

Simulink® Requirements™

Getting Started Guide



MATLAB® & SIMULINK®

R2018a



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



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

Simulink® Requirements™ Getting Started Guide

© COPYRIGHT 2017–2018 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 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 2017	Online Only	New for Version 1.0 (Release 2017b)
March 2018	Online Only	Revised for Version 1.1 (Release 2018a)

1

Getting Started Content

Work with Requirements in the Simulink Editor	1-2
Author Requirements in Simulink	1-2
Link Blocks and Requirements	1-4
Link to Test Cases from Requirements	1-9
Introduction to Simulink Requirements	1-12
Link Between Requirements and Implementation	1-12
Link Between Requirements and Simulink Test	1-13
Additional Requirements Traceability Links	1-14
Share and Reuse Requirements	1-14

Getting Started Content

- “Work with Requirements in the Simulink Editor” on page 1-2
- “Link to Test Cases from Requirements” on page 1-9
- “Introduction to Simulink Requirements” on page 1-12

Work with Requirements in the Simulink Editor

Simulink Requirements enables you to author, organize, and edit requirements in the Simulink Editor. A specialized view, the Requirements Perspective, helps you to visualize the links between requirements and the parts of a model. Using an integrated environment simplifies linking requirements to the parts of your model that implement them.

This integrated environment has other advantages. For details, see “Introduction to Simulink Requirements” on page 1-12.

Author Requirements in Simulink

In Simulink Requirements, you organize your requirements in groups called requirements sets. In each requirements set, you can create additional levels of hierarchy for when you need more requirements to describe the details of a requirement.

In this tutorial, you use the Requirements Editor to create a requirements set to organize related requirements and add requirements to the set.

Suppose that you are writing requirements for a controller model of an automobile cruise control system. You develop these requirements, using your company’s numbering standard for requirements (R1, R2, and so on).

ID and Description	Rationale
R1: The maximum input throttle is 100%	The maximum value of the throttle from the acceleration pedal can be no greater than 100%.
R2: Cruise control has a speed operation range	Cruise control has a minimum and maximum operating speed.
R2.1: The vehicle speed must be at least 40 km/h	The speed of the vehicle must be at least 40 km/h for the cruise control system to engage.
R2.2: The vehicle speed cannot be greater than 100 km/h	The maximum operational speed of the cruise control system for the vehicle is 100 km/h.

Add these requirements to a model called `crs_controller`.

- 1 The model and supporting files are in a Simulink project. Open the project. At the MATLAB® command prompt, enter:

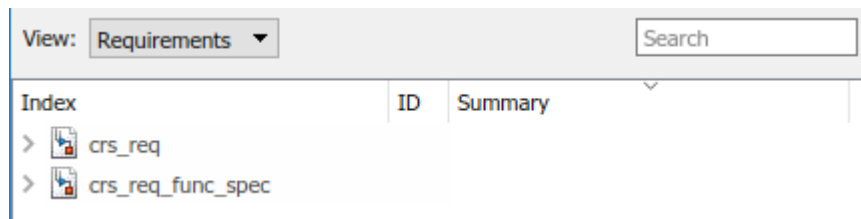
```
slreqCCProjectStart
```



- 2 Open the model. At the command prompt, enter:

```
open_system('models/crs_controller')
```

- 3 Open the Requirements Editor. In the Simulink Editor, select **Analysis > Requirements > Requirements Editor**.

The Requirements Editor opens with the requirements in view in the requirements browser, arranged by requirements set. The `crs_controller` model has two requirements sets that you can see in the browser: `crs_req_func_spec` and `crs_req`.



- 4 Add a requirements set in the requirements browser. From the Requirements Editor toolbar, click **New Requirement Set** .
- 5 You save requirements sets to external files. Save your requirements set to a writable location and name it `cruise_control_reqset.slreqx`.
- 6 When you save requirements sets to external files, you can share requirements with other models.
- 7 Add a requirement to your requirements set. Select the requirements set and click **Add Requirement** .
- 8 In the **Properties** pane, enter details for the requirement. You can copy and paste or drag requirements from another source to the **Properties** pane.
 - **Custom ID:** R1
 - **Summary:** Max input throttle %
 - **Description:** The maximum input throttle is 100%.

If you do not specify a custom ID, the Requirements Editor numbers requirements in order. Custom IDs let you use your company standards for labeling requirements and set the numeric order. (Custom IDs cannot contain a # character.) You can also use

an ID to help locate a requirement by searching. Keywords also aid in searching for a requirement.

9 Create the requirement R2. Right-click R1 and select **Add Requirement After**. Enter details for the requirement:

- **Custom ID:** R2
- **Summary:** Cruise control speed operation range
- **Description:** Cruise control has a minimum and maximum operating speed.

10 Create the child requirement of R2. Right-click R2 and select **Add Child Requirement**. Enter details for the requirement:

- **Custom ID:** R2.1
- **Summary:** Maximum vehicle speed
- **Description:** The speed of the vehicle must be at least 40 km/h for the cruise control system to engage.

Index	ID	Summary
> crs_req		
> crs_req_func_spec		
▼ cruise_control_reqset*		
1	R1	Max input throttle %
▼ 2	R2	Cruise control speed operation range
2.1	R2.1	Minimum vehicle speed

Add the other child to requirement R2 if you want.

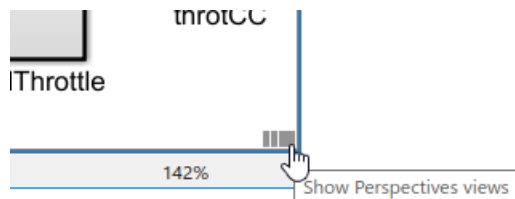
Tip You can rearrange the hierarchy by using the or by dragging requirements.

Link Blocks and Requirements

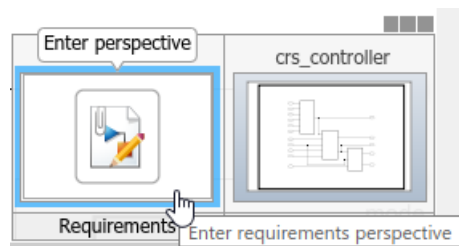
You can create links between your requirements and the Simulink blocks or Stateflow® objects that implement them. By linking between the model elements that implement requirements, you can track whether all your requirements have been implemented. You can also keep requirements and implementation in sync, for example, if a requirement changed or if the implementation causes you to revise your requirements.

In this tutorial, you use the Requirements Perspective to work with your requirements in the cruise control model. As in the Requirements Editor, you can add requirements sets and requirements in the Requirements Perspective. In both views, you can link requirements with blocks. In the Requirements Perspective, visual elements help you to see the links between requirements and blocks.

- 1 In the `crs_controller` model, to preview the perspectives associated with the model, click the perspectives control in the lower-right corner.

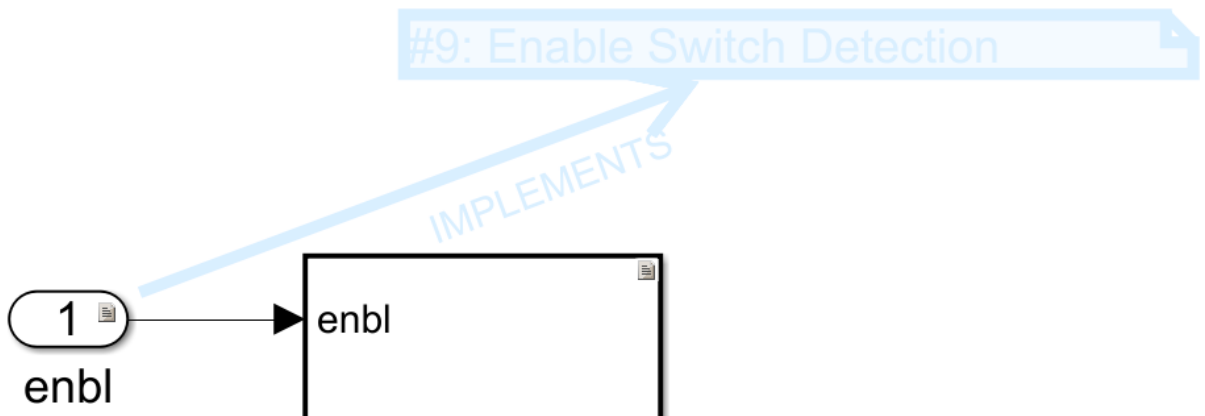


- 2 To open the Requirements Perspective, click the **Requirements** image.

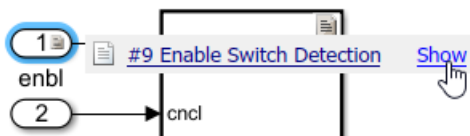


The Requirements Perspective includes the Requirements Browser, docked at the bottom of the Simulink editor. When you select a requirement, the Property Inspector displays the Simulink Requirements **Properties** pane.

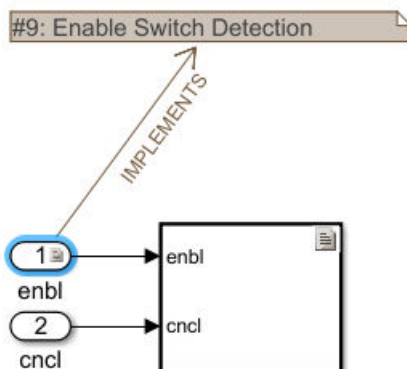
- 3 Link a requirement to the model element that implements it. Search for the requirement that you want to link from. In the Requirements Browser search box, enter `Enable Switch Detection`. As you type, matches are highlighted.
- 4 To create a link, select the matching requirement and drag it onto the `enbl` inport block. A markup guide appears on the canvas as a preview of the requirement annotation. Click the markup guide to place the requirement annotation on the canvas. You can also click anywhere else to create the link but not have the annotation display on the canvas.



- 5 A badge appears on the block to show that a link was created. You see badges only in the Requirements Perspective. To see more information about the requirement, click the badge and select **Show**.

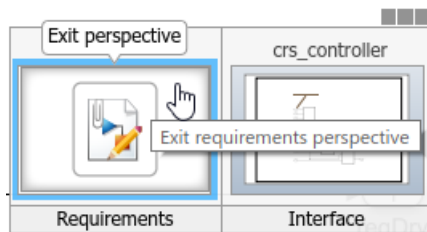


Clicking **Show** displays a requirement annotation on the model canvas.



The annotation displays the ID, the summary, and the relationship between the block and the requirement. In this example, the block implements the requirement. For other possible relationships, see “Requirement Links”.

- To see the requirement description, double-click the annotation.
 - To edit the requirement, right-click the annotation and select **Select in Requirements Browser**. The requirement properties appear in the Property Inspector, where you can edit them.
- 6** Exit the Requirements Perspective. Click the perspectives control and click the requirements image.



Convert Simulink Annotations to Requirements

You can convert the annotations in your Simulink models to requirements by using the context menu in the Requirements Perspective and by using the API. See `slreq.convertAnnotation` for more information on converting annotations to requirements by using the API.

To convert annotations to requirements by using the context menu in the Requirements Perspective:

- 1** Open the Simulink model and enter the Requirements Perspective.
- 2** Select a requirements set from the Requirements Browser. This is the destination requirements set for the new requirement.
- 3** Right click the annotation you want to convert to a requirement and click **Convert to Requirement**.
- 4** The annotation is converted to a requirement and is linked to the system or subsystem at which the annotation was present.

See Also

More About

- “Introduction to Simulink Requirements” on page 1-12
- “Requirement Links”

Link to Test Cases from Requirements

If you have Simulink Test™ and Simulink Requirements, you can link your requirements to test cases in the Test Manager. Linking to tests lets you see how requirements are confirmed with tests.

These links also help you to see your progress toward verifying the implementation of each requirement. Simulink Requirements can display verification status. Verification status helps you to see the requirements that are verified, the requirements that are missing tests, and whether tests associated with your requirements passed or failed.

In this tutorial, you link requirements to test cases associated with a controller model of an automobile cruise control system, managed in a Simulink project. After you run the tests, you include the results in the Requirements Editor.

- 1 Open the project. At the MATLAB command prompt, enter:

```
slreqCCProjectStart
```

- 2 Open the controller model. At the command prompt, enter:

```
open_system('models/crs_controller')
```

- 3 Open the Test Manager. In the model, select **Analysis > Test Manager**.

- 4 In the Test Manager, from the **tests** folder, open the **DriverSwRequest_Tests.mldatx** test file.

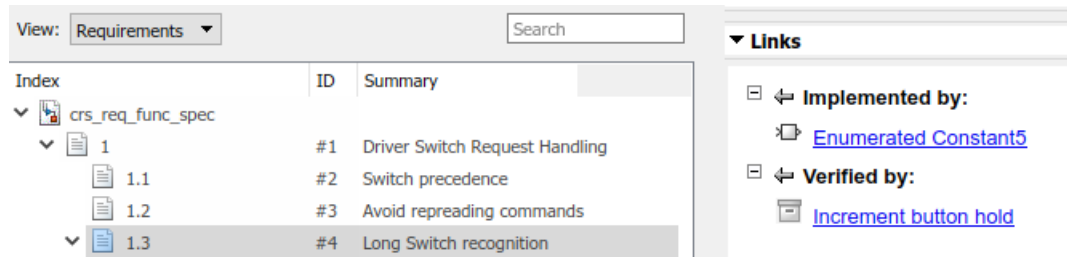
- 5 In the test browser, expand the test case hierarchy. The test file contains the test cases for several of the requirements in the **crs_controller** model. Most of these tests cases already link to requirements.

You want to link the **Increment** button hold test to a requirement. Select that test case.

- 6 In the model, select **Analysis > Requirements > Requirements Editor**. Expand the requirements set **crs_req_func_spec**.

- 7 Right-click the requirement with the ID **1.3 (Long Switch recognition)** and select **Link from Selected Test Case**.

The link to the test case appears in the **Links** pane.



- 8 Look at some of the other requirements, for example 1.8, 1.6, 1.9, and 1.4. Select each requirement and look at the **Links** pane. You can see the test case links under **Verified by**.
- 9 Run all the tests that verify the linked requirements, including the linked test that you added. In the Test Manager, right-click the test file `DriverSwRequest_Tests` and select **Run**.

The **Results and Artifacts** pane in the Test Manager shows that 7 tests passed and 1 test failed. Expand the results. The `Cancel button` test failed and the other tests passed.

NAME	STATUS
Results: 2017-Jul-07 14:51:58	7 1
DriverSwRequest_Tests	7 1
Unit test for DriverSwRequest	7 1
Enable button	
Cancel button	
Set button	
Resume button	
Increment button short	
Increment button hold	
Decrement button short	
Decrement button hold	

- 10 In the Requirements Editor, select **Display > Verification Status**. Resize the requirements browser window to see the **Verified** column.

This view shows you the progress toward verification. Some requirements do not yet have tests. Some requirements are fully verified. The width of the verified display shows the proportion of subrequirements that have links to verification. The color of the display indicates the proportion of tests that have passed, failed, or not run.

Here, the **Cancel Switch Detection** verification status shows that the test failed. Some requirements are partially verified because the subrequirements are not yet verified, such as requirement 1.8 in the figure. Hover over the verification status bar for details about the subrequirements.

Index	ID	Summary	Verified
crs_req_func_spec			
1	#1	Driver Switch Request Handling	Progress bar (mostly green)
1.1	#2	Switch precedence	Progress bar (mostly green)
1.2	#3	Avoid re-reading commands	Progress bar (mostly green)
1.3	#4	Long Switch recognition	Progress bar (mostly green)
1.3.1	#5	Waiting state for Long Increment swit...	Progress bar (mostly green)
1.3.2	#6	Waiting state for Long Decrement swi...	Progress bar (mostly green)
1.4	#7	Cancel Switch Detection	Progress bar (red)
1.5	#8	Set Switch Detection	Progress bar (mostly green)
1.6	#9	Enable Switch Detection	Progress bar (mostly green)
1.7	#10	Resume Switch Detection	Progress bar (mostly green)
1.8	#11	Increment Switch Detection	Progress bar (mostly green)
1.9	#15	Decrement Switch Detection	Progress bar (mostly green)

Passed: 8, Justified: 0, Failed: 1, Stale: 0, Unknown: 0, No Test: 9, Total: 18

See Also

More About

- “Requirement Links”
- “Link Test Cases to Requirements Documents”
- “Work with Requirements in the Simulink Editor” on page 1-2

Introduction to Simulink Requirements

In this section...
“Link Between Requirements and Implementation” on page 1-12
“Link Between Requirements and Simulink Test” on page 1-13
“Additional Requirements Traceability Links” on page 1-14
“Share and Reuse Requirements” on page 1-14

Simulink Requirements integrates requirements authoring and management with your modeling environment. You can author requirements in Simulink in the Requirements Editor, where you can organize and manage them. You can also import them from Microsoft Word or Excel. For details, see “Import Requirements from Third-Party Tools”.

Link Between Requirements and Implementation

You can link from requirements to the Simulink blocks or Stateflow objects that implement them. The connection is bidirectional, meaning that you can locate a requirement from a model element and a model element from a requirement.

You can:

- See your progress toward implementing requirements, including identifying missing implementations.
- React to requirement changes, for example, by updating model elements as requirements change, and clarifying requirements as your model evolves. With Simulink Requirements, you can find changed requirements by using a single command.
- Ensure that changes that you make to the model conform to the associated requirement.

The screenshot displays the Simulink Requirements Editor interface. On the left, a list of requirements is shown: #1: Driver Switch Request Handling, #2: Switch precedence, #3: Avoid reprecading commands, #4: Long Switch recognition, #7: Cancel Switch Detection, #8: Set Switch Detection, #9: Enable Switch Detection, and #10: Resume Switch Detection. An arrow labeled 'IMPLEMENTS' points from the 'enbl' requirement to the '#1: Driver Switch Request Handling' requirement. On the right, a table shows the implementation status for each requirement.

Index	ID	Summary	Implemented
> crs_req			
crs_req_func_spec			
1	#1	Driver Switch Request Handling	
1.1	#2	Switch precedence	
1.2	#3	Avoid reprecading commands	
> 1.3	#4	Long Switch recognition	
1.4	#7	Cancel Switch Detection	
1.5	#8	Set Switch Detection	
1.6	#9	Enable Switch Detection	
1.7	#10	Resume Switch Detection	

For more information, see “Requirement Links”.

Link Between Requirements and Simulink Test

If you have Simulink Test, you can link between requirements and tests that verify them. You can associate a requirement or set of requirements with tests that you create in Test Manager. When you run a test in Test Manager that you linked to a requirement, you can see the pass/fail results in the Requirements Editor.

Because you can track test results in Simulink Requirements, you can see the progress toward verification. The verification status also helps you to identify missing information and clusters of requirements associated with failing tests. You can use this information to understand the impact and complexity of those requirements.

The screenshot shows the Simulink Requirements tool interface. At the top, there is a 'View: Requirements' dropdown and a search box. Below this is a table with columns for 'Index', 'ID', 'Summary', and 'Verified'. The 'Index' column shows a tree view of requirements, with '1' selected. The 'ID' column contains IDs from #1 to #15. The 'Summary' column contains descriptions of requirements. The 'Verified' column shows progress bars for each requirement. A tooltip for requirement #1 shows 'Passed: 8, Justified: 0, Failed: 1, Stale: 0, Unknown: 0, No Test: 9, Total: 18'. On the right side, there is a 'Links' panel with 'Implemented by:' and a link to 'DriverSwRequest'. Below that is a 'Comments' panel.

Index	ID	Summary	Verified
1	#1	Driver Switch Request Handling	Passed: 8, Justified: 0, Failed: 1, Stale: 0, Unknown: 0, No Test: 9, Total: 18
1.1	#2	Switch precedence	Passed
1.2	#3	Avoid re-reading commands	Passed
1.3	#4	Long Switch recognition	Passed
1.3.1	#5	Waiting state for Long Increment swit...	Passed
1.3.2	#6	Waiting state for Long Decrement swi...	Passed
1.4	#7	Cancel Switch Detection	Failed
1.5	#8	Set Switch Detection	Passed
1.6	#9	Enable Switch Detection	Passed
1.7	#10	Resume Switch Detection	Passed
1.8	#11	Increment Switch Detection	Passed
1.9	#15	Decrement Switch Detection	Passed

For more information, see “Link Test Cases to Requirements Documents”.

Additional Requirements Traceability Links

With Simulink Requirements, you can create several other types of traceability links and establish many relationships within your model and to external documents. You can create these types of traceability links:

- Implements, in which a design element implements a requirement
- Verifies, in which a test case verifies a requirement
- Related to, in which you establish a trace relationship between a model element and a requirement
- Derives, in which a requirement is derived from another requirement
- Refines, in which one requirement refines another requirement

You can link between other types of documents, for example, HTML or DOORS® items, and requirements and to additional model elements such as dictionary objects.

For more information, see “Requirement Links”.

Share and Reuse Requirements

You save requirements files separately from your model files. You can then reference requirement files from multiple models, and each model can reference multiple

requirement files. Saving requirements in separate files lets you modularize common requirements across models while also managing requirements that are model-specific. This approach avoids copy-and-paste errors and keeps the requirements in sync across the models that share them.

You can compare requirements files (.slreqx files) by using the MATLAB file comparison tool. This tool helps you to identify differences in similar requirement sets. For more information, see “Compare Requirements Sets”.

You can also include requirements files in your Simulink projects. When you open a project, you can load any included requirements sets into the Requirements Editor from the project explorer. For more information, see “Manage Requirements Across a Team”.

Another way to share information about requirements is to generate a report that includes the requirements definition, links, implementation details, verification status, and so on. For more information, see “Generate Requirements Reports Using Simulink”.

See Also

More About

- “Work with Requirements in the Simulink Editor” on page 1-2
- “Import Requirements from Third-Party Tools”
- “Link to Test Cases from Requirements” on page 1-9

