022a	Not applicable	Not applicable
When a real-time application stops, its parameter values are saved to parameter set file `autoSaveOnStop`.	You can import the `autoSaveOnStop` parameter set to the development computer by using the `importParamSet` function and load it to the real-time application. For more information, see `importParamSet`.	Not applicable	Not applicable
The `import` function imports a file log that was created by the same release of MATLAB as the function.	If using `import` on a file log from a previous MATLAB release, an error is displayed.	Not applicable	Not applicable

3

R2021b

Application Compiler for standalone executable instrument panels

In R2021b, you can use the Application Compiler to compile standalone executable instrument panels for real-time applications. For more information, see Create Standalone Instrument Panel App by Using Application Compiler.

App Designer components for operations interfacing to real-time applications

In R2021b, the component library in App Designer includes a set of Simulink Real-Time components that support common operations that interface to a real-time application. These components support selecting a target computer, connecting or disconnecting a target computer, starting or stopping a real-time application, and other operations. For more information, see Create App Designer Instrument Panels by Using Simulink Real-Time Components.

Bind mode for signals to instruments in Simulink Editor

In R2021b, you can bind signals to instruments (also referred to as instrumenting a signal) by using 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. For more information, see Add Instruments to Real-Time Application from Simulink Model.

Additional target object utility functions

In R2021b, these `Target` object utility functions have been added:

- `getAllInstruments`
- `getApplicationFile`
- `getInstalledApplications`
- `getLastApplication`
- `isConnected`
- `isLoaded`
- `isRunning`
- `reset`

Additional target object ECU and XCP page functions

In R2021b, the `Target` object ECU and XCP memory page functions have been added:

- `copyPage`
- `getECUPage`
- `getNumPages`
- `getXCPPage`
- `setECUAndXCPPage`
- `setECUPage`
- `setXCPPage`

If the ECU page and the XCP page selections do not match, the **Parameters** tab in Simulink Real-Time Explorer is disabled. When this occurs, you can use the **Enable Parameter Table** button in Simulink Real-Time Explorer to fix the mismatch. For more information, see Simulink Real-Time Explorer documentation.

Compatibility Considerations

To support these functions, the default storage class for new models has changed from `default` for model parameters and external parameters. For model parameters, the default storage class is `PageSwitching`. For external parameters, the default storage class is `PageSwitching (slrealtime)`.

Compu method conversion for XCP blocks

In R2021b, to add Compu method conversion support, the **Force Datatypes and Allow Non-Scalar** parameter for the XCP CAN Data Acquisition block, XCP CAN Data Stimulation block, XCP UDP Data Acquisition block, and XCP UDP Data Stimulation block has been replaced with the **Block Input/Output Settings** parameter. Using this parameter, you can select whether to apply Compu method conversion to block input or output. The table provides a comparison of the operations provided by the previous parameter as compared to the new parameter. For more information, see the XCP CAN and XCP UDP block reference pages. For more information about changes to the `xcpA2L` (Vehicle Network Toolbox) function and `xcp.A2L` (Vehicle Network Toolbox) properties, see “A2L Version 1.7.1: xcpA2L function creates A2L version 1.7.1 object” (Vehicle Network Toolbox).

Previous XCP Block Parameter and Operations	New XCP Block Parameter and Operations
Force Datatypes and Allow Non-Scalar — Disabled	**Block Input/Output Settings** — Raw values as double (no Compu method conversion)
Force Datatypes and Allow Non-Scalar — Enabled	**Block Input/Output Settings** — Raw values (no Compu method conversion)
Force Datatypes and Allow Non-Scalar — Disabled Issue warning for no Compu method support.	**Block Input/Output Settings** — Physical values (apply Compu method conversion)

Compatibility Considerations

If porting a model from R2021b to a previous release, the model issues a warning and disables values for XCP CAN/UDP STIM/DAQ blocks that use the **Block Output Settings** or **Block Input Settings** parameter set to `Physical values (apply Compu method conversion)`.

LIN Pack and LIN Unpack blocks for LIN bus data

In R2021b, the LIN Pack block and LIN Unpack block provide support for working with local interconnect network (LIN) bus data.

EtherCAT Get Device State and Set Device State blocks for control of EtherCAT terminal devices state

In R2021b, you can use the EtherCAT Get Device State and EtherCAT Set Device State blocks to control the state of EtherCAT terminal devices.

Execution profiler for task schedule display

In R2021b, you can use the `executionProfile.schedule()` function to display task scheduling in the Simulation Data Inspector. For more information, see https://www.mathworks.com/help/releases/R2021b/slrealtime/ug/profiling-target-application-execution.html.

Disk usage display

In R2021b, the target computer disk usage appears on the **Target Configuration** tab of the Simulink Real-Time Explorer and appears on the target computer status monitor. For more information, see Simulink Real-Time Explorer and Target Computer Status Monitor.

Custom messages in system log

In R2021b, you can use C++ function calls in S-functions to add custom messages in the target computer system log. For more information, see `SystemLog`.

ASAM XIL protocol for Test Cases

In R2021b, you can interface Simulink Real-Time models with test cases from third-party software that supports the ASAM XIL API. To use an ASAM XIL simulation workflow for your real-time applications, install the Simulink Real-Time XIL API support package. Follow the workflow in Install the Simulink Real-Time Support Package for ASAM XIL Standard.

API for C# program control of real-time applications

The Simulink Real-Time XIL API support package provides an API that lets you use a C# program to control real-time applications on the target computer. For more information, see https://www.mathworks.com/help/releases/R2021b/slrealtime/ug/control-real-time-application-by-using-c-sharp.html.

API for Python program control of real-time applications

In R2021b, you can create Python programs that run real-time applications on the target computer. For more information, see https://www.mathworks.com/help/releases/R2021b/slrealtime/ug/run-real-time-application-by-using-python.html.

Parameter structure for getparam and setparam functions

In R2021b, the operation of the `getparam` function and `setparam` function supports dot notation for:

- Specifying a field of a struct for `getparam`. It has the same support as `setparam`

- Specifying an element of an array or matrix for `getparam` and `setparam`
- Specifying one field of a struct when any substructure is an array of structs for `getparam` and `setparam`

For more information, see `getparam` function and `setparam` function examples.

USB to serial for RS232 blocks

In R2021b, the RS232 mainboard blocks support connecting USB serial adapter ports. The `Serial port` parameter of the Legacy Serial Read block, Legacy Serial Write block, Legacy Serial Setup block, Legacy Serial Port block, and Legacy Serial Port F block means that you can select a `USB Serial Port` or a `Legacy Serial Port`.

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
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.	When you open the TET Monitor, it appears as a tab in Simulink Real-Time explorer.	N/A	N/A
The parameters for the UDP Send block have changed. For unicasting, broadcasting, and multicasting, the parameters **Local IP address** and **Local port** are optional. The description of the UDP Send block has been updated to describe the effect of the Speedgoat configuration utility **Default Gateway** setting. The value range for the Local port parameter has changed to 0–65535.	The **Local IP address** and **Local port** parameters are optional. For a full description of their use, see the UDP Send block.	N/A	N/A

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
The parameters for the TCP Client block have changed. The parameters **Client IP address** and **Client port** are optional.	The **Client IP address** and **Client port** parameters are optional. For a full description of their use, see the TCP Client block.	N/A	N/A
The operation of the `Target.FileLog` object has changed. Now, the file log import process is synchronous in MATLAB, which means that while data import is occurring, the MATLAB status is busy. The `abort` function has been removed. The `ImportProgress` property has been removed.	The file log import process is synchronous in MATLAB. If used, the `abort` function issues a warning.	N/A	N/A
The `getTargetInfo` function is removed.	The `getTargetInfo` function is removed.	Use the `slrealtime.getSupportInfo` function.	N/A
The default storage class has changed from `default` for model parameters and external parameters. For model parameters, the default storage class is `PageSwitching`. For external parameters, the default storage class is `PageSwitching (slrealtime)`.	Do not use storage class `default` if using the page switching functions in "Additional target object ECU and XCP page functions" on page 3-2.	Use the new storage classes if using the page switching functions in "Additional target object ECU and XCP page functions" on page 3-2.	The storage class `default` is not compatible with page switching.
In R2021b, on the **Real-Time** tab, the **Batch Mode** button label changed to **Hold Updates**.	This change affects the appearance of the button. There is no operational change.	N/A	N/A

4

R2021a

Parameter set operations

In R2021a, the saveParamSet function and loadParamSet function let you save the model parameters to a file from a real-time application on the target computer and reload the parameters from the file. Using the ParameterSet object, you can work with the parameters from the development computer and transfer updated parameters to a model. For more information, see Save and Reload Parameters by Using the MATLAB Language and the ParameterSet object functions.

Target computer passwords

In R2021a, you can set the password for the user account slrt on the target computer. For more information, see Change Password for Target Computer.

GCC -ffast-math performance

In R2021a, you can improve real-time application performance by selecting the **Compile with GCC -ffast-math** configuration option. For more information, see Compile with GCC -ffast-math.

File log data import from target computer

In R2021a, you can copy file logs for simulation runs on the target computer to the development computer and import the log data into the Simulation Data Inspector. This workflow eliminates the possibility of losing data if a communications interruption occurs during automatic file log upload. For more information, see the slrealtime.fileLogList function and the slrealtime.fileLogImport function.

Bus signal names display in Simulation Data Inspector

In R2021a, the limitation on viewing overridden signal names in nonvirtual buses is removed. Signal names for nonvirtual buses display the bus object names or overridden names (if set) in the Simulation Data Inspector when live streaming data.

Simulink Real-Time Explorer application operations

In R2021a, 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.

For more information, see **Simulink Real-Time Explorer**.

Status Monitor disk usage display

In R2021a, the Status Monitor includes a display of the disk usage of the target computer. For more information, see Target Computer Status Monitor.

Model root inports stimulation on target computer

In R2021a, support for root inport stimulation is enhanced by the addition of the `Target.Stimulation` object and functions. This stimulation APIs lets you control the stimulation of root inport signal data on the target computer and get the stimulation status of the root-input ports on the target computer. You can start, pause, stop, and restart the stimulation on the target computer without reloading the real-time application. For more information, see Control and Update Stimulation of Inports to Real-Time Application.

Logitech G29 Steering Wheel block

In R2021a, the Logitech G29 Steering Wheel block is supported. You can read the data from a Logitech G29 steering wheel into the model. For more information, see Logitech G29 Steering Wheel.

UDP Multicast Receive block

In R2021a, the UDP Multicast Receive block is supported, and the UDP Send block has multicast support. These blocks provide multicast message support for the UDP communications protocol. For more information, see UDP Multicast Receive and UDP Send.

Real-time application signal values

In R2021a, the `getsignal` function returns the signal values from signals in the real-time application. For more information, see `getsignal`.

EtherCAT device signal names

In R2021a, the `slrealtime.EtherCAT.getSignalNames` function returns the signal names for EtherCAT devices connected to a model. For more information, see `slrealtime.EtherCAT.getSignalNames`.

Target computer support information

In R2021a, the `getTargetInfo` function returns target computer information from a target object. For more information, see `getTargetInfo`.

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Change of recommended IP address for development and target computers. In R2021a, the recommended IP address for the development computer is `192.168.7.2`, and the recommended IP address for the default target computer is `192.168.7.5`. This change makes the IP address recommendations in the Simulink Real-Time documentation consistent with the Speedgoat documentation.	Not applicable	Not applicable	Not applicable
Added support for logging Stateflow states from real-time applications to the Simulation Data Inspector. For more information, see Instrument a Stateflow Subsystem.	Not applicable	Not applicable	Not applicable

R2020b

Release notes from previous releases

In R2020b, Simulink Real-Time had a major release. The information from release notes from previous releases do not apply for R2020b and future releases.

For release notes from releases previous to R2020b, see Archived MathWorks Documentation.

64-bit POSIX compliant real-time operating system for target computers

Starting in R2020b, target computers use a 64-bit POSIX compliant real-time operating system. In R2020a and earlier, Simulink Real-Time used a 32-bit operating system.

In R2020b, you can install multiple real-time applications on your target computer. You can control these applications by using Simulink Real-Time Explorer, the MATLAB Command Window, or the target computer command-line interface. For more information, see Simulink Real-Time Workflows and Real-Time Application and Target Computer Modes.

You can connect to the target computer by using SSH and control real-time applications through the target computer command-line interface. For more information, see Target Computer Command-Line Interface and Execute Target Computer RTOS Commands at Target Computer Command Line.

Tools to compile real-time applications

To compile real-time applications, the cross-compiler is provided free of charge and is installed through the Simulink Real-Time Target Support package. For more information, see Install Development Computer Software.

Graphical instrument panels and applications

R2020b expands MATLAB-based API and App Designer support. You can create graphical instrument panels and other custom applications for monitoring signals and tuning parameters on your target computer. The APIs for the `Instrument` and `Target` objects support streaming data from a real-time application to an instrument panel application. Simulink Real-Time Explorer can generate code that creates these connections. For an example, see https://www.mathworks.com/help/releases/R2020b/slrealtime/ug/add-app-des-app-to-inverted-pendulum-mdl.html.

New Simulink Real-Time Explorer

In R2020b, Simulink Real-Time Explorer provides a single point of contact for interacting with real-time applications. You can monitor and trace signals, tune parameters and stream data to the Simulation Data Inspector. For more information, see Configure and Control Real-Time Application by Using Simulink Real-Time Explorer.

Upgrade Advisor for Simulink Real-Time

Use the Upgrade Advisor to upgrade models from previous releases. For more information, see Troubleshoot Model Upgrade for R2020b. Using the Upgrade Advisor eliminates many manual steps in upgrading a model, such as manually changing the code generation target to `slrealtime.tlc`.

Note that for models with their code generation target set to `slrt.tlc`, builds do not process and selecting **AppsSimulink Real-Time** does not set the code generation target to `slrealtime.tlc`.

New File Log block and other new blocks in Simulink Real-Time block library

To support the 64-bit POSIX compliant real-time operating system for target computers in R2020b, the blocks in the Simulink Real-Time block library have been updated and new blocks added. An upgrade advisor helps you upgrade your models to use the updated blocks. For a list of all blocks, see Simulink Real-Time Blocks.

The new File Log block provides enhanced signal logging capabilities with full Simulink data type support. For an example, see https://www.mathworks.com/help/releases/R2020b/slrealtime/ug/parameter-tuning-and-data-logging.html.

License change to support test engineering workflows

In R2020b, Simulink Real-Time does not require Simulink, Simulink Coder™, and MATLAB Coder to develop apps and run scripts to interact with a real-time application. This support enables new workflows for test engineers who need to develop instrumentation, test benches, or other tooling for pre-built real-time applications.

Simulink, Simulink Coder, MATLAB Coder and the Simulink Real-Time Target Support Package are required to create models and build real-time applications.

For more information regarding product dependencies, see Product Requirements & Platform Availability for Simulink Real-Time. For a description of software installation and licensing, see Install Development Computer Software.

Target Computer Upgrade by Using Speedgoat Software

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.

Bus Signal Names Display Limitation in Simulation Data Inspector

For Simulink Real-Time models, there are some limitations on displaying signal names and grouping signals for live streaming to the Simulation Data Inspector. These limitations apply only to live streaming. When simulating the model or using the File Log block for signal data logging, signal names and signal grouping appear correctly in the Simulation Data Inspector.

- Signal names for non-virtual buses display the bus object names (not overridden names) in the Simulation Data Inspector when live streaming.
- Signals that are grouped into a virtual bus display as individual signals in the Simulation Data Inspector when live streaming.

Global Data Stores for Referenced Models

In R2020b, Simulink Real-Time model builds support global data stores for referenced models. For more information, see Data Store Memory and Local and Global Data Stores (Simulink).

Functionality Being Removed or Changed

The functionality being removed or changed table indicates a number of removed product features due to the operating system change to the QNX® Neutrino® real-time operating system for target computers.

If any of the removed functionality is important to your workflow, please provide feedback through the **Why did you choose this rating?** feedback for this help page.

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Block library `xpcethernetlib` and blocks are removed: • Real-Time Ethernet Configuration • Create Ethernet Packet • Ethernet Tx • Ethernet Rx • Extract Ethernet Packet • Header Extract • Filter Address • Filter Type	The model Upgrade Advisor warns that these blocks are unavailable. In R2020b, no replacement suggestions are available for these blocks. For more information, see Troubleshoot Model Upgrade for R2020b.	Consider using UDP and TCP blocks instead of raw Ethernet blocks.	

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Block library `xpcvideousblib`, block library `xpcvideoutillib` and blocks are not available: • From USB Video Device • USB Video Device List • Video Display • Image Receive • Image Transmit • JPEG Compression • JPEG Decompression	The model Upgrade Advisor warns that these blocks are unavailable. In R2020b, no replacement suggestions are available for these blocks. For more information, see Troubleshoot Model Upgrade for R2020b.		
Blocks from library `slrtlib` are removed: • Scope removes Target Scope, Host, Scope, and File Scope blocks • To Target • From Target • From File • Video Display • Task Execution Time • Elapse Time • Time Stamp Delta • CPU Temperature • Minimum Available Stack Size • Current Available Stack Size • Current Available Stack Size	The model Upgrade Advisor warns that these blocks are unavailable. In R2020b, no replacement suggestions are available for these blocks. For more information, see Troubleshoot Model Upgrade for R2020b.		

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Block from library `logitechG29` is not available: - Steering Wheel Read	The model Upgrade Advisor warns that this block is unavailable. In R2020b, no replacement suggestion is available for this block. For more information, see Troubleshoot Model Upgrade for R2020b.		
Support is removed for signals marked for buffered logging with the Simulation Data Inspector. All signals are marked for immediate logging (also referred to as signal streaming).	The model Upgrade Advisor warns that signals marked for buffered logging are not supported. In R2020b, the Upgrade advisor offers to change the marked signals to immediate logging. For more information, see Troubleshoot Model Upgrade for R2020b.	For buffered logging, connect a File Log block in place of marking a signal for buffered logging.	
Support is removed for the Simulink Real-Time API for Microsoft .NET Framework and for the Simulink Real-Time C API.	The model build warns that these features are not supported.		
Support is removed for the Simulink Real-Time Performance Advisor Checks.			

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations

Support is removed for these MATLAB objects and their related functions:
`SimulinkRealTime.fileScope`
`SimulinkRealTime.hostScope`
`SimulinkRealTime.targetScope`
`SimulinkRealTime.fileSystem`
`SimulinkRealTime.crashInfo`
`...`
	When run, scripts and callbacks that use these objects and related functions warn that these features are not supported.	Mark signals for logging with the Simulation Data Inspector or connect the signal to a File Log block for viewing in the Simulation Data Inspector. For more information, see https://www.mathworks.com/help/releases/R2020b/slrealtime/ug/inspect-signals-in-external-mode-using-simulation-data-inspector.html.	

Support is removed for some properties of these MATLAB objects: Scopes property of `SimulinkRealTime.target`	When run, scripts and callbacks that use these properties warn that these features are not supported.		

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations

Support is removed for these MATLAB functions:
`viewTargetScreen`
`slrtsetCC`
`slrtgetCC`
`SimulinkRealTime.utils.minimumSampleTime`
`SimulinkRealTime.utils.createInstrumentationModel`
`SimulinkRealTime.utils.getFileScopeData`
`getPCIInfo`
`addscope`
`getscope`
`remscope`
`getDiskSpace`
`createStandAlone`
`getlog`
`getscope`
`SimulinkRealTime.copyFileToHost`
`SimulinkRealTime.copyFileToTarget`
`display(crashinfo_object)`
`update(crashinfo_object)`
	When run, scripts and callbacks that use these functions warn that these features are not supported.		

Support is removed for the `slrtexplr` command to start Simulink Real-Time Explorer.	If you type `slrtexplr` in the MATLAB Command Window, you are prompted to use the `slrtExplorer` command.	Use the `slrtExplorer` command to start Simulink Real-Time Explorer.	

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Support is removed for Simulink Real-Time Explorer based instrument panels (`*.slrtip`) .	In R2020b, Simulink Real-Time Explorer cannot open instrument panel files.	Use instrumentation object based instrument panels. For more information, see Instrumentation Apps for Real-Time Applications.	
Support is removed for Simulink Real-Time Explorer based signal logging to files.	In R2020b, Simulink Real-Time Explorer cannot create these files.	Connect a File Log block in place of using Simulink Real-Time Explorer for signal logging to files.	
Support is removed for FORTRAN S-functions.	The model build warns that these features are not supported.	Use C/C++ S-functions.	
Support is removed for Target object properties `TimeLog`, `OutputLog`, `TETLog`, and `StateLog`.	When run, scripts and callbacks that use these target object properties warn that these features are not supported.	Use the TET Monitor or execution profiling to observe this information. For more information, see slrtTETMonitor and getProfilerData.	
Support is removed for the `slrt.tlc` code generation target and its associated configuration parameters. The Simulink Real-Time pane is removed from the Code Generation tab of the Configuration Parameters dialog box.	The model Upgrade Advisor warns when a model uses unavailable configuration parameters. When you save the model, this configuration parameter is removed.	The `slrealtime.tlc` code generation target replaces `slrt.tlc`.	
Support for Visual C/C++ compiler is removed.	The toolchain for the `slrealtime.tlc` code generation target uses the QNX Neutrino C/C++ compiler.	Use QNX Neutrino C/C++ compiler that is installed with support package. If your workflow to build real-time applications applies external code integration, re-compile and link all external sources with the QNX Neutrino C/C++ compiler.	

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Communications setup support for the development computer and target computer has changed.	Until the target computer is updated to use the QNX Neutrino operating system, the development computer cannot connect to the target computer.	For information about communications setup for the development computer, see Development Computer Requirements and Enable Development Computer Communication (Windows). For information about communications setup for the target computer, see the Speedgoat documentation.	
Target computer boot method support has changed.	Until the target computer is updated to use the QNX Neutrino operating system, the target computer cannot boot.	For information about boot method support for the target computer, see the Speedgoat documentation.	

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Support for target computer keyboard (console) commands has changed. The target computer command-line interface changes include: Removed target object property commands, including `getpar`, `setpar`, `stoptime`, `sampletime`. Removed scope and video object function commands, including `addscope`, `remscope`, `startscope`, `stopscope`, `addsignal`, `remsignal`, `show`, `hide`. Removed scope object property commands, including `numsamples`, `decimation`, `grid`, `scopemode`, `triggermode`, `prepostsamples`, `triggersignal`, `triggersample`, `triggerlevel`, `triggerslope`, `triggerscope`, `triggerscopesample`, `ylimit`. Removed aliasing with variable command commands, including `setvar`, `getvar`, `delvar`, `delallvar`, `showvar`.	With the target computer operating system change to QNX Neutrino, previous target computer operating system commands are not recognized.	For information about commands that you can run on the target computer by using an SSH utility on the development computer or by using the target computer keyboard, see Target Computer Command-Line Interface.	
Support for MATLAB compiler deployment of MATLAB applications to control real-time applications is disabled.	In R2020b, MATLAB application that control real-time applications cannot be compiled with the MATLAB compiler.		

Functionality	What Happens When You Use This Functionality?	Use This Functionality Instead	Compatibility Considerations
Generation of ASAP2 (A2L) data description file has changed.	The **Generate INCA/ CANape extensions** configuration parameter is not supported or required.	In R2020b, the real-time application contains the A2L file information. To extract the generated A2L file from real-time application file, use the `extractASAP2` function.	
Support for the `setparam` function has changed. This function does not support the return history option that was supported.	Not available	Not available	