

Simulink® Report Generator™ Release Notes



MATLAB® & SIMULINK®



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® Report Generator™ Release Notes

© COPYRIGHT 2002–2021 by The MathWorks, Inc.

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

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

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks 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.

R2021b

slreportgen.report.CFunction: Report on C Function blocks in Report API reports	1-2
C Function Component: Report on C Function blocks in Report Explorer reports	1-2
Report on Stateflow ports with the Report API	1-2
Report on Stateflow ports with Report Explorer	1-2
slreportgen.utils.HierarchyNumber: Generate number that represents subsystem position in model hierarchy	1-3
Improved reporting of configurations by data dictionary reporter	1-3
slreportgen.report.SystemIO reports signal port number and block name for models and subsystems	1-3

R2021a

slreportgen.report.Bus: Report on buses selected or created by blocks	2-2
Find and report on signals used by model or block	2-2
slreportgen.utils.traceSignal: Trace output signals and multiple signals, and specify nonvirtual or graphical sources or destinations	2-2
slreportgen.report.MATLABFunction: Use new properties to report code of user-defined supporting functions	2-2
Report Explorer MATLAB Function Component: Use new option to report code of user-defined supporting functions	2-3
slreportgen.report.DocBlock: Use new property to embed HTML and RTF content in reports	2-3
Report Explorer Documentation Component: Use new options to include HTML and RTF content in reports	2-3

Report Inport and Outport block properties with slreportgen.report.SystemIO reporter	2-3
slreportgen.report.Reporter methods: Create custom reporters and templates	2-4
slwebview: Include requirements or coverage view in a web view	2-4

R2020b

Find and report on Simulink data dictionaries	3-2
slreportgen.report.ExecutionOrder: Report on system task and block execution order	3-2
slreportgen.report.ModelConfiguration: Report on active configuration set of model	3-2
slreportgen.report.TestSequence: Report on a test sequence block	3-3
slreportgen.utils.hasDiagram: Check if object has a diagram	3-3
slreportgen.utils.isModel: Check if object is model	3-3
New property for specifying title generated by slreportgen.report.ModelVariable reporter	3-3
append: Use append to add content to Report API objects	3-3
Features being removed or changed	3-3
add method is not recommended	3-3

R2020a

slreportgen.report.SystemIO: Report on input and output signals	4-2
slreportgen.report.Notes: Report on model notes	4-2
Take snapshot of part of a diagram	4-2

R2019b

Model variable finder and reporter	5-2
Bus object reporter	5-2
DocBlock reporter	5-3
System hierarchy reporter	5-3
Embedded web view report generation with Report API	5-3
Items in embedded web view report table of contents are not numbered	5-3
SVG supported by diagram and element diagram reporters for Microsoft Word reports	5-4
Accessing Simulink Report Generator features from the Simulink toolstrip	5-5

R2019a

Report API examples	6-2
----------------------------------	-----

R2018b

Truth Table Reporter: Report on Simulink Truth Table blocks and Stateflow Truth Table functions	7-2
Element Diagram Reporter: Include a snapshot of a Simulink or Stateflow element in a report	7-2
Diagram reporter method added to DiagramElementResult and BlockResult finders	7-2

R2018a

MATLAB Function Reporter: Report on Simulink MATLAB Function block and Stateflow MATLAB Function object	8-2
--	-----

Lookup Table Reporter: Report on Simulink Lookup Table block	8-2
Simulink Web View Goto and Data Store Memory Support: Navigate between related blocks	8-2
Batch Web View Export: Export all models in a folder to Web views	8-2

R2017b

Simulink Reporters: Use MATLAB report objects to generate object property tables, diagram snapshots, and other Simulink report elements	9-2
Simulink Finders: Use MATLAB report objects to find diagrams, blocks, charts, states, and other model objects	9-2
Compare and Merge	9-3

R2017a

Embedded Web View: Generate HTML report containing interactive view of a Simulink model	10-2
Rich Annotation Component: Preserve model annotation formatting in generated reports	10-2

R2016b

Searchable Diagrams: Search for parameters and properties in a PDF report	11-2
Hyperlinked Diagrams: Navigate by clicking block diagrams in PDF reports	11-2
Numbering Template: Number sections in Simulink Design Description reports	11-2
Diff and Merge: Reuse open MATLAB sessions for diff and merge from external source control tools	11-2

Scalable Report Generation: Generate PDF reports as big as 10,000 pages	12-2
Three-Way Model Merge: Resolve conflicts between revisions and ancestor models using Simulink projects	12-2
Table of Contents: Add TOCs programmatically	12-3
Page Numbers: Create page number placeholders programmatically ..	12-3
HTML Text Component: Convert HTML to Word or PDF	12-3
pptview Function: Open PowerPoint presentation or convert it to PDF	12-3
Test Sequence Options: Include description, action, transition table, and requirements in report	12-3
Cross-Platform PDF Viewer: View PDF with built-in viewer	12-3
Functionality Being Removed or Changed	12-4

Bug Fixes

Web view search	14-2
Programmatic interface for adding content to PowerPoint presentations	14-2
Simulink data dictionary report component	14-2
Test Sequence block support	14-2
Several times faster model comparison analysis	14-2
DOM API object display options	14-2
DOM API horizontal rule	14-3

R2015a

Support for appending HTML string or file to a Word or PDF report generated by the Document Object Model (DOM) API	15-2
Single-file output option for HTML reports generated by DOM API	15-2
Simplified table formatting with DOM API	15-2
Container for generating high-level HTML elements	15-3
Function symbol data for MATLAB Function component	15-3
Images and text for DOM report links	15-3

R2014b

Report formatting based on Word and HTML templates	16-2
MATLAB report objects for creating report scripts	16-2
Fast file converter with reduced memory requirements	16-2
Fill-in-the-blanks Word and HTML forms for generating reports	16-2
Color settings preferences in Simulink XML comparisons	16-2
Optional Web views for model coverage and requirements	16-3
Improved performance for code generation report creation	16-3

R2014a

Object Inspector in Web view to view block parameter and signal property values	17-2
Options for name, location, and packaging of Web view files	17-2
Enhanced Web view export dialog, with context-sensitive help	17-2
Extended and simplified slwebview command	17-2

R2013b

Merge capability for MATLAB Function blocks from XML comparison report	18-2
Protected model Web view	18-2

R2013a

Toolstrip that replaces menus and toolbars in XML Comparison Tool	19-2
Scope Snapshot display of additional scopes	19-2

R2012b

Inclusion or exclusion of model references from System Design Description reports	20-2
Simulink functions in reports	20-2

R2012a

XML Comparison of Configuration Parameters	21-2
Print Filtered XML Comparison Results	21-2

R2011b

Use Full Page Option for PDF Reports	22-2
Filter Systems for the System Filter Component	22-2
Filter Properties for the Simulink Workspace Variable Component	22-2

R2011a

Merge Simulink Models From Comparison Report	23-2
Export XML Comparison Results to the Workspace	23-2
Improved PDF Images of Models and Charts	23-2

R2010b

Simulink Workspace Variable and Simulink Workspace Variable Loop Components Added	24-2
MATLAB Function Component Added	24-2
Model Configuration Set Component Added	24-2
System Design Description Report Enhancements	24-2
Transposable Columns for Summary Table Components	24-2
Improved XML Comparison Report	24-2

R2010a

Model Reference Support for Model Loop Component	25-2
Improved XML Comparisons for Files with the Same Name	25-2

R2009b

System Design Description Report	26-2
Blocks Sortable by Model and Chart Layout	26-2
Reporting Compiled Simulink Model Data	26-2
New Components for the Requirements Management Interface	26-2

Goto and From Block Information Added to Simulink Summary Table	26-2
Improved Images in Word and RTF Reports on Windows Platforms	26-3
Required Products Information for MATLAB/Toolbox Version Number Component	26-3
Block Summary Table Link to Data Object Property Information	26-3
Adobe Illustrator Image File Format No Longer Supported	26-3
Navigation Controls for XML Comparison Report	26-4

R2009a

New XML Comparison Example	27-2
Print a Sample Time Legend	27-2

R2008b+

Comparison of XML Files Exported From Simulink Models	28-2
--	------

R2008b

Simulink Web View Support for MATLAB Function Blocks in Stateflow	29-2
Simulink Web View Support for Simulink Functions in Stateflow	29-2

R2008a+

Bug Fixes

R2008a

Simulink Web View Enhancements	31-2
Performance and Scalability Enhancements	31-2
DocBlock Support	31-2
Support for Embedded URLs in Simulink Annotations	31-2
MATLAB Function Block Support	31-2

R2007b

Bug Fixes

R2007a

Simulink Web View Enhancements	33-2
Adobe Scalable Vector Graphics (SVG) Format Support in System Snapshot and Stateflow Snapshot Components	33-2

R2006b

New Real-Time Workshop Components	34-2
--	-------------

R2006a+

Bug Fixes

R2006a

Simulink Models Can Be Exported as Web Views	36-2
Browser Requirements and Limitations	36-2
Model Loop Component Reports on Libraries	36-3

Model Advisor Results Reported by New Component	36-3
User Interface and Performance Enhanced	36-3
Empty Columns Removed from Summary Tables	36-3

R14SP3

Bug Fixes

R14SP2+

Table Cell Spanning	38-2
Dialog Snapshot Components	38-2
Annotation Loop Component	38-2
Improved Graphical User Interface	38-2

R2021b

Version: 5.11

New Features

Bug Fixes

Compatibility Considerations

slreportgen.report.CFunction: Report on C Function blocks in Report API reports

Use an object of the `slreportgen.report.CFunction` class to report on a C Function block.

By default, a `CFunction` object reports:

- A table that includes the **Description** parameter and any custom mask parameters
- A table that lists the contents of the **Symbols** parameter
- Sections for the C code defined by the **Output Code**, **Start Code**, and **Terminate Code** parameters

Use the object properties to exclude or modify the reported information.

C Function Component: Report on C Function blocks in Report Explorer reports

Use the C Function component to include information about C Function blocks in Report Explorer reports.

By default, a `CFunction` component reports:

- A table that includes the **Description** parameter and any custom mask parameters
- A table that lists the contents of the **Symbols** parameter
- Sections for the C code defined by the **Output Code**, **Start Code**, and **Terminate Code** parameters

You can specify the parameters to report and customize the generated tables.

Report on Stateflow ports with the Report API

In R2021b, when you report on a Stateflow® chart, you can include information about the entry and exit ports, which are represented as `Stateflow.Port` objects.

By default, an `slreportgen.finder.DiagramElementFinder` or `slreportgen.finder.StateflowDiagramElementFinder` object finds all Stateflow elements in a diagram, including the ports. To include the ports in a custom search, specify:

- `Stateflow.Port` or `sf_port` in the `Types` property of a `DiagramElementFinder` object
- `Stateflow.Port` or `port` in the `Types` property of a `StateflowDiagramElementFinder` object

By default, an `slreportgen.report.StateflowObjectProperties` object reports the `LabelString`, `Description`, and `PortType` properties of a `Stateflow.Port` object. To change the properties that are reported or the formatting of the properties table, use the properties of the `StateflowObjectProperties` object.

Report on Stateflow ports with Report Explorer

In R2021b, you can include information about `Stateflow.Port` objects in Report Explorer reports.

To control whether the Object Loop component includes `Stateflow.Port` objects, select or clear the **Report on "Port" objects** check box. This check box is selected by default.

The Stateflow Automatic Table component reports the `LabelString`, `Description`, and `PortType` properties of a `Stateflow.Port` object. To control which properties are reported or customize the property table formatting, use the Stateflow Property Table component and edit the Port preset table.

To control how the Stateflow Hierarchy component shows ports, use the **Show ports** menu in the **Children** section.

slreportgen.utils.HierarchyNumber: Generate number that represents subsystem position in model hierarchy

Use an object of the `slreportgen.utils.HierarchyNumber` class to generate numbers that represent subsystem positions in a model hierarchy. You can use the numbers to label report elements, such as section titles.

Improved reporting of configurations by data dictionary reporter

In R2021b, an `slreportgen.report.DataDictionary` object uses an `slreportgen.report.ModelConfiguration` object to report on the **Configurations** section of a data dictionary. As a result, you can more easily identify components in the Configuration Details section and customize the section formatting.

The titles of the tables in the Configuration Details section now use the component name. In previous releases, the title identified the component by an array index instead of by name. For example, the title of the Solver component table is Solver in R2021b. In previous releases, the title was `vdp_config_ref.Components(1)`.

You can customize the formatting of the Configuration Details section by customizing the `slreportgen.report.ModelConfiguration` object used by the `DataDictionary` object. To access the `ModelConfiguration` reporter, use the new `ConfigurationReporter` property of the `DataDictionary` object.

To report the **Configurations** section of a data dictionary, set the `ShowConfigurations` property of the `DataDictionary` reporter to `true`. In R2021b and previous releases, the `ShowConfigurations` property is `false` by default.

slreportgen.report.SystemIO reports signal port number and block name for models and subsystems

By default, an `slreportgen.report.SystemIO` object now reports this information for input or output signals in both models and subsystems:

- Port number
- Inport or Outport block that corresponds to the signal
- Source or destination
- Name
- Data type

In previous releases, by default, a `SystemIO` object reported the Inport or Outport block for a model and the port number for a subsystem. The object also reported the source or destination, name, and data type.

Compatibility Considerations

In previous releases, a `SystemIO` object always reported the Inport or Outport block for a model signal and the port number for a subsystem signal, regardless of the values of the `InputSummaryProperties` or `OutputSummaryProperties` properties. In R2021b, to report the port number, you must explicitly include `Port` in the property list. To report the Inport or Outport block that corresponds to the signal, you must include `Inport Block` or `Outport Block` in the list.

In the `InputSummaryProperties` or `OutputSummaryProperties` property, you can now specify `Inport Block` and `Outport Block` instead of `Inport Block Name` and `Outport Block Name`. `Inport Block Name` and `Outport Block Name` are still valid.

R2021a

Version: 5.10

New Features

Bug Fixes

slreportgen.report.Bus: Report on buses selected or created by blocks

Use an object of the `slreportgen.report.Bus` class to report on buses that are selected or created by Simulink blocks.

An `slreportgen.report.Bus` reporter finds buses to report by searching a model, subsystem, block, or signal for blocks that select signals from a bus or combine signals to create a bus. For a model or subsystem, a `Bus` reporter reports all of the buses selected or created in the model or subsystem. For a block, a `Bus` reporter reports the bus that is selected or created by the block. For a signal, a `Bus` reporter reports all of the buses selected or created by the blocks that are connected to the port specified by the signal. By default, a `Bus` reporter generates:

- A table with details about the selected signals or the signals used to create the bus.
- Paragraphs with details about bus objects and connected blocks for each bus-related block found.

Use the reporter properties to customize the content and appearance of the generated report.

Find and report on signals used by model or block

You can use objects of these classes to find and report on signals used by a model or block:

- `slreportgen.finder.SignalFinder` — Signal finder
- `slreportgen.finder.SignalResult` — Result for a found signal
- `slreportgen.report.Signal` — Reporter for a signal

You can create your own `Signal` reporter and specify the signal to report. Alternatively, you can use a `SignalFinder` object to find signals. Constrain the search by using the properties of the `SignalFinder` object. Add the resulting `SignalResult` objects directly to a report or customize the `Signal` reporters for the results and then add the reporters to the report. To access the `Signal` reporter for a result, use the `getReporter` method of the `SignalResult` object.

For more information, see the `slreportgen.finder.SignalFinder`, `slreportgen.finder.SignalResult`, and `slreportgen.report.Signal` reference pages.

slreportgen.utils.traceSignal: Trace output signals and multiple signals, and specify nonvirtual or graphical sources or destinations

In R2021a, you can use the `slreportgen.utils.traceSignal` function to trace output signals to their destinations. In previous releases, you could trace only input signals. You can trace multiple signals by specifying an array of the input or output signals.

Use the new "NonVirtual" option to indicate whether you want to trace a signal to its nonvirtual or to its graphical source or destination.

slreportgen.report.MATLABFunction: Use new properties to report code of user-defined supporting functions

In previous releases, when you used an `slreportgen.report.MATLABFunction` object to report on MATLAB Function blocks or Stateflow MATLAB[®] functions, you could include the names of user-defined supporting functions, but you could not include the function code. In R2021a, you can include the code of user-defined supporting functions by setting the `IncludeSupportingFunctionsCode`

property of the `MATLABFunction` reporter to `true`. The default value of the property is `false`. The code is syntax-highlighted. Use the new `SupportingFunctionsCode` property to customize the formatting of the code, such as the font family or size. Use the new `SupportingFunctionsCodeTitle` property to customize the title that precedes the code.

Report Explorer MATLAB Function Component: Use new option to report code of user-defined supporting functions

In previous releases, when you used the Report Explorer MATLAB Function component to report on MATLAB Function blocks, you could include the names of user-defined supporting functions, but you could not include the function code. In R2021a, you can include the code of user-defined supporting functions by selecting the **Include code of user-defined supporting functions** check box in the **MATLAB Function** properties pane of the Report Explorer. By default, the check box is cleared.

`slreportgen.report.DocBlock`: Use new property to embed HTML and RTF content in reports

Use the new `EmbedFile` property of an `slreportgen.report.DocBlock` reporter to embed DocBlock HTML or RTF content in a PDF report and RTF content in an HTML report. If the value of the `EmbedFile` is `true`, the DocBlock reporter embeds the DocBlock content in the report and includes a link to the embedded content. The default value of the `EmbedFile` property is `false`.

Report Explorer Documentation Component: Use new options to include HTML and RTF content in reports

In R2021a, you can use these new options of the Report Explorer Documentation component to include HTML and RTF DocBlock content in template-based reports:

- **ConvertHTML** — Select this option to convert HTML content to a DOM object that is appended to the report. This option applies only to Word and PDF reports. This option is selected by default.
- **EmbedFile** — Select this option to embed HTML or RTF content in a PDF report and RTF content in an HTML report. The report includes a link to the embedded file. This option is cleared by default.

For information about template-based reports, see Report Templates.

Report Inport and Output block properties with `slreportgen.report.SystemIO` reporter

When you use an `slreportgen.report.SystemIO` object to report on the inputs and outputs of a system, you can now include the properties of the Inport and Output blocks that correspond to the input and output signals in the system.

You can specify these Inport block properties in the `InputSummaryProperties` property of the `SystemIO` object:

- "Inport Block Name"
- "Inport Block Description"

You can specify these Output block properties in the `OutputSummaryProperties` property of the `SystemIO` object:

- "Output Block Name"
- "Output Block Description"

slreportgen.report.Reporter methods: Create custom reporters and templates

The `slreportgen.report.Reporter` class has new methods that help you to define custom Simulink reporters and templates:

- `slreportgen.report.Reporter.createTemplate` creates a copy of the `slreportgen.report.Reporter` template for a specific report output type.
- `slreportgen.report.Reporter.customizeReporter` creates a class definition file that defines a subclass of the `slreportgen.report.Reporter` class. The method also creates copies of the default template file for each report output type.
- `slreportgen.report.Reporter.getClassFolder` returns the location of the folder that contains the `slreportgen.report.Reporter` class definition file.

slwebview: Include requirements or coverage view in a web view

When you use `slwebview` to export a model to a web view, you can include these optional views:

- Model requirements (requires Simulink Requirements™)
- Model coverage (requires Simulink Coverage™)

To include an optional view, use the new `'OptionalViews'` argument. Specify one or both of these values in a cell array:

- `'requirements'`
- `'coverage'`

For example:

```
htmlFileName = slwebview(gcs,'OptionalViews',{'requirements' 'coverage'});
```

For information about how to access an optional view in a web view, see [Include Model Requirements and Coverage Data in a Web View](#).

R2020b

Version: 5.9

New Features

Bug Fixes

Compatibility Considerations

Find and report on Simulink data dictionaries

You can use objects of these classes to find and report on Simulink data dictionaries:

- `slreportgen.finder.DataDictionaryFinder` — Data dictionary finder
- `slreportgen.finder.DataDictionaryResult` — Result for a found data dictionary
- `slreportgen.report.DataDictionary` — Reporter for a data dictionary

You can create your own `DataDictionary` reporter and specify the data dictionary to report. Alternatively, you can use a `DataDictionaryFinder` object to find data dictionaries. Constrain the search by using the properties of the `DataDictionaryFinder` object. Add the resulting `DataDictionaryResult` objects directly to a report or customize the `DataDictionary` reporters for the results and then add the reporters to the report. To access the `DataDictionary` reporter for a result, use the `getReporter` method of the `DataDictionaryResult` object.

For each section of a data dictionary, a `DataDictionary` reporter generates a table that summarizes the entries. For each entry, the reporter generates a table with details of the entry value. Use the properties of the reporter to specify the report options, such as which data dictionary sections to report, whether to report referenced data dictionaries, and the formatting options.

For more information, see the `slreportgen.finder.DataDictionaryFinder`, `slreportgen.finder.DataDictionaryResult`, and `slreportgen.report.DataDictionary` reference pages.

`slreportgen.report.ExecutionOrder`: Report on system task and block execution order

Use an object of the `slreportgen.report.ExecutionOrder` class to report on the tasks of a model or nonvirtual subsystem and the blocks in each task, sorted by execution order.

By default, an `ExecutionOrder` reporter generates a table of the tasks, sorted in execution order. For each task, the reporter generates a list of the blocks, sorted in execution order. Conditionally executed blocks appear in a separate table after the block execution order list. In the block execution order lists, the atomic subsystems, except for those that represent Stateflow charts, link to the block execution order for the same task reported by a separate `ExecutionOrder` reporter for the subsystem.

Use the `ExecutionOrder` object properties to filter and customize the formatting of the reported content.

For more information, see the `slreportgen.report.ExecutionOrder` reference page.

`slreportgen.report.ModelConfiguration`: Report on active configuration set of model

Use an object of the `slreportgen.report.ModelConfiguration` class to report on the active configuration set of a model. By default, a `ModelConfiguration` reporter generates a table for the configuration set properties and tables for the component properties. Use the `ModelConfiguration` object properties to filter and customize the formatting of the reported content.

For more information, see the `slreportgen.report.ModelConfiguration` reference page.

slreportgen.report.TestSequence: Report on a test sequence block

Use an object of the `slreportgen.report.TestSequence` class to report on a Simulink Test Sequence block.

By default, a `TestSequence` reporter generates:

- Property tables for all of the symbols — input, output, local, constant, parameter, and data store memory
- The step hierarchy as a nested list
- Details for each step including the step description, `When` condition, action statements, and a table of transition conditions and next steps

Use the `TestSequence` object properties to filter the reported content and customize the formatting of the content.

For more information, see the `slreportgen.report.TestSequence` reference page.

slreportgen.utils.hasDiagram: Check if object has a diagram

Use the `slreportgen.utils.hasDiagram` function to check whether a Simulink object has a diagram. You can check whether an object has a diagram before you use the object with an `slreportgen.report.Diagram` reporter.

slreportgen.utils.isModel: Check if object is model

Use the `slreportgen.utils.isModel` function to check whether a Simulink object is a model. You can use `slreportgen.utils.isModel` to validate an object before you use the object as an input to a finder or reporter that requires a model as input.

New property for specifying title generated by slreportgen.report.ModelVariable reporter

Use the new `Title` property of an `slreportgen.report.ModelVariable` object to specify a title for the variable reported by the reporter. If you do not specify a value for the `Title` property, the reporter uses the variable name as the title.

append: Use append to add content to Report API objects

To add content to `slreportgen.report.Report`, `mlreportgen.report.Chapter`, and `mlreportgen.report.Section` objects, you can now use `append` instead of `add`. For compatibility considerations, see “add method is not recommended” on page 3-3.

Features being removed or changed

add method is not recommended

Still runs

Starting in R2020b, use the `append` method instead of the `add` method to add content to objects of these Report API classes:

- `slreportgen.report.Report`
- `mlreportgen.report.Chapter`
- `mlreportgen.report.Section`

To add content to a DOM API object, such as an `mlreportgen.dom.Paragraph` object, continue to use the `append` method of the DOM object. The advantage of using `append` to add content to Report API objects is that you use the same method name as you use to add content to DOM API objects.

There are no plans to remove the `add` methods of the `Report`, `Chapter`, or `Section` classes. Report API programs that use the `add` methods will continue to run.

To update existing code, replace the method name `add` with `append` as shown by the examples in the table.

Not Recommended	Recommended
<pre>import mlreportgen.report.* import mlreportgen.dom.* rpt = slreportgen.report.Report("myrpt", "pdf"); ch = Chapter("My Chapter"); sect = Section("My Section"); para = Paragraph("My Content "); append(para, "more Content"); add(sect, para); add(ch, sect); add(rpt, ch); close(rpt); rptview(rpt);</pre>	<pre>import mlreportgen.report.* import mlreportgen.dom.* rpt = slreportgen.report.Report("myrpt", "pdf"); ch = Chapter("My Chapter"); sect = Section("My Section"); para = Paragraph("My Content "); append(para, "more Content"); append(sect, para); append(ch, sect); append(rpt, ch); close(rpt); rptview(rpt);</pre>

R2020a

Version: 5.8

New Features

Bug Fixes

slreportgen.report.SystemIO: Report on input and output signals

Use an object of the `slreportgen.report.SystemIO` class to report on input and output signals for a model or subsystem. By default, a `SystemIO` reporter generates:

- Summary tables for the input and output signals. The input summary table lists the sources of the input signals. The output summary table lists the destinations of the output signals.
- A signal details section that lists the properties of the input and output ports.

Use the `slreportgen.report.SystemIO` object properties to customize the content and appearance of the report.

slreportgen.report.Notes: Report on model notes

Use an object of the `slreportgen.report.Notes` class to report on model notes associated with a Simulink or Stateflow diagram. See Report Model Notes.

Take snapshot of part of a diagram

The `slreportgen.report.Diagram` reporter has a new property, `SnapshotArea`, that you can use to specify which part of a diagram to include in a snapshot.

To specify that you want to take a snapshot of the current Simulink Editor view, set the `SnapshotArea` property to the output of the new `slreportgen.utils.getCurrentEditorView` function.

R2019b

Version: 5.7

New Features

Bug Fixes

Compatibility Considerations

Model variable finder and reporter

You can use these new classes to find and report on variables used by a model:

- `slreportgen.finder.ModelVariableFinder` — Model variable finder
- `slreportgen.finder.ModelVariableResult` — Result for a found variable
- `slreportgen.report.ModelVariable` — Reporter for a found variable

You can add the results directly to a report. For example:

```
...  
finder = slreportgen.finder.ModelVariableFinder(model_name);  
results = find(finder);  
add(report, results);  
...
```

By default, the following information is displayed:

- The value of the variable
- Blocks that use the variable
- Workspace where the variable is defined

Alternatively, you can customize the reported information by setting the properties of the reporter associated with a found variable. For example:

```
...  
finder = slreportgen.finder.ModelVariableFinder(model_name);  
  
while hasNext(finder)  
    result = next(finder);  
    reporter = getReporter(result);  
    reporter.NumericFormat = "%.4f";  
    add(report, reporter);  
end  
...
```

Bus object reporter

You can report on Simulink.Bus objects by using the `slreportgen.report.BusObject` reporter class. Use this procedure:

- 1 Find all of the variables used by a model.
- 2 Identify the variables that are bus objects.
- 3 For each bus object, create a bus object reporter.

In 2019b, you can simplify the process of finding the bus objects by using the new model variable finder, `slreportgen.finder.ModelVariableFinder`.

See [Create a Simulink Bus Object Report](#).

DocBlock reporter

You can include a DocBlock in a report by using the `slreportgen.report.DocBlock` reporter class. The reporter supports HTML, RTF, and Text DocBlock blocks. The reporter includes the content of the DocBlock, or a link to the content, depending on the DocBlock type and the report type.

System hierarchy reporter

You can include the system hierarchy of a Simulink model in a report by using the `slreportgen.report.SystemHierarchy` reporter class. The reporter generates a nested list of systems that shows the ancestors, descendants, and peers of the specified model or system.

Embedded web view report generation with Report API

In R2019b, when you generate HTML reports that contain embedded web views, you can take advantage of the Report API. You can use the Report API finders and reporters in the `fillContent` method of your custom `slreportgen.webview.EmbeddedWebViewDocument` class. In previous releases, you were limited to using the DOM API.

You can use a new method, `getReportObject`, to get the report object for the embedded web view report. Once you have the report object, you can get the DOM implementation of a reporter by calling the reporter `getImpl` method.

Items in embedded web view report table of contents are not numbered

In R2019b, items in the table of contents in an embedded web view report are not numbered. For example:

[Cov](#)
[Ka](#)
[Kf](#)
[Ki](#)

In previous releases, the items were numbered. For example:

1. [Cov](#)
2. [Ka](#)
3. [Kf](#)
4. [Ki](#)

In R2019b, if your report includes chapters and sections, the table of contents looks like this:

[Chapter 1. Variables](#)
[1.1. Cov](#)
[1.2. Ka](#)
[1.3. Kf](#)
[1.4. Ki](#)

The numbers come from the chapters and sections, not from the table of contents.

Compatibility Considerations

In R2019b, to generate a table of contents with numbered items:

- 1 Copy the default template for an embedded web view report to the current folder.

```
copyfile(fullfile(matlabroot, ...  
'/toolbox/slreportgen/webview/resources/templates/embedded_webview.htmx'))
```

- 2 Unzip the template into the subfolder myTemplate.

```
unzipTemplate('embedded_webview.htmx', 'myTemplate');
```

- 3 In myTemplate, navigate to the stylesheets subfolder. Open the root.css file and remove this line from the ol.toc style:

```
list-style-type: none;
```

- 4 Save the root.css file.

- 5 Navigate to the folder that contains the original zipped template file. Package the unzipped template files into a zipped template file.

```
zipTemplate('Newtemplate.htmx', 'myTemplate');
```

- 6 Set the TemplatePath property of the embedded web view report generator object to the path and name of the new template file. For example, suppose that MyEmbeddedWebView is a subclass of slreportgen.webview.EmbeddedWebViewDocument. Here is the code to set the TemplatePath property of a MyEmbeddedWebView object:

```
rpt = MyEmbeddedWebView(rptName,model);  
rpt.TemplatePath = 'NewTemplate.htmx';
```

SVG supported by diagram and element diagram reporters for Microsoft Word reports

In R2019b, the diagram reporter (`slreportgen.report.Diagram`) and the element diagram reporter (`slreportgen.report.ElementDiagram`) can export snapshot images in Scalable Vector Graphics (SVG) format for Microsoft® Word reports. In previous releases, SVG format was supported for only HTML or PDF reports.

In an `slreportgen.report.Diagram` object and an `slreportgen.report.ElementDiagram` object, the default value of the `SnapshotFormat` property is now 'svg'. In previous releases, it was 'auto'. The value 'auto' now indicates 'svg' for all report types. In previous releases, 'auto' indicated 'svg' for HTML and PDF reports and 'emf' or 'png' for Word reports, depending on the platform.


Compatibility Considerations

Word reports that contain SVG images require Word 2016 or later. In MATLAB R2019b, to generate reports with diagram images that are compatible with earlier versions of Word, set the `SnapshotFormat` property of an `slreportgen.report.Diagram` or `slreportgen.report.ElementDiagram` object to a value other than 'svg'. To specify the image format used by default in previous releases of MATLAB, set `SnapshotFormat` to:

-
- 'emf' for a Windows® platform
 - 'png' for a UNIX® or Mac platform

Accessing Simulink Report Generator features from the Simulink toolstrip

In R2019b, the Simulink Toolstrip replaces the Simulink menu bar. Access Simulink Report Generator features and predefined reports from the toolstrip as follows:

- To open the Report Explorer, in the **Apps** tab, in the **Simulation Graphics and Reporting** section, click **Report Generator**.
- To export a model to a web view, on the **Simulation** tab, in the **File** section, click **Save** . Under **Export Model To**, select **Web View**.
- To generate a System Design Description report, on the **Modeling** tab, in the **Design** section, click **System Design Report**.

R2019a

Version: 5.6

New Features

Bug Fixes

Report API examples

Examples have been added to the documentation under Simulink Report Generator Task Examples. These examples illustrate using the Report API to generate and format common types of report content, such as Simulink system hierarchy, customizing block hyperlinks, and tiling Simulink diagrams.

R2018b

Version: 5.5

New Features

Bug Fixes

Truth Table Reporter: Report on Simulink Truth Table blocks and Stateflow Truth Table functions

You can use the `TruthTable` reporter to format and include Simulink Truth Table blocks and Stateflow Truth Table functions. For a given Truth Table, the reporter generates:

- A table listing the conditions of the Truth Table
- A table listing the actions that occur when a matching condition occurs

Element Diagram Reporter: Include a snapshot of a Simulink or Stateflow element in a report

You can use the `ElementDiagram` reporter to include a captioned snapshot of a block diagram or chart element in a report.

Diagram reporter method added to DiagramElementResult and BlockResult finders

A `getDiagramReporter` method has been added to the `DiagramElementResult` and `BlockResult` classes. If the result is a graphical element, this method returns an `ElementDiagram` reporter for the Diagram Element Result or a `Diagram` reporter for the Block Result. You use the result to include a snapshot of the element in a report.

R2018a

Version: 5.4

New Features

Bug Fixes

MATLAB Function Reporter: Report on Simulink MATLAB Function block and Stateflow MATLAB Function object

You can use the `slreportgen.report.MATLABFunction` reporter to report on MATLAB Function blocks and Stateflow MATLAB Function objects. The reporter reports this information:

- Properties
- Summary of input and output argument properties
- Details of input and output argument properties
- Function script
- Function symbol data
- Supporting functions

See `slreportgen.report.MATLABFunction` for a detail about each of these information types, which you can include in your report.

Lookup Table Reporter: Report on Simulink Lookup Table block

You can use the `slreportgen.report.LookupTable` reporter to report information on Simulink blocks that use lookup tables to compute their outputs from their inputs. The Lookup Table reporter includes tabular and graphical representations of the lookup table data. These Simulink blocks are supported:

- 1-D Lookup Table
- 2-D Lookup Table
- n-D Lookup Table
- Direct Lookup Table (n-D)
- Lookup Table Dynamic
- Interpolation Using Prelookup

Simulink Web View Goto and Data Store Memory Support: Navigate between related blocks

Simulink Web views now include navigation between Goto and From blocks, and between Data Store Memory, Data Store Read, and Data Store Write blocks. When you select one of these blocks in a Web view, a link to the corresponding block appears. Click the link to navigate to the corresponding block.

Batch Web View Export: Export all models in a folder to Web views

`slwebview` now supports exporting Simulink models in a folder and, optionally, its subfolders, to Web views. To export a folder of models to Web views, specify the path to that folder. For example, use `myWebView = slwebview('myModels')` to create a Web view for each model in the `myModels` folder. To include subfolders in the export operation, use the `RecurseFolder` property, for example, `myWebView = slwebview('myModels','RecurseFolder',true)`.

R2017b

Version: 5.3

New Features

Bug Fixes

Simulink Reporters: Use MATLAB report objects to generate object property tables, diagram snapshots, and other Simulink report elements

You can use the Report API to create MATLAB programs to generate reports. This API simplifies creating reports by providing objects for common report elements, such as Simulink property tables and diagram snapshots. You can use both Report API and DOM API objects in the same program. Using DOM API objects, you can customize the formatting of the provided Report API elements and add additional elements to your report.

When you create a report program, you can use the following Report API objects to generate diagram snapshots and object property tables.

- `slreportgen.report.Diagram` generates a diagram snapshot with an optional caption. A diagram is a Simulink block diagram or a Stateflow chart.
- `slreportgen.report.SimulinkObjectProperties` generates a table of properties for Simulink models, blocks, and annotations.
- `slreportgen.report.StateflowObjectProperties` generates a table of properties for charts, transitions, junctions, and other Stateflow objects.
- `slreportgen.report.Report` contains the report.

Simulink Finders: Use MATLAB report objects to find diagrams, blocks, charts, states, and other model objects

You can use Simulink finders to find Simulink model and Stateflow chart elements to include in your report. Simulink finders are part of the Report API. Finders return results as Report API objects that you can add directly to a report.

When you create a report program, you can use the following Report API objects to find Simulink elements.

- `slreportgen.finder.DiagramFinder` finds Simulink systems, Stateflow charts, and model references
- `slreportgen.finder.SystemDiagramFinder` finds Simulink block diagrams
- `slreportgen.finder.DiagramElementFinder` finds all graphical elements of a diagram, where a diagram is a Simulink block diagram or a Stateflow chart. Graphical Simulink elements include blocks, annotations, and lines. Stateflow graphical elements include annotations, transitions, junctions, states, boxes, truth tables, functions, Simulink functions, and embedded MATLAB functions.
- `slreportgen.finder.BlockFinder` finds all blocks in a Simulink block diagram
- `slreportgen.finder.AnnotationFinder` finds all annotation elements in a Simulink block diagram
- `slreportgen.finder.ChartDiagramFinder` finds Stateflow charts.
- `slreportgen.finder.StateflowDiagramElementFinder` finds graphical elements in a Stateflow chart. Graphical elements include annotations, transitions, junctions, states, boxes, truth tables, functions, Simulink functions, and embedded MATLAB functions.
- `slreportgen.finder.StateFinder` finds all states in a Stateflow chart.

-
- `slreportgen.finder.DiagramResult` holds `DiagramFinder`, `SystemDiagramFinder`, or `ChartFinder` results to add to a report document.
 - `slreportgen.finder.DiagramFinder` holds `AnnotationFinder`, `BlockFinder`, `DiagramElementFinder`, `StateFinder`, or `StateflowObjectFinder` results to add to a report document.
 - `slreportgen.finder.BlockResult` holds `BlockFinder` results to add to a report document.
 - `slreportgen.finder.DiagramElementResult` holds diagram element results to add to a report document.

Compare and Merge

As of R2017b, the compare and merge feature in Simulink Report Generator has moved into Simulink. See Model Comparison.

R2017a

Version: 5.2

New Features

Bug Fixes

Embedded Web View: Generate HTML report containing interactive view of a Simulink model

You can use `slreportgen.webview.webviewdocument` methods to embed interactive Simulink Web Views of models into HTML reports. The default Embedded Web View contains an autogenerated table of contents (TOC), a document panel, and the Web View. You can define your report content in the document panel and create two-way links between the document and the Web View. You can use the Embedded Web View report generator to create HTML reports, such as interactive model-based calibration guides and traceability reports for code and requirements.

Rich Annotation Component: Preserve model annotation formatting in generated reports

Generated reports now preserve Simulink model annotation formatting. Annotations that contain rich text content (text with HTML markup) and LaTeX equations now use the same format as the model annotations. Previously, annotations in reports appeared only as plain text.

R2016b

Version: 5.1

New Features

Bug Fixes

Compatibility Considerations

Searchable Diagrams: Search for parameters and properties in a PDF report

You can now search for Simulink model parameters and properties in a direct to PDF report. Use a System Snapshot component and set the snapshot format to `Automatic SL format (Scalable Vector Graphics)` to enable searching in the resulting report.

See the System Design Description report for an example. In a Simulink model, select **File > Reports > System Design Description**. Generate the report with the file format set to `Direct PDF (from template)`. To view the System Snapshot component, in the System Design Description dialog box, click **Customize Content** and locate the component in the report setup.

Hyperlinked Diagrams: Navigate by clicking block diagrams in PDF reports

In Simulink PDF reports, blocks in diagrams created using the System Snapshot component link to information about the associated block. To enable these links, in the System Snapshot component, use the `Automatic SL format (Scalable Vector Graphics)` format.

See the System Design Description report for an example. In a Simulink model, select **File > Reports > System Design Description**. Generate the report with the file format set to `Direct PDF (from template)`. To view the System Snapshot component, in the System Design Description dialog box, click **Customize Content** and locate the component in the report setup.

Numbering Template: Number sections in Simulink Design Description reports

When you generate a report from the Simulink Editor using **Reports > System Design Description**, you can now select a new template when your file output format is one of the `from template` options. The template `Default Numbered numbers chapter and subsections of the report`. The default template numbers only chapters. When you use this template, the subsections have numbers like 1.1, 1.2, and 1.1.1.

Diff and Merge: Reuse open MATLAB sessions for diff and merge from external source control tools

Save time by using open MATLAB sessions for XML comparison and merge of Simulink models from your external source control client.

Compatibility Considerations

In previous releases, you could use the `fileComparisonDriver` function to customize external source control tools, but you could not reuse open MATLAB sessions. The `fileComparisonDriver` will be removed in a future release.

For current instructions, see: [Customize External Source Control to Use MATLAB for Diff and Merge](#).

R2016a

Version: 5.0

New Features

Bug Fixes

Scalable Report Generation: Generate PDF reports as big as 10,000 pages

In previous releases, the DOM API used Microsoft Word to generate PDF documents from Word documents. In R2016a, the DOM API generates PDF documents directly. The direct-to-PDF capability allows you to programmatically generate large PDF documents on all platforms supported by MATLAB. This release also adds a new output type to Report Explorer, `Direct PDF (from template)`, based on the direct-to-PDF capability. The new output type speeds up PDF generation and expands and simplifies PDF formatting.

Direct-to-PDF allows you to use an HTML template to specify the fixed content and format of a report and holes for inserting generated content. The template can use a predefined set of HTML tags to specify a report's page layout, table of contents location, and the content of page headers and footers. You can also programmatically specify page layout, table of contents location, and page headers and footers. The DOM API automatically generates table of contents and page numbers. See [Create an HTML or PDF Template](#).

See these new classes associated with the new PDF capability in the DOM API:

- `mlreportgen.dom.StyleRef`

Use this class to create running page headers and footers in Word and PDF documents. See [Create Running Page Headers and Footers](#)

- `mlreportgen.dom.PDFPageLayout`
- `mlreportgen.dom.PDFPageHeader`
- `mlreportgen.dom.PDFPageFooter`
- `mlreportgen.dom.Page`
- `mlreportgen.dom.NumPages`
- `mlreportgen.dom.CSSProperty`
- `mlreportgen.dom.CSSProperties`
- `mlreportgen.dom.FOProperty`
- `mlreportgen.dom.FOProperties`
- `mlreportgen.dom.PageRef`
- `mlreportgen.dom.Heading1`
- `mlreportgen.dom.PageBreak`
- `mlreportgen.dom.Leader`
- `mlreportgen.dom.ListStyleType`

Three-Way Model Merge: Resolve conflicts between revisions and ancestor models using Simulink projects

In a Simulink project under source control, if your changes in a model conflict with another user, you can open the Three-Way Model Merge tool to resolve the conflicts. You can examine your local file compared to the conflicting revision and the base ancestor file, and decide which changes to keep. You can resolve the conflict and submit your changes.

For details, see [Merge Simulink Models from the Comparison Report](#).

Table of Contents: Add TOCs programmatically

In R2016a, you can create a placeholder for a table of contents in your DOM API report program. Use the `mlreportgen.dom.TOC` class to create the placeholder. For more information on creating a table of contents, see [Create a Table of Contents](#).

Page Numbers: Create page number placeholders programmatically

In R2016a, you can create page number placeholders programmatically in Word and PDF reports. Previously you used only a template to create the placeholders. Updating a Word document replaces the placeholders with automatically generated page numbers. Closing a PDF DOM document similarly replaces the placeholders with automatically generated page numbers. For more information, see the `mlreportgen.dom.Page` class.

HTML Text Component: Convert HTML to Word or PDF

The new HTML Text component allows you to include HTML markup in a Report Explorer report. If you specify Word or PDF as the report output type, the included HTML text is converted to Word or PDF. The conversion preserves the text's HTML-specified format. You can enter the HTML text in the component or specify a file or workspace variable that contains the text.

pptview Function: Open PowerPoint presentation or convert it to PDF

The `pptview` function opens a PPT presentation in PowerPoint® on Windows or Apache OpenOffice™ on Linux. If you have Microsoft Office installed, you can also use this function to convert a PowerPoint presentation to PDF format.

Test Sequence Options: Include description, action, transition table, and requirements in report

The `Test Sequence` component now includes options to include the description, action, transition table, and requirements in the report, or a combination of these elements.

Cross-Platform PDF Viewer: View PDF with built-in viewer

Previously, the Report Generator displayed PDF reports in Adobe® Acrobat® on Windows and in the PDF viewer you specified on other platforms. Now the Report Generator displays PDF reports in the same built-in viewer on all platforms. This change avoids the need for you to interact with a new Acrobat security dialog box on Windows every time you display a report. You also do not need to set up a viewer on Linux® and Macintosh systems.

Note The new PDF viewer does not display a report's bookmarks.

Functionality Being Removed or Changed

Functionality	Result	Use Instead	Compatibility Considerations
<code>mreportgen.dom.DOCXPageMargins</code> class	Still works for Microsoft Word output	<code>mreportgen.dom.PageMargins</code>	To prevent your code from being output dependent, replace instances of <code>DOCXPageMargins</code> with <code>PageMargins</code> .
<code>mreportgen.dom.DOCXPageSize</code> class	Still works for Word output	<code>mreportgen.dom.PageSize</code>	Replace instances of <code>DOCXPageSize</code> with <code>PageSize</code> .
<code>mreportgen.dom.DOCXRawFormat</code> class	Still works for Word output	<code>mreportgen.dom.PageRawFormat</code>	Replace instances of <code>DOCXRawFormat</code> with <code>PageRawFormat</code> .
<code>mreportgen.dom.DOCXSection</code> class	Still works for Word output	<code>mreportgen.dom.DOCXPageLayout</code> or <code>mreportgen.dom.PDFPageLayout</code>	Replace instances of <code>DOCXSection</code> with <code>DOCXPageLayout</code> . If appropriate, add a <code>PDFPageLayout</code> object.
<code>CurrentDOCXSection</code> property	Still works for Word output	<code>CurrentPageLayout</code>	Replace instances of <code>CurrentDOCXSection</code> property with <code>CurrentPageLayout</code> property.

R2015aSP1

Version: 4.1.1

Bug Fixes

R2015b

Version: 4.2

New Features

Bug Fixes

Web view search

You can perform a search in a Web View of a model. You can:

- Enter the same kinds of search strings as you can in most other Simulink search dialog boxes. For example, you can specify criteria for the search, such as restricting the search to block types.
- View the matched results in the model viewer.
- Navigate from search results to systems containing matched results.
- Sort search results based on name or path.

For details, see [Search a Web View](#).

Programmatic interface for adding content to PowerPoint presentations

You can use the PPT (PowerPoint) API to add generated content to an existing PowerPoint presentation or to create a complete PowerPoint presentation programmatically. Your presentation can capture dynamic information from a MATLAB program without you making manual updates to the presentation. Use templates, slide masters, and styles to format your presentation. To format the layout and general look of your presentation, use template slide masters, style layouts, and styles. To format specific presentation content, you can use the PPT API format objects and properties.

For examples of PPT API code, see:

- [Update Presentation Content Programmatically](#)
- [Create a Presentation Programmatically](#)

Simulink data dictionary report component

You can use the Simulink Data Dictionary Loop and Simulink Data Dictionary components to include data from Simulink data dictionaries in a report.

Test Sequence block support

You can use the Test Sequence component to include data from Simulink Test™ Test Sequence blocks in a report.

Several times faster model comparison analysis

Performance improvements in model comparisons result in typical comparisons running 2-3 times faster.

DOM API object display options

For Microsoft Word reports, you can use an `mlreportgen.dom.Display` object to specify whether to display an `mlreportgen.dom.Text` object. For HTML reports, you can use a `Display` object to specify the how to display DOM objects such as text, paragraphs, images, list items. For HTML reports, some display options are:

-
- Display as an inline or block element.
 - Display the element similar to a table column, table caption, list element, or a few other elements.
 - Suppress the display of the object.

For details, see `mlreportgen.dom.Display`.

DOM API horizontal rule

You can use an `mlreportgen.dom.HorizontalRule` object to specify a horizontal line to visually divide report content.

For details, see `mlreportgen.dom.HorizontalRule`.

R2015a

Version: 4.1

New Features

Bug Fixes

Support for appending HTML string or file to a Word or PDF report generated by the Document Object Model (DOM) API

You can append HTML markup or the contents of a whole HTML file to a programmatic report written with the Document Object Model (DOM) API to:

- Convert an existing HTML report to a Microsoft Word or PDF report.

You can append the HTML markup for a report to a DOM report, which you can then generate in Word or PDF format. For example, you can use a MATLAB `printf` statement to generate an HTML report file, and then append that HTML file to a DOM report to generate a PDF or Word version of the report. Another example is to convert a report generated by Model Advisor from HTML to Word or PDF.

- Add content based on HTML markup.

You can append the HTML code for a report to a DOM report, using the HTML code as a building block in a DOM report that includes other report elements.

Perform these steps to append HTML to a DOM report.

- 1 Ensure that the HTML code complies with HTML standards, such as including beginning and end tags (for example, `<p>` and `</p>`). Consider using a utility such as HTML Tidy.
- 2 In a DOM report, use `addHTML` to append HTML code or `addHTMLFile` to append an HTML file to a document or document part.

For example, this DOM code creates a Word report that displays Hello World, based on the HTML code that you append to the report.

```
import mlreportgen.dom.*;
d = Document('MyReport', 'docx');
addHTML(d, '<p style="color:blue"><b>Hello World</b></p>');
```

For information about using the `addHTML` or `addHTMLFile` methods, see `mlreportgen.dom.Document.addHTML` or `mlreportgen.dom.Document.addHTMLFile`.

Single-file output option for HTML reports generated by DOM API

To simplify display of an HTML report, you can specify that the DOM API generate the report as a single HTML file that includes the report images.

Specify `'html-file'` for the `type` input argument for a `Document` or `DocumentPart` constructor. For details, see the `mlreportgen.dom.Document` reference page.

Simplified table formatting with DOM API

In R2015a, the DOM API includes several enhancements to simplify creating and formatting tables.

- You can use an `mlreportgen.dom.ResizeToFitContents` format object in the `Style` property for `mlreportgen.dom.Table` or `mlreportgen.dom.FormatTable` objects. The `ResizeToFitContents` specifies whether to resize columns to fit the widest content in a column.
- You can specify an `mlreportgen.dom.Height` format object in the `Style` property of an `mlreportgen.dom.TableEntry` object. The height of a row is the tallest height specified for a

table entry in that row. If no entries in the row specify a height, then the row height is specified by the `Height` property of the `TableRow` object.

- Table entries can now inherit the border formatting specified in a `TableEntriesStyle` property of `mlreportgen.dom.Table` or `mlreportgen.dom.Table` objects.

Container for generating high-level HTML elements

You can use the `mlreportgen.dom.Container` class to create high-level HTML elements not otherwise supported by the DOM API, such as `div`, `section`, or `article`.

You can also use a `Container` object to simulate format inheritance in Word reports.

For details, see `mlreportgen.dom.Container`.

Function symbol data for MATLAB Function component

You can use the MATLAB Function component property **Include function symbol data** to generate a table with MATLAB function symbol data. The table includes information on the user-defined and (optionally) built-in MATLAB variables and functions invoked by the MATLAB function that computes the block outputs.

Images and text for DOM report links

You can append text and images to `mlreportgen.dom.ExternalLink` and `mlreportgen.dom.InternalLink` objects.

For details, see `mlreportgen.dom.ExternalLink.append`.

R2014b

Version: 4.0

New Features

Bug Fixes

Report formatting based on Word and HTML templates

You can use Microsoft Word or HTML templates for reports that you create using:

- The Report Explorer. For details, see [Generate a Report Using a Template](#).
- MATLAB report objects. See [Document Object Model](#).

MATLAB report objects for creating report scripts

You can use the DOM (Document Object Model) API to create MATLAB programs to generate reports. The DOM API provides a set of objects for creating text, paragraphs, tables, images, lists, and other kinds of report content. When you create a report program, you can:

- Produce Microsoft Word, HTML, and PDF output using the same code.
- Specify formatting for report objects, such as color and bold for text objects.
- Use format objects (for example, `Bold` and `FontFamily`) to format report objects.
- Add MATLAB data directly to report objects (for example, a char array to a `Text` object).
- Use formatting and fixed content that is defined in a Word or HTML template.
- Create form-based reports that contain fixed and generated content.
- Modularize a report into parts for generation of reports that have repetitive elements of the same format.

For information about how to create report programs, see [Document Object Model](#).

Fast file converter with reduced memory requirements

Generate template-driven reports much faster than in previous releases, without using Java memory.

To take advantage of these performance advantages, use one of these **File format** report options:

- HTML (from template)
- PDF (from template)
- Word (from template)

For details, see [Report Generation Using Templates](#).

Fill-in-the-blanks Word and HTML forms for generating reports

You can use the DOM (Document Object Model) report objects in a MATLAB program along with Word and HTML fill-in-the-blanks templates that contain fixed and holes to include generated content. The report program uses report objects to generate the variable information. See [Form-Based Reporting](#).

Color settings preferences in Simulink XML comparisons

You can now change and save your diff color preferences for the MATLAB Comparison Tool. Apply your color preferences when comparing model files, text files, MAT-files, variables, or zip files or folders.

For details, see Change Color Preferences.

Optional Web views for model coverage and requirements

If you have Simulink Verification and Validation™ installed, you can include model coverage and requirements information in a Web view. In the Web View dialog box, open the **Optional Views** tab and select **Model Coverage** or **Requirements** or both.

To view the model coverage or requirements optional Web view information in a Web view, you must have Simulink Verification and Validation installed. To view the information, in the Web view click a highlighted block (for example, blocks with an orange border have requirements information). The information for that block appears in the **Informers** pane below the model.

For more information, see Optional Web Views.

Improved performance for code generation report creation

When you use `codegen . rpt` to create code generation reports with Simulink Report Generator, in the Report Options dialog box on the **Properties** pane, the **Compile model to report on compiled information** check box is selected by default. With this option, the software updates a model only once when creating the report. You get much faster report generation, especially for models with many atomic subsystems.

R2014a

Version: 3.16

New Features

Bug Fixes

Compatibility Considerations

Object Inspector in Web view to view block parameter and signal property values

When exploring a Web view, you can view click a block or signal to view block parameter and signal property values in a separate pane. For details, see [View Block Parameters and Signal Properties](#).

Options for name, location, and packaging of Web view files

Specify the file name and folder to use for storing the exported Web view. You can also specify whether to package the Web view files a zip file, in a folder, or in both a zip file and a folder. For details, see [Web View Files](#).

Enhanced Web view export dialog, with context-sensitive help

The Web View export dialog box has been enhanced to make it easier to specify export options. To get detailed information about an option, right-click the option and select "What's This?" For details, see [Export Models to Web View Files](#).

Extended and simplified slwebview command

The `slwebview` command, which you can use to export a Web view, includes an option for specifying whether to export MathWorks® built-in library blocks.

The interface for whether to export masked subsystems has been simplified.

Compatibility Considerations

The default for the `FollowLinks` argument has been changed to `Off`.

The syntax for specifying whether to export masked subsystems has been simplified. The following argument values relating to masked subsystems are supported for compatibility with releases before R2014a. Use `false` or `true`.

Previous 'LookUnderMasks' Argument Value	Meaning
None	Equivalent to <code>false</code>
Graphical (Default)	Equivalent to <code>true</code>
Functional	Equivalent to <code>true</code>
All	Equivalent to <code>true</code>

R2013b

Version: 3.15

New Features

Bug Fixes

Merge capability for MATLAB Function blocks from XML comparison report

From the XML comparison report, you can compare the contents of MATLAB Function blocks. You can now merge selected changes in the block code. This enables you to merge changes made by developers in different parts of the MATLAB Function block code without leaving the Simulink environment.

In the XML comparison report of the parent model, click **Compare** next to the MATLAB Function block to open a new text comparison report showing the differences in the code inside the blocks. Use the merge controls on the Comparison tab to merge selected differences.

For details, see Merge Simulink Models from the Comparison Report.

Protected model Web view

You can view a Web view of a protected model. For details, see Protected Web View.

R2013a

Version: 3.14

New Features

Bug Fixes

Toolbar that replaces menus and toolbars in XML Comparison Tool

The Comparison Tool is redesigned for more efficient workflow and access to tools. The toolbar contains components that were previously available in menus and toolbars. Tools are unified across different comparison types such as text, MAT, variable, and Simulink XML, to make it easier to find controls for displaying, filtering, and navigating differences.

For details, see [Explore the XML Comparison Report for Simulink Models](#).

Scope Snapshot display of additional scopes

The Scope Snapshot component, in addition to inserting images of the Simulink library Scope and XY Graph blocks, now supports:

- Spectrum Analyzer and Time Scope blocks (DSP System Toolbox™)
- Video Viewer (Computer Vision System Toolbox™)
- Blocks in the Simulink Control Design™ Linear Analysis Plots library (for example, the Bode Plot block)

R2012b

Version: 3.13

New Features

Bug Fixes

Inclusion or exclusion of model references from System Design Description reports

The System Design Description dialog box provides an option for you to include or exclude model references from a System Design Description report. Prior to R2012b, model references were always included. In R2012b, by default reports include model references, but you can choose to exclude them.

Simulink functions in reports

R2012b provides enhanced Stateflow reporting capabilities for Simulink functions:

- The System Loop component provides an option to exclude from a report a subsystem that implements a Simulink function.
- The new Simulink Function System Loop component provides information on the function-call subsystem hierarchy that implements a Simulink function.

R2012a

Version: 3.12

New Features

Bug Fixes

XML Comparison of Configuration Parameters

In R2012a, you can view changes in Configuration Parameters dialog box options in the Simulink XML Comparison report.

After comparing XML text files exported from Simulink models, you can use the report to explore differences in the Configuration Parameters. Click report items to view changed options in the Parameters pane of the report. The Parameters pane displays the label text from the dialog controls and the parameter values. If you have reverse annotation on, the report also displays the appropriate root node pane of both Configuration Parameters dialog boxes.

For details, see Display Items in Original Models in the Simulink Report Generator User's Guide.

Print Filtered XML Comparison Results

In R2012a, the printable XML Comparison report has improved speed and filtered results. If you set filters, you can now create a printable report containing the filtered results only.

For details, see Filter Out Differences in the Simulink Report Generator User's Guide.

R2011b

Version: 3.11

New Features

Bug Fixes

Use Full Page Option for PDF Reports

To display full-page images in PDF reports, set the **Paper orientation** property to the new **Full page image (PDF only)** option. The following components provide this new option:

- Look-Up Table
- Scope Snapshot
- Stateflow Snapshot
- System Snapshot
- To Workspace Plot

Filter Systems for the System Filter Component

You can use the new Custom filtering MATLAB code property of the **System Filter** component to extend the system filtering criteria beyond the existing **System Filter** properties (for example, the **Report only if system has at least N blocks** property).

Filter Properties for the Simulink Workspace Variable Component

You can define a filter for selecting the variable properties that the **Simulink Workspace Variable** component displays. You can choose from a set of properties for a class, or you can define a custom MATLAB function to specify a filter.

R2011a

Version: 3.10

New Features

Bug Fixes

Merge Simulink Models From Comparison Report

You can now merge Simulink models from the XML text comparison report. You can merge individual parameters, blocks, or entire subsystems. This feature enables you to merge two versions of a design modeled in Simulink.

You can merge from the left (or base) model to the right (or edited) model using the XML text files. If necessary, you can swap sides. Click **Undo** to revert all merge operations.

For details, see Merge Simulink Models From the Comparison Report in the Simulink Report Generator User's Guide.

Export XML Comparison Results to the Workspace

You can now export XML comparison results to the MATLAB base workspace. You can use the results data for tasks such as further analysis or incorporating into different reports.

For details, see Export Results to the Workspace in the Simulink Report Generator User's Guide.

Improved PDF Images of Models and Charts

The Simulink Report Generator includes Simulink models and Stateflow charts in PDF reports as Scalable Vector Graphics (SVG) images. SVG images substantially improve the readability of Simulink models and Stateflow diagrams.

R2010b

Version: 3.9

New Features

Bug Fixes

Simulink Workspace Variable and Simulink Workspace Variable Loop Components Added

The new Simulink Workspace Variable and Simulink Workspace Variable Loop components report on Simulink workspace variable objects.

MATLAB Function Component Added

The new MATLAB Function component inserts information about MATLAB Function blocks into a report.

Model Configuration Set Component Added

The new Model Configuration Set component inserts the active configuration set of a model into a report.

System Design Description Report Enhancements

The System Design Description report, which you can generate from the Simulink Model Editor, includes additional information about these elements of a model:

- Simulink workspace variables
- Configuration sets
- MATLAB Function blocks
- Truth tables

The System Design Description dialog box includes a new stylesheet option, for specifying the report output format. Customizing the System Design Description report is simpler.

For details, see [Generate System Design Description Reports](#).

Transposable Columns for Summary Table Components

For the Fixed Point Summary Table, Stateflow Summary Table, and Requirements Summary Table components, use the new **Transpose table** check box to change the summary table rows into columns in the generated report. Specifying this option puts the property names in the first column and the values in the other columns.

Improved XML Comparison Report

Reports for comparisons of XML files exported from Simulink models have a new faster user interface, with changed parameters displayed in a separate panel for easier review. The enhanced report now has color highlighting for new and changed items. The report provides additional filter options, allowing you to filter categories of changes, such as nonfunctional changes (such as position or font), line changes, and changes in block parameter defaults.

For details, see [Explore the XML Comparison Report](#) in the Simulink Report Generator User's Guide.

R2010a

Version: 3.8

New Features

Bug Fixes

Model Reference Support for Model Loop Component

The Model Loop component has a new Model Reference option for automatically reporting on referenced models.

Improved XML Comparisons for Files with the Same Name

Comparisons of XML files exported from Simulink models with the same name now only contain information about user changes to the files. In previous releases, the comparison process introduced confusing artificial changes to both files, such as file name changes. Comparing the XML exported from two versions of the same file is a common workflow when using Configuration Management tools. The XML comparison now produces better results for this workflow.

R2009b

Version: 3.7

New Features

Bug Fixes

Compatibility Considerations

System Design Description Report

From the Simulink Model Editor, you can access a new System Design Description report. The report provides summary or detailed information about the system design represented by a model. You can specify report options using the report dialog box, or you can create a customized version of the report using the Report Explorer. For details, see [Generate System Design Description Reports](#).

Blocks Sortable by Model and Chart Layout

For looping components, you can sort blocks based on how the blocks are arranged in the model or chart. The two new **sort blocks** options are: **By layout (left to right)** and **By layout (top to bottom)**. These options reflect the model or chart layout horizontally across rows or vertically down columns.

The components with these new block sorting options are:

- Block Loop
- Chart Loop
- Fixed Point Block Loop
- Fixed Point Summary Table
- Missing Requirements Block Loop
- Requirements Block Loop
- Requirements Summary Table
- Simulink Summary Table
- Stateflow Summary Table

Reporting Compiled Simulink Model Data

The new **Compile model to report on compiled information** report output option ensures that a report reflects compiled values for Simulink parameters. For details, see [Report Explorer Report Options](#) in the Simulink Report Generator documentation.

New Components for the Requirements Management Interface

New components that support the Requirements Management Interface are:

- Missing Requirements Block Loop — Apply all child components to blocks that do not have requirements
- Missing Requirements System Loop — Loop only on systems and subsystems that do not have associated requirements
- Requirements Documents Table — Insert table of linked requirements documents
- Requirements Signal Loop — Run child components for each signal with requirements in current system, model, or block

Goto and From Block Information Added to Simulink Summary Table

The Simulink Summary Table component now reports additional information about Goto and From blocks.

For Goto block properties, the new columns are:

- **From Blk** — Name of the From block connected to the Goto block; this name links to that block's properties information in the report
- **From Blk Location** — Path to From block location in the model

For From block properties, the new columns are:

- **Goto Blk Name** — Name of the Goto block connected to the From block; this name links to that block's properties information in the report
- **Goto Blk Location** — Path to Goto block location in the model

If a Goto or From block is not connected to a corresponding From or Goto block, you see "Unconnected" in the Goto or From block name and location columns.

Improved Images in Word and RTF Reports on Windows Platforms

To provide better graphics quality in reports using the Word document format or RTF (Rich Text format) on Windows platforms, the default format for Handle Graphics®, Simulink, and Stateflow images is now Windows metafile format (the default was black and white TIFF).

You can control the image format using the MATLAB preferences or the Simulink Report Generator Preferences, or from snapshot components such as the Axes Snapshot component.

Compatibility Considerations

To change the image format back to the previous default of black and white TIFF, change the preferences.

Required Products Information for MATLAB/Toolbox Version Number Component

The MATLAB/Toolbox Version Number component now supports options to display information based on whether a Simulink model or Stateflow chart requires a MathWorks product.

Block Summary Table Link to Data Object Property Information

For Simulink data object variables, in the Block Summary table, the block variables column (third column) now provides a link from the variable name to the Simulink Functions and Variables table row for that variable. The Model Variables table includes information about the data object, such as the data type and the minimum and maximum values.

Adobe Illustrator Image File Format No Longer Supported

Compatibility Considerations

The **Image file format** options for these components no longer include Adobe Illustrator:

- Look-Up Table

- Scope Snapshot
- Stateflow Dialog Snapshot
- Stateflow Snapshot
- System Snapshot
- To Workspace Plot

If you wish to integrate a Simulink Report Generator image into Adobe Illustrator, specify a image file format supported by Adobe Illustrator, such as PDF, and then open the image file in Adobe Illustrator.

Navigation Controls for XML Comparison Report

The XML comparison report has new navigation controls to step through differences in the report. You can use the toolbar buttons or the **XML** menu to move to the next or previous group of changes. As you step through selected changes in the report, Simulink Report Generator can highlight the corresponding items in the original models.

Other usability improvements in the report include:

- Enhanced XML comparison algorithm with reduced memory consumption
- Improved matching results for many types of Simulink blocks and Stateflow elements
- Ability to filter out nonfunctional differences in the comparison result
- Simplified toolbar and **XML** menu

For details, see *Explore the XML Comparison Report* in the Simulink Report Generator User's Guide.

R2009a

Version: 3.6

New Features

Bug Fixes

New XML Comparison Example

The following new example shows you how to use Simulink Report Generator to export XML text files from two Simulink models and perform an XML comparison on the exported files:

Simulink XML Comparison

Print a Sample Time Legend

Version 3.4 introduced the Sample Time Legend to the Simulink Report Generator. With Version 3.6 you can easily print a Sample Time Legend.

R2008b+

Version: 3.5

New Features

Bug Fixes

Comparison of XML Files Exported From Simulink Models

New comparison report for pairs of XML files, including an option to compare XML files exported from Simulink models.

XML comparison report can display highlighted results in the original models.

For details, see [Comparing XML Files Exported from Simulink Models](#) in the Simulink Report Generator documentation.

See also the following new examples:

- [Simulink XML Comparison, Including Stateflow](#)
- [Simulink XML Comparison, Including Truth-Tables](#)

R2008b

Version: 3.4

New Features

Bug Fixes

Simulink Web View Support for MATLAB Function Blocks in Stateflow

Web views support Stateflow MATLAB Function blocks, enabling the display of Stateflow MATLAB Function block code in Web views.

Simulink Web View Support for Simulink Functions in Stateflow

Web views support Stateflow Simulink functions, allowing display of function contents in Web views.

R2008a+

Version: 3.3.1

Bug Fixes

R2008a

Version: 3.3

New Features

Bug Fixes

Simulink Web View Enhancements

Performance and Scalability Enhancements

- The Simulink Report Generator software now loads only visible systems into Web views, resulting in a faster viewing experience for Web views of large Simulink models.
- Clicking the **Show All Layers** button in a Web view now displays all layers as thumbnail images in png format, resulting in faster loading of the Show All Layers view.

DocBlock Support

You can now export information contained in DocBlock blocks from Simulink models to Web views.

For more information on DocBlock blocks, see the DocBlock reference page in the Simulink documentation.

Support for Embedded URLs in Simulink Annotations

You can now specify the MATLAB web command in the format:

```
web('http://[url]')
```

as the callback function in a Simulink annotation. When you click on the annotation in the model's Web view, the Simulink Report Generator software launches the Web page with the specified URL.

For more information about specifying functions in annotations, see Annotation Callback Functions in the Simulink documentation.

MATLAB Function Block Support

You can place your cursor over a MATLAB Function block in a Web view to display the MATLAB code contained in the block.

For more information on MATLAB Function blocks, see the MATLAB Function reference page in the Simulink documentation.

R2007b

Version: 3.2.1

Bug Fixes

R2007a

Version: 3.2

New Features

Bug Fixes

Simulink Web View Enhancements

Support for hierarchical Web view navigation, which provides you with:

- Easy navigation through layers of your Web view
- Visual indications of what context you are in when you navigate through your Web view

Adobe Scalable Vector Graphics (SVG) Format Support in System Snapshot and Stateflow Snapshot Components

The System Snapshot and Stateflow Snapshot components now support Adobe Scalable Vector Graphics (SVG) format, providing enhanced graphics resolution in HTML reports.

R2006b

Version: 3.1

New Features

Bug Fixes

New Real-Time Workshop Components

Two new components report on code generated by the Real-Time Workshop® product:

Code Generation Summary

Produces version information, a list of the generated files, code generation configuration information, and, if the model includes subsystems, a subsystem map.

Import Generated Code

Imports source and header (include) files generated by Simulink Coder™ software. Also imports custom files you specify as part of your model.

R2006a+

Version: 3.0.1

Bug Fixes

R2006a

Version: 3.0

New Features

Bug Fixes

Simulink Models Can Be Exported as Web Views

You can now create a Web view of a Simulink model or Stateflow chart. A Web view is an interactive rendition of the model or chart that you can view in a Web browser. A Web view uses Scalable Vector Graphics (SVG) to render and navigate a model.

Creating a Web view requires the Simulink Report Generator product. Using a Web view in a Web browser requires SVG support in the browser, implemented natively or through a plug-in, but it does not require MathWorks products to be installed.

Because they require only Web browsers with widely available SVG support, Web views are useful for presenting models to audiences and for sharing models with colleagues who do not have MathWorks products installed.

For details, see *Export Simulink Models to Web Views* in the Simulink Report Generator documentation.

Browser Requirements and Limitations

Web views in V3.0 (R2006a) are supported only in these Web browsers:

- Firefox Version 1.5 or later, which has native support for Scalable Vector Graphics (SVG). To obtain the Firefox browser, visit <https://www.mozilla.com/>.
- Microsoft Internet Explorer® with the Adobe SVG Viewer plug-in. To obtain the SVG Viewer plug-in, visit <https://www.adobe.com/svg/>.

Web views have these limitations in Firefox:

- There is no shortcut menu.
- When you move the mouse over exported systems and diagrams, they are not highlighted, but the cursor changes.
- Fonts may be too large in some circumstances.
- Compressed SVG files are not supported.

Web views have these limitations in Internet Explorer with the SVG Viewer plug-in:

- In panning mode, the cursor does not change.
- On a Japanese Microsoft Windows system, you must edit the Windows registry to obtain Japanese text in menus. Define this key:

```
HKEY_CURRENT_USER\Software\Adobe\Adobe SVG Viewer\version\Locale
```

substituting the version of your Adobe SVG Viewer plug-in for *version*. For example, if the plug-in version is 3.03, define this key:

```
HKEY_CURRENT_USER\Software\Adobe\Adobe SVG Viewer\3.03\Locale
```

Then, define a string value for this key. The value name is `systemLanguage`, and the value data is `ja`.

Model Loop Component Reports on Libraries

The Model Loop component can now report on libraries. The Model Loop attribute pane in Report Explorer specifies which libraries to include in the report.

Model Advisor Results Reported by New Component

A new Model Advisor Report component runs Model Advisor and includes its report in the report you are generating.

User Interface and Performance Enhanced

The Report Explorer interface has a number of enhancements. The **View** menu has commands to increase and decrease font size and to show the message window if it is not displayed.

Simulink Report Generator performance has improved. Reports are often generated more rapidly and with less chance of running out of memory.

Empty Columns Removed from Summary Tables

Simulink Summary Table components now have an option to remove empty columns from the report. The option is selected by default.

R14SP3

Version: 2.3.1

Bug Fixes

R14SP2+

Version: 2.3

New Features

Bug Fixes

Table Cell Spanning

You can create advanced table layouts for Simulink and Stateflow property tables. The advanced layouts include cells that span multiple rows or columns.

Dialog Snapshot Components

You can take screen captures of Simulink and Stateflow editor dialog boxes. You can choose to display the current settings associated with an object, or document the appearance of your custom mask dialog boxes.

See Simulink Dialog Snapshot and Stateflow Dialog Snapshot in the Simulink Report Generator documentation.

Annotation Loop Component

You can include Simulink annotations in your reports. A Simulink annotation is a text field that you can edit directly in the block diagram. You typically use an annotation to display brief information about a subsystem.

For more details, see Annotation Loop in the Simulink Report Generator documentation.

Improved Graphical User Interface

The following changes and improvements have been made to the graphical user interface:

- There is an improved user interface for searching according to property/value pairs in Simulink System Loop and Block Loop components.