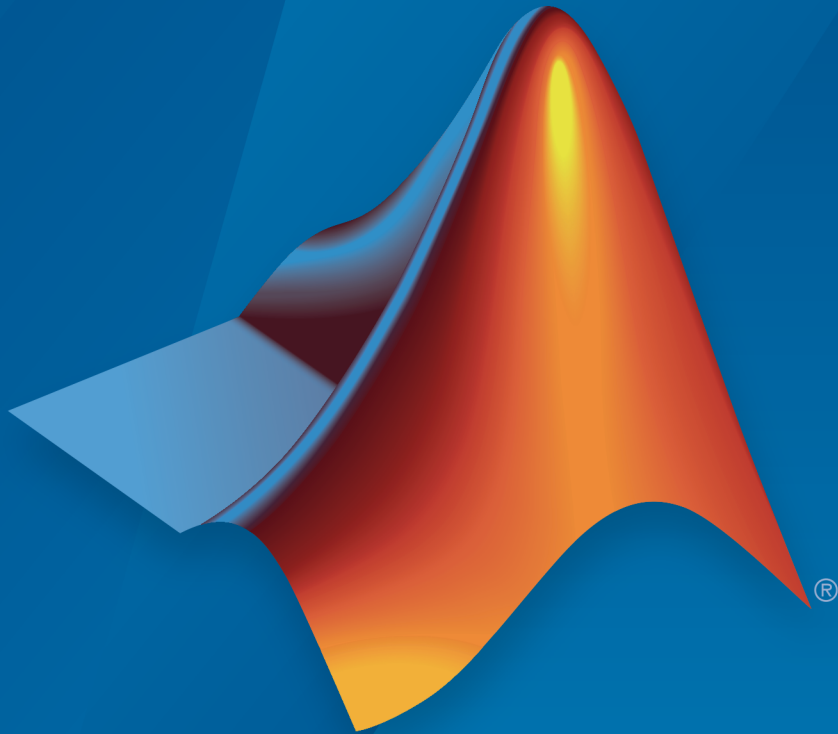# Simulink® Report Generator™

User's Guide

MATLAB®&SIMULINK®

R2016b

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

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

Phone: 508-647-7000

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

Contents

Generate System Design Description Reports

2

System Design Description

3

Creating Simulink Reports

4

Export Simulink Models to Web Views

5

Compare Simulink Model XML Files

6

7 **Components — Alphabetical List**

8 **Functions – Alphabetical List**

Getting Started

Simulink Report Generator Product Description
Design and generate reports from models and simulations

Simulink Report Generator lets you design and generate richly formatted Microsoft®
Word, Microsoft PowerPoint, HTML, and PDF reports from Simulink models and
simulations. The report generator lets you automatically create artifacts for Model-Based
Design, such as system design descriptions, generated code, requirements traceability,
and testing reports. It can produce artifacts for DO-178, ISO 26262, IEC 61508, and
related industry standards. An interactive rendition of your model can be generated for
viewing in a web browser.

Using the Simulink Report Generator you can compare models, review comparison
results in an interactive XML report, merge model differences, and create difference
reports.

Key Features

- Automatic capture of simulation results and model specifications
- Report formatting based on Microsoft Word and HTML report templates
- Interactive reports for viewing models, generated code, and analysis results in web browsers
- Report designer app for creating custom Word, HTML, PDF, RTF, and XML reports
- Artifacts for DO-178, IEC 61508, and ISO 26262, including system design, model and code verification, and requirements documentation
- API for forms-based Word, PowerPoint, HTML, and PDF report generation
- Model differencing and merging (two-way and three-way) with XML comparison tool

System Design Documentation and Results Reporting

In this section...
"Types of Reports" on page 1-3
"System Design Documentation" on page 1-3
"Results Reporting" on page 1-4

Types of Reports

Two common goals for creating reports are:

- System design documentation — Capture information about the design decisions, structure, implementation, and operational details of a system.
- Results reporting — Present results of running a system.

You use a similar workflow for creating and generating reports for both goals. However, some components are particularly useful for each use case.

System Design Documentation

System documentation helps you to:

- Capture design decisions
- Record implementation details
- Communicate the system design and interfaces among groups

When you create a Simulink Report Generator report to provide system design documentation, the report captures information about the system design from the model. Each time that you generate the report, you see up-to-date documentation for the design.

The table includes examples of components that are useful for system design documentation reports.

System Information	Examples of Components to Use
Requirements	`Requirements Summary Table` (for requirements specified with Simulink Verification and Validation™)
System layout	`System Hierarchy`, `System Snapshot`

System Information	Examples of Components to Use
Model configuration	`Model Configuration Set`, `Model Advisor Report`
Block parameter settings	`Simulink Dialog Snapshot`, `Block Loop`
Properties	`Simulink Property Table`, `Simulink Summary Table`
Variables	`Variable Table`, `Simulink Workspace Variable`
System documentation included in a model	`Documentation`, `Simulink Name`

Results Reporting

Capturing results from simulating a model is useful for:

- Model regression testing
- Verifying and validating designs
- Exploring design alternatives
- Optimizing designs

The table includes examples of components that are useful in results reports.

Results Information	Examples of Components to Use
Signal values	`Scope Snapshot`, `Block Loop`
Simulation processing	`Model Simulation`, `Model Configuration Set`, `Fixed-Point Logging Options`
Figures generated with MATLAB®	`Figure Snapshot`, `To Workspace Plot`
Generated code	`Code Generation Summary`, `Import Generated Code`

You can use components such as the `Model Simulation` component to control how the model simulates. Other components, such as the `Scope Snapshot`, show the results of the simulation.

Generate Reports Without Customizing

In this section...
"Predefined Standard Reports" on page 1-5
"Web View" on page 1-5
"XML Comparison Report" on page 1-6

You can use Simulink Report Generator without customizing reports by using:

- Predefined standard reports
- Web view
- XML Comparison report

Predefined Standard Reports

Simulink Report Generator comes with two predefined, standard reports for Simulink:

- System Design Description
- Design Requirements (requires Simulink Verification and Validation)

You generate these reports using the **File** > **Reports** menu in the Simulink Editor. The System Design Description report provides summary or detailed information about a system design represented by a model. You can choose report options by using the report dialog box, or you can create a customized version using the Report Explorer. You can use the System Design Description report setup file as a starting point for creating a setup file for your own report. For details, see "Generate a System Design Description Report" on page 2-6.

The Design Requirements report includes information about all the requirements associated with the model and its objects. You must have the Simulink Verification and Validation product installed to use the Design Requirements report. For details, see "Customize Requirements Traceability Report for Model" in the Simulink Verification and Validation documentation.

Web View

A Web view is a view of a model that you can explore in a Web browser. Web views are useful for presenting models to audiences and for sharing models with colleagues who do

not have MathWorks® products installed. You can use Web views to navigate subsystems and see properties of blocks and signals. For details, see "Export Models to Web View Files" on page 5-4.

XML Comparison Report

You can use Simulink Report Generator software to compare XML text files of different versions of a Simulink model to explore differences between versions. You can also compare the XML text files of two different models. For details, see "Model Comparison".

Approaches to Creating Reports

You can create and generate reports:

- Interactively, using the Report Explorer
- Programmatically, using the DOM (Document Object Model) API
- Programmatically using the PPT (Microsoft PowerPoint®) API

You can use the Report Explorer interface to create reports without having to write code.

Using the programmatic approach, you can integrate report generation into analysis and testing applications. For more information, see "Programmatic Report Creation".

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 your making manual updates to the presentation. You can use templates, slide masters, and styles to format your presentation. For more information, see "Programmatic PowerPoint Presentation Creation".

Interactive Report Creation Workflow

Use this general approach for creating reports interactively.

1 Open the Report Explorer. In the Simulink Editor, select **Tools** > **Report Generator**.

2 Create a report setup file for your new report design.

3 Add components to the report setup file. Components determine the behavior and contents of your report. You can use the supplied components and you can create your own custom components.

4 Choose a Microsoft Word, HTML, or PDF template or a Report Explorer stylesheet to associate styles with the report setup file.

5 Generate the report.

Related Examples

· "Generate a Report Using a Template"

· "Add Report Content with Components" on page 4-5

More About

· "Report Setup"

· "Layout Stylesheets"

Report Components

In this section...
"About Report Components" on page 1-9
"Report Structure Components" on page 1-10
"System-Based Components" on page 1-10
"User-Supplied Information Components" on page 1-11
"Dynamic Reporting Components" on page 1-11
"Format Control at the Component Level" on page 1-12

About Report Components

Include components in a report setup file to specify report behavior and insert content, such as tables, lists, and figures, into a report. Use the Report Explorer to add components to a report and to specify their behavior.

Use a combination of these types of components in your report setup file.

Component Type	Description
"Report Structure Components" on page 1-10	Include a title page, sections, and other components to organize the content of a report.
"System-Based Components" on page 1-10	Include components that obtain information directly from a model to include in a report.
"User-Supplied Information Components" on page 1-11	Include text and graphics that you supply, independent of a model.
"Dynamic Reporting Components" on page 1-11	Set up dynamic control for when to include components and what information to report on for a component, based on data from a model or on other conditions that you specify.

Report Structure Components

To add a title page, use a `Title Page` component. You can include an abstract and legal notice information. For an example, see "Add a Title Page" on page 4-12.

To organize a report into sections, use `Chapter/Subsection` components. For an example, see "Create a Section for Each Iteration" on page 4-28.

System-Based Components

The Simulink Report Generator includes components that get information from a model to include in a report. Using system-based components allows your report to describe the current state of a model. Once the setup file contains these components, you can generate the report whenever you want to capture the latest version of a model.

Property table components display property name/property value pairs for objects in tables. Summary table components insert tables that include specified properties for objects into generated reports. The tables contain one object per row, with each object property appearing in a column.

To use descriptive information from DocBlock blocks, use the `Documentation` component.

A few examples of system-based components include:

- `MATLAB Property Table`
- `Simulink Workspace Variable`
- `System Hierarchy`
- `Simulink Summary Table`
- `Simulink Dialog Snapshot`
- `Block Execution Order List`
- `Model Loop`
- `Model Configuration Set`
- `Scope Snapshot`

For examples of using system-based components, see:

- "Property Table Components"

- "Summary Table Components"
- "Create the Body of the Report" on page 4-20

The Simulink Report Generator also includes system-based components that contain model elements from the following Simulink products:

- Stateflow®
- Fixed-Point Designer™
- Simulink Coder™
- Simulink Verification and Validation

User-Supplied Information Components

In addition to using system-based components to extract data from a system and insert that information into a report, you can also add content that you, or others, supply. For example, to include text, use the `Paragraph` and `Text` components.

To insert a graphic from a file, use an `Image` component. To insert ASCII text, use an `Import File` component.

To include notes about the report source files, use a `Comment` component.

For an example, see "Add Introductory Text to the First Chapter".

Dynamic Reporting Components

Dynamic reporting components execute conditionally, enabling you to decide when a child component executes or how many times a child component executes. To control the report generation flow, use logical and flow components such as `Logical If`, `Logical Then`, `While Loop`, or `For Loop`.

A looping component runs its child components a specified number of times. There are several looping components, including logical loops, Handle Graphics® loops, and model and chart loops. For model and chart loops, you can control aspects such as the order in which the report sorts blocks.

For examples, see:

- "Logical and Looping Components"

Format Control at the Component Level

The output format and stylesheet that you select for a report determines most aspects of the generated report formatting. For details, see "Report Output Format".

In addition to stylesheets that control the format and layout of the report, for some components you can set properties to specify formatting details for that specific instance of a component. For example, for the `Simulink Property Table`, you can specify whether to display table borders or specify the alignment of text in table cells.

Working with the Report Explorer

About the Report Explorer

The *Report Explorer* is the MATLAB Report Generator and Simulink Report Generator graphical interface. It allows you to:

- Create and modify report setup files.
- Apply stylesheets to format the generated report.
- Specify the report file format.
- Generate reports.

To open the Report Explorer, enter `report` in the MATLAB Command Window.

- The *Outline pane* on the left shows the hierarchy of components in currently opened report setup files. Report components can reside within other report components, creating parent, child, and sibling relationships.
- The *Library pane* in the middle lists the objects available in the context of the Outline pane.

Outline Pane Context	Library Pane Contents
No report setup file is open.	Reports
Report setup file is open.	Components
Stylesheet is open.	Stylesheet attributes

- The *Properties pane* contents depend on the Outline pane context. If no report setup file is open, on the right displays tasks the Report Explorer can perform. If a report setup file is open, the Properties pane displays the properties for the item that is currently selected in the Library pane.

Outline Pane Context	Properties Pane Contents
No report setup file is open.	Tasks that the Report Explorer can perform
Report setup file is open.	Properties for the item that is currently selected After you create a report setup file, the Properties pane initially displays

Outline Pane Context	Properties Pane Contents
	properties for the report setup file as a whole.

Tip If the Report Explorer window opens with only two panes, one of the panes is hidden. You can move the vertical boundaries between the panes to reveal any hidden pane, or to make visible panes wider or narrower.

2

Generate System Design Description Reports

System Design Description

In this section...
"Predefined Standard Reports" on page 2-2
"What Is the System Design Description?" on page 2-2
"What You Can Do with the Report" on page 2-3
"Report Contents" on page 2-3

Predefined Standard Reports

From the Simulink Editor, you can access two predefined, standard Simulink Report Generator reports called:

- System Design Description
- System Requirements Traceability

The System Design Description report provides summary or detailed information about a system design represented by a model. You can choose report options using the report dialog, or you can create a customized version using the Report Explorer. For details, see "Generate a System Design Description Report" on page 2-6.

You can use the System Design Description report setup file as a starting point for creating a setup file for your own report.

You can also generate an HTML model report for Stateflow charts. For details, see "Generate a Model Report".

The System Requirements Traceability report requires that you have the Simulink Verification and Validation product installed. The System Requirements Traceability report includes information about all the requirements associated with the model and its objects. For details, see "Customize Requirements Traceability Report for Model" in the Simulink Verification and Validation documentation.

What Is the System Design Description?

The System Design Description is a prebuilt Simulink Report Generator report that describes the system design represented by a Simulink model.

By default, the Simulink Report Generator generates the report for the model from which you invoke the System Design Description report option.

What You Can Do with the Report

You can use the System Design Description to

- Review a system design without having the model open
- Generate summary and detailed descriptions of the design
- Assess compliance with design requirements
- Archive the system design in a format independent of the modeling environment
- Build a customized version of the report, using the Report Explorer

Note: To view step-by-step tutorials for creating and generating a report, see the Introduction to System Design Description Reports example.

Report Contents

You can specify what kinds of information to include in the report, in terms of:

- What elements of a model to include in the report (for example, whether to include subsystems from custom libraries)
- Whether to generate a summary version or a detailed version of the System Design Description report.

For details, see "Generate a System Design Description Report" on page 2-6.

Summary Version

Section	Information
Report Overview	Model version
Root System	• Block diagram representing the algorithms that compute root system outputs • Description (if available from model) • Interface: name, data type, and other properties of the system input and output signals

Section	Information
	• Subsystems: the path and a block diagram for each subsystem • State charts • Requirements (optional)
Subsystems	• Path • Block diagram
System Design Variables	• Design variables • Functions in design variable expressions

Detailed Version

The detailed version of the report includes all the information that is in the summary form of the report, as well as more information about the system components. The atomic subsystem information is more detailed than virtual subsystem information.

Section	Information
Report Overview	Model version
Root system	• Block diagram representing the algorithms that compute root system outputs • Description (if available from model) • Interface: name, data type, and other properties of the root system input and output signals • Block parameters • Includes detailed information about MATLAB Function blocks • Block execution order for root system and atomic subsystems • Look-up tables • Simulink workspace variables • Model configuration sets • State charts • Requirements (optional)

Section	Information
Subsystems	The same type of information as the information for the root system, as well as: • Path of the subsystem in the model • (For atomic subsystems) Checksum that indicates whether the version of an atomic subsystem that generates the report differs from other versions of the subsystem • Referenced models (optional) • Subsystems from custom libraries (optional)
State Charts	• State chart • States • Transitions between the states • Junctions • Events that trigger state transitions • Data types • Targets • Truth tables

Report Captures Documentation Included in a Model

The System Design Description reports documentation included in a model, including:

• The model description (from the model properties)
• The block property `Description`
• DocBlock model documentation blocks

To enrich the generated System Design Description, add corresponding information in the model.

Generate a System Design Description Report

Generate a system design description report to create a standard report of your model from the Simulink Editor. The System Design Description report provides summary or detailed information about a system design represented by a model.

When you generate the report, you can specify layout and content options for:

- Title page contents
- Report content
- Report file format and storage location

Tip For faster report generation, set **File format** to one of the `from template` options. For example, select `Direct PDF (from template)` to output to PDF.

---

1 Open the model or subsystem for which you want to generate a report. The model must compile without error for the report to generate.

2 From the **File** menu, select **Reports** > **System Design Description**.

3 In the System Design Description dialog box, specify layout and content options for the report. To display detailed information about each option, right-click the label and select **What's This**.

4 Click **Generate**.

To create a customized version of the report, click **Customize Content**. This option creates a copy of the report setup file and opens the copy in the Report Explorer. See "Customize the System Design Description" on page 2-7.

More About

- "System Design Description" on page 2-2

Customize the System Design Description

In this section...
"Using the Report Explorer to Customize the Report" on page 2-7
"Building a Dialog Box for a Custom Report Setup File" on page 2-8

You can create customized versions of the System Design Description report by using the Report Explorer and, optionally the MATLAB tools for building graphical user interfaces.

By default, when you open a customized version of the report, the System Design Description dialog box does not open.

Using the Report Explorer to Customize the Report

To customize the System Design Description setup file in the Simulink Report Generator using the Report Explorer:

1 In the System Design Description dialog box, click the **Customize Content** button to open the Report Explorer.
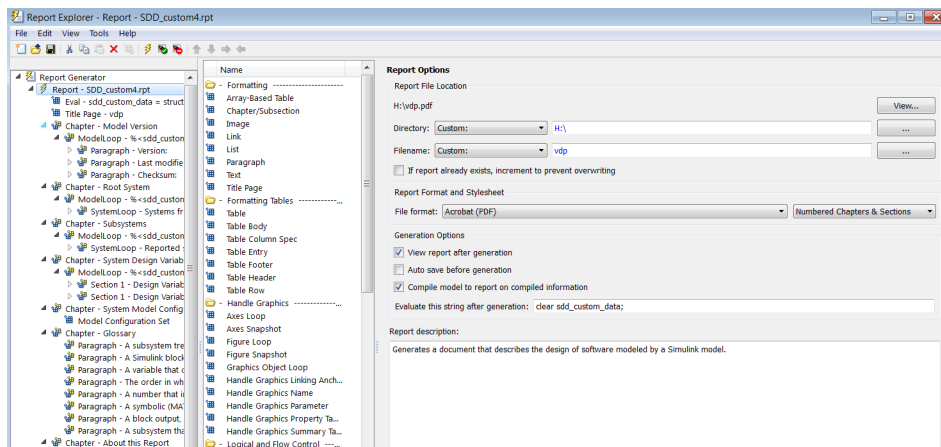
The Report Explorer reflects any changes (for example, a different report name) that you made in the System Design Description dialog box.

2 In the Report Explorer, add or modify components. See "Add Report Content with Components" on page 4-5 and "Information Components".

- Do not remove the `sdd_custom_data` structure, which is defined as:

  `sdd_custom_data = struct('model',bdroot,'rootSystem',gcs);`

  You can modify the `model` argument, which is the model for which you generated the report and the `rootSystem` argument, which is the system level in the model at which, and below which, you want to use to generate the report.

- Do not remove or modify functions that begin with `StdRpt`, such as `%StdRpt.getChecksum`

3 Optionally modify a style sheet (see "Layout Stylesheets").

4 Save the customized report with a name other than `SDD_custom.rpt`.

Building a Dialog Box for a Custom Report Setup File

To provide options for your custom report, you can create a dialog box, like the System Design Description dialog box. The dialog box that you create for your custom report can allow others to adapt the report to meet their needs, without their having to use the Report Explorer.

System Design Description

System Design Description Dialog Box

System Design Description Overview

Choose options for the content, format, and location of the generated System Design Description report. To customize the report template, click the **Customize** button to open the report in the Report Explorer.

To get help on an option

1   Right-click the option's text label.
2   Select **What's This** from the popup menu.
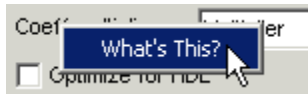
See Also

"Use Predefined Reports"

Title

Provide the title text. For PDF, Word, and RTF format reports, the title is on the title page. For HTML format reports, the title is at the top of the report.

Settings

Default: <Model name>

- Title can include letters, numbers, and special characters.
- Length is unlimited.

See Also

"Use Predefined Reports"

Subtitle

Provide the subtitle text. For PDF, Word, and RTF format reports, the subtitle is under the title on the title page. For HTML format reports, the subtitle is under the title at the top of the report.

Settings

Default: System Design Description

- If you do not want a subtitle, delete the default setting and leave the field blank.
- Subtitle can include letters, numbers, and special characters.
- Length is unlimited.

Tip

If you generate both summary and detailed versions of the report, consider reflecting the type of report in the subtitle.

See Also

"Use Predefined Reports"

Authors

List the names of the creators of the system for which you are generating the design description. List of authors is under the subtitle.

Settings

Default: Value of the `ModifiedBy` parameter for the model. The `ModifiedBy` parameter indicates the last person who updated the model.

Tip

To find the creator of the system, click the menu **File** > **Model Properties**, and then click **History** tab.

See Also

"Use Predefined Reports"

Image

Include an image on the title page (for PDF, Word, and RTF format reports) or near the top of the report (for the HTML format).

Settings

No Default

- Specify the full path to the image file that you want to include in the report or click the **Select Image** button to browse to the image file.
- Supported image file formats include:
    - GIF
    - JPEG
    - BMP
    - PNG
    - TIF

Tip

An example of an image you might want to use is a logo or other graphic for a company, division, or project involved in the system design.

See Also

"Use Predefined Reports"

Legal Notice

Provide legal notification text. For PDF, Word, and RTF format reports, the legal notice appears at the bottom of the title page (second page). For HTML format reports, the legal notice appears near the top of the report.

Settings

Default: For Internal Distribution Only

- Length is unlimited.
- The Legal Notice field does not support text formatting (such as bold or italics).

See Also

"Use Predefined Reports"

Design details

To generate a detailed system design description report, use the default (enabled). To generate a summary description, disable this option.

Settings

Default: Enabled (generate a detailed report)

- The summary report provides system design information about the root system and block diagrams for the subsystems in the model.

  - The information about the root system includes:

    - Block diagram
    - Interface: name, data type, and other properties of the system input and output signals
    - Look-up tables
    - State charts
    - Requirements (optional)

- The detailed version of the report includes all the information that is in the summary form of the report. The detailed version includes the following information, in addition to the summary information. Atomic subsystem information is more detailed than virtual subsystem information.

  - Block parameters
  - Block execution order for root system and atomic subsystems
  - (For atomic subsystems) Checksum that indicates whether the version of an atomic subsystem used to generate the report differs from other versions of the subsystem

See Also

"Use Predefined Reports"

Model references

To include model references, use the default (enabled). To exclude model references, disable this option.

Settings

Default: Enabled

See Also

- "Use Predefined Reports"
- "Model Referencing"

Subsystems from custom libraries

Include library links to subsystems defined in custom (user-created) libraries.

Settings

Default: Enabled

See Also

- "Use Predefined Reports"

Requirements traceability

Include links from blocks to the requirements that the blocks meet.

Settings

Default: Enabled

- To capture requirements links in the report, the model must include requirements links. Use the Simulink Verification and Validation product to establish requirements links.
- If you use the default (enabled) and there are no requirements links in the model, the generated report omits the Requirements section.

See Also

• "Use Predefined Reports"

Glossary and report explanation

Include a glossary of terms in the report and a description of the report contents. The glossary includes definitions of Simulink terms such as "atomic subsystem," "block diagram," "signal." The glossary helps readers who are unfamiliar with Simulink to understand the system design description. The glossary and report explanation sections appear at the end of the report and are three pages long (in PDF).

Settings

Default: Enabled

Tip

The report explanation ("About this Report") describes the information in each report section.

See Also

"Use Predefined Reports"

File format

Specify the output format for the report. Generating a report formatted by a template is generally faster than generating a report formatted by a Report Explorer stylesheet.

Settings

`Direct PDF (from template)`

Generate a PDF report from a template.

`PDF (from Word template)`

Generate a PDF report using a Word template.

`HTML (from template)`

Generate an HTML report from a template. You can choose a **Package type** option.

`Word (from template)`

Generate a Word report using a template.

`Acrobat (PDF)`

Generate a PDF report using an XSL stylesheet.

`Web (HTML)`

Generate an HTML report using an XSL stylesheet.

`Word Document`

Generate a Rich Text Format (RTF) document using a DSSSL stylesheet and convert the RTF document to a Word `.doc` file. Available only on Microsoft Windows[®].

`Rich Text Format`

Generate a Rich Text Format (RTF) document using a DSSSL stylesheet.

See Also

"Use Predefined Reports"

Stylesheet or Template

Specify the stylesheet or template to use for the report.

Settings

The settings depend on the **File format** option you choose.

When you use a `from template` option, you can use:

- The default template for the output type
- The `Default Numbered` template, which numbers sections in chapters with numbers like 1.1, 1.2, and 1.1.1
- A customized template provided at your site

When you use one of the other output options, you can choose from several standard style sheets or any customized versions provided at your site.

For the `Acrobat(PDF)` file format, these are the stylesheet options:

Default: `Default print stylesheet`

`Default print stylesheet`
>   Displays title page, table of contents, list of titles

`Standard Print`
>   Displays title page, table of contents, list of titles

`Simple Print`
>   Suppresses title page, table of contents, list of titles

`Compact Simple Print`
>   Minimizes page count, suppresses title, table of contents, list of titles

`Large Type Print`
>   Uses 12-point font (slightly larger than Standard Print)

`Very Large Type Print`
>   Uses 24-point font and landscape paper orientation

`Compact Print`
>   Minimizes white space to reduce page count

`Unnumbered Chapters & Sections`

Chapters and sections are not numbered

`Numbered Chapters & Sections`

Chapters and sections are both numbered

`Paginated Sections`

Sections are printed with page breaks

`Custom Header`

Lets you specify custom headers and footers

`Custom Titlepage`

Lets you specify custom title page content and presentation

`Verbose Print`

Lets you specify advanced print options

For the **Web (HTML)** file format, these are the stylesheet options:

Default for Web (HTML): `Default HTML stylesheet`

`Default HTML stylesheet`

HTML on a single page

`Simulink book HTML stylesheet`

HTML on multiple pages; suppresses chapter headings and table of contents

`Truth Table HTML stylesheet`

HTML on multiple pages; suppresses chapter headings and table of contents

`Multi-page Web`

HTML, with each chapter on a separate page

`Single-page Web`

HTML on a single page

`Single-page Unnumbered Chapters & Sections`

HTML on a single page; chapters and sections are not numbered

`Single-page Numbered Chapters & Sections`

HTML on a single page; chapters and sections are numbered

`Single-page Simple`

HTML on a single page; suppresses title page and table of contents

`Multi-page Simple`

HTML on multiple pages; suppresses title page and table of contents

`Multi-page Unnumbered Chapters & Sections`

HTML on multiple pages; chapters and sections are not numbered

`Multi-page Numbered Chapters & Sections`

HTML on multiple pages; chapters and sections are not numbered

For the **Rich Text Format** and **Word** file formats, these are the stylesheet options:

Default for Rich Text Format and **Word** file formats: `Standard Print`

`Standard Print`

Displays title page, table of contents, list of titles

`Simple Print`

Suppresses title page, table of contents, list of titles

`Compact Simple Print`

Minimizes page count, suppresses title, table of contents, list of titles

`Large Type Print`

Uses 12-point font (slightly larger than Standard Print)

`Very Large Type Print`

Uses 24-point font and landscape paper orientation

`Compact Print`

Minimizes white space to reduce page count

`Unnumbered Chapters & Sections`

Chapters and sections are not numbered

`Numbered Chapters & Sections`

Chapters and sections are both numbered

See Also

"Use Predefined Reports"

File name

Provide a name for the generated report file.

Settings

Default: <Name of model>

Do not include the file format extension. (For example, enter `MyReport`, but *not* `MyReport.pdf`.

Tip

If you generate both summary and detailed versions of the report, consider reflecting the type of report in the file name.

See Also

"Use Predefined Reports"

Folder

Provide a path to the folder in which to store the generated report file.

Settings

No Default

- Use a full path name
- Click the **Select Folder** button to browse to the folder where you want to store the generated report.

See Also

"Use Predefined Reports"

If report exists, increment name to prevent overwriting

Increment the file name to preserve an existing report.

Settings

Default: Enabled (increments the file name to prevent overwriting of an existing report file)

Tips

If you generate both summary and detailed versions of the report, consider reflecting the type of report in the file name.

See Also

"Use Predefined Reports"

Package type

Packaging to use for reports generated using an HTML template

Settings

Default: Zipped

Zipped

    Package report files in a single compressed file that has the report name, with a `.zip` extension.

Unzipped

    Generate the report files in a subfolder of the current folder. The subfolder has the report name.

Both zipped and unzipped

    Package the report files as both zipped and unzipped.

Dependency

To use the **Packaging type options**, set **File format** to HTML `(from template)`, choose a packaging options for the output files.

See Also

"Use Predefined Reports"

4

Creating Simulink Reports

Create a Simulink Report Generator Report

This example shows how to use the Report Explorer to design a report setup file and generate a report that does the following:

- Opens a Simulink model for the van der Pol equation, called the `vdp model`.
- Sets the `Gain` parameter for the `Mu` block to five different values.
- Simulates the model each time the `Gain` parameter is set.
- Collects the results. Results that fall within a specified range appear in a table in the generated report.

You do not need to know MATLAB or Simulink software to create and run this example report. However, knowledge of these products might be helpful for understanding the MATLAB code and model simulation that executes.

To create this report, you perform these main tasks:

- "Specify Report Options in the Setup File" on page 4-3
- "Add Report Content with Components" on page 4-5

This example includes separate sections for different kinds of report creation and generation task. Each section builds on the previous sections. However, if you want to see the report setup components for a later section without doing the previous sections, in MATLAB you can view the completed report setup file by opening `Dynamic Simulink Report`. The report is for the `vdp` model.

Note: For another set of step-by-step examples for creating and generating a report, see the Introduction to System Design Description Reports example.

Specify Report Options in the Setup File

To create and configure the report setup file:

1 Start a Simulink software session.

2 Open the Report Explorer. From the MATLAB Toolstrip, in the **Apps** tab, in the **Database Connectivity and Reporting** section, click **Report Generator**.

3 Select **File** > **New** to create a report setup file.

4 Save the report setup file.

In the Properties pane:

a Specify where to save the report setup file. To save it in the current working folder, select `Present Working Directory` from the **Directory** selection list.

b Specify the report format. In the **File format** selection list, select `Acrobat (PDF)`.

c Enter a description for the report. In the **Report description** text box, replace the existing contents with the following text.

Tip Copy and paste this code from the HTML documentation into the Report Explorer.

```
Simulink Dynamic Report

This report opens up a model, sets a block parameter
several times, simulates the model, and collects the
results. Results that fall within a specified range are
displayed in a table after the test is complete.

The report is configured to test the vdp model only.
By selecting the Eval String component immediately
below the Report component, you can modify
* model
* block
* parameter
* tested values
```

5 Click **File** > **Save As** to save the report setup file as `simulink_tutorial.rpt`.

The Outline pane on the left displays the new file name.
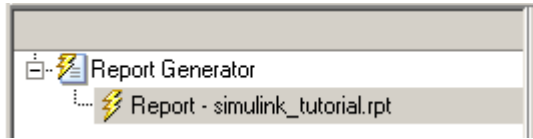
To create the content for the report, see "Add Report Content with Components" on page 4-5.

Add Report Content with Components

In this section...
"Report Components" on page 4-5
"Add MATLAB Code" on page 4-7
"Add a Title Page" on page 4-12
"Open the Simulink Model" on page 4-14
"Add Logical Then and Logical Else Components" on page 4-16
"Error If Model Cannot Be Opened" on page 4-17
"Create the Body of the Report" on page 4-20
"Process with a Model Loop Component" on page 4-21
"Add a Paragraph for Each Model" on page 4-23
"Insert a Snapshot of the Model" on page 4-24
"Add a Loop for Processing the Model" on page 4-25
"Block Parameter Value from a MATLAB Expression" on page 4-27
"Create a Section for Each Iteration" on page 4-28
"Insert the Block Value" on page 4-29
"Set a Parameter Value" on page 4-30
"Check Value Using a Logical If Component" on page 4-32
"Simulate the Model Using a Model Simulation Component" on page 4-35
"Create a Post-Test Analysis Section" on page 4-41

Report Components

Report *components* specify what information to include in the report. Components are self-contained, modular MATLAB objects that control the report-generation process and insert elements, such as tables, lists, and figures, into a report setup file. Use components to customize the appearance and output of reports.

For more information, see "Report Components" on page 1-9.

The following figure shows a sample page from the report you create in this example, and which components you use to produce this output.

Note: Do not deactivate report components that you add to the report setup file.

Chapter/Subsection component ——

Chapter 1. Model - vdp

Paragraph component ——

This report demonstrates Simulink Report Generator's ability to experiment with Simulink systems and auto-document the results. In this report, you load the model vdp and simulate it length times. This report modifies the vdp/Mu block's "Gain" value, setting it to the values [-1 0 0.5 1 2] . Each iteration of the test includes a set of scope snapsnots in the report.
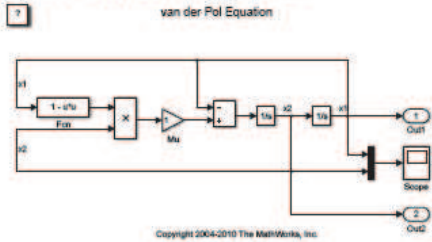
System Snapshot component ——

Chapter/Subsection component ——

Processing the vdp model

Insert Variable component ——

Iteration_Value .   -1

Figure 1.1. Scope

Scope Snapshot component ——
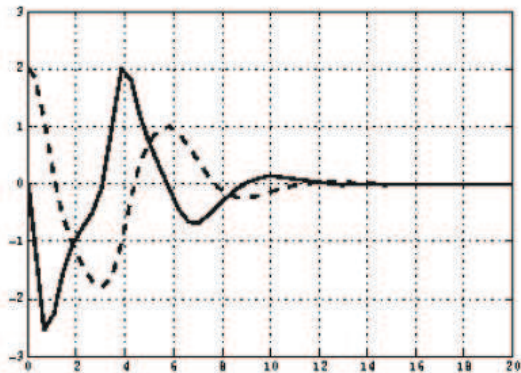
Add MATLAB Code

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

The first component to add is the `Evaluate MATLAB Expression` component, which evaluates MATLAB commands in the workspace. The code in this component assigns initial values to variables used in this example.

1 In the Outline pane on the left, select `simulink_tutorial.rpt`.
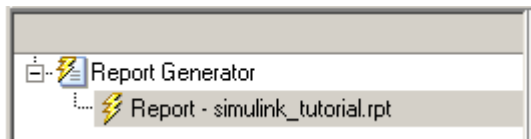
2 In the Library pane in the middle, under the MATLAB category, select `Evaluate MATLAB Expression`.

3 In the Properties pane on the right, click the icon next to **Add component to current report** to insert the component into the report.

Note: You cannot edit the component information in the Properties pane on the right until you add the component to the report.

In the Outline pane on the left, the `Evaluate MATLAB Expression` component appears under the `simulink_tutorial` report setup file. The Simulink Report Generator software abbreviates the component name to `Eval`.
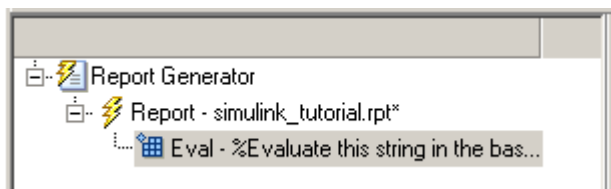
The icon in the upper-left corner of the `Eval` component's icon indicates that this component cannot have child components. By default, any components you add while the `Eval` component is selected are siblings of this component.

The options for the `Evaluate MATLAB Expression` component appear in the Properties pane on the right.

Evaluate MATLAB Expression

☑ Insert MATLAB expression in report

☑ Display command window output in report

Expression to evaluate in the base workspace: [Eval Now]

```
%Evaluate this string in the base workspace
```

☑ Evaluate this expression if there is an error:

```
rptgen.displayMessage(sprintf('Error during
%rptgen.displayMessage('Halting generation',
```

◄   ►

[Revert]   [Help]
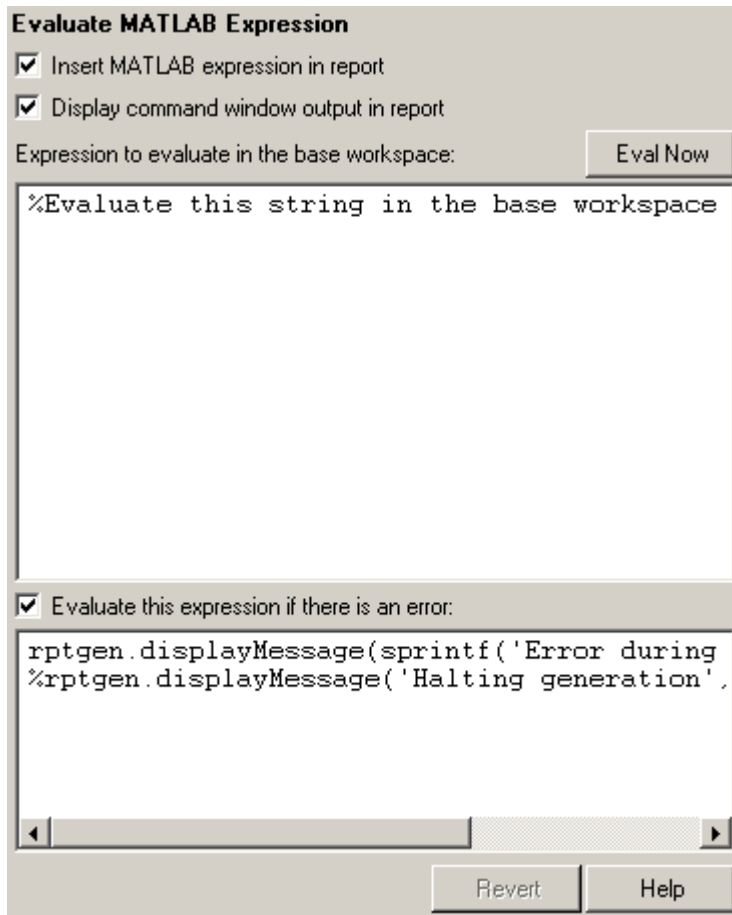
4 Clear the **Insert MATLAB expression in report** and the **Display command window output in report** check boxes so you do not include MATLAB code or output in this report.

5 Add MATLAB code to the **Expression to evaluate in the base workspace** text box to specify the following values:

- The model name
- The block name
- The block parameter
- Parameter values
- Other initial values required for processing the vdp model

Replace the existing text with the following MATLAB code.

```
%The name of the model
%that will be changed
expModel='vdp';

%The name of the block in the model
%that will be changed
expBlock='vdp/Mu';

%The name of the block parameter
%that will be changed
expParam='Gain';

%The values that will be set
%during experimentation
expValue=[-1 0 .5 1 2];

%expValue can be either a vector
%or a cell array

testMin=2.1;
testMax=3;

%---- do not change code below line ---

try
   open_system(expModel);
end

expOkValues=cell(0,2);
```

> **Note:** When you change a field in the Properties pane on the right, the field background changes color (the default is a cream color), indicating that there are unapplied changes to that field. As soon as you perform operations on another component, the Simulink Report Generator software applies the changes, and the background color becomes white again.

6 Select the **Evaluate this expression if there is an error** check box.

7 In the field under the check box, replace the existing text with the following text:

```
disp(['Error during eval: ', evalException.message])
```
The Report Explorer window now looks as follows.

Evaluate MATLAB Expression

☐ Insert MATLAB expression in report

☐ Display command window output in report

Expression to evaluate in the base workspace: [Eval Now]

```
%The name of the model
%that will be changed
expModel='vdp';

%The name of the block in the model
%that will be changed
expBlock='vdp/Mu';

%The name of the block parameter
%that will be changed
expParam='Gain';

%The values that will be set
%during experimentation
expValue=[-1 0 .5 1 2];

%expValue can be either a vector
%or a cell array

testMin=2.1;
testMax=3;

%---- do not change code below line ---
```

☑ Evaluate this expression if there is an error:

```
disp(['Error during Eval: ', evalException.message])
```

[Revert]   [Help]

Tip To run the commands that you specified in your MATLAB expression, click the **Eval Now** button. This button is located at the upper-right corner of the Report

Explorer. This is an easy way to ensure that your commands are correct and will not cause report generation problems.

8   Click **File** > **Save** to save the report setup file.

For information about handling error conditions, see "Error Handling for MATLAB Code".

Add a Title Page

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

Create a custom title page for your report using the `Title Page` component.

1   In the Outline pane on the left, select the `Eval` component.
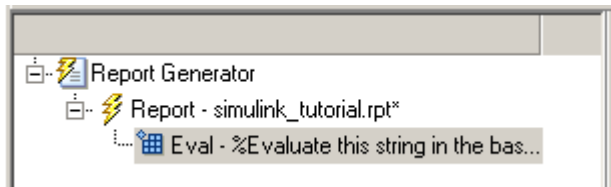
2   In the Library pane in the middle, under the `Formatting` category, click `Title Page`.
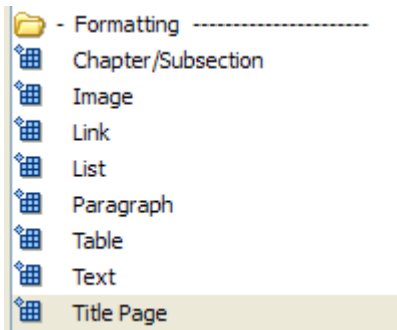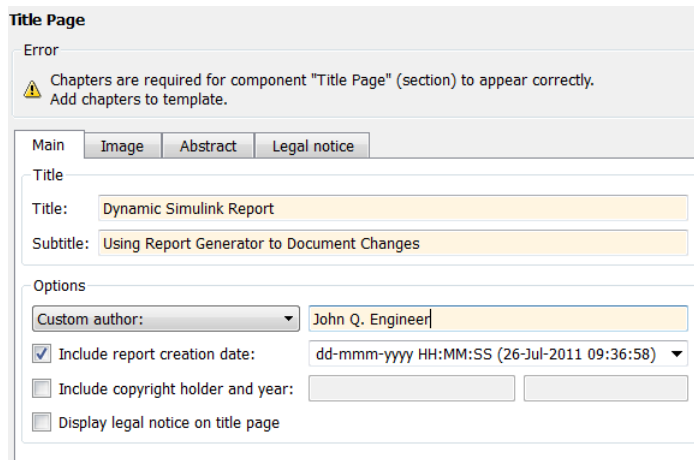
3   Click the icon next to **Add component to current report**.

The `Title Page` component appears in the Outline pane.
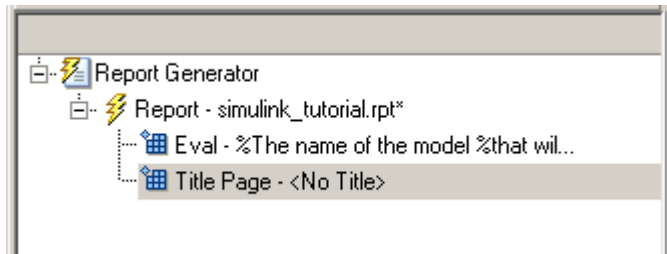
Note: To use the Title Page component, you need to have a Chapter component in your report . You have not yet added a Chapter component, so the Properties pane displays a message indicating that chapters are required for the Title Page component to appear correctly. Because later in this example you add Chapter components to this report, you can ignore that message.

---

4   In the Properties pane on the right:

   a   In the **Title** text box, enter:

      `Dynamic Simulink Report`

   b   In the **Subtitle** text box, enter:

      `Using Simulink Report Generator to Document Changes`

   c   In the **Options** section, choose `Custom Author` from the selection list.

   d   Enter your name in the text box.

   e   Select the **Include report creation date** check box.

   f   Select the default date and time format from the selection list. The Properties pane on the right looks as follows.

5 Save the report setup file.

Open the Simulink Model

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

The following statement in the `Evaluate MATLAB Expression` component that you created in "Add MATLAB Code" on page 4-7 tries to open the vdp model:

```
try
    open_system(expModel);
end
```

Tip Select the `Eval` component in the Outline pane on the left to look at this code again.

To see if the `vdp` model was successfully opened, test the result of the `open_system` command using a `Logical If` component.

1 In the Outline pane on the left, select the `Title Page` component.

2 In the Library pane in the middle, under the `Logical and Flow Control` category, select `Logical If`. This component checks to see if a given condition is true or false; in this case, if the model opened successfully.
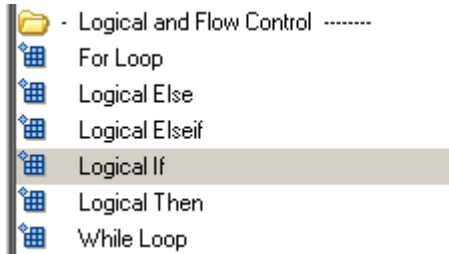
3 In the Properties pane on the right, click the icon next to **Add component to current report**. The `Logical If` component appears as `if` in the Outline pane.
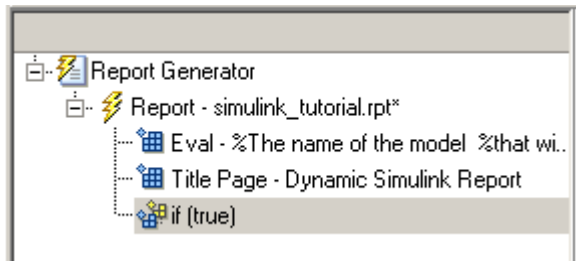
These components are child components of the report and siblings of one another. Components can have parent, child, and sibling relationships.

This component can have child components. "Add Logical Then and Logical Else Components" on page 4-16 explains how to add two child components to the `if` component.

4 In the Properties pane on the right, in the **Test expression** text box, replace the default text, `true`, with the following text:

```
strcmp(bdroot(gcs),expModel)
```
The `strcmp` function compares the name of the open Simulink model and the value of `expModel`, which was set to `'vdp'`. It tests to see if the vdp model opened successfully. `strcmp` returns `1` (`true`) if the two strings match, and `0` (`false`) if not.

5 Save the report setup file.

The if component name in the Outline pane changes to include the expression that you added.
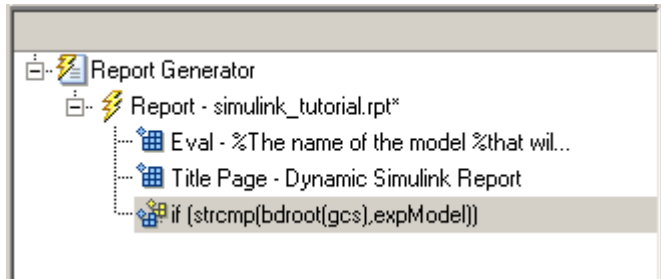
Add Logical Then and Logical Else Components

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

The `if strcmp(bdroot(gcs), expModel))` component has two possible results. Add two child components to the report setup file to process these cases.

1 In the Outline pane on the left, select the if component.
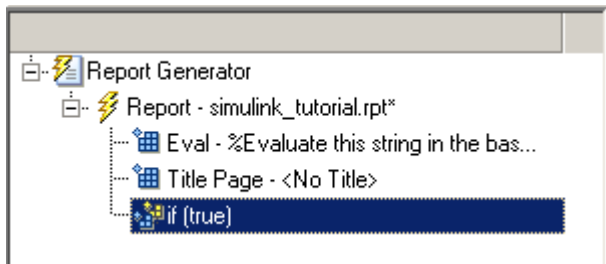
2 In the Library pane in the middle, under the `Logical and Flow Control` category, double-click `Logical Then`.

3 In the Outline pane on the left, select the if component again.

4 In the Library pane in the middle, under the `Logical and Flow Control` category, double-click `Logical Else`.

Both elements are added as child components to the if component, as shown in the Outline pane.
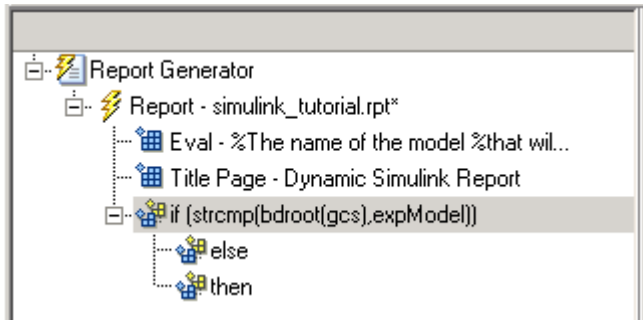
5 To move the else component under the `then` component, select the else component and click the **down** arrow on the toolbar once. The Outline pane on the left looks as follows.
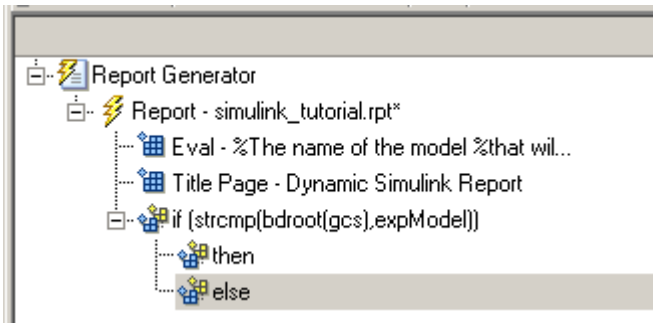
6 Save the report setup file.

Error If Model Cannot Be Opened

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

If the `if strcmp(bdroot(gcs), expModel))` component fails (the vdp model cannot open), the `else` component executes. Display an error message in the report using the `Chapter/Subsection` component.

1 In the Outline pane on the left, select the `else` component.
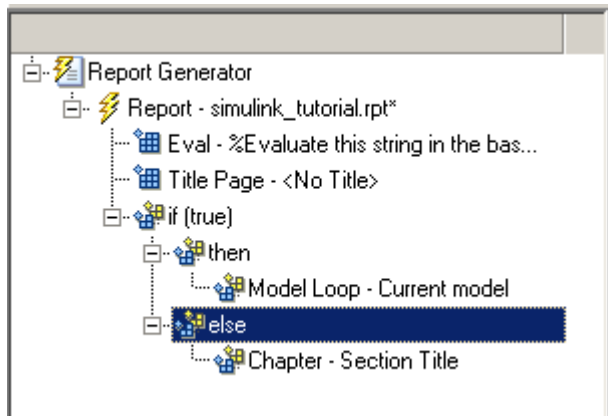
2 In the Library pane in the middle, under the `Formatting` category, double-click `Chapter/Subsection` to add it as a child of the `else` component. This component displays an error message if an error occurs when opening the vdp model.
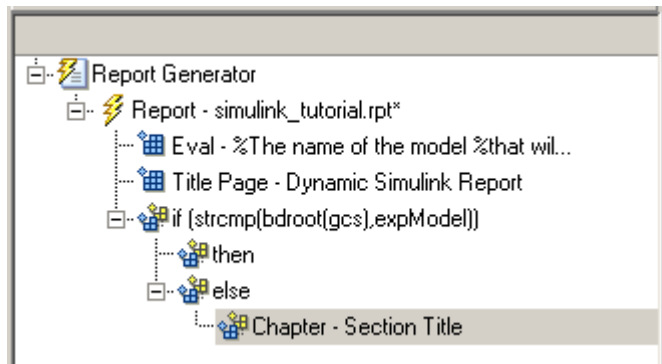
> **Note:** When you add a component to a report, it is added by default as a child component unless the selected component cannot have child components.

3 In the Properties pane on the right, choose `Custom` from the **Title** selection list, and then enter the following text in the text box:

```
Load Model Failed.
```

Save the report file.

The Outline pane looks as follows.
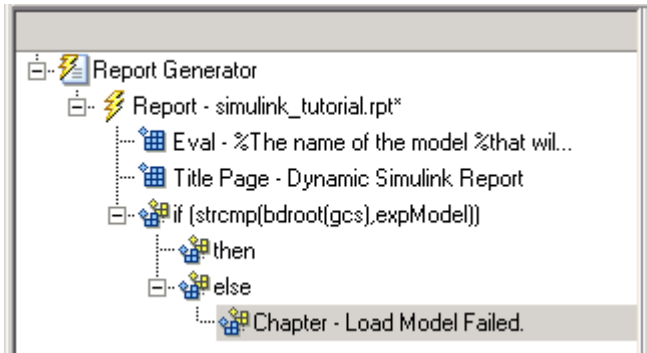
4 In the Outline pane on the left, select the `Chapter` component.

5 In the Library pane in the middle, under `Formatting`, double-click `Paragraph`.

6 In the Properties pane on the right, enter the following text in the **Paragraph Text** text box to display the following error message:

```
Error: Model %<expModel> could not be opened.
```
The expression `%<expModel>` indicates that the value of the workspace variable `expModel` is inserted into the text, as in the following example.

```
Error: Model vdp could not be opened.
```

7 In the Outline pane on the left, select the `Chapter/Section` component.

8 Save the report setup file.

The Outline pane looks as follows.
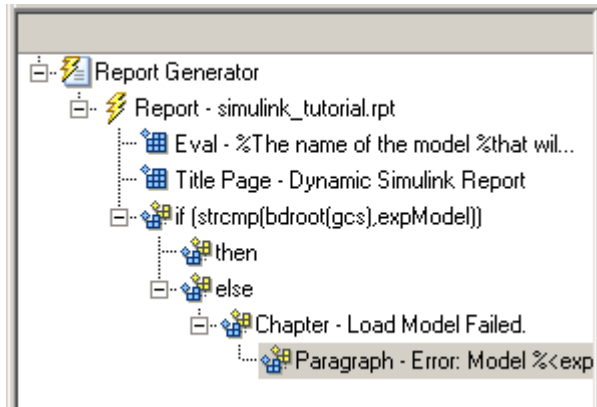
Create the Body of the Report

**Note:** This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

Creating the body of the report involves setting up components and code for dynamic execution of report components. In this example, you perform the following tasks:

- "Process with a Model Loop Component" on page 4-21
- "Add a Paragraph for Each Model" on page 4-23
- "Insert a Snapshot of the Model" on page 4-24
- "Add a Loop for Processing the Model" on page 4-25
- "Block Parameter Value from a MATLAB Expression" on page 4-27
- "Create a Section for Each Iteration" on page 4-28
- "Insert the Block Value" on page 4-29
- "Set a Parameter Value" on page 4-30
- "Check Value Using a Logical If Component" on page 4-32

- "Simulate the Model Using a Model Simulation Component" on page 4-35
- "Create a Post-Test Analysis Section" on page 4-41

Each action requires a separate component under the `then` component. For information about the then component in this report, see "Add Logical Then and Logical Else Components" on page 4-16.

Process with a Model Loop Component

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`.

The report changes the `Gain` parameter for the `Mu` block in the vdp model several times. This task requires a `Model Loop` component.

1 In the Outline pane on the left, select the `then` component.

2 In the Library pane in the middle, scroll down to the `Simulink` category, and then double-click `Model Loop`. It is added as a child of the `then` component.
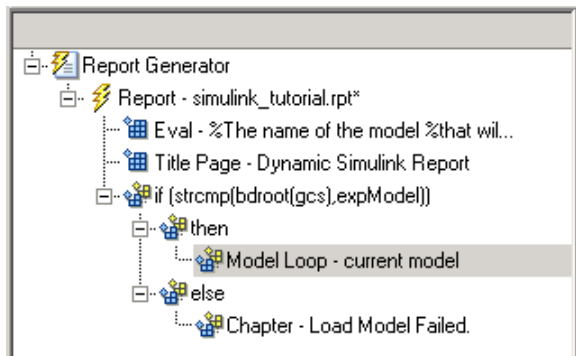
The Properties pane on the right looks as follows.
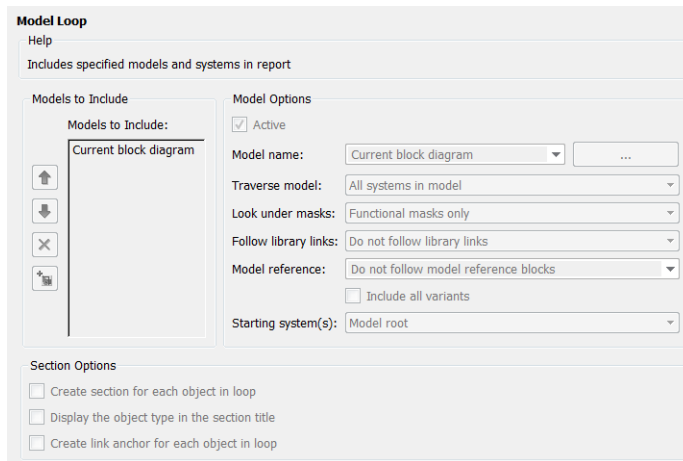
3 In the Properties pane on the right:

a Select the **Active** check box to process the vdp model.

b In the **Traverse model** selection list, select `Selected system(s) only` to traverse only the vdp model.

c Select `Model root` from the **Starting system(s)** selection list.

d At the bottom of the Properties pane on the left, select the **Create section for each object in loop** check box to create a chapter or section for each model. When you select this check box, the component name in the Outline pane on the left changes to `Model Loop Chapter`.
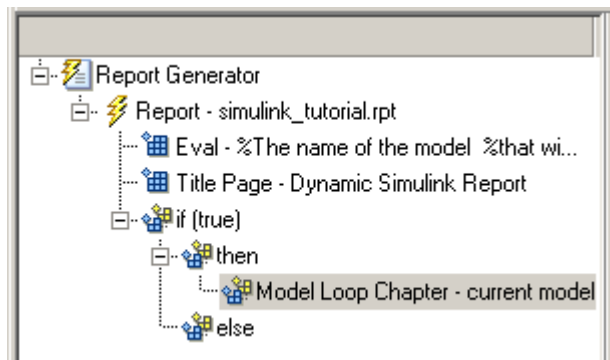
    e   Select the **Display the object type in the section title** check box to include the object type (in this example, model) in the title name.

    f   Clear the **Create link anchor for each object in loop** check box.

4   Save the report setup file.

Add a Paragraph for Each Model

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

In each Model Loop Chapter, add an explanation using the `Paragraph` component.

1   In the Outline pane on the left, select the `Model Loop Chapter` component.

2   In the Library pane in the middle, scroll up to the `Formatting` category, and then double-click `Paragraph`. The `Paragraph` component is added as a child of the `Model Loop Chapter` component.
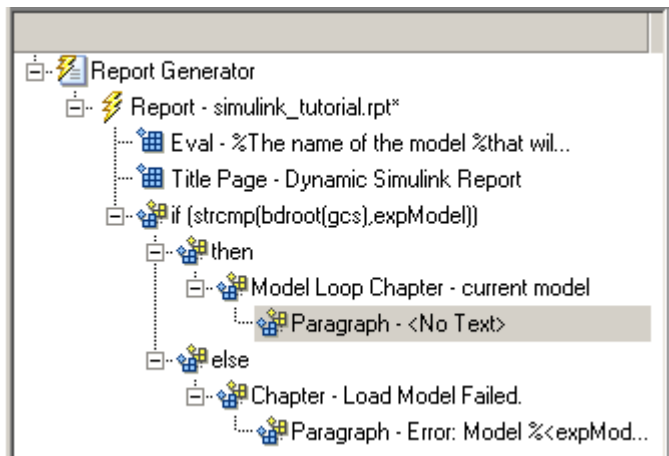
3   In the Properties pane on the right, in the **Paragraph Text** text box, enter the following text:

```
This report demonstrates Simulink Report Generator's ability
to experiment with Simulink systems and auto-document
the results. In this report, you load the model %<expModel>
and simulate it %<length> times. This report modifies the
%<expBlock> block's "%<expParam>" value, setting it to the
values %<expValue>. Each iteration of the test includes
a set of scope snapsnots in the report.
```
When this report is generated, the variable names preceded by percent signs (%)
and enclosed in brackets (<>) are replaced with the values of those variables in the
MATLAB workspace.

4   Save the report setup file.

Insert a Snapshot of the Model

Note: This section builds on the previous tasks described in the step-by-step example
summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is
for the `vdp` model.

Inside each `Model Loop Chapter` component, include a snapshot of the current model
using the `System Snapshot` component.

1   In the Outline pane on the left, select the `Model Loop Chapter` component.

2   In the Library pane in the middle, scroll down to the `Simulink` category, and then
double-click the `System Snapshot` component.

This component inserts an image of the current model into your report. The
Properties pane on the right looks as follows.
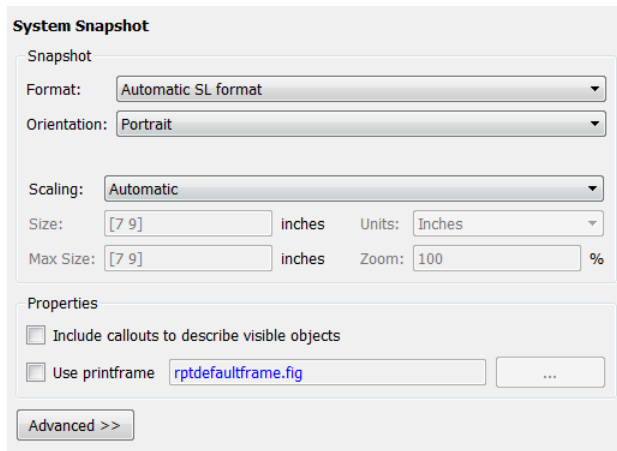
3 In the Properties pane on the right:

a Select Zoom from the **Scaling** selection list.

b Enter 70 as the **%** value.

4 In the Outline pane on the left, select the System Snapshot component.

5 Click the **down** arrow on the toolbar once to move it under the Paragraph component.
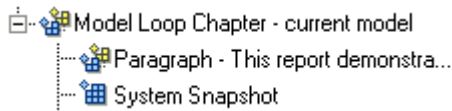
6 Save the report setup file.

Add a Loop for Processing the Model

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open Simulink Dynamic Report. The report is for the vdp model.

Create a loop to process the model `%length` times using the `For Loop` component.

1 In the Outline pane on the left, select the `System Snapshot` component.

2 In the Library pane in the middle, under the `Logical and Flow Control` category, double-click `For Loop`. The `For Loop` component is added as a sibling of the `System Snapshot` component.
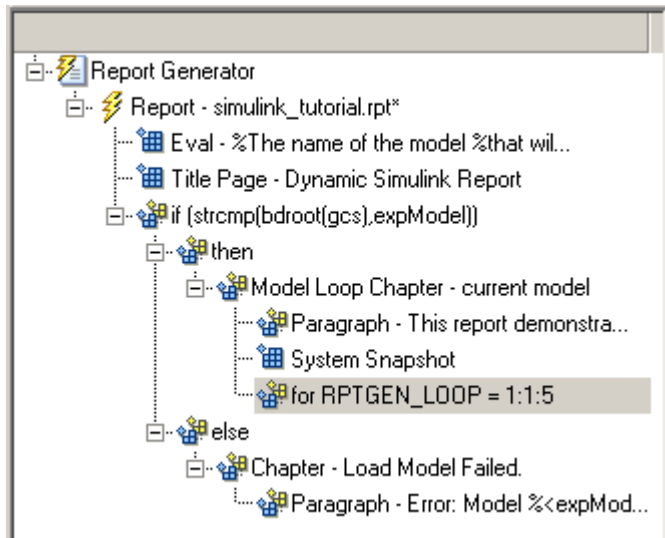
3 In the Properties pane on the right:

a In the **End** text box, replace the existing text with the following text:

```
length(expValue)
```
`expValue` is the array of `Gain` parameter values assigned in the `Eval` component with the command `expValue=[-1 0 0.5 1 2];`. The expression `length(expValue)` evaluates to `5` in this example.

b In the **Variable name** text box, replace the existing text with the name of the `for loop` variable. Enter the following text:

```
expIteration
```
The name of the `For` component in the Outline pane on the left changes to reflect the loop variable and the termination value.

4 Save the report setup file.

Block Parameter Value from a MATLAB Expression

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

For each iteration, get a value from the `expValue` array to use as the `Gain` parameter value. This task requires an `Evaluate MATLAB Expression` component.

1   In the Outline pane on the left, select the `for` component.
2   In the Library pane in the middle, under the `MATLAB` category, double-click `Evaluate MATLAB Expression`. In the Outline pane, the component name is shortened to `Eval`.
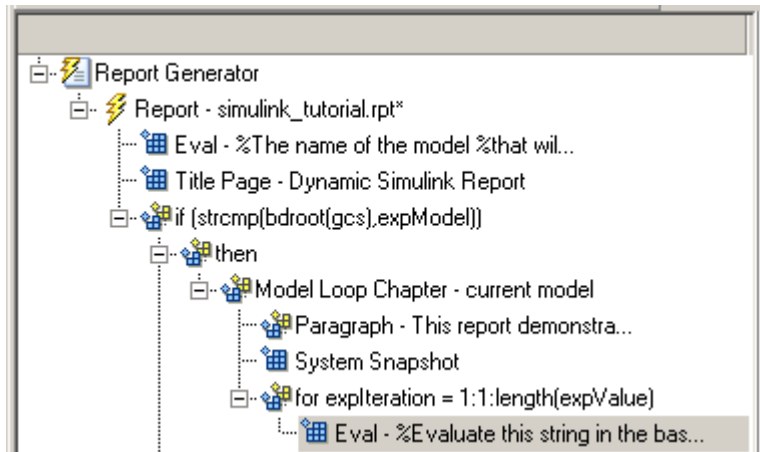
3   On the Properties pane on the right:

   a   Clear the **Insert MATLAB expression in report** and **Display command window output in report** check boxes.

   b   Enter the following text in the **Expression to evaluate in the base workspace** text box:

```
%Evaluate this string in the base workspace
```

```
if iscell(expValue)
   Iteration_Value=expValue{expIteration};
else
  Iteration_Value=...
     num2str(expValue(expIteration));
end
```

The `Iteration_Value` variable represents the designated array element.

   c Clear the **Evaluate this expression if there is an error** check box.

4 Save the report setup file.

Create a Section for Each Iteration

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

Create a separate section for each iteration of the loop that includes the data using the `Chapter/Subsection` component.

1 In the Outline pane on the left, under the `for` component, select the `Eval` component.

2 In the Library pane in the middle, under the `Formatting` category, double-click the `Chapter/Subsection` component to add it as a sibling. This component is automatically added as `Section 1` because it is inside a `Chapter` component (the `Model Loop Chapter` component).
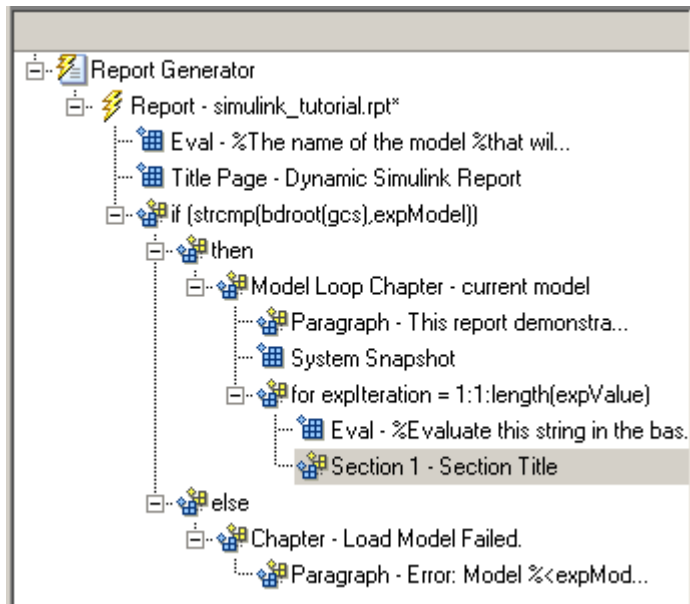
3   In the Properties pane on the right:

   a   In the **Title** selection list, select `Custom`.

   b   In the text box, enter the following title:

   ```
   Processing the vdp model
   ```

   This indicates that the section title comes from the first child component. Do not
   change any other properties.

4   Save the report setup file.

Insert the Block Value

Note: This section builds on the previous tasks described in the step-by-step example
summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is
for the `vdp` model.

Insert the `Gain` value that is used for each simulation.

1 In the Outline pane on the left, select the `Section 1` component.

2 In the Library pane in the middle, under the `MATLAB` category, double-click `Insert Variable`.

3 In the Properties pane on the right:

    a In the **Variable name** text box, enter `Iteration_Value`.

    b In the **Display as** selection list, select `Paragraph`.

The Properties pane on the right looks as follows.

4 Save the report setup file.

Set a Parameter Value

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

For each iteration, set the `Gain` parameter to the value that you extracted from the `expValue` array.

1   In the Outline pane on the left, select the `Variable` component.

2   In the Library pane in the middle, under the `MATLAB` category, double-click `Evaluate MATLAB Expression`. This component is added as a sibling of the `Variable` component.
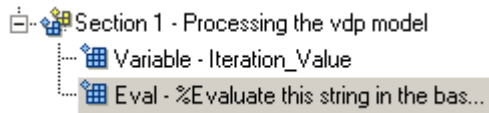
3   In the Properties pane on the right, clear the **Insert MATLAB expression in report** and **Display command window output in report** check boxes.

4   In the **Expression to evaluate in the base workspace** text box, replace the existing text with the following text.

```
set_param(expBlock,expParam,Iteration_Value);
okSetValue=(1);
```

The `set_param` command sets the value of the `Gain` parameter for the `Mu` block in the vdp model to the value of `Iteration_Value`.

5   Make sure you select **Evaluate this expression if there is an error**. Enter the following text into the text box:

```
okSetValue=logical(0);
```

If the `set_param` command works, `okSetValue` is set to 1. If an error occurs, `okSetValue` is set to `0`. The next component then reports the error and terminates processing.

6   Save the report setup file.

The Outline pane on the left looks as follows.
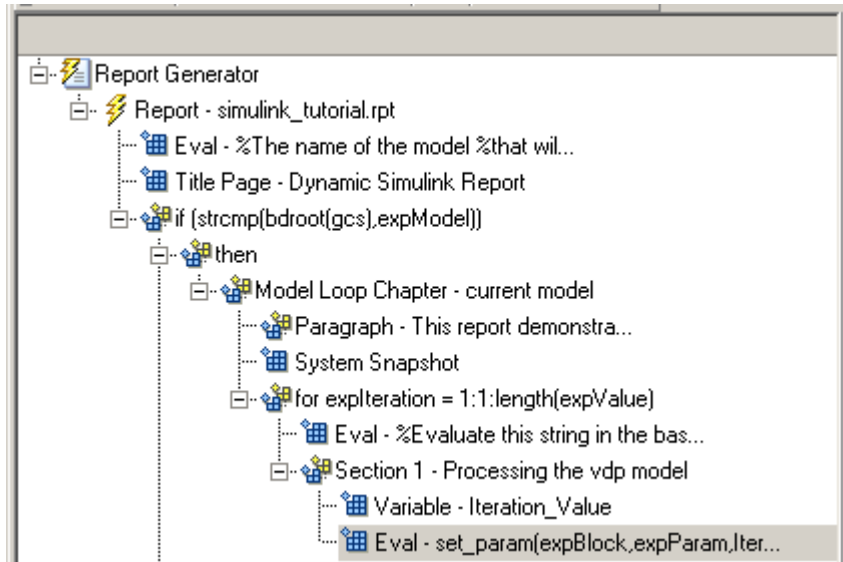
## Check Value Using a Logical If Component

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

Check the value of `okSetValue` using a `Logical If` component. If the value is `0`, the simulation cannot proceed because the `Gain` parameter could not be set.

1 In the Outline pane on the left, select the `Eval` component for the `set_param` command.

2 In the Library pane in the middle, under the `Logical and Flow Control` category, double-click `Logical If`. The component is added as a sibling of `Eval`.
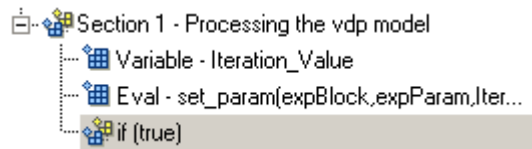
3 In the Properties pane on the right, in the **Test expression** text box, replace `true` with `okSetValue`.

`okSetValue` can be `1` (`true`) or `0` (`false`), so insert two components — `Logical Then` and `Logical Else` — to process those conditions:

1 In the Outline pane on the left, select the `if(okSetValue)` component.

2 To insert `Logical Then` and `Logical Else` in the correct order:

   a In the Library pane in the middle, double-click the `Logical Else` component.

   b Select the `if(okSetValue)` component again.

   c Double-click the `Logical Then` component. The Outline pane on the left looks as follows.
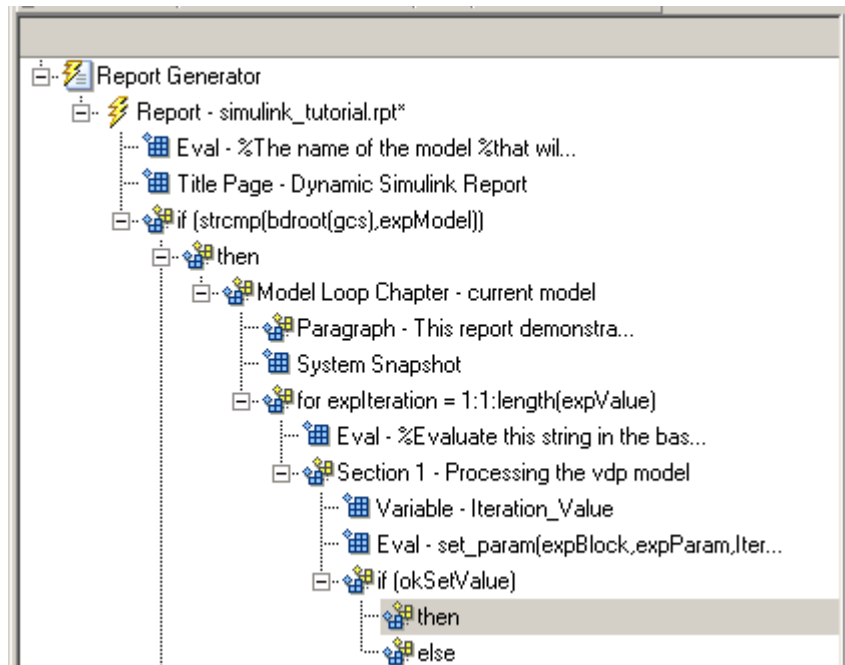
3. In the Outline pane on the right, select the `else` component.

4. In the Library pane in the middle, double-click `Paragraph`.

   If `okSetValue = 0`, the `Gain` parameter value is not set and the report displays an error.

5. In the Properties pane on the right:

   a. Choose `Custom title` from the **Title Options** selection list.

   b. Enter `Error` in the text box next to the selection list.

   c. Enter the following text into the **Paragraph Text** text box:

   ```
   Could not set %<expBlock> "%<expParam>" to value
   %<Iteration_Value>.
   ```

6. Save the report.

Simulate the Model Using a Model Simulation Component


Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

Now that the model is open and the `Gain` parameter is set, use the `Model Simulation` component to simulate the vdp model.

1   In the Outline pane on the left, select the `then` component under the `if (okSetValue)` component.

2   In the Library pane, under the `Simulink` category, double-click `Model Simulation`. In the Outline pane on the left, this component is renamed `Simulate model`.
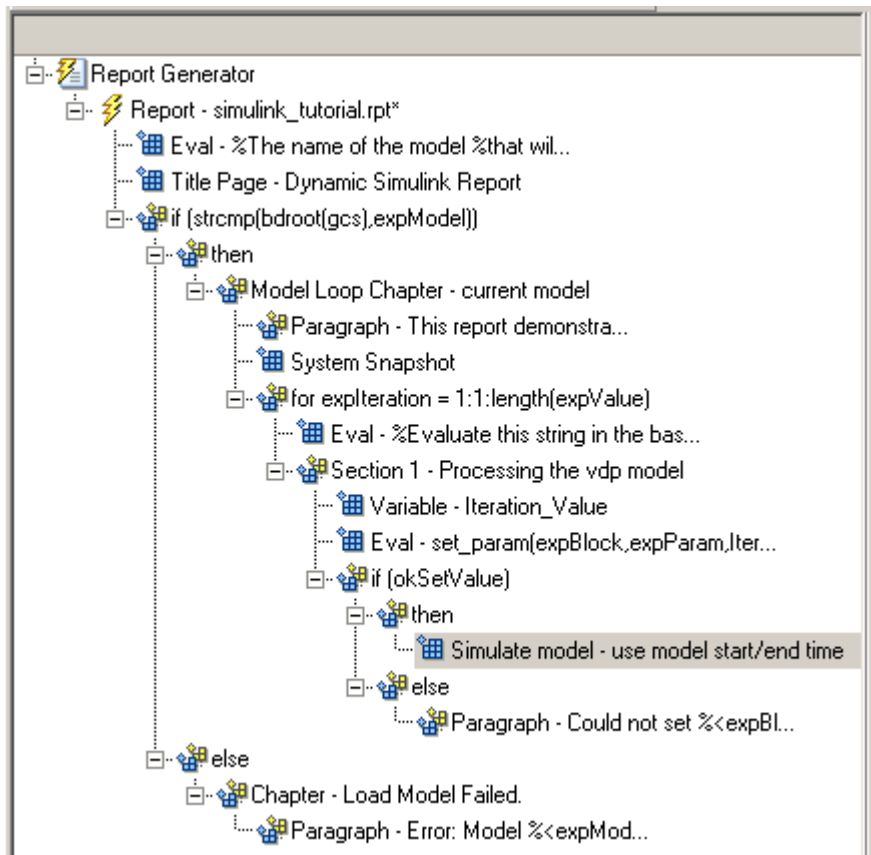
3 In the Properties pane on the right:

a Clear the **Use model's workspace I/O variable names** check box.

b In the **Time** text box, enter `dynamicT`.

c In the **States** text box, enter `dynamicX`.

d In the **Output** text box, enter `dynamicY`.

The Properties pane on the right looks as follows.

Model Simulation

I/O Parameters

☐ Use model's workspace I/O variable names

Time: dynamicT

States: dynamicX

Output: dynamicY

Timespan

☑ Use model's timespan values

Start: 0

Stop: 60

Simulation Options

☐ Compile model before simulation

Simulation status messages: Display to command line

Simulation Parameters:

Revert   Help
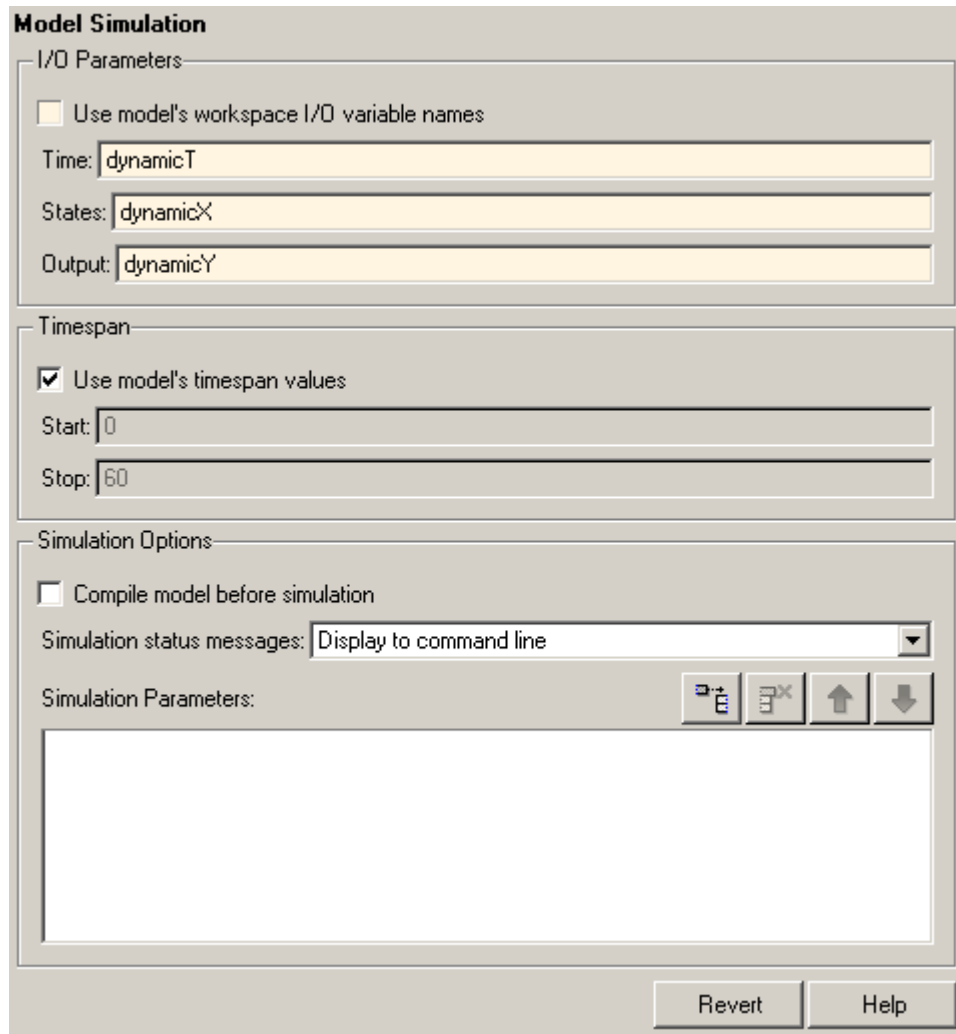
4   In the Outline pane on the left, select the `Simulate model` component.

5   In the Library pane in the middle:

   a   Scroll down to the `Simulink Blocks` category.

   b   Double-click `Scope Snapshot` to add it as a sibling of the `Simulink Model` component.

This component captures the scope for each iteration.
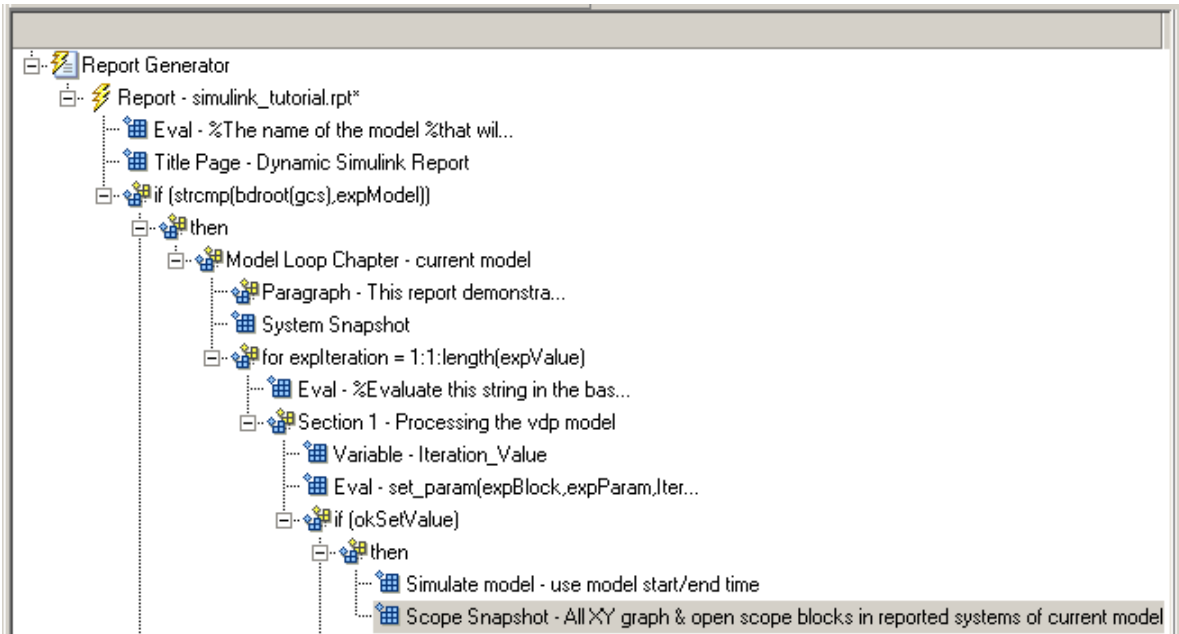
6   In the Properties pane on the right:

   a   In the **Paper orientation** selection list, select Portrait.

   b   For the **Image size**, enter [5 4].

   c   In the **Scaling** selection list, select Zoom.

   d   Enter 75 for the % value.

7   Save the report setup file.

8   To test to see if the signal data falls within a specified range, add another Logical If component:

   a   In the Outline pane on the left, select the Scope Snapshot component.

   b   In the Library pane in the middle, scroll up to the Logical and Flow Control category.

   c   Double-click the Logical If component.

9 To test the signal data, replace `true` in the **Test expression** text box with the following in the Properties pane on the right:

```
max(dynamicX(:,2))>testMin & max(dynamicX(:,2))
```

10 Save the report.

The Outline pane looks as follows:
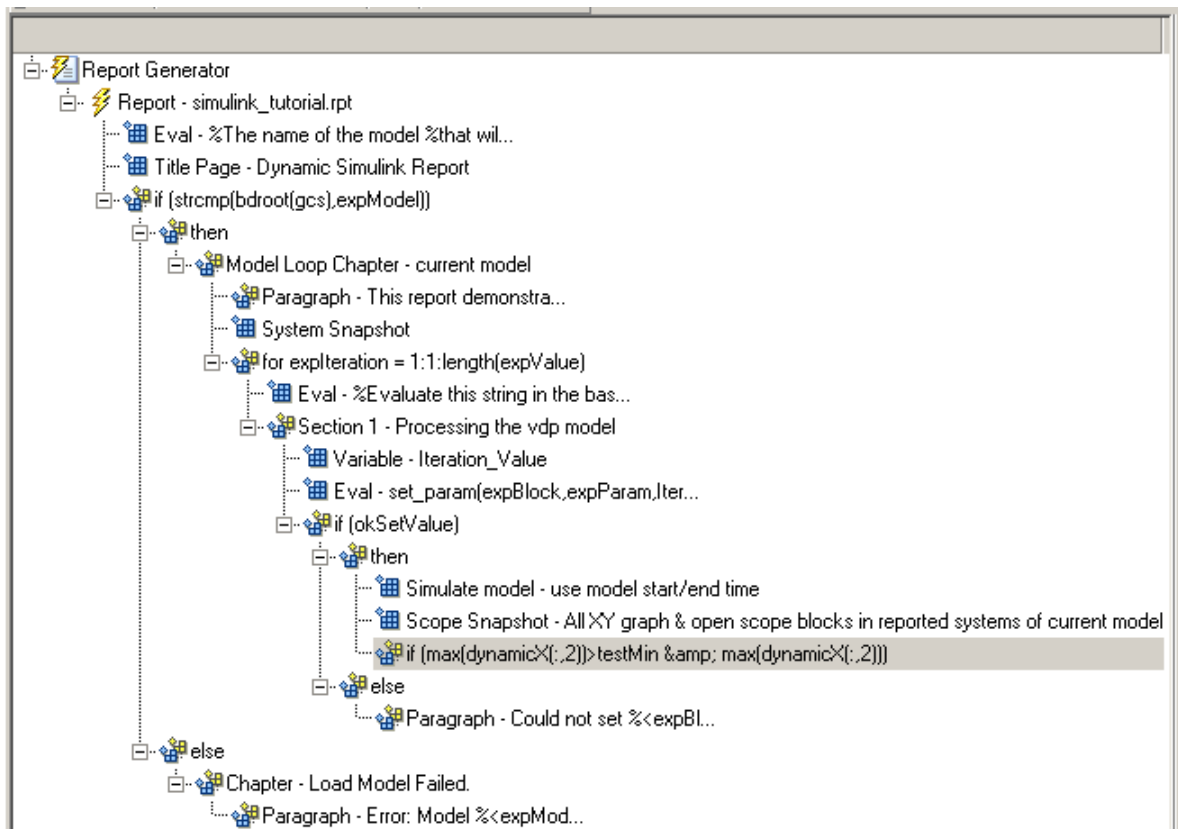
11 If this condition is true, the signal data falls within the desired range. Add a `Paragraph` component to print information about the signal data in the report.

   a In the Outline pane on the left, select the `if` component you just added.

   b In the Library pane in the middle, under the `Formatting` category, double-click `Paragraph` so that it becomes a child of the `if` component.

   c In the Properties pane on the right:

      i From the **Title Options** selection list, select `Custom title`.

      ii Type `Success` in the text box.

      iii Enter the following text in the **Paragraph text** text box.

```
The conditioned signal has a maximum value
of %<max(dynamicX(:,2))>, which lies in the
desired range of greater than %<testMin> and
less than %<testMax>.
```

12 To save the success values to insert into a table at the end of the iterations, use an `Evaluate MATLAB Expression` component.

   a In the Outline pane on the left, select the `Paragraph` component.

   b In the Library pane in the middle, under the `MATLAB` category, double-click `Evaluate MATLAB Expression`.

   An unintended result occurs: the new component is a child of the `Paragraph` component.
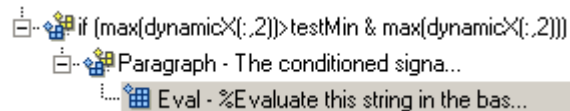
   c To make the new component a *sibling* of the `Paragraph` component, in the Outline pane on the left, select the `Eval` component, and then Click the left arrow on the toolbar. The `Eval` component becomes a sibling of the `Paragraph` component.
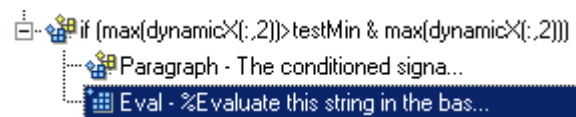
13 In the Properties pane on the right, for the `Eval` component:

a Clear the **Insert MATLAB expression in report** and **Display command window output in report** check boxes.

b In the **Expression to evaluate in the base workspace** text box, enter the following to save the desired signal values in the `expOkValues` array:

```
expOkValues=[expOkValues;...
     {Iteration_Value,max(dynamicX(:,2))}];
```

c Make sure you select **Evaluate this expression if there is an error**. Insert the following text in the text box:

```
disp(['Error during eval: ', evalException.message])
```

14 Save the report setup file.
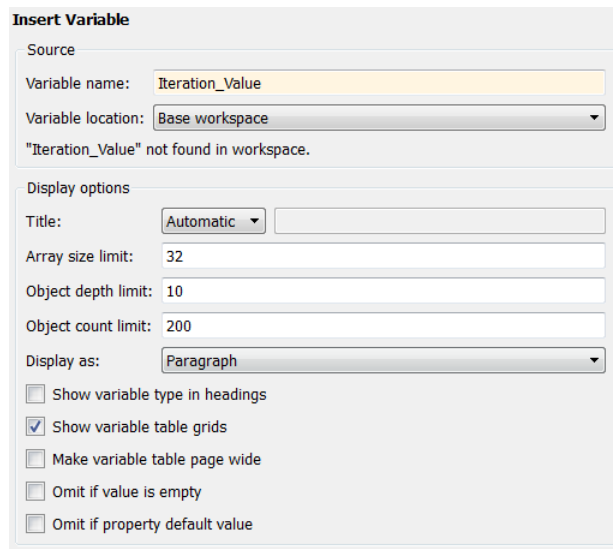
Create a Post-Test Analysis Section

Note: This section builds on the previous tasks described in the step-by-step example summarized in "Create a Simulink Report Generator Report" on page 4-2.

To see the completed report setup file, open `Simulink Dynamic Report`. The report is for the `vdp` model.

Now that you have collected all the desired values, create the post-test analysis section by creating a table and inserting it into your report at the end of this chapter.

1 In the Outline pane on the left, select the `Model Loop Chapter` component.

```
┌─────────────────────────────────────────────────────────────────────────────┐
│                                                                               │
│ ⊟  ⚡ Report Generator                                                          │
│     ⊟  ⚡ Report - simulink_tutorial.rpt                                        │
│         ┈ ▦ Eval - %The name of the model %that wil...                        │
│         ┈ ▦ Title Page - Dynamic Simulink Report                             │
│     ⊟  ▦ if (strcmp(bdroot(gcs),expModel))                                    │
│         ⊟  ▦ then                                                             │
│             ⊟  ▦ Model Loop Chapter - current model                          │
│                 ┈ ▦ Paragraph - This report demonstra...                     │
│                 ┈ ▦ System Snapshot                                          │
│                 ⊟  ▦ for expIteration = 1:1:length(expValue)                  │
│                     ┈ ▦ Eval - %Evaluate this string in the bas...           │
│                     ⊟  ▦ Section 1 - Processing the vdp model                 │
│                         ┈ ▦ Variable - Iteration_Value                       │
│                         ┈ ▦ Eval - set_param(expBlock,expParam,Iter...       │
│                         ⊟  ▦ if (okSetValue)                                  │
│                             ⊟  ▦ then                                         │
│                                 ┈ ▦ Simulate model - use model start/end time │
│                                 ┈ ▦ Scope Snapshot - All XY graph & open scope blocks in reported systems of current model │
│                                 ⊟  ▦ if (max(dynamicX(:,2))>testMin &amp; max(dynamicX(:,2))) │
│                                     ┈ ▦ Paragraph - The conditioned signa...  │
│                                     ┈ ▦ Eval - expOkValues=[expOkValues;...  ...│
│                             ⊟  ▦ else                                         │
│                                 ┈ ▦ Paragraph - Could not set %<expBl...      │
│         ⊟  ▦ else                                                            │
│             ⊟  ▦ Chapter - Load Model Failed.                                │
│                 ┈ ▦ Paragraph - Error: Model %<expMod...                     │
│                                                                               │
└─────────────────────────────────────────────────────────────────────────────┘
```

2   In the Library pane in the middle, under the `Formatting` category, double-click `Chapter/Subsection`.

The new section appears at the beginning of the chapter.
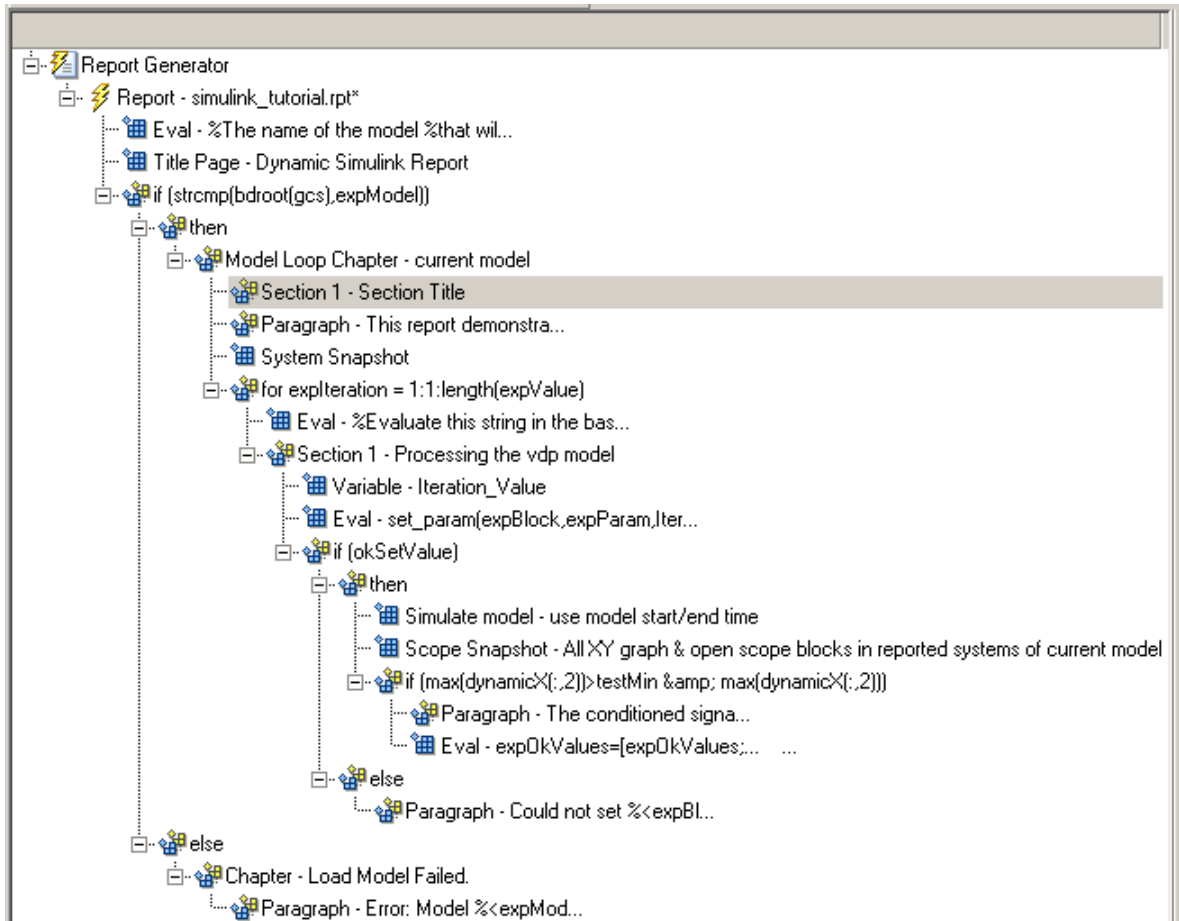
Click the **down** arrow three times so `Section 1` moves to the end of the `Model Loop Chapter` component.
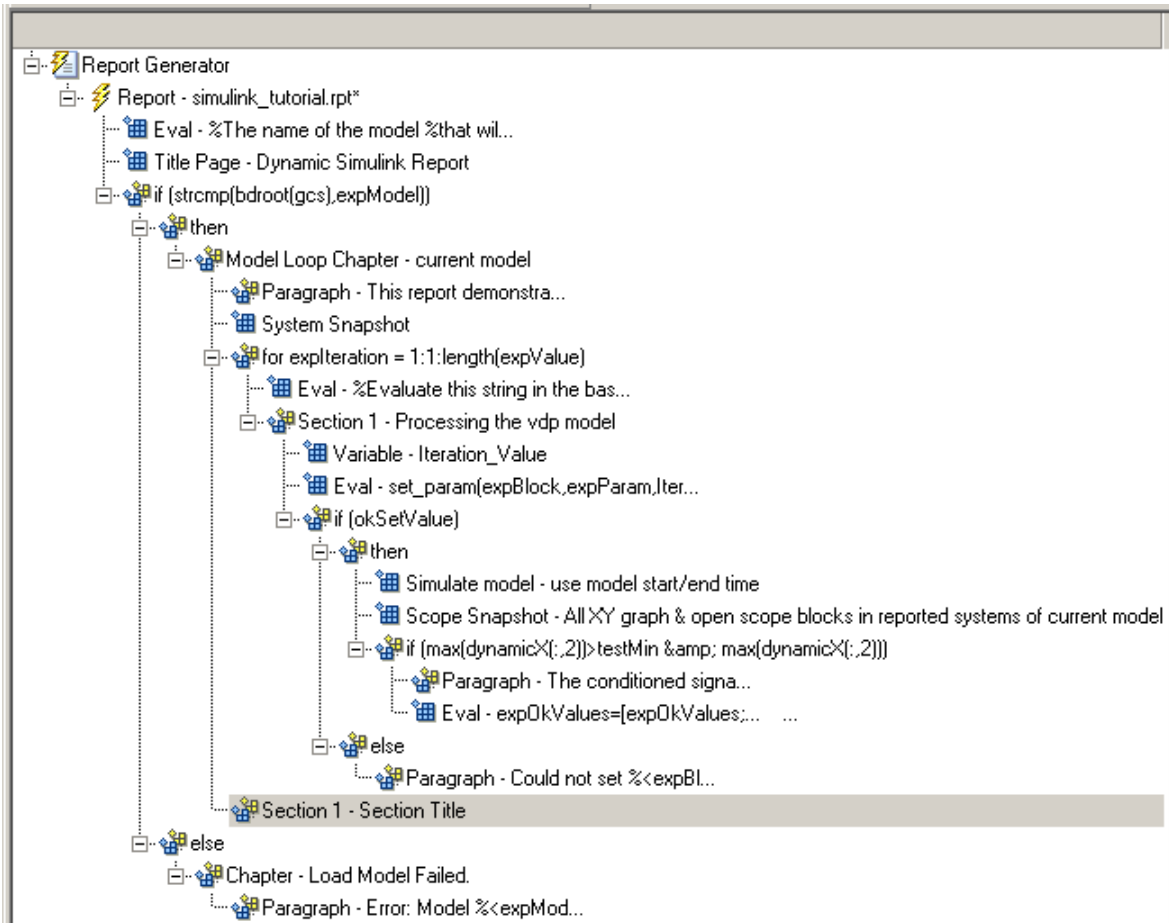
3   In the Properties pane on the right:

   a   Select `Custom` in the **Title** selection list.

   b   Enter `Post-Test Analysis` in the text box.

4   In the Outline pane on the left, select the new `Section 1` component.

5   In the Library pane in the middle, under the `Formatting` category, double-click `Paragraph`. Do not change its properties.

6 To check whether there are any signal values within the desired range, check the array `expOkValues` with a `Logical If` component. If `expOkValues` is empty, there are no signal values in the desired range. Report the result of this check.

    a In the Outline pane on the left, select the `Paragraph` component and add a `Logical If` child component.

    b In the Properties pane on the right, enter the expression to evaluate in the **Test expression** text box:

        `~isempty(expOkValues)`
    This expression evaluates to `0` (`false`) if `expOkValues` is empty; otherwise, it evaluates to `1` (`true`).

    c In the Outline pane on the left, select the `if(~isempty(expOkValue))` component and add the `Logical Else` component as a child.

    d Select the `if(~isempty(expOkValue))` component again and add the `Logical Then` component as a child.

    The two components are siblings in the Outline pane on the left.
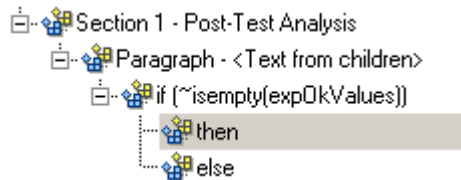
7 Save the report setup file.

8 Now, insert report components to handle the case where `expOkValues` is empty; that is, where no signal values fall within the designated range.

    a In the Outline pane on the left, select the `else` component.

    b In the Library pane in the middle, double-click the `Text` component to add it as a child of the `else` component.

    c In the Properties pane on the right, in the **Text to include in report** text box, enter the following:

        `None of the selected iteration values had`
        `a maximum signal value between %<testMin> and %<testMax>.`

9 Now handle the case where `expOkValues` is not empty and you want to insert a table of the acceptable signal values.

a In the Outline pane on the left, select the `then` component.

b Add a `Text` component as a child to the `then` component.

c In the Properties pane on the right, in the **Text to include in report** text box, enter the following text.

```
%<size(expOkValues, 1)> values for %<expBlock> were
found that resulted in a maximum signal value greater
than %<testMin> but less than %<testMax>. The following
table shows those values and their resulting signal maximum.
```

d In the Outline pane on the left, select the `Text` component under the `then` component of the `if(~isempty(expOkValues))` component.

10 To create an array for use when formatting the table, use the `Evaluate MATLAB Expression` component.

a In the Library pane in the middle, double-click `Evaluate MATLAB Expression`.

b In the Properties pane on the right:

i Clear the **Insert MATLAB expression in report** and **Display command window output in report** check boxes.

ii The next component of the report uses the strings `Mu Value` and `Signal Maximum` as table header values. Add the strings to the front of the `expOkValues` cell array by entering the following text into the **Expression to evaluate in the base workspace** text box:

```
expOkValues=[{'Mu Value','Signal Maximum'} expOkValues];
```

iii Make sure you select the **Evaluate this expression if there is an error** check box. Enter the following text into the text box:

```
disp(['Error during eval: ', evalExpression.message])
```

11 In the Outline pane on the left, select the `Eval` component.

12 In the Library pane in the middle, under the `Formatting` category, double-click the `Table` component so it becomes a sibling of the `Text` and `Eval` components.
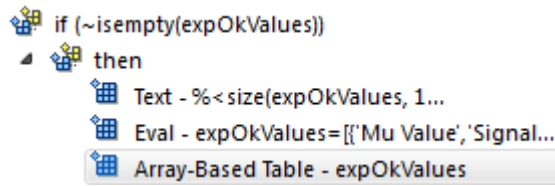
13 In the Properties pane on the right:

    a In the **Workspace variable name** text box, enter `expOkValues`. The Simulink Report Generator software uses the contents of `expOkValues` to construct the table.

    b In the **Table title** text box, enter `Valid Iteration Values`.

14 Save the report setup file.

The Outline pane on the left looks as follows.

- Report Generator
  - Report - simulink_tutorial.rpt
    - Eval - %The name of the model %that wil...
    - Title Page - Dynamic Simulink Report
    - if (strcmp(bdroot(gcs),expModel))
      - then
        - ModelLoop Chapter - current model
          - Paragraph - This report demonstra...
          - System Snapshot
          - for expIteration = 1:1:length(expValue)
            - Eval - %Evaluate this string in the bas...
            - Section 1 - Processing the vdp model
              - Variable -Iteration_Value
              - Eval - set_param(expBlock,expParam,Iter...
              - if (okSetValue)
                - then
                  - Simulate model - use model start/end time
                  - Scope Snapshot - All XY graph & open scope blocks in reported systems
                  - if (ax(dynamicX(:,2))>testMin & max(dynamicX(:,2)))
                    - Paragraph - The conditioned signa...
                    - Eval - expOkValues=[expOkValues;... ...
                - else
                  - Paragraph - Could not set %<expBl...
        - Section 1 - Post-Test Analysis
          - Paragraph - <Text from children>
            - if (~isempty(expOkValues))
              - then
                - Text - %<size(expOkValues, 1...
                - Eval - expOkValues=[{'Mu Value','Signal...
                - Array-Based Table - expOkValues
              - else
                - Text - None of the selected ...
      - else
        - Chapter - Load Model Failed
          - Paragraph - Error: Model %<expMod...

Generate the Report

To generate the report, click the Report icon on the toolbar. The following occurs:

1   A Message List window appears, displaying informational and error messages as the report is processed. Specify the level of detail you would like the Message List window to display while the report is being generated. Options range from 0 (least detail) to 6(most detail). Click the selection list located under the title bar of the Message List window to choose an option.

    `Message level 3 (Important messages)` is used for the remainder of this example.
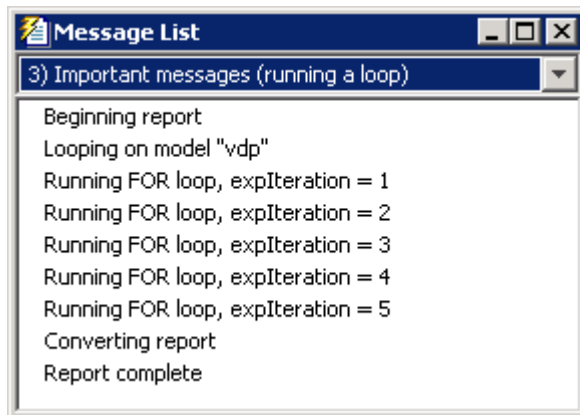
2   The vdp model appears. You can see each time it is simulated.

3   The scope window appears. The scope graph changes each time the parameter value changes.

4   Each component of the report is highlighted as it executes, in the Outline pane on the left in the Report Explorer window.

When the report is complete, Adobe® Acrobat® Reader opens your report in PDF format.
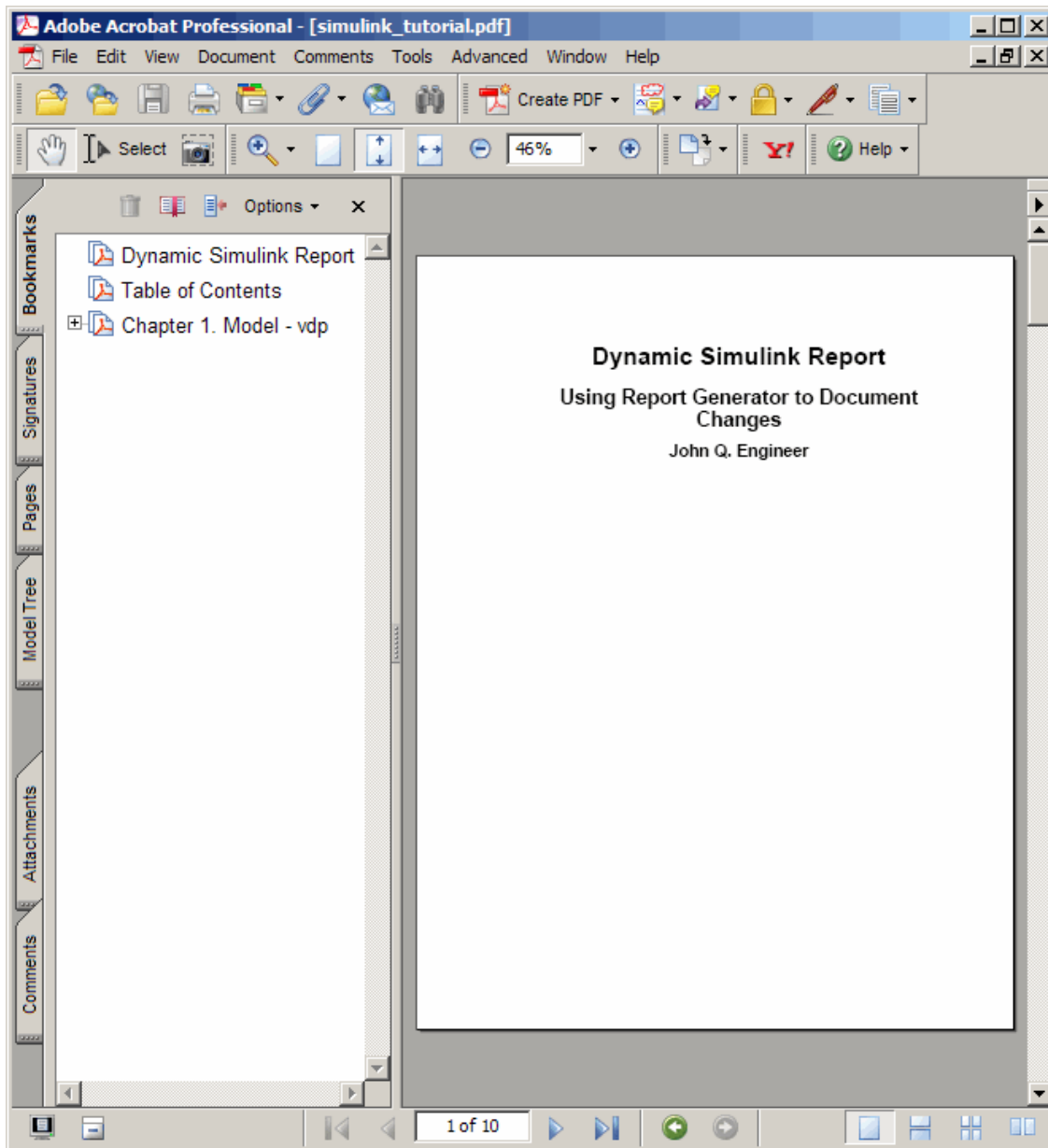
Generate a Report Associated with a Model

You can associate a report with your model. By associating a report with a model, you can use a script to generate the report without specifying the name of the report. You can change the association without modifying the script.

Associating a report with a model sets the model parameter `ReportName` to the name of the report. Each model can have only one report associated with it. You can associate the same report with more than one model.

1   Open the model you want to associate with a report.
2   In the Report Explorer, from the hierarchy view, select `Report Generator`. The library pane lists your reports.
3   Select the report you want to associate with your model.
4   In the properties pane, under **Simulink**, click **Associate report with Simulink system**. The report and model names are part of the button label.
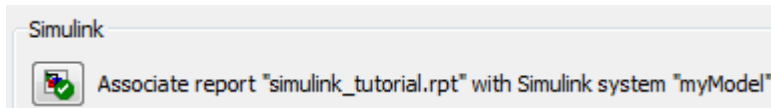
5   Save the model.
6   Create a script to generate the report using `report`. Make sure the model and report template are on the MATLAB path when you run the script.

```
load_system('myModel')
report(get_param('myModel','ReportName'))
bdclose('myModel')
```

You can clear the association clicking **Un-associate Simulink system**.

See Also
report

Related Examples
•   "Select Report Generation Options"

Logical and Looping Components

Logical and looping components execute conditionally, determining when a child component executes or how many times a child component executes.

A *looping component* runs its child components a specified number of times. There are several looping components, such as logical loops, Handle Graphics loops, and model and chart loops. For model and chart loops, you can control aspects such as the order in which the report sorts blocks.

For an example that uses loop components, see "Edit Figure Loop Components".

You can use loop context functions with loop components. For details, see:

- "Filter with Loop Context Functions" on page 4-53
- "Loop Context Functions" on page 4-55

Filter with Loop Context Functions

In this section...
"Create and Save the Setup File" on page 4-53
"Add Components" on page 4-53
"Run the Report" on page 4-54

Use loop context functions to filter the modeling elements to report on and to perform special reporting on specific elements.

In the following example, in a Block Loop component, you use `RptgenSL.getReportedBlock` in a Logical If component to report on targeted blocks within a Block Loop component.

For a summary of loop context functions, see "Loop Context Functions" on page 4-55.

Create and Save the Setup File

1 Open the f14 model.
2 At the MATLAB command prompt, enter:

   `report`
3 In the Report Explorer, select **File** > **New**.
4 In the Properties pane, set **Directory** to `Present working directory`.
5 Save the setup file as `inport_outport.rpt`.

Add Components

Add these components to the report, in order.

From this Library Folder	Add this Component	Set this Property
`Simulink`	Model Loop	N/A
`Formatting`	Chapter	**Title** to `Inport Blocks`
`Simulink`	Block Loop	N/A
`Logical and Flow Control`	Logical If	**Test Expression** to `strcmp(get_param...`

From this Library Folder	Add this Component	Set this Property
		`(RptgenSL.getReportedBlock,'BlockType'),...` `'Inport')`
Simulink	Simulink Property Table	N/A

The report setup file looks like this:

```
Report Generator
   Report - inport_outport.rpt
      ModelLoop - current model
         Chapter - Input Blocks
            BlockLoop Section 1 - All blocks in reported systems of current model
               if (strcmp(get_param(RptgenSL.getReportedBlock, 'BlockType'), 'Inport'))
                  Model Prop Table - %<Name> Infor...
```
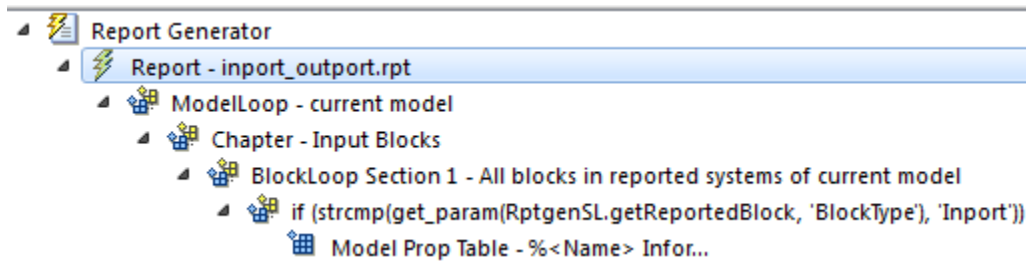
Run the Report

1 Select `inport_outport.rpt`.

2 From the context menu, select **Report**.

The report includes a chapter with properties for the Inport blocks only.

If you wish, create a second chapter that reports on Outport blocks only, as shown below.

```
Report Generator
   Report - inport_outport.rpt
      ModelLoop - current model
         Chapter - Input Blocks
            BlockLoop Section 1 - All blocks in reported systems of current model
               if (strcmp(get_param(RptgenSL.getReportedBlock, 'BlockType'), 'Inport'))
                  Model Prop Table - %<Name> Infor...
         Chapter - Output Blocks
            BlockLoop Section 1 - All blocks in reported systems of current model
               if (strcmp(get_param(RptgenSL.getReportedBlock, 'BlockType'), 'Outport'))
                  Model Prop Table - %<Name> Infor...
```

Loop Context Functions

In this section...
"For Simulink Modeling Elements" on page 4-55
"For Stateflow Modeling Elements" on page 4-55

You can use these loop context functions in similar ways as shown in "Filter with Loop Context Functions" on page 4-53.

For Simulink Modeling Elements

Modeling Element	Looping Component	Function
Simulink modeling elements		
Block	Block Loop	RptgenSL.getReportedBlock
Signal	Signal Loop	RptgenSL.getReportedSignal
System	System Loop	RptgenSL.getReportedSystem
Model	Model Loop	RptgenSL.getReportedModel

For Stateflow Modeling Elements

Modeling Element	Looping Component	Function
Object	Object Loop	RptgenSF.getReportedObject
State	State Loop	RptgenSF.getReportedState
Chart	Chart Loop	RptgenSF.getReportedChart

5

Export Simulink Models to Web Views

Web Views

What Is a Web View?

A Web view is an interactive rendition of a model that you can view in a Web browser. You can use Web views to navigate hierarchically to specific subsystems and see properties of blocks and signals. Web views provide a simple way to interactively explore a model. For example, you can view block parameter values without opening a block parameter dialog box.

Use Web views to share models with people who do not have Simulink installed.

You can save Web views of a model over time, creating snapshots of the model as it changes during the development process.

System Requirements

Although you use Simulink Report Generator software to create Web views. you can display a Web view in a browser, even if you do not have Simulink Report Generator installed.

By default, when you export a Web view, that Web view automatically displays in your default Web browser. Web views require a Web browser that supports SVG natively.

Web View Files

By default, exporting a Web view creates a zip file that includes the Web view HTML file, as well as files that support Web view display. Supporting files include files include `.svg` and `.png` files. Zip file packaging compresses the files and consolidates the Web view and supporting files into one zip file.

You can choose to export the Web view files as the Web view HTML file and the supporting files, in a folder, without being zipped. You can open the Web view HTML file directly, without having to open a non-zipped file. You can also choose to export the Web view files as both a zip file and as non-zipped files.

The default name of the zip file or folder that contains the non-zipped Web view files is the name of the model that contains the systems to export. You can specify a different file or folder name.

The default location for storing Web view files is the MATLAB current folder. You can choose a different folder.

If you send Web view files to someone else, consider whether you need to explain how to access the Web view file.

Related Examples

- "Export Models to Web View Files" on page 5-4
- "Display and Navigate a Web View" on page 5-6
- "Create and Use a Web View" on page 5-14

More About

- "Optional Web Views" on page 5-21

Export Models to Web View Files

In this section...
"Open the Web View Dialog Box" on page 5-4
"Export a Model to a Web View" on page 5-4

Open the Web View Dialog Box

To export a Web view, use the Web View dialog box. The way you access the dialog box differs, depending on whether you are using the Simulink Editor or the Report Explorer. You can also open the dialog box from the command line.

Interface	What You Do
Simulink Editor	Select **File** > **Export Model to** > **Web**.
Report Explorer	Select **Tools** > **Export Simulink to Web**.
Command line	Use the `slwebview` function without arguments.

Export a Model to a Web View

1 Open the model to export.

2 In the Simulink Editor, select **File** > **Export Model to** > **Web**.

3 In **Systems to Export**, select the levels of the model to export, in relationship to the system currently displayed or chart currently selected in the Simulink Editor.

4 For the systems in the levels that you are exporting, in **Include Options**, select any kinds of systems you want the Web viewer user to be able to navigate below the Subsystem or Model block, to the underlying blocks or models.

If you select more than one kind of system, the criteria for exporting information for interacting with the contents of the systems are applied downward through the model hierarchy. For example, if you left **Referenced Models** unchecked when you exported the model, regardless of how you set the **Library Links** option, in the Web view you cannot interact with a library link block that is inside of a referenced model.

5 In the **Systems to Exclude** list of the systems you have selected to export, select any systems that you do not want to export. To select multiple systems, press the **Ctrl** key and select systems.

6 To avoid overwriting existing exported Web view files, select **If file exists, increment name to prevent overwriting**.

7 In **Package Type**, specify whether you want to package the Web view as a zipped file (the default packaging). In **Package name**, you can specify a name for the zip file or for the folder for the Web view files.

8 Click **Export**.

Note: If you use the Web View export option in the Report Explorer Properties pane, then click **Export model**. If you change the visible system or chart while the Report Explorer **Web View** pane is visible, the pane does not automatically change to show information about the newly visible system. To update the **Web View** pane, click **Refresh**.

---

9 If you have Simulink Verification and Validation installed, you can use the **Optional Views** tab to select **Model Coverage** or **Requirements**, or both, to export the associated kind of information.

See Also

Functions
slwebview

Related Examples

- "Display and Navigate a Web View" on page 5-6
- "Create and Use a Web View" on page 5-14

More About

- "Web Views" on page 5-2
- "Web View Files" on page 5-2
- "Optional Web Views" on page 5-21

Display and Navigate a Web View

In this section...

Display a Web View When You Export It

When you export a Web view using the Web View dialog box or from the Report Explorer **Web View** pane, the Web view appears in your system Web browser.
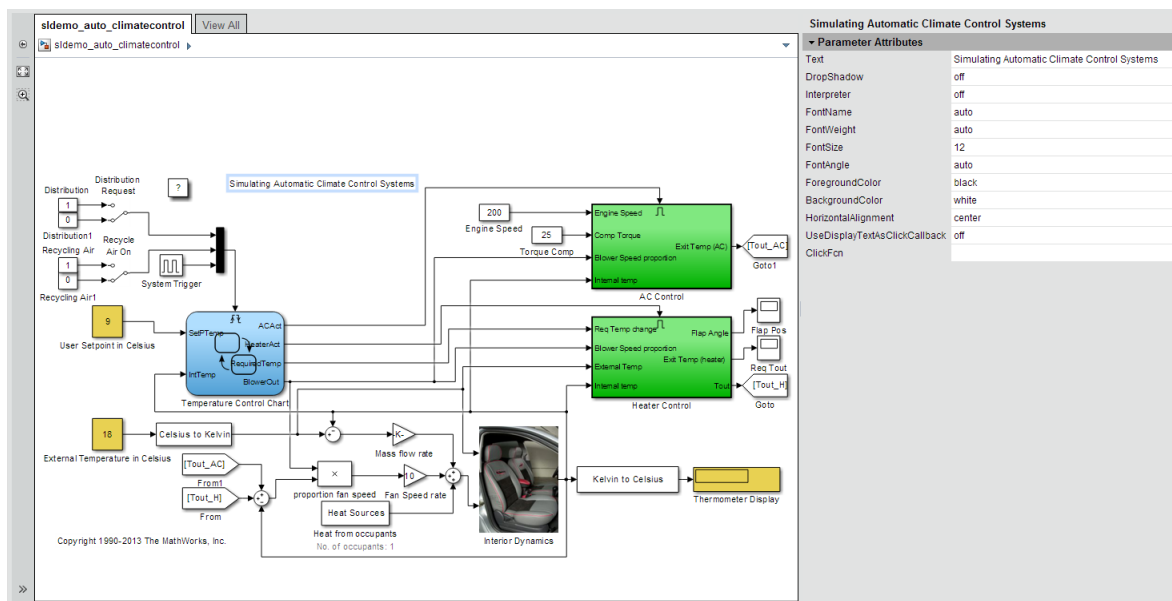
Open a Web View File in a Web Browser

To open a Web view file to display in a Web browser, from the folder that contains the Web view files, select the HTML file.

Open the `webview.html` file to display the Web view. For details about file packaging and location, see "Web View Files" on page 5-2.

To use a Google® Chrome Browser, you need to do some setup.

Open a Web View in a Google Chrome Browser on a Windows Platform

1   If you do not already have a shortcut set up for Google Chrome, click the Windows **Start** button and search for `Chrome`.
2   Right-click and drag the Google Chrome icon to an open area on your desktop.
3   Right-click the icon and select **Create shortcut**.
4   Right-click the shortcut and select **Properties**.
5   In the **Target** edit box, append the following text: `--allow-file-access-from-files`. Be sure to use two hyphens at the beginning. Click **OK**.
6   Close all open Google Chrome browsers.
7   Use the shortcut to open a Google Chrome browser.
8   Open the Web view file.

Open a Web View in a Google Chrome Browser on a Macintosh Platform

1   Run **Terminal**. You can find it using **Spotlight**, in `Applications/Utilities`.
2   Enter the following text:

    `open/Applications Google\Chrome.app --allow-file-access-from-files`

Open a Web View in a Google Chrome Browser on a Linux Platform

1   Run **terminal**.
2   Enter the following text:

    `./chromium-browser --allow-file-access-from-files`

View Contents of a System

To see a thumbnail of the contents of all of systems in the Web view, click the **View All** tab.

To view the contents of a specific system, use one of these approaches:

- In the model viewer, double-click the system.
- In the model browser, select a system. To expose this pane, click **Hide/Show Model Browser** » .
- Click the **View All** tab and click the thumbnail of a system.

To open a system in a separate tab, press **CTRL** and click the system.

View Block Parameters and Signal Properties

Click a block or signal in the model to see its parameters or properties in the **Object Inspector** pane.
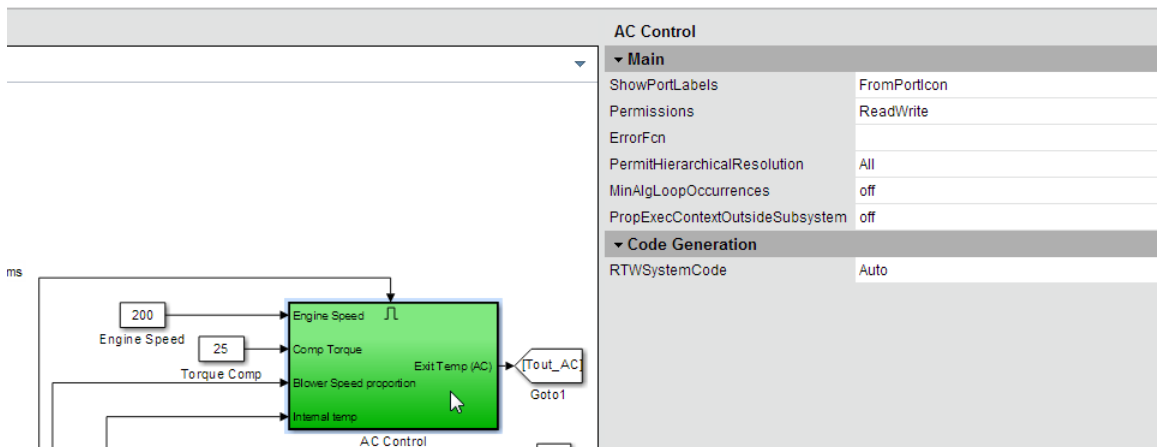
Access Optional Web View Information

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

See Also

Functions
`slwebview`

Related Examples

·   "Create and Use a Web View" on page 5-14

More About

·   "Web Views" on page 5-2
·   "Web View Files" on page 5-2
·   "Optional Web Views" on page 5-21

Search a Web View

In this section...
"Perform a Search" on page 5-11
"Sort Search Results" on page 5-13
"Navigate Between Search Results and Model Elements" on page 5-13

Perform a Search

1   In a Web View, at the top of the displayed tab, click the search button [Q] .

2   In the search box, enter the search term.

Search strings are case-insensitive. The search treats the string as a partial string.

3   To specify search criteria, click the search criteria button and select the types of model element you want to search in.
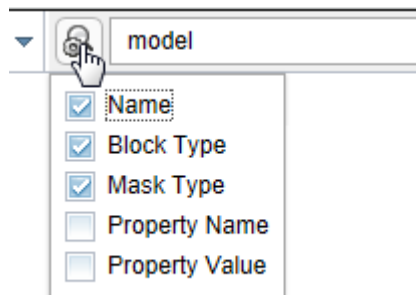
4   Press **Enter**.

The elements of the model that the search returns appear highlighted. The search results include the name and parent for each returned element.

Sort Search Results

You can sort the search results in alphabetical order. In the search results table, click the **Name** or **Parent** column.

Navigate Between Search Results and Model Elements

To see the corresponding search result for a highlighted model element, click the element.

To highlight the model element for a search result, click the search result.

The **Object Inspector** pane to the right of the model updates to reflect the selected model element or search result.

Create and Use a Web View

In this section...
"About This Tutorial" on page 5-14
"Export Specific Systems" on page 5-14
"Navigate the Web View" on page 5-16
"Display Parameters and Properties of Blocks and Signals" on page 5-17
"Open the Web View" on page 5-19

About This Tutorial

This tutorial takes you through the steps to export a Simulink model to a Web view.

You create a Web view from the Simulink model window using the `sldemo_auto_climatecontrol` model, which is provided with the Simulink software. This model simulates the working of an automatic climate control system in a car.

Export Specific Systems

When you create the Web view, you can specify export options.

1 At the MATLAB command prompt, enter `sldemo_auto_climatecontrol` to open the Simulink model.

2 In the Simulink Editor, select **File** > **Export Model to** > **Web**.
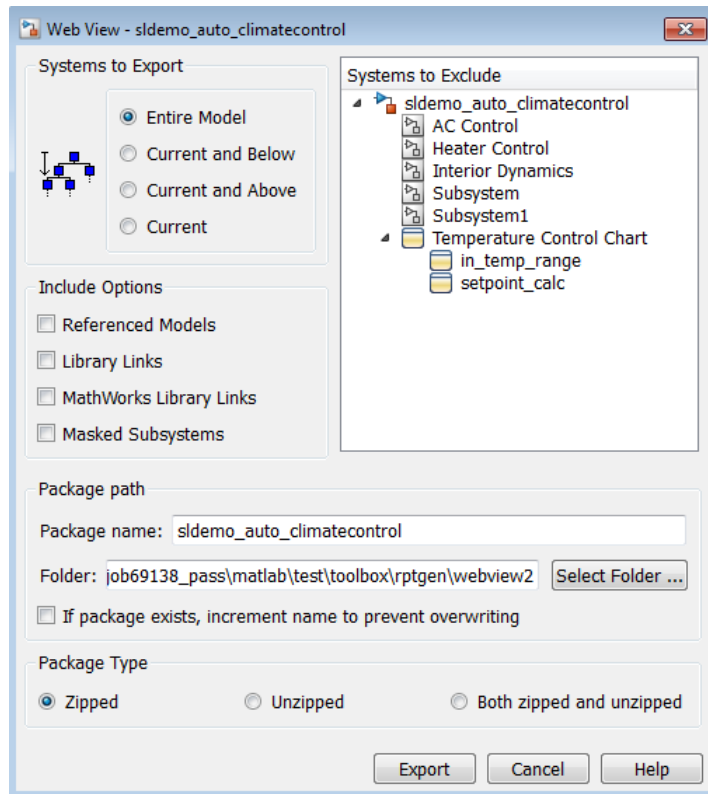
3 In the **Include Options**, select **Masked Subsystems**. This enables users of the Web view to interact with masked blocks.

4 In **Systems to Exclude**, select `Temperature Control Chart`.

5 Use the **Folder** edit box to specify `climate_control_webview` as the name for the zip file for the exported Web view files.

6 Select the **If file exists, increment name to prevent overwriting** check box. Selecting this option prevents overwriting the Web view files if you export multiple Web views from the same model.

7 Click **Export**.

Exporting the selected systems to a Web view creates several support files, as well as an HTML file for displaying the systems. In this example, you change the defaults for the naming of the files.

The Web view files are exported, and the Web view appears in a Web browser.
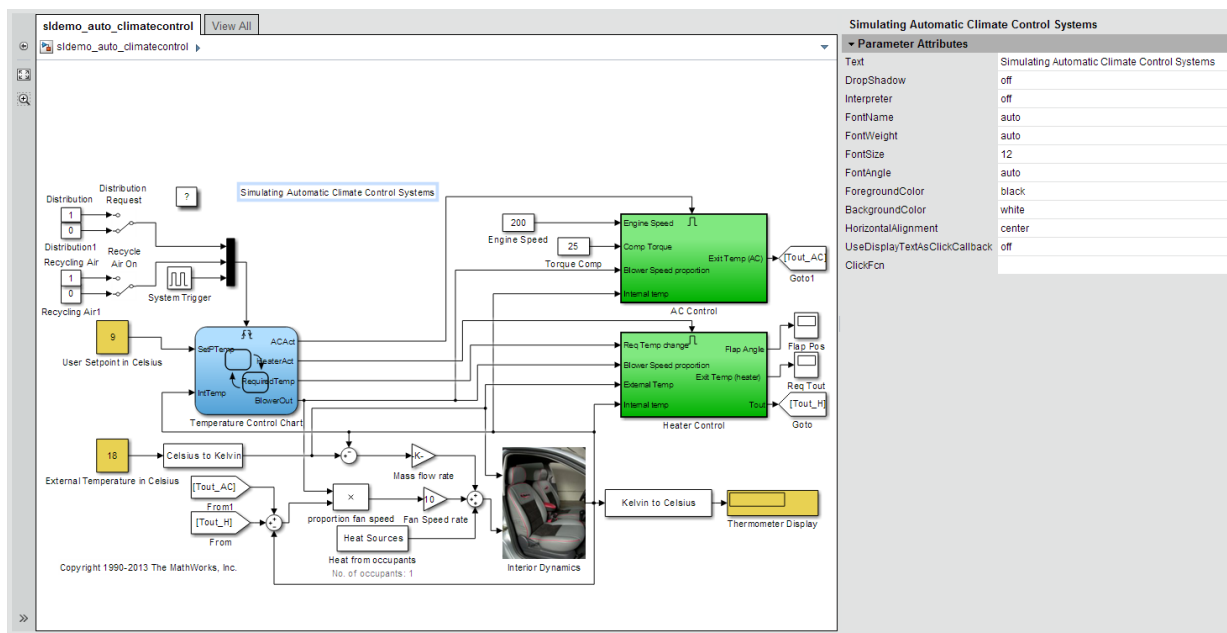
The Temperature Control Chart appears in the top level of the model, but you cannot open that chart in the Web view to see its contents.

Navigate the Web View

By default, if you export a whole model to a Web view, the **Model Viewer** pane shows the whole model. You can display specific systems in the Web view. For example:

1   In the model viewer, double-click the AC Control subsystem.
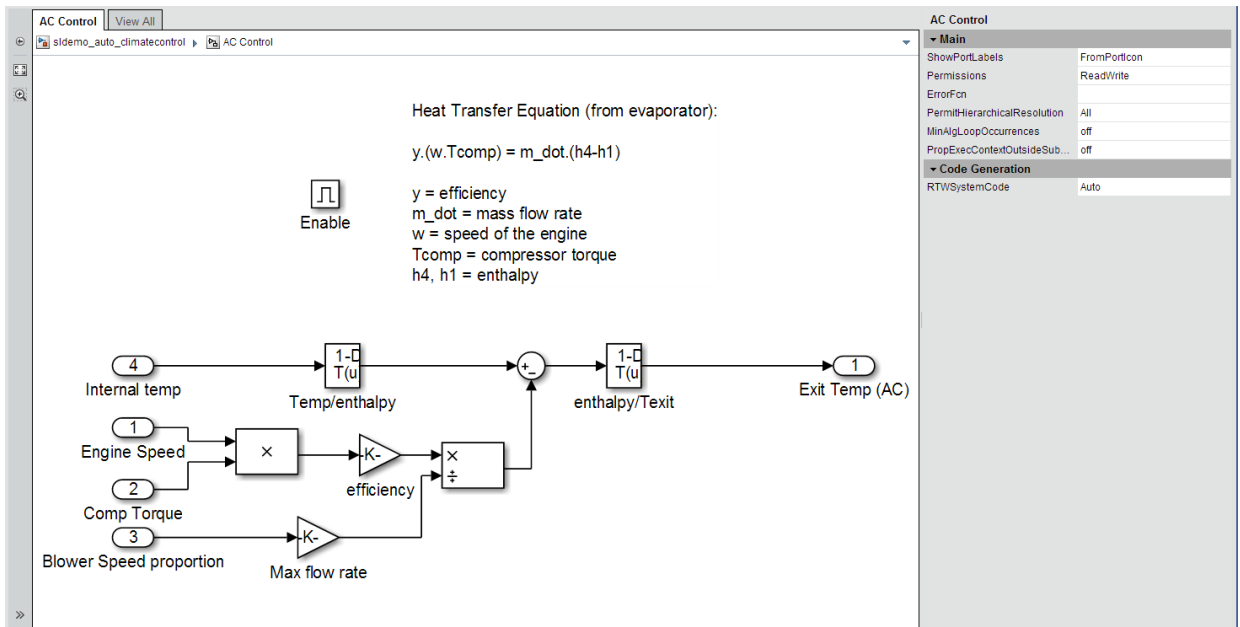
The AC Control subsystem appears in the model viewer. The tab label reflects the name of the currently displayed subsystem.

2 Open the model browser (hidden by default). Click **Hide/Show Model Browser** ».

3 Open another system, in a separate tab. In the model browser, **CTRL+click** select the Heater Control system.

4 Drag the AC Control system to the top of model viewer. Place the cursor in the display area, hold down the mouse scroll wheel, and drag.

5 Zoom the display with the mouse scroll wheel.

Display Parameters and Properties of Blocks and Signals

1 In the model browser, select `sldemo_auto_climatecontrol`.

2 Double-click the AC Control subsystem.

3 Click the Temp/enthalpy block to view the block parameter values. The **Object Inspector** pane groups the block parameters by the block parameter dialog box tabs.

Heat Transfer Equation (from evaporator):

$$y \cdot (w \cdot Tcomp) = m\_dot \cdot (h4-h1)$$

y = efficiency
m_dot = mass flow rate
w = speed of the engine
Tcomp = compressor torque
h4, h1 = enthalpy

4 Click the input signal for the Exit Temp (AC) block to display the signal properties.

Try navigating to other parts of the Web view.

5   Close the Web view.

Open the Web View

In the MATLAB current folder (or wherever you saved the zip file when you performed the steps in "Export Specific Systems" on page 5-14), extract the `climate_control_webview` zip file contents and open the `webview.html` file.

Related Examples

More About

Optional Web Views

Optional Web views provide information about a model in addition to the standard Web view information about blocks and signals in a model.

The Simulink Report Generator includes the following optional Web views that you can capture and view if you have Simulink Verification and Validation installed:

- Requirements
- Model coverage

These optional views display requirements or model coverage information associated with the current Web view.

To include model coverage information in a Web view, set up a coverage report for the model and simulate the model before selecting the model coverage optional view option.

Related Examples

- "Capture and View Optional Web View Information" on page 5-22
- "Export Models to Web View Files" on page 5-4

More About

- "Optional Web Views" on page 5-21

Capture and View Optional Web View Information

In this section...
"Capture Optional Web View Information for a Model" on page 5-22
"View Optional Web View Information" on page 5-22

Capture Optional Web View Information for a Model

When you create a Web view, you can include optional Web view information.

To add the model coverage or requirements optional view information to a Web view for a model, you must have Simulink Verification and Validation installed.

1   In the Web View dialog box, open the **Optional Views** tab.
2   Select each optional view (for example, **Model Coverage** or **Requirements**) that you want to capture the associated information for.

Tip The **Optional Views** tab appears only if you can access an optional Web view.

Alternatively, in the Simulink Editor, click **Analysis** > **Requirements Traceability** > **Generate Web View**.

View Optional Web View Information

To view the model coverage or requirements optional Web view information in a Web view, you must have Simulink Verification and Validation installed.

In a Web view, click a highlighted block (for example, blocks with an orange border have requirements information). The information for that block appears in the **Informer** pane below the model.

Related Examples
·   "Export Models to Web View Files" on page 5-4

More About
·   "Optional Web Views" on page 5-21

Web View

Web View Export Dialog Box Overview

Use Web View dialog box to set export options, including:

- Systems to export (for example, the current system or current system and systems below)
- Whether to export information to support interacting with contents of referenced models, library links, or masked subsystems
- Where to store the Web view files
- How to package the Web view files
- Capture and optional view information

To get help on an option

1   Right-click the option's text label.

2   Select **What's This** from the popup menu.

See Also

"Export Models to Web View Files" on page 5-4

Systems to Export

Select which Simulink systems or Stateflow charts to export.

Note: The systems to export are relative to the system that is currently visible in the Simulink Editor or Stateflow Editor.

Include Systems Option Value	Meaning
`Entire Model` (Default)	Export all Simulink systems or Stateflow charts in the model
`Current and below`	Export the system or Stateflow chart that is displayed in the Simulink Editor or in the Stateflow Editor and all subsystems or subcharts that it contains
`Current and above`	Export the displayed Simulink system or Stateflow chart and all systems or charts that contain it
`Current`	Export only the Simulink system that is displayed in the Simulink Editor or in the Stateflow Editor

Referenced Models

Export models referenced by the systems that you select to export. For example, if you select the **Entire Model** option, any models referenced by a Model block the exported model.

Library Links

Export library blocks linked to from the systems that you select to export. For example, if you select the **Entire Model** option, all library blocks linked to in the exported model.

If you select this option, do not also select the **MathWorks Library Links** option.

MathWorks Library Links

Export MathWorks built-in library blocks linked to systems that you select to export.

For example, if you select the **Entire Model** option, all MathWorks library blocks linked to in the exported model.

If you select this option, along with the **Masked Subsystems** option, right-clicking a built-in library block in a Web view displays the parameter values for that block.

If you select this option, do not also select the **Library Links** option.

Masked Subsystems

Allow Web view interaction with masked subsystem contents in the systems that you select to export. Select this option to view the contents of the masked subsystem and open masked subsystem dialog boxes.

Package name

The default name is the name of the model that contains the systems to export.

If you use the default **Package Type** setting of `Zipped`, the **Package name** field specifies the name of the zip file.

If you set **Package Type** to `Unzipped`, the **Package name** field specifies the name of the folder.

If the zip file or folder with the specified name in the specified folder already exists, Simulink Report Generator overwrites the existing zip file or folder contents. To avoid overwriting an existing Web View files, consider selecting the **If file exists, increment name to prevent overwriting** option.

Folder

The folder in which to store the exported Web view files. The default folder is the MATLAB current folder.

Specify a full path to the folder or click **Select Folder** to navigate to the folder.

If package exists, increment name to prevent overwriting

Increment the package name by appending a system-generated number to the end of the zip file or folder name.

For example, suppose you use the following settings:

- For **Package Type**, you use the default setting of `Zipped`.
- For **Package name**, you use `vdp_web_view`.

When you export the Web view the first time, the resulting zip file is called `vdp_web_view`. If you export the Web view again, Simulink Report Generator creates a `vdp_web_view1.zip` file, and preserves the original `vdp_web_view` file.

Package Type

Specify the file packaging to use when exporting the Web view.

- **Zipped** — (Default) Export as a zipped file that includes the Web view file, along with files that support the display of the Web view. Zip file packaging compresses the files and consolidates the Web view and supporting files into one file.
- **Unzipped** — Export as the Web view file and supporting files, without being zipped. You can open an unzipped Web view file directly.
- **Both zipped and unzipped**— Export as both a zipped file and an unzipped file

Include Model Coverage view

This option only appears if you have Simulink Verification and Validation software installed.

Provides additional information about the model, based on the Model Coverage report.

Include Embedded Coder view

This option only appears if both these conditions exist:

- You have the Embedded Coder® software installed.
- There is generated code for the systems that you export.

This view provides additional information about the generated code for a block in the Web view.

Click a block in the Web view. The generated code for that block appears in the Web Informer pane below the displayed model.

Include Requirements view

Captures model requirements information. For details, see "Capture and View Optional Web View Information" on page 5-22.

This option only appears if you have Simulink Verification and Validation software installed.

Include Coverage view

Captures model coverage information based on the Model Coverage report. For details, see "Capture and View Optional Web View Information" on page 5-22.

This option only appears if you have Simulink Verification and Validation software installed. You must set up a coverage report for the model and simulate the model before you can use this option.

6

Compare Simulink Model XML Files

About Simulink Model XML Comparison

In this section...
"Creating XML Comparison Reports" on page 6-2
"Using XML Comparison Reports" on page 6-2

Creating XML Comparison Reports

If you have Simulink Report Generator software, you can compare XML text files from Simulink models.

You can select a pair of Simulink models to compare their XML files. You can use models from any version of Simulink. The XML comparison tool produces a comparison report based on the SLX files. You can use the report to explore the differences, view the changes highlighted in the original models, and merge differences.

You can access the XML comparison tool from:

- The MATLAB Current Folder browser context menu
- The MATLAB Comparison Tool
- The MATLAB command line
- The Simulink Editor **Analysis** menu
- The Simulink Project Modified Files view

The Simulink Report Generator XML comparison functionality is an extension of the MATLAB Report Generator XML comparison feature.

You can use the XML comparison tool with both model file formats, SLX and MDL. If the selected files are `.mdl` files, the XML comparison tool first exports the `.mdl` files to SLX files in a temporary directory. The XML comparison tool then produces a comparison report based on the SLX files.

For more information on creating reports, see "Select Simulink Models for XML Comparison" on page 6-5.

Using XML Comparison Reports

You can display XML comparison reports in the MATLAB Comparison Tool. The comparison tool processes the output of the XML comparison into an interactive report

with links that you can click to reverse annotate from the XML tag comparison to the corresponding Simulink models. "Reverse annotation" means when you click items in the report, Simulink Report Generator displays the corresponding items highlighted in the original models, as shown in the following example.

The XML comparison report shows a hierarchical view of the portions of the two XML files that differ. The report does not show sections of the files that are identical.

If the files are identical you see a message reporting there are no differences.

If files have not been saved, you see an error message informing you that you must save modified or newly created models before running an XML comparison.

Note: It might not be possible for the analysis to detect matches between previously corresponding sections of files that have diverged too much.

Change detection in the Chawathe analysis is based on a scoring algorithm. Items match if their Chawathe score is above a threshold. The Simulink Report Generator implementation of Chawathe's algorithm uses a comparison pattern that defines the thresholds assigned to particular node types (e.g., "block"). For more information, see "How the Matching Algorithm Works" in the MATLAB Report Generator documentation.

For more information on using the report, see "Compare Simulink Model XML Files" on page 6-8.

To control reverse annotation, see "Display Items in Original Models" on page 6-17.

To merge differences, see "Merge Simulink Models from the Comparison Report" on page 6-21.

For more information about the Comparison Tool, see "Comparing Files and Folders" in the MATLAB documentation.

Related Examples

- "Select Simulink Models for XML Comparison" on page 6-5
- "Compare Simulink Model XML Files" on page 6-8
- "Display Items in Original Models" on page 6-17
- "Merge Simulink Models from the Comparison Report" on page 6-21
- "Compare Project or Model Templates" on page 6-41

Select Simulink Models for XML Comparison

In this section...
"Select Files from the Simulink Editor" on page 6-5
"Select Files from the Current Folder Browser" on page 6-5
"Select Files from a Simulink Project" on page 6-6
"Select Files from the Comparison Tool" on page 6-6
"Select Files from the Command Line" on page 6-6
"Choose a Comparison Type" on page 6-7
"Examples of XML Comparison" on page 6-7

To learn what you can do with XML comparison reports, see "About Simulink Model XML Comparison" on page 6-2.

Select Files from the Simulink Editor

To compare files using the Simulink Editor:

1 Select **Analysis** > **Compare Simulink XML Files**.

   The Select Files or Folders for Comparison dialog box opens.

2 If the Editor currently displays a model, the current model name and path appear automatically selected in the **First file or folder** edit box. Use the browse buttons to locate and select files for the first and second model files.

3 When you click **Compare**, the XML comparison tool performs the analysis, and displays the resulting report in the Comparison Tool.

Select Files from the Current Folder Browser

To compare two files from the Current Folder browser:

· For two files in the same folder, select the files, right-click and select **Compare Selected Files/Folders**.

· To compare files in different folders:

   1 Select a file, right-click and select **Compare Against**

2   Select the second file to compare in the Select Files or Folders for Comparison dialog box.

3   Leave the default **Comparison type**, `Simulink XML text comparison`.

4   Click **Compare**.

If the selected files are XML or model files, the XML text comparison tool performs a Chawathe analysis and displays a report in the Comparison Tool.

For more information about comparisons of other file types (e.g., text, MAT, or binary) with the Comparison Tool, see "Comparing Files and Folders" in the MATLAB documentation.

Select Files from a Simulink Project

If you have a Simulink Project using source control, you can create an XML comparison report from the Modified Files view of the Simulink Project Tool. For details, see "Project Management".

Select Files from the Comparison Tool

To compare files using the Comparison Tool, from the MATLAB Toolstrip, in the **File** section, select the **Compare** button. In the dialog box select files to compare.

If the selected files are XML or model files, the XML text comparison tool performs a Chawathe analysis and displays a report in the Comparison Tool.

Select Files from the Command Line

To compare XML files from the command line, enter

```
visdiff(filename1, filename2)
```

where `filename1` and `filename1` are XML files or Simulink models.

If the files are models, the XML comparison tool performs the comparison on the XML files. This XML comparison functionality is an extension to the MATLAB `visdiff` function. `visdiff` produces a report in the Comparison Tool.

To create an `xmlcomp.Edits` object at the command line without opening the Comparison Tool, enter:

```
Edits = slxmlcomp.compare(modelname_A,modelname_B)
```
See "Export Results to the Workspace" on page 6-30 for information about the `xmlcomp.Edits` object.

Choose a Comparison Type

If you specify two XML or model files to compare using either the Current Folder Browser or the `visdiff` function, then the Comparison Tool automatically performs the default comparison type. The defaults are `XML text comparison` for XML files and `Simulink XML text comparison` for model files.

To change comparison type, either create a new comparison from the Comparison Tool, or use the **Compare Against** option from the Current Folder browser. You can change comparison type in the Select Files or Folders for Comparison dialog box. For example, if you want the MATLAB text differences report for XML or model files, change the comparison type to `Text comparison` in the dialog before clicking **Compare**.

Examples of XML Comparison

For examples with instructions, see:

- `slxml_radar_matlab_function`
- `slxml_sfcar`
- `slxml_truthtables`

For information on using and understanding the report and the XML comparison functionality, refer to "Compare Simulink Model XML Files" on page 6-8.

Compare Simulink Model XML Files

In this section...

Navigate the Simulink XML Comparison Report

You can select a pair of Simulink models to compare their XML files. You can use models from any version of Simulink. The XML comparison tool produces a comparison report based on the SLX files. You can use the report to explore the differences, view the changes highlighted in the original models, and merge differences.

The XML Comparison report shows changes only, not the entire XML text file contents. The report shows a hierarchical view of the portions of the two XML files that differ. The report does not show sections of the files that are identical. To learn about the report, see "About Simulink Model XML Comparison" on page 6-2.
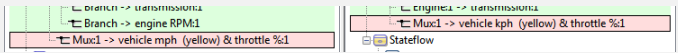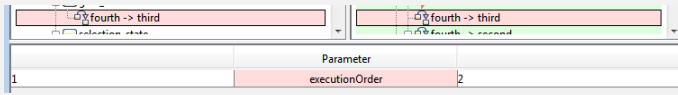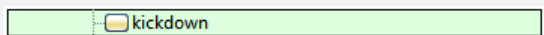
To *step through differences*, use the **Comparison** tab on the toolstrip. To move to the next or previous group of differences, on the **Comparison** tab, in the **Navigate** section, click the arrow buttons to go to the previous or next difference. See "Step Through Changes" on page 6-10.

You can also click to select items in the hierarchical trees and observe the following display features:

- Selected items appear highlighted in a box.
- If the selected item is part of a matched pair it is highlighted in a box in both left and right trees.

- When you select an item, the original model displays and the corresponding item is highlighted. See "Explore Changes in the Original Models" on page 6-11.

Report item highlighting indicates the nature of each difference as follows:

Type of report item	Highlighting	Notes
Modified	Pink	Modified items are matched pairs that differ between the two files. When you select a modified item it is highlighted in a box in both trees. Example of a modified pair of nodes: Changed parameters for the selected pair are displayed in a separate **Parameters** panel for review. If strings are too long to display in the **Parameters** table, right-click and select **Compare as Text** to open a new comparison of the parameters. Example of modified parameters:
Unmatched	Green	When you select an unmatched item it is highlighted in a box in one tree only. Example of an unmatched node:
Container	None	Rows with no highlighting indicate a container item that contains other modified or unmatched items. Example of a container node:

Icons indicate the category of item, for example: model, subsystem, Stateflow machine or chart, block, line, parameter, etc.

To expand or filter the tree view, use the **View** tab controls on the toolstrip for the following functions:

• **Expand All** — Expands every item in the tree.

Tip Right-click to expand or collapse the hierarchy within the selected tree node.

• **Collapse All** — Collapses all items in the tree to the most compact view possible.

• **Filter** — Opens the Filter list. Select check boxes to enable or disable display of categories of changes in the report. Use the filters to show only the changes you are interested in. By default the report hides all nonfunctional changes, such as repositioning of items. Turn off filters to explore *all* differences including nonfunctional changes. See "Filter Out Differences" on page 6-13.

If you want to swap the files, on the **Comparison** tab, select **Swap Sides**. The report swaps the sides and reruns the comparison. **Refresh** also runs the analysis again.

To create a new report, see "Select Simulink Models for XML Comparison" on page 6-5.

For examples with instructions, see also "Examples of XML Comparison" on page 6-7 .

Step Through Changes

On the **Comparison** tab, in the **Navigate** section, when you click the **Next** arrow button (or press the Down key when the report has focus), you step through groups of changes in the report, in the following order:

1 The first time you click **Next**, it selects the first changed (pink) or inserted (green) node on the left tree.

2 Step through the differences with the **Next** button.

   • When selected items have a match in the right tree then they are also highlighted.

   • Next skips white nodes with no color background. White nodes are parts of the hierarchy that contain no differences.

   • If there is an insertion or deletion with child nodes, **Next** skips the child nodes if they are all also insertions or deletions. For example, if you insert a subsystem, **Next** selects the top subsystem node, then skips all the nodes inside the subsystem (if they are all also insertions) and selects the next difference.

- **Next** minimizes context switching when highlighting in models. When you click **Next**, the report steps through all differences at the same level of the model, subsystem, or chart, in both left and right trees in the report, before moving to the next level of the report. For example, you step through all differences in a subsystem in the left and right trees, before moving to another subsystem.

3 When you have stepped through all changes, **Next** returns to the beginning of the left tree.

If you click an item in the report, the **Next/Previous** controls will step through changes from the point you selected.

Explore Changes in the Original Models

When you compare the XML text files from Simulink® models, you can choose to display the corresponding items in the original models when you select report items. You can use this reverse annotation function to explore the changes in the original models. When you select an item, the report invokes reverse annotation to the original model and highlights the corresponding item in the model.

Control the display by using the **View** tab **Highlight in Models** button and the **Always Highlight** check box.

Tip Click a Subsystem contents node to see the report highlight all visible modified objects in the subsystem.

For details, see "Display Items in Original Models" on page 6-17.

Merge Differences

Tip You can only merge from left to right. If you want to merge into the other file, use **Swap Sides** before you start merging. Swap Sides reverts any merges already made and creates a new comparison report for the original files.

To merge a selection, use the following buttons on the **Comparison** tab, in the **Merge** section:

- **Merge Node** — Merge the selected node from the left side of the report to the right.
- **Merge Parameter** — Merge the selected parameter from the left side of the report to the right.
- **Undo All** — Revert all merge operations.

For more information, see "Merge Simulink Models from the Comparison Report" on page 6-21.

Open Child Comparison Reports for Selected Nodes

If additional comparisons are available for particular nodes, you see a **Compare** button to open a report for that pair of nodes. For example, if there are differences in the Model Workspace, you can click **Compare** to open a new report to explore differences in variables.
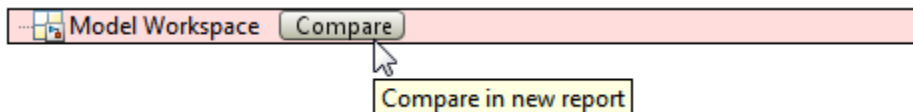
You can open child reports for parameters, MATLAB Function blocks, truth tables and Model Workspaces.

- To compare parameters, click the Parameters pane, then on the **Comparison** tab select **Compare Selected Parameter**. This opens a new report for the currently selected pair of parameters. Use this when the report cannot display all the details in the Parameters pane, e.g., long strings or a script.
- If the original models contain MATLAB Function block components, and if differences are found, the XML comparison tool lists them in the Stateflow section of the report. Click the **Compare** button at the end of the MATLAB Function block report items to open new comparisons in the Comparison Tool, showing the text difference reports for the MATLAB Function block components. You can merge differences in MATLAB Function block code from the text comparison report. See "Merge Simulink Models from the Comparison Report" on page 6-21, and the example `slxml_radar_matlab_function`.
- If the original models contain truth tables, and if differences are found, the XML comparison tool lists them in the Stateflow section of the report.

  - Click the **Compare** button at the end of the MATLAB Function node to see a summary of all changes.

- Click the `truthtable` node to reverse annotate and display both truthtable editors.
- Click the **Compare** button at the end of the `Condition Table` node to open a new text comparison showing only Condition differences.
- Similarly click the **Compare** button for `Action Table` to view only Action changes.

See the example `slxml_truthtables`.

Understand the Report Hierarchy and Matching

Hierarchical node tags (such as subsystem tags in the `.xml` file) appear twice in the tree as nested nodes. This is because the container node and the contents can have separate differences in their properties. This feature of the XML report allows you to distinguish between property differences of the node itself, and differences contained within nodes nested inside.

To understand matching results within an XML comparison report, see "How the Matching Algorithm Works" in the MATLAB Report Generator documentation.

Note: It might not be possible for the analysis to detect matches between previously corresponding sections of files that have diverged too much.

If you cannot see changes you expected to see in the report, on the **View** tab, click the **Filter** button to turn off filters and see *all* identified changes. See "Filter Out Differences" on page 6-13.

Comparing Signal Builder Blocks

You can see differences in Signal Builder blocks when you have not touched them. This occurs because Signal Builder blocks generate new handles when you load a model. The report shows that the handles have changed. To view the differences, click the **Compare** button in the report and observe that handles called `blockH` and `modelH` are different.

Filter Out Differences

You can use the **Filter** button on the **View** tab to control display of categories of changes. Turn off filtering to view *all* identified changes.

In the **Filter** list, select check boxes to enable or disable display of categories of changes in the report. Use the filters to show only the changes you are interested in. By default the report hides all nonfunctional changes, such as repositioning of items. Turn off filters to explore all differences including nonfunctional changes. Try this if you cannot see changes you expected to see in the report.

Categories for filtering include:

• **Hide changes in lines**. Hide all changes to signal lines including functional changes.

• **Hide nonfunctional changes**. The report processing identifies certain items in the XML file as nonfunctional, for example, tags representing parameters such as block, system, chart or label positions, font and color settings for blocks and lines, and system print and display settings. The report processing tries to identify "consequential" changes as nonfunctional (that is, changes as a consequence of another change). For example, if a block name changes from `block_A` to `block_B`, a line emerging from that block has a change in its source block parameter. This change in the line parameters is considered nonfunctional. Lines are highly functional, but line changes can be very noisy because of changes in blocks they connect to.

• **Hide changes in graphical interface**. This information is a summary of inports and outports at the top level of the model. Filter graphical interface changes to avoid duplication in the report, as any changes in root ports are also reported as functional changes where you can use reverse annotation.

• **Hide changes in block parameter defaults**. Hiding changes in defaults can avoid duplication in the report, as any changes in blocks are also reported as functional changes where you can use reverse annotation. Block parameter defaults are an undocumented part of the Simulink XML file that store the default parameters for the blocks used in a model.

Exceptions

The report does *not* filter out changes to Block and System names, annotations, and Stateflow Notes as nonfunctional, even though changes to these items do not affect the outcome of simulation. The report always displays these changes to facilitate review of code changes, because they can contain important information about users' intentions.

In certain rare cases the report filters out changes that can impact the behavior of the design. By default moves are filtered as nonfunctional, but in the following cases moves can change design behavior:

• Moving blocks can in some cases change the execution order.

- In a Stateflow chart, if you move states or junctions so that they intersect, the model fails to simulate.

To view these types of changes in the report, turn off the filter for nonfunctional changes.

Change Color Preferences

You can change and save your diff color preferences for the Comparison tool. You can apply your color preferences to all comparison types.

1 On the MATLAB Home tab, click **Preferences**.

2 In the Preferences dialog box, under **MATLAB**, click **Comparison**.

3 Edit color settings as desired for differences and merges. View the colors in the **Sample** pane.

   The **Active Settings** list displays **Default (modified)**.

4 To use your modified settings in the comparison, click **Apply** and refresh the comparison report.

5 To return to the default color settings, in the Preferences dialog box, click **Reset** and click **Apply**. Refresh the comparison report.

6 If you want to save your modified color preferences for use in future MATLAB sessions, click **Save As**. Enter a name for your color settings profile and click **OK**.

   After saving settings, you can select them in the **Active Settings** list.

Save Comparison Results

To save your comparison results, use these **Comparison** tab buttons:

- **Save As** > **HTML**, **Word**, or **Basic HTML** — Open the Save dialog box, where you can choose to save a printable version of the XML comparison report. See "Save Printable HTML Report" on page 6-30.

- **Save As** > **Workspace variable** — Export XML comparison results to workspace. See "Export Results to the Workspace" on page 6-30.

Related Examples

- "Select Simulink Models for XML Comparison" on page 6-5

- "Display Items in Original Models" on page 6-17
- "Merge Simulink Models from the Comparison Report" on page 6-21
- "Compare XML from Models Managed with Subversion" on page 6-37
- "Compare Revisions"
- "Source Control in Simulink Project"

More About

- "About Simulink Model XML Comparison" on page 6-2
- "Comparing XML Files from Models with Identical Names" on page 6-33
- "Work with Referenced Models and Library Links" on page 6-35

Display Items in Original Models

In this section...
"Highlighting in Models" on page 6-17
"Control Highlighting in Models" on page 6-19
"View Changes in Model Configuration Parameters" on page 6-20

Highlighting in Models

When you compare the XML text files from Simulink models, you can choose to display the corresponding items in the original models when you select report items. You can use this *reverse annotation* function to explore the changes in the original models. When you select an item, the report invokes reverse annotation to the original model and highlights the corresponding item in the model.

Tip If you click a Subsystem contents node, the report highlights all visible modified objects in the subsystem.

Click a report entry to view the highlighted item (or its parent) in the model:

- If the item occurs in both models, they both appear with highlighting.
- If there is no match for the item, the unmatched report item row is green. It is considered unique and appears highlighted by itself.

  When there is no match in the other model, the report highlights the first matched ancestor to show the context of the missing item. If this ancestor is a system or subsystem, then the report highlights all visible modified objects in the system.

- If the XML comparison tool cannot highlight an item directly (e.g., configuration parameters), then it highlights the nearest ancestor of the selected node.

The following screenshots show reverse annotation of Simulink and Stateflow items in original models using the example slxml_sfcar.

Control Highlighting in Models

To control highlighting in models, on the **View** tab in the Comparison Tool, select or clear the check box **Always Highlight**. You can click the **Highlight in Models** button to highlight the currently selected report node at any time. This can be useful if you turn off automatic highlighting and only want to display specific nodes.

By default, models display to the right of the comparison report, with the model corresponding to the left side of the report on top, and the right below. If you move or resize the models your position settings are respected by subsequent model highlighting operations within the same session. The tool remembers your window positions.

If you want to preserve window positions across sessions, position the window, and then enter:

```
slxmlcomp.storeWindowPositions
```

This preserves the placement of any Simulink windows, Stateflow windows, and truth table windows.

To stop storing window positions and return to the defaults, enter:

```
slxmlcomp.clearWindowPositions
```

View Changes in Model Configuration Parameters

You can use the report to explore differences in the model Configuration Parameters. If you select a Configuration Parameter item, the report displays the appropriate root node pane, if possible, of both Configuration Parameters dialog boxes.

The Parameters pane of the report displays the label text from the dialog controls (or the parameter name if it is command line only), and the parameter values. Some configuration parameters have a different hierarchy in the XML file and the dialog box. You can right-click to merge a selected parameter value in the Parameters pane.

Related Examples

- "Select Simulink Models for XML Comparison" on page 6-5
- "Compare Simulink Model XML Files" on page 6-8
- "Merge Simulink Models from the Comparison Report" on page 6-21
- "Compare Revisions"

More About

- "About Simulink Model XML Comparison" on page 6-2

Merge Simulink Models from the Comparison Report

In this section...
"Resolve Conflicts Using Three-Way Model Merge" on page 6-21
"Use Three-Way Merge with External Source Control Tools" on page 6-26
"Open Three-Way Merge Without Using Source Control" on page 6-26
"Two-Way Model Merge" on page 6-26
"Merge MATLAB Function Block Code" on page 6-28

Merge tools enable you to:

- Resolve conflicts in model files under source control using three-way merge. Open by selecting **View Conflicts**.
- Merge any two model files using two-way merge. Open by selecting **Compare** context menu items.
- Merge MATLAB Function block code using text comparison reports.

Resolve Conflicts Using Three-Way Model Merge

If you have a conflicted model file under source control in Simulink Project or in the Current Folder browser, right-click and select **View Conflicts**. You can resolve the conflicts in the Three-Way Model Merge tool. 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.

1 To try an example three-way merge, enter `slxml_three_way_merge`.
2 In the Simulink Project locate the conflicted model file, right-click and select **View Conflicts**. You can only see **View Conflicts** in the context menu if your file is marked conflicted by the source control.
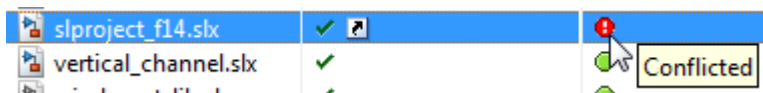
The Merge tool automatically resolves every difference that it can, and shows the results in the **Target** pane. Review the automerge choices, edit if desired, and decide how to resolve any remaining conflicts.

1   Examine the Merge report columns.

  •   At the top, **Theirs**, **Base**, and **Mine** columns show the differences in the conflicting revision, your revision, and the base ancestor of both files.

  •   Underneath, the **Target** shows the local file in your sandbox that you will merge changes into. The Merge tool already automerged the differences it can merge.
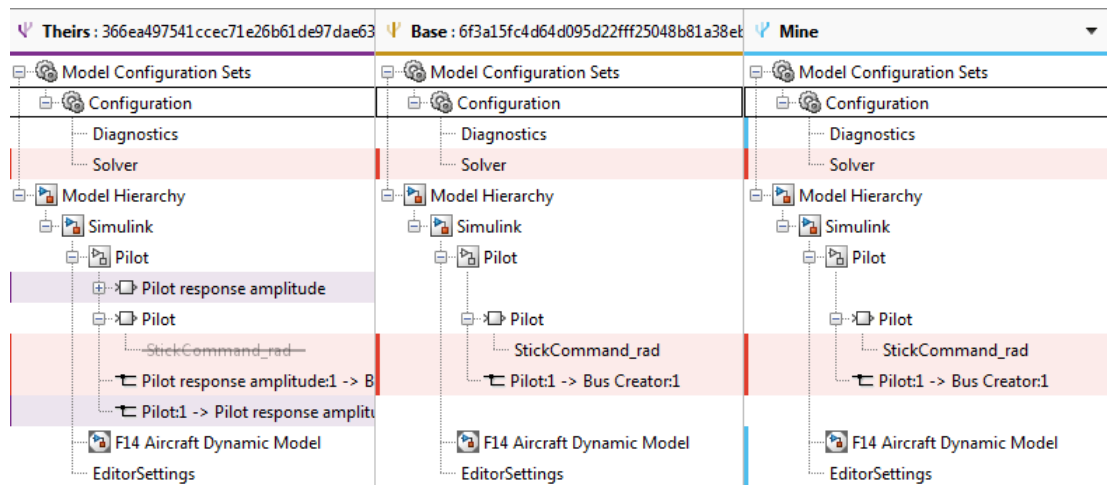
2   Examine a difference by clicking **Next** or by clicking a row in the **Theirs**, **Base**, and **Mine** columns.

The merge tool displays two models (or if you selected a configuration setting, you see two model Configuration Parameters dialog boxes). By default, you see **Theirs** and **Target** models.
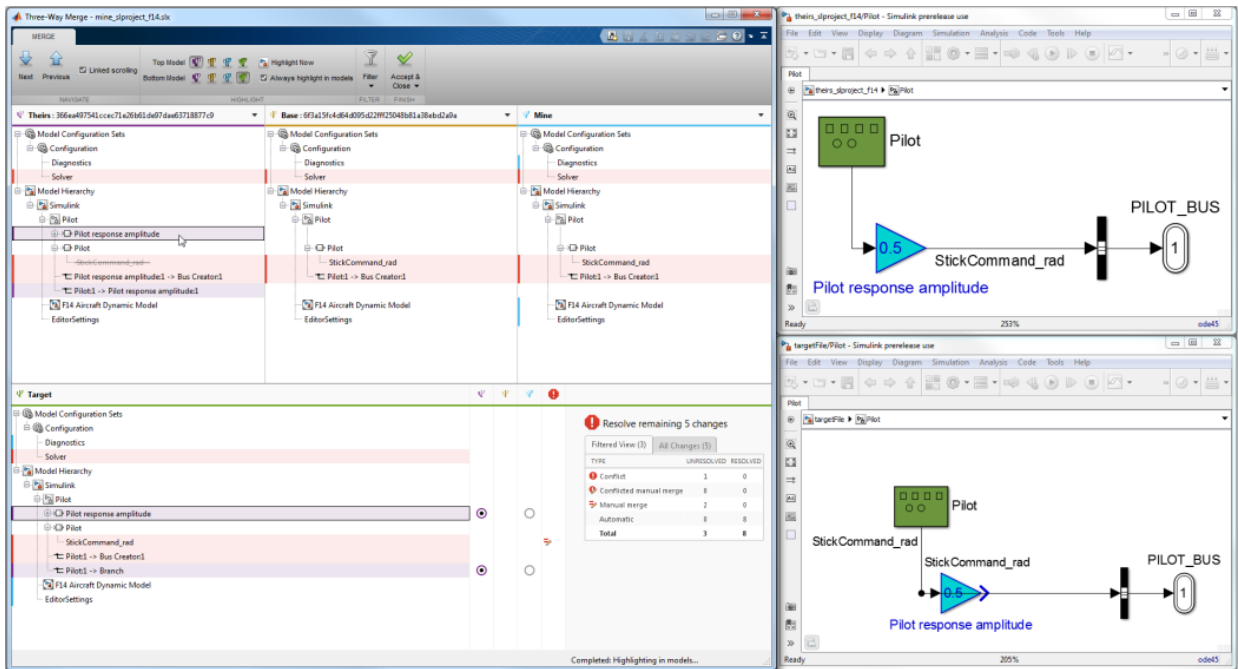
3. Choose the models to display with the toolstrip buttons on the **Merge** tab: **Top Model** or **Bottom Model**. View the models to help you decide what to merge.
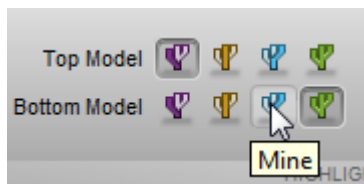
Note: If you open the merge tool using **View Conflicts**, then the models **Theirs**, **Base**, and **Mine** are temporary files showing the conflicting revisions. Examine them to decide how to merge. The **Target** model is a copy of **Mine** containing the results of your merges in the report.

4. Select a version to keep for each change by clicking the buttons in the **Target** pane. You can merge modified, added, or deleted nodes, and you can merge individual

parameters. The Merge tool selects a choice for every difference it could resolve automatically. Review the selections and change them if you want.
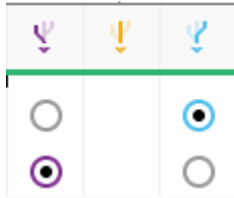
Look for warnings in the Conflicts column. Select a button to use **Theirs**, **Base**, or **Mine** for each conflicted item.
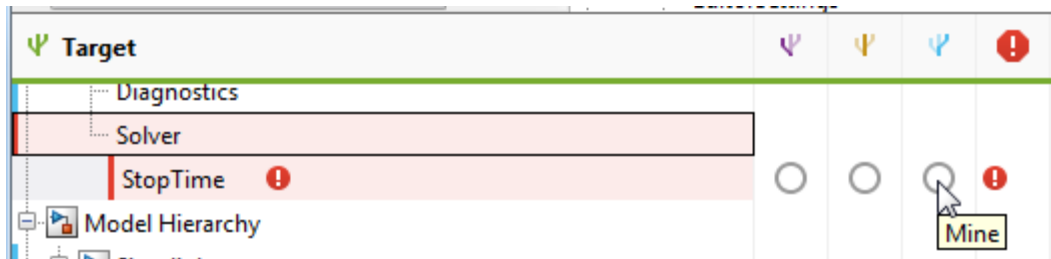
Tip Merge blocks before lines, and merge states and junctions before merging transitions. Merge tool then attempts to connect all lines to blocks for you.

5   Some differences you must merge manually. In the **Target** pane, look for the manual merge icon in the Conflicts column that shows you must take action.
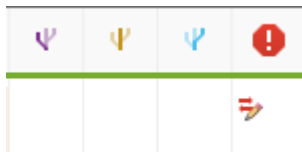
Make manual changes in the Editor. The comparison report cannot update to show any changes that you make in the Editor, so try to make manual changes after addressing all the simpler merges in the report.

After you have resolved the conflict using the Editor, in the **Target** pane, select the check option to mark the node as complete.

6 Examine the summary table to see the number of automatic merges and remaining conflicts you need to resolve.

Check for changes that are filtered out of the current view by looking at the summary table tab titles. The Filtered View and All Changes tab titles show the number of changes. By default the report hides all nonfunctional changes. Turn off active filters to view all identified changes.

7 When you are happy with your merge selections and any manual merges in the **Target** file, click **Accept and Close**. This action saves the target file with all your merges and marks the conflicted file resolved in the source control tool.

To save and not mark the conflict resolved, select **Accept and Close** > **Save and Close**.

To learn more about resolving conflicts in a change list of modified files in a Simulink project, see "Resolve Conflicts".

Use Three-Way Merge with External Source Control Tools

If you are using source control outside of MATLAB, then you can customize external source control tools to open Three-Way Merge (or two-way merge for diffs).

For instructions, see "Customize External Source Control to Use MATLAB for Diff and Merge".

Open Three-Way Merge Without Using Source Control

If you are not using source control or you want to choose three files to merge, then you can open Three-Way Merge using the function `slxmlcomp.slMerge`. Specify the files to merge, for example:

```
slxmlcomp.slMerge(baseFile, mineFile, theirsFile, targetFile);
```
Three-Way Merge opens, where you can merge the changes in `baseFile`, `mineFile`, and `theirsFile` into the `targetFile`.

Two-Way Model Merge

You can merge two Simulink models from an XML text comparison report. The **Compare** context menu items open a two-way model merge. If you are using source control and want to resolve conflicts using a three-way model merge instead, see "Resolve Conflicts Using Three-Way Model Merge" on page 6-21.

The merge feature enables you to merge two versions of a design modeled in Simulink. You can merge individual parameters, blocks, or entire subsystems.

Tip With the two-way merge, you can only merge from left to right. If you want to merge into the other file, use Swap Sides before you start merging. Swap Sides reverts any merges already made and creates a new comparison report for the original files.
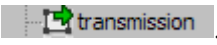
You can merge from the left model to the right model using the XML text files. You can click Undo to revert all merge operations. You can merge modified, added, or deleted nodes in the report as follows:

1   Select a report item you want to merge.
2   On the **Comparison** tab, in the **Merge** section, click **Merge Node** to merge the selected node. Merge is disabled if you cannot merge the selected node. For

example, you cannot merge the top-level model nodes, data nodes, or nodes within configuration settings.

Tip Merge blocks before lines, and merge states and junctions before merging transitions. See "Merging Tips" on page 6-27.

---
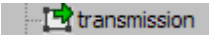
3 View the results in the report and the models.

The report merges the selected node from the left side of the report to the right. Merged report nodes have gray row highlighting, and a green merge arrow if the node has an icon, as shown on this example node: .

The merge copies the change (a modified, added, or deleted item) from the left model to the right model. If the node exists only in the left tree, then the merge inserts it into the right tree. The software attempts to connect all lines to blocks after the merge.

4 To merge individual parameters, right-click an item in the **Parameters** pane and select **Merge Left to Right**. Alternatively, click the **Merge Parameter** button in the **Merge** section of the toolstrip.

You cannot insert or delete parameters, and not all parameters can be merged.

If you merge all possible parameters for a node, then the report marks that node as merged, as shown on this example node: . If you partially merge some parameters of a node, the report marks the node as partially merged with a green merge arrow icon and no gray row highlighting.

5 (Optional) To revert all merge operations, on the **Comparison** tab, in the **Merge** section, click **Undo All**. A dialog box prompts you to confirm that you want to throw away all merge operations and revert the report and models to their original state.

6 Inspect your merge changes in the Simulink Editor. If necessary, connect any lines that the software did not connect automatically. The comparison report does not update to show any changes that you make in the Editor.

7 Save the model in the Editor or in the comparison report.

Merging Tips

- You must merge blocks before lines in the Simulink part of the report. In the Stateflow section, you must merge states and junctions before merging transitions, or the report cannot make the connections.

For an example showing how to merge a change involving multiple nodes, see `slxml_sfcar`.

- If you want to merge subsystems, be aware that in XML text files, subsystems are represented by two nodes — the container and the contents. The two nodes have the same name but different properties. For example, name changes are a property of the container node. You can merge the container parameters and contents independently. If you want to merge a subsystem and all its properties, merge both the container and the contents nodes.

- You cannot insert or delete parameters, and not all parameters can be merged. For example, you cannot merge Simulink Identifier (`SID`) parameters.

- If you change filter settings after any merge operations, you will lose your merge changes. A dialog box prompts you to confirm that you want to throw away all merge operations and revert the report and models to their original state. If you click **Yes** to continue, the analysis runs again and you see a new report with the new filtering applied.

- For information on merging between models with identical names, see "Comparing XML Files from Models with Identical Names" on page 6-33.

Merge MATLAB Function Block Code

1 To merge differences in MATLAB Function block code, create a comparison report for the parent models.

2 (Optional) On the View tab, turn off **Highlight in Models**. Otherwise, the parent models display each time you merge a difference in the MATLAB Function block code text comparison.

3 Next to the MATLAB Function block node in the XML comparison report, click the **Compare** button.

A new text comparison report opens.

4 In the text comparison report, select a difference in the code and click **Merge** to copy the selected difference from the left block to the right block.

5 After you finish merging differences, save the parent model in the Editor.

Related Examples

- "Resolve Conflicts with Simulink Three-Way Merge"

- "Compare Simulink Model XML Files" on page 6-8
- "Display Items in Original Models" on page 6-17
- "Source Control in Simulink Project"
- "Resolve Conflicts"
- "Compare Revisions"
- "Customize External Source Control to Use MATLAB for Diff and Merge"

Export, Print, and Save XML Comparison Results

Save Printable HTML Report

To save a printable version of an XML comparison report,

1   On the **Comparison** tab, in the **Comparison** section, select **Save As** > **HTML**, **Word**, or **Basic HTML**.

   The Save dialog box opens, where you can choose to save a printable version of the XML comparison report.

2   Select a file name and location to save the report.

The report is a noninteractive HTML document of the differences detected by the algorithm for printing, sharing, or archiving a record of the comparison. If you have applied filters, your filtered results appear in the printable report.

Export Results to the Workspace

To export the XML comparison results to the MATLAB base workspace,

1   On the **Comparison** tab, in the **Comparison** section, select **Save As** > **Workspace variable**.

   The Input Variable Name dialog box appears.

2   Specify a name for the export object in the dialog and click **OK**. This action exports the results of the XML comparison to an `xmlcomp.Edits` object in the workspace.

The `xmlcomp.Edits` object contains information about the XML comparison including file names, filters applied, and hierarchical nodes that differ between the two XML files.

To create an `xmlcomp.Edits` object at the command line without opening the Comparison Tool, enter:

```
Edits = slxmlcomp.compare(modelname_A,modelname_B)
```

Property of `xmlcomp.Edits`	Description
`Filters`	Array of filter structure arrays. Each structure has two fields, Name and Value.
`LeftFileName`	File name of left model exported to XML.
`LeftRoot`	`xmlcomp.Node` object that references the root of the left tree.
`RightFileName`	File name of right model exported to XML.
`RightRoot`	`xmlcomp.Node` object that references the root of the right tree.
`TimeSaved`	Time when results exported to the workspace.
`Version`	MathWorks release-specific version number of `xmlcomp.Edits` object.

Property of `xmlcomp.Node`	Description
`Children`	Array of `xmlcomp.Node` references to child nodes, if any.
`Edited`	Boolean — If `Edited = true` then the node is either inserted (green) or part of a modified matched pair (pink).
`Name`	Name of node.
`Parameters`	Array of parameter structure arrays. Each structure has two fields, Name and Value.
`Parent`	`xmlcomp.Node` reference to parent node, if any.
`Partner`	If matched, `Partner` is an `xmlcomp.Node` reference to the matched partner node in the other tree. Otherwise empty `[]`.

Save Comparison Log Files in a Zip File

Temporary comparison files accumulate in *tempdir*/`MatlabComparisons`/ `XMLComparisons/TempDirs/`. These temporary files are deleted when you close the related comparison report.

You can zip the temporary files (such as log files) created during XML text comparisons for sharing or archiving. While the comparison report is open, enter:

```
xmlcomp.zipTempFiles('c:\work\myexportfolder')
```

The destination folder must exist. The output reports the zip file name:

```
Created the zipfile "c:\work\myexportfolder\20080915T065514w.zip"
```

To view the log file for the last comparison in the MATLAB Editor, enter:

```
xmlcomp.showLogFile
```

Comparing XML Files from Models with Identical Names

You can compare XML text from files of the same name. To complete the operation, the XML comparison tool copies one of the models to a temporary folder, because Simulink cannot have two models of the same name in memory at the same time. The XML comparison tool creates a read-only copy of one model named *modelname*_TEMPORARY_COPY, and compares the resulting XML files.

Warning When you use reverse annotation from the report, one of the models displayed is a temporary copy with a new name. The temporary copy is read-only, to avoid making changes that can be lost.

Alternatively, you can run the comparison by renaming or copying one of the files.

All *merge* operations merge from left to right, so you cannot accidentally merge to a temporary copy. Merge operations on models with identical names copy changes from the left (temporary copy) model to the right model. If you swap sides, the report always places a new temporary copy on the left side of the report, so any merges change the original model file and never a temporary copy.

If one of the models is open when you try to compare XML files, a dialog box appears where you can click **Yes** to close the file and proceed, or **No** to abort. You must close open models before the XML comparison tool can compare XML files from two models with the same name. The problem requiring you to close the loaded model is called "shadowed files". In some cases, another model with the same name might be in memory, but not visible. See "Shadowed Files" in the Simulink documentation for more information.

If you want to automatically close open models of the same name when comparing XML files and not see the dialog box again, run these commands:

```
opt = slxmlcomp.options
opt.setCloseSameNameModel(true)
```
This is persistent across MATLAB sessions. To revert to default behavior and be prompted whether or not to close the open model every time, enter:

```
opt = slxmlcomp.options
opt.setCloseSameNameModel(false)
```

If you open a comparison report from Simulink Project (for example, using **Compare to Revision**), the project handles files of the same name and does not prompt you to close models.

Work with Referenced Models and Library Links

The XML comparison report applies only to the currently selected models, and does not include changes to any referenced models or linked libraries. For compatibility with source control and peer review workflows, the comparison report shows only changes in the files selected for comparison.

Tip If you want to examine your whole hierarchy instead, try using a Simulink Project, where you can examine modified files and dependencies across your whole project, and compare to selected revisions. See "Project Management".

If you are comparing models that contain referenced models with the same name, then your MATLAB path can affect the results. For example, this can happen if you generate an XML comparison report for the current version of your model and a previous baseline. Make sure that your referenced models are not on your MATLAB path before you generate the report.

The reason why results can change is that Simulink records information in the top model about the interface between the top model and the child model. This interface information in the top model enables incremental loading and diagnostic checks without any need to load child models.

When you load a model (for example, to compare XML) then Simulink refreshes the interface information for referenced models if it can find the child model. Simulink can locate the child model if it is on the path. If another model of the same name is higher on the path, Simulink updates the interface information for that other model before comparing XML. This can produce entries for interface changes for model reference blocks in the comparison report. Make sure your referenced models are not on your path before you generate the report, to avoid these interface changes in the results. If both model versions are off the path, the interface information in the top model is not refreshed during the XML comparison process. Instead the cached information is used, resulting in a valid XML comparison report.

With library links, Simulink does not update the cached interface information when comparing XML, and so the report correctly captures library interfaces. However with both referenced models and library links, Simulink updates the information when displaying the model. When displaying report items in original models, you may see that Simulink finds another model or library that is higher in the path. To obtain the clearest results, make sure that the models and associated libraries are temporarily removed

from the path. By removing the files from the path you will see unresolved library links and referenced models when you view the original models, but their interfaces will be correct and will correctly align with the comparison report.

Compare XML from Models Managed with Subversion

In this section...
"Work with Subversion" on page 6-37
"Configure TortoiseSVN" on page 6-38
"Test TortoiseSVN Setup" on page 6-39

Note: The functionality on this page will be removed in a future release. Instead, use either of these approaches:

- Built-in SVN source control integration in Simulink Project. See "Project Management".

- "Customize External Source Control to Use MATLAB for Diff and Merge"

Work with Subversion

Simulink Projects provide built-in Subversion source control integration. You can create an XML comparison report from the Modified Files view of the Simulink Project Tool. See "Project Management".

Note: Alternatively, you can customize your external Configuration Management tools to call the XML comparison functionality in the Simulink Report Generator. The functionality below on this page describes a previous approach and will be removed in a future release. Instead, use the instructions here:"Customize External Source Control to Use MATLAB for Diff and Merge".

Comparing two versions of the same file is a common workflow when using Configuration Management tools. If your Configuration Management tool is configurable, you can customize your *Diff* operations on Simulink model files from your Configuration Management tool to call the XML comparison functionality in the Simulink Report Generator. This allows you to compare two versions of the XML from the same model file and generate a report of the differences.

TortoiseSVN and Subversion are a popular suite of open-source version control tools. The following example describes how to configure TortoiseSVN to use the Simulink

Report Generator XML comparison. You can register the XML comparison function with TortoiseSVN as an extension-specific diff program to use for model files. When you perform a TortoiseSVN diff on a model file, TortoiseSVN uses the XML comparison to generate a report. This workflow describes a typical usage of Subversion on a Windows PC.

1 Configure TortoiseSVN to use the `fileComparisonDriver` function for model files.

2 When you perform a TortoiseSVN diff on a model file, the `fileComparisonDriver` function invokes the `visdiff` function to generate an XML comparison report.

Optionally, you can also configure TortoiseSVN to use the same function to call the Comparison Tool for `.mat` files and for Simulink manifest files (`.smf` files).

Configure TortoiseSVN

This example is compatible with Release 2008b+ onwards and was tested with Subversion 1.7.7 on Windows 7 Enterprise.

Configure TortoiseSVN to use the XML comparison tool for model files, as follows:

1 Right-click a file in Windows Explorer and select **TortoiseSVN > Settings**.

2 In the TortoiseSVN Settings dialog box, click **Diff Viewer** under **External Programs** in the tree, then click the **Advanced** button.

3 In the Advanced Diff settings dialog box, add an entry to specify what to do with model files by clicking **Add**.

4 In the Add extension specific diff program dialog box, enter `.mdl` or `.slx` for the **Extension** and enter the following command in the **External Program** edit box:

```
"matlabroot\bin\matlab.exe" -r fileComparisonDriver("'%base'","'%mine'") -nosplash
```

Replace *matlabroot* with the path to the specific location on your computer for your MATLAB installation, for example, `C:\Program files\MATLAB\R2009a`.

The following example shows this setup on an R2013b installation.
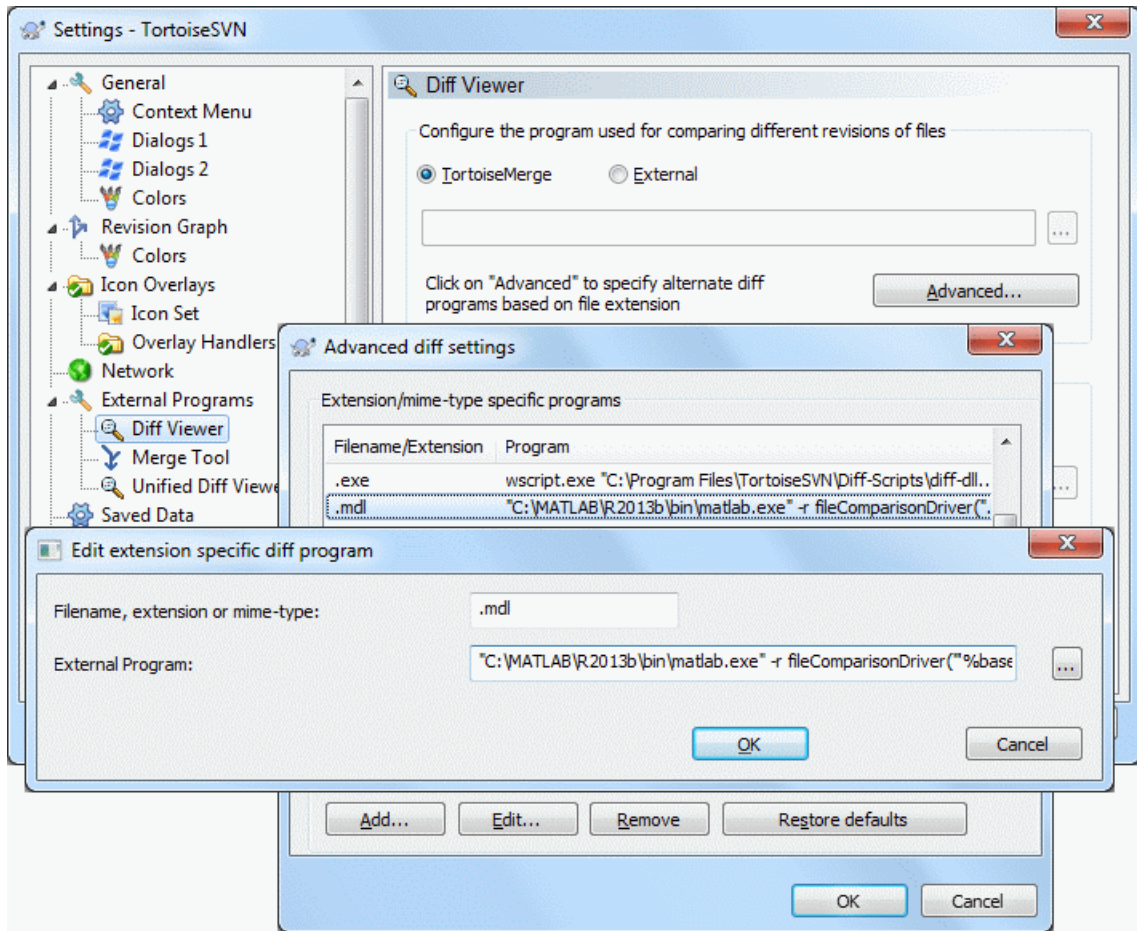
5. Click **OK** to apply your changes and close all the Tortoise SVN dialog boxes.

6. If you also want to use the MATLAB Comparison Tool to compare MATLAB files, such as .m and .mat files, or Simulink manifest files (.smf files) you can repeat the steps to add exactly the same **External Program** command for .m, .mat and .smf files.

Test TortoiseSVN Setup

To test your setup, follow these steps:

1   Start MATLAB and open, modify, and save a Simulink model that is managed in a Subversion archive. This creates a local working copy that is different to the head repository copy.

2   In Windows Explorer, right-click your modified file, and select **TortoiseSVN** > **Diff**.

TortoiseSVN runs a new instance of MATLAB. MATLAB loads and runs the `fileComparisonDriver.m` file, located in the folder `matlab\toolbox\rptgenext \rptgenextdemos\slxmlcomp`. The `fileComparisonDriver` function performs these steps:

1   Creates temporary copies of the current working version of the Simulink model and the previously stored version of the model.

2   Compares the XML text files from both models and generates a comparison report displayed in the MATLAB Comparison Tool.

The function must preprocess the file names by creating renamed temporary copies because Subversion uses a temporary file naming convention that is not compatible with Simulink because of invalid delimiting characters. The branch and version information is embedded in the temporary model names. See also "Comparing XML Files from Models with Identical Names" on page 6-33 for information about using the report and a warning to avoid losing work in the temporary models.

Other TortoiseSVN workflows using Diff operations are also supported, such as comparing two versions in an archive.

Related Examples

·   "Customize External Source Control to Use MATLAB for Diff and Merge"

Compare Project or Model Templates
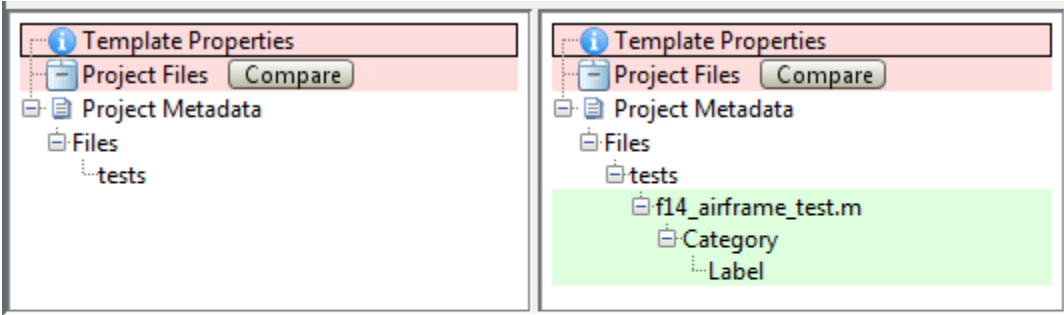
In this section...
"Compare Project Templates" on page 6-41
"Compare Model Templates" on page 6-42

Compare Project Templates

You can compare project templates (SLTX files). If you select two project template files to compare, you see a comparison report showing differences in template properties and project metadata. You can open a new report to investigate project file and folder differences.

- Click **Template Properties** to view differences in the Parameters, such as the description or date modified.
- Expand the **Project Metadata** node to view metadata differences such as label changes.
- Next to **Project Files**, click **Compare** to open a folder comparison where you can investigate changed, added, or removed files and folders.

Comparing template deploy.sltx vs. template labels.sltx

Left file list	Contents of template C:\Work\deploy.sltx
Right file list	Contents of template C:\Work\labels.sltx

Click on a column header to sort the table

		In left list (template deploy.sltx)		In right list (template labels.sltx)		
Type	**File Name ▲**	**Size (bytes)**	**Last Modified Date**	**Size (bytes)**	**Last Modified Date**	**Difference Summary**
MATLAB Code File	utilities/set_up_project.m (open: left | right)	583	2014-12-10 12:34:26	583	2014-12-10 12:34:26	*identical*
MATLAB Code File	utilities/rebuild_s_functions.m (open: left | right)	2213	2014-12-10 12:34:26	2213	2014-12-10 12:34:26	*identical*
MATLAB Code File	utilities/clean_up_project.m (open: left | right)	442	2014-12-10 12:34:26	442	2014-12-10 12:34:26	*identical*
MATLAB Code File	tests/f14_airframe_test.m (open)	*(not in this list)*		1775	2014-12-10 12:34:26	*added*
folder	tests/	-	-	*(not in this list)*		*removed*

Compare Model Templates

You can compare model templates (SLTX files). If you select two model template files to compare, you see a comparison report showing differences in template properties. You can open a new comparison report to compare XML text files from the Simulink models.

· Click **Template Properties** to view differences in the Parameters, such as the description or date modified.

· Next to **Model**, click **Compare** to open a Simulink model XML comparison report where you can investigate differences.

2014-12-10T12:27:38Z	Parameter	2014-12-10T12:24:04Z
	created	
2014-12-10T12:27:38Z	created	2014-12-10T12:24:04Z
This template creates a model containing a continuous time plant model and a controller.	description	This template creates a model containing a continuous time plant model and a controller. The plant model is a simple damped, second order system. A discrete PID controller is used to make the output of the plant track an input reference signal. The model is configured to log data that can be displayed using the Simulation Data Inspector.
2014-12-10T12:27:38Z	modified	2014-12-10T12:24:04Z
Feedback Control	title	Discrete Feedback Control

Related Examples

- "Compare Simulink Model XML Files" on page 6-8
- "Comparing Folders and Zip Files"
- "Create a Template from the Current Project"
- "Create a Template from a Model"

Components — Alphabetical List

For a list of MATLAB Report Generator components, see the MATLAB Report Generator documentation.

Annotation Loop

Run child components multiple times for each Simulink annotation in current context

Description

This component runs its child components multiple times for each Simulink annotation in the current context. The parent component determines the context.

- `Model Loop`: Reports on all annotations inside the reported portion of the reported model.
- `System Loop`: Reports on all annotations inside the current system.
- `Block Loop` or `Signal Loop`: Does nothing.

Loop Options

The Loop Options pane displays information about the current context. You can sort `Alphabetically by text` or `In traversal order`.

Child components of the `Annotation Loop` consider their context to be annotations when the report is running.

For example, the following components report on the looped annotation:

- `Simulink Automatic Table`
- `Simulink Linking Anchor`
- `Simulink Name`
- `Simulink Property`
- `Simulink Property Table`

Use a Summary Table component to show annotation objects in reports. Each Summary Table component creates a single table with each reported annotation on a single row of the table.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.

- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.

- **Create link anchor for each object in loop**: Create a link target for each annotation in the loop so that other parts of the report can link to it.

See Also

`Block Loop`, `Model Loop`, `Signal Loop`, `System Loop`, `Simulink Linking Anchor`, `Simulink Name`, `Simulink Property`, `Simulink Property Table`, `Simulink Summary Table`

Block Execution Order List

Create a list or table of all nonvirtual blocks in the model, showing order in which they execute

Description

This component creates a list or table of all nonvirtual blocks in the model, showing the order in which they execute.

For more information about virtual and nonvirtual blocks, see "Nonvirtual and Virtual Blocks" in the Simulink documentation.

Properties

- **List Title**:

    - `Automatic`: Generates a list or table title automatically.
    - `Custom`: Enables you to enter a title.

- **Include block type information**: Include each block's `BlockType` property in the list or table.

- **Look under nonvirtual subsystems**: The default is `Automatic (On for models, Off for systems)`. Set it to `On` or `Off`.

Insert Anything into Report?

Yes. List.

Class

rptgen_sl.csl_blk_sort_list

See Also

Block Loop

Block Loop

Run child components for each block in the current system, model, or signal

Description

This component runs its child components for each block contained in the current system, model, or signal.

For conditional processing based of blocks, you can use the `RptgenSL.getReportedBlock` function. For more information, see "Loop Context Functions" on page 4-55.

Report On

This pane describes the type of object on which this component operates.

- **Automatic list from context**: Report on all blocks in the current context. The parent component of the Block Loop determines its context. If this component does not have the `Model Loop`, `System Loop`, `Signal Loop`, or `Block Loop` as its parent, selecting this option causes this component to report on all blocks in all models.

  - `Model Loop`: Reports on all blocks in the current model.
  - `System Loop`: Reports on all blocks in the current system.
  - `Signal Loop`: Reports on all blocks connected to the current signal.
- **Custom - use block list**: Enables you to specify a list of blocks on which to report. Enter the full path of each block.

Loop Options

Choose block sorting options and reporting options in this pane.

- **Sort blocks**:

  Use this option to select how to sort blocks (applied to each level in a model):

  - `Alphabetically by block name`. Sorts blocks alphabetically by their names.

- `Alphabetically by system name`. Sorts systems alphabetically. The report lists blocks in each system, but in no particular order.

- `Alphabetically by full Simulink path`. Sorts blocks alphabetically by Simulink path.

- `By block type`. Sorts blocks alphabetically by block type.

- `By block depth`. Sorts blocks by their depth in the model.

- `By layout (left to right)`: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- `By layout (top to bottom)`: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- `By traversal order`. Sorts blocks by traversal order.

- `By simulation order`. Sorts blocks by execution order.

- `%<VariableName>`: Inserts the value of a variable from the MATLAB workspace. The `%<>` notation can denote a string or cell array. The following example reports on the `theta dot` integrator block and the `theta` integrator block in the model `simppend`, using the variable `Z={ 'simppend/theta'}`:

```
simppend/theta dot
%<Z>
```
The generated report includes information about the following blocks:

- `simppend/theta dot`
- `simppend/theta`

For more information, see `%<VariableName> Notation` on the `Text` component reference page in the MATLAB Report Generator documentation.

- **Search for Simulink property name/property value pairs**: Reports only on Simulink blocks with specified property name/property value pairs.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each block found in the loop.
- **Display the object type in the section title**: Automatically inserts the object type into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each block in the loop so that other parts of the report can link to it. For example, the image created by a `System Snapshot` component can link to the block information only if you select this check box.

Insert Anything into Report?

Yes, inserts a section if you select the **Create section for each object in loop** option and a link target if you select **Create link anchor for each object in loop**.

Class

`rptgen_sl.csl_blk_loop`

See Also

Model Loop, Signal Loop, System Loop, Simulink Linking Anchor, Simulink Name, Simulink Property, Simulink Property Table, Simulink Summary Table

Block Type Count

Count number of each block type in the current model or system

Description

This component counts the number of each block type in the current model or system. Within a model, this component counts blocks underneath masks and inside library links.

For more information about block types, see "Nonvirtual and Virtual Blocks" in the Simulink documentation.

Count Types

The parent of this component determines where to count block types:

- `Model Loop`: Reports all block types in the current model:

    - `All blocks in model`: Counts block types in the entire model.
    - `All blocks in reported systems`: Counts block types only in systems that appear in the report.

- `System Loop`: Reports all block types in the current system.

Table Content

- **Table title**: Allows you to enter the table title.
- **Show block names in table**: Includes a column that displays all block names in the table.
- **Sort table**:

    - `Alphabetically by block type`: Sorts blocks alphabetically by block type.
    - `By number of blocks`: Sorts by decreasing number of occurrences.

- **Show total count**: Displays total number of block types.

Insert Anything into Report?

Yes. Table.

Class

`rptgen_sl.csl_blk_count`

See Also

`Block Loop`, `Model Loop`, `System Loop`

Bus

Create list of signals exiting from `Bus Selector` block

Description

This component creates a list of signals exiting a `Bus Selector` block. The list contains signals leaving from the reported block or downstream buses and signals.

The parent of this component determines which buses appear in the report:

- `Model Loop`: Includes all buses in the current model.
- `System Loop`: Includes all buses in the current system.
- `Block Loop` : If the current block is a bus block, then the report includes that block.
- `Signal Loop`: Includes all buses connected to the current signal.

If the Bus component does not have a looping component as its parent, it reports on all buses in all open models.

Properties

- **Show Bus Hierarchy**: Specifies whether the list displays downstream buses hierarchically.
- **Insert linking anchor for bus blocks**: Inserts a linking anchor for each bus block. This property designates the list item as the location to which other links for that block point. (For more information, see the `Simulink Linking Anchor` or `Link` component reference pages.) Do not use this option if you have already specified an anchor location for the bus block with an `Object Linking Anchor` component.
- **Insert linking anchor for signals**: Inserts a linking anchor for each signal. This property designates the list item as the location to which other links for that signal point. For more information, see the `Simulink Linking Anchor` or `Link` component reference pages.) Do not use this option if you have already specified an anchor location for the signal with an `Object Linking Anchor` component.

- **Title**: Inserts a title before each list. This attribute supports the `%<varname>` notation. For more information, see `%<VariableName> Notation` on the `Text` component reference page in the MATLAB Report Generator documentation.

Insert Anything into Report?

Yes. List.

Class

```
rptgen_sl.csl_blk_bus
```

See Also

`Block Loop`, `Model Loop`, `Signal Loop`, `Simulink Linking Anchor`, `System Loop`,

Chart Loop

Run child components for specified Stateflow charts

Description

This component runs its children for specified Stateflow charts.

For conditional processing for a chart, you can use the `RptgenSF.getReportedChart` function. For more information, see "Loop Context Functions" on page 4-55.

Report On

- **Automatic list from context**: Report on all chart blocks in the context set by the parent of this component.

    - `Model Loop`: Reports on all Stateflow chart blocks in the current model.
    - `System Loop`: Reports on all Stateflow chart blocks in the current system.
    - `Signal Loop`: Reports on all Stateflow chart blocks connected to the current signal.
    - `Machine Loop`: Reports on the current block if it is in a Stateflow chart.

    If the `Chart Loop` component has any other type of component as its parent, selecting this option causes it to report on all Stateflow chart blocks.

- **Custom - use block list**: Reports on a specified list of Stateflow chart blocks.

Loop Options

Choose chart block sorting options and reporting options in this pane.

- **Sort blocks**: Specifies how to sort blocks (applied to each level in a model). This option is available if you select `Automatic list from context` in the **Report On** section, or if you select `Custom - use block list` and the `Sort blocks` option.

    - `Alphabetically by block name`. Sorts blocks alphabetically by name.
    - `Alphabetically by system name`. Sorts systems alphabetically by name. Lists blocks in each system, but in no particular order.

- **Alphabetically by full Simulink path**. Sorts models alphabetically by their full paths.

- **By block type**. Sorts blocks alphabetically by block type.

- **By depth**. Sorts blocks by their depth in the model.

- **By layout (left to right)**: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- **By layout (top to bottom)**: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- **By simulation order**. Sorts blocks by execution order.

- **%<VariableName>**: Inserts the value of a variable from the MATLAB workspace. The %<> notation can denote a string or cell array. For more information, see **%<VariableName> Notation** on the **Text** component reference page in the MATLAB Report Generator documentation.

- **Search for Simulink property name/property value pairs**: Reports on Simulink blocks with specified property name/property value pairs.

- **Search Stateflow**: Reports on Stateflow charts with specified property name/ property value pairs.

7-15

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.

- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.

- **Create link anchor for each object in loop**: Create a link target for each chart in the loop so that other parts of the report can link to it. For example, the image created by a `Stateflow Snapshot` component can link to this information only if you select this check box.

Insert Anything into Report?

Yes, inserts a section if you select the **Create section for each object in loop** option.

Class

rptgen_sf.csf_chart_loop

See Also

`Block Loop`, `Machine Loop`, `Model Loop`, `Signal Loop`, `System Loop`, `Simulink Function System Loop`

Code Generation Summary

Insert version number information, list of generated files, tables summarizing code generation configuration information, and subsystem maps into report

Description

This component reports the following information:

- Version number information
- List of generated files
- Code generation configuration information
- Subsystem map

Summary

- **General information**: Includes the following information in the report:

  - Model name and version
  - Simulink Coder version number
  - List of full paths of generated files
- **Configuration settings**: Includes tables that list optimization and Simulink Coder target selection and build process Configuration Parameter settings.
- **Subsystem map**: Includes in the report a unique mapping between subsystem numbers and subsystem labels in the model.

Traceability Report

Use settings from model: When you select this option, the report uses all of the following configuration settings, as specified in your model. Deselecting this option allows you to turn off one or more of these settings as needed:

- **Eliminated/virtual blocks**
- **Traceable blocks**

- **Traceable StateFlow Objects**
- **Traceable MATLAB Function Blocks**

For more information on these configuration settings, see "Model Configuration Parameters: Code Generation Report" in the Simulink Coder documentation.

Insert Anything into Report?

Yes. Tables and list.

Class

RptgenRTW.CCodeGenSummary

See Also

Import Generated Code

Data Dictionary Traceability Table

Insert a table that links data dictionary information to requirements

Description

This component inserts a table into the report. The table links data dictionary information to corresponding requirements. This component reports on the currently open data dictionary. Place this component inside a section, paragraph, or table component.

To use this component, your report setup must include `Eval` statements that open a data dictionary or determine the data dictionary that is open. To open a template report that shows an example of these `Eval` statements, at the MATLAB command prompt, enter:

```
setedit([matlabroot '/toolbox/slrequirements/+rmide/rmide.rpt'])
```

Find the `Eval` statements in the `if` condition at the beginning of the report setup.

Table Options

Specify information about the table.

- **Table title**: Specify the table title.

    - `No title` — Do not include a table title.
    - `Object name` — Use the name of the data dictionary in the title.
    - `Custom` — Specify your own table title.

Table Columns

Specify the table columns that you want to include in the report. The **Document name**, **Locations within document**, and **Requirement keyword** check boxes correspond to properties on the Requirements Management Interface Link Editor dialog box.

- **Description** — Include the description of the requirement. The description helps you to identify the requirement the table is linking to. Leave this box selected to improve the readability of your table.

- **Document name** — Include the name of the document where the requirement is located.
- **Locations within document** — Include the identifier of a location in the document.
- **Requirement keyword** — Include the requirement keyword.

Insert Anything into Report?

Yes. Table.

Class

`RptgenRMI.DDReqTable`

See Also

`MATLAB Code Traceability Table`, `Simulink Test Suite Traceability Table`, `rmi`

Documentation

Insert text extracted from `DocBlock` blocks in Simulink models

Description

This component inserts text extracted from `DocBlock` blocks in Simulink models. It can have the following components as its parent:

- `Model Loop`
- `System Loop`
- `Block Loop`

The specified report format determines the format of the `DocBlock` block data inserted into the report:

- `HTML`: Imports HTML data into the report.

Note: For non-English HTML DocBlock text that you want to include in a Documentation component, use UTF-8 file encoding. Use a simple text editor to create the HTML code.

- `RTF`: Imports RTF data into the report.

Properties

- **Import file as**: Specifies how to format the imported information. The following example shows how each option works, using the following text as input:

```
First row.
  Second row.

Third row follows blank line.
```

  - `Plain text (ignore line breaks)`. Imports plain text without any line breaks (no paragraphs), as in this example:

    ```
    First row. Second row. Third row follows blank line.
    ```

- `Paragraphs defined by line breaks`. Imports the text contained in paragraphs defined by line breaks (hard returns or carriage returns), as in this example:

  ```
  First row.
  Second row.

  Third row follows blank line.
  ```

- `Paragraphs defined by empty rows`. Imports text contained in paragraphs defined by empty rows (rows that do not contain text), as in this example:

  ```
  First row. Second row.

  Third row follows blank line.
  ```

- `Text (retain line breaks)`. Imports plain text, including line breaks, as in this example:

  ```
  First row.
  Second row.

  Third row follows blank line.
  ```

- `Fixed-width text (retain line breaks)`. Imports fixed-width text (all letters have the same width or size) including line breaks, as in this example:

  ```
  First row.
    Second row.

  Third row follows blank line.
  ```

  Tip This option is useful for importing MATLAB files.

- **Insert linking anchor for blocks**: Inserts a linking anchor for each `DocBlock` block that designates the location where other links for that block point. (See the `Simulink Linking Anchor` or `Link` component reference pages for more help.) Do not use this option if you have already specified an anchor location for a `DocBlock` block with an `Object Linking Anchor` component.

Insert Anything into Report?

Yes. Text, paragraph, or external RTF/HTML data.

Class

`rptgen_sl.csl_blk_doc`

See Also

`Block Loop`, `Model Loop`, `Simulink Linking Anchor`, `System Loop`

Fixed Point Block Loop

Run child components for the Simulink model, system, or signal defined by parent component

Description

This component runs its children for the Simulink model, system, or signal that its parent defines. Options for the parent component are:

- `Model Loop`
- `System Loop`
- `Signal Loop`

Report On

- **Automatic list from context**: Reports on all fixed-point blocks in the context of the parent of this component. For example, if the parent component is the `System Loop`, then this component reports on all fixed-point blocks in the current system. If this component does not have a looping component as its parent, then selecting this option causes the component to report on all fixed-point blocks in all models.
- **Custom - use block list:**: Reports on a specified list of blocks.

Loop Options

Choose block sorting options and reporting options in this pane.

- **Sort blocks**: Specifies how to sort blocks (applied to each level in a model). This option is available if you select the `Automatic list from context` option in the **Report On** section, or if you select `Custom - use block list` and the `Sort blocks` options.

  - `Alphabetically by block name`. Sorts blocks alphabetically by name.
  - `Alphabetically by system name`. Sorts systems alphabetically. Lists blocks in each system, but in no particular order.

- `Alphabetically by full Simulink path`. Sorts blocks alphabetically by Simulink path.

- `By block type`. Sorts blocks alphabetically by block type.

- `By block depth`. Sorts blocks by their depth in the model.

- `By layout (left to right)`: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- `By layout (top to bottom)`: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- `By traversal order`. Sorts blocks by traversal order.

- `By simulation order`. Sorts blocks by execution order.

- `%<VariableName>`: Inserts the value of a variable from the MATLAB workspace. For more information, see `%<VariableName> Notation` on the `Text` component reference page in the MATLAB Report Generator documentation.

- **Search for Simulink name/property value pairs**: Reports only on blocks with the specified property name/property value pairs. To enable searching, click the check box. In the first row of the property name and property value table, click inside the

edit box, delete the existing text, and type the property name and value. To add a row, clck the **Add row** button.

For information about subsystem property names and values, in "Block-Specific Parameters", see the "Ports & Subsystems Library Block Parameters" section.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each fixed-point block in the loop so that other parts of the report can link to it. For example, the image created by a `System Snapshot` component can link to this information only if you select this check box.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

`rptgen_fp.cfp_blk_loop`

See Also

`Block Loop`, `Model Loop`, `Signal Loop`, `Simulink Linking Anchor`, `System Loop`

Fixed Point Logging Options

Set fixed-point options like in Fixed Point Tool

Description

This component sets fixed-point options like those set in the Fixed Point Tool (invoked by running the `fxptdlg` function).

This component must be a child of the `Model Loop` component. Use this component to set the following options on the current model:

- Data type override
- Fixed-point instrument mode
- Logging type

This component can have child components. It is a good practice to use this component with a `Model Simulation` component as its child. This approach sets fixed-point properties for the model for the purpose of the simulation, and then restores them to their original values after the simulation is complete.

Data Type Override

- **Use local settings**: Overrides data types according to the value of this parameter set for each subsystem. Otherwise, settings for parent systems override those of child systems.
- **Scaled double**: Overrides the output data type of all blocks in the current system or subsystem with doubles. However, this option maintains the scaling and bias specified in the mask of each block.
- **Doubles**: Overrides the output data type of all blocks in the current system or subsystem with doubles. The overridden values have no scaling or bias.
- **Singles**: Overrides the output data type of all blocks in the current system or subsystem with singles. The overridden values have no scaling or bias.
- **Off**: Does not perform any data type override on any block in the current system or subsystem.

Fixed-Point Instrumentation Mode

Specify logging options in this section. For logged blocks, minimum and maximum simulation values are written to the workspace.

- **Use local settings**: Logs data according to the value of this parameter set for each subsystem. Otherwise, settings for parent systems always override those of child systems.
- **Min, max, and overflow**: Logs minimum value, maximum value, and overflow data for all blocks in the current system or subsystem.
- **Overflow only**: Logs only overflow data for all blocks in the current system or subsystem.
- **Force off**: Logs no data for any block in the current system or subsystem. Use this selection to work with models containing fixed point-enabled blocks, if you do not have a Fixed-Point Designer license.

For more information on logging simulation results, see "Propose Fraction Lengths Using Simulation Range Data" in the Fixed-Point Designer documentation.

Logging Type

Specify how to record logs in this section:

- **Overwrite log**: Clears information in the logs before new logging data is entered.
- **Merge log**: Merges new logging data with previously logged information.

Insert Anything into Report?

No.

Class

rptgen_fp.cfp_options

See Also

Model Simulation

Fixed Point Property Table

Insert table that reports on Fixed-Point Designer block property name/property value pairs

Description

This component inserts a table that reports on Fixed-Point Designer block property name/property value pairs.

Table

Select a preset table, which is already formatted and configured, in the **preset table** list in the upper-left corner of the attributes page.

- **preset table**

    Specifies the type of object property table.

    - `Default`
    - `Mask properties`
    - `Block limits`
    - `Out-of-range errors`
    - `All fixed-point properties`
    - `Blank 4x4`

    To apply the specified table, select the table and click **Apply**.

- **Split property/value cells**: Split property name/property value pairs into separate cells. For the property name and property value to appear in adjacent horizontal cells in the table, select the **Split property/value cells** check box. In this case, the table is in split mode, so there only one property name/property value pair can exist in a cell. If there is more than one name/property pair in a cell, only the first pair appears in the report. The report ignores all subsequent pairs.

    For the property name and property value to appear together in one cell, clear the **Split property/value cells** check box. That option specifies nonsplit mode. Nonsplit mode supports more than one property name/property value pair and text.

Before switching from nonsplit mode to split mode, make sure that there is only one property name/property value pair per table cell. If you have more than one property name/property value pair or text in one cell, only the first value pair appears in the report. Subsequent pairs and text are omitted.

- **Display outer border**: Display the outer border of the table in the generated report.

Table Cells

Select table properties to modify. The selection in this pane affects the available fields in the **Cell Properties** pane.

Cell Properties

- **Contents**

  Modify the contents of the table cell selected in the **Table Cells** pane.
- **Show as**: Specifies the format for the contents of the table cell.

  - `PROPERTY Value`
  - `Value`
  - `Property Value`
  - `Property: Value`
  - `PROPERTY: Value`
  - `Property - Value`
  - `PROPERTY - Value`
- **Alignment**: Aligns the contents of the table cell.

  - `Center`
  - `Left`
  - `Right`
  - `Double justified`
- **Lower border**: Displays the lower border of the table in the generated report.
- **Right border**: Displays the right border of the table in the generated report.

Creating Custom Tables

To create a custom table, edit a preset table, such as the `Blank 4x4` table. Add and delete rows and add properties. To open the Edit Table dialog box, click **Edit**.

For details about creating custom property tables, see "Property Table Components".

Insert Anything into Report?

Yes. Table.

Class

`rptgen_fp.cfp_prop_table`

See Also

`Fixed Point Summary Table`

Fixed Point Summary Table

Table of specified fixed-point block properties or parameters

Description

This component displays properties or parameters of specified fixed-point blocks in a table.

Properties

Table title

Choose a table title in the generated report:

- `Automatic`: Generates a title automatically from the parameter.
- `Custom`: Specifies a custom title.

Property Columns

Property name

This field displays the object properties to include in the Summary Table in the generated report.

- To add a property:

   1 Select the appropriate property level in the menu

   2 Select the property to add from the selection list and click **Add**.
- To delete a property, select the property name and click the **Delete** button.
- To move properties up and down in the list, click the **Up** and **Down** buttons.

Note: Some entries in the list of available properties (such as `Depth`) are "virtual" properties that you cannot access using the `get_param` command. The properties used

---

for property/value filtering in the block and system loop components must be retrievable by the `get_param`. Therefore, you cannot configure your Summary Table to report on all blocks of `Depth == 2`.

Transpose table

Enabling this check box changes the summary table rows into columns in the generated report, putting the property names in the first column and the values in the other columns.

Object Rows

- **Insert anchor for each row**: Inserts an anchor for each row in the summary table.
- **Report On**: Specifies blocks on which to report:

  - `Automatic list from context`. Reports on all blocks in the current context.
  - `Custom - use block list`. Reports on a specified list of blocks. To include a given block in the report, specify its full path.

Loop Options

- **Sort blocks**: Specifies how to sort blocks (applied to each level in a model):

  - `Alphabetically by block name`. Sorts blocks alphabetically by name.
  - `Alphabetically by system name`. Sorts systems alphabetically. Lists blocks in each system, but in no particular order.

  - `Alphabetically by full Simulink path`. Sorts blocks alphabetically by Simulink path.
  - `By block type`. Sorts blocks alphabetically by block type.
  - `By block depth`. Sorts blocks by their depth in the model.
  - `By layout (left to right)`: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The

other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- By `layout (top to bottom)`: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.
- By `traversal order`. Sorts blocks by traversal order.
- By `simulation order`. Sorts blocks by execution order.
- **Search for Simulink property name/property value pairs**: Reports only on Simulink blocks with specified property name/property value pairs.

Insert Anything into Report?

Yes. Table.

Class

rptgen_fp.cfp_summ_table

See Also

Fixed Point Property Table

Import Generated Code

Import source and header files generated by Simulink Coder software, and custom files specified as part of model

Description

This component imports source and header files generated by Simulink Coder software. It also imports custom files that you specify as part of your model.

Properties

- **Source files (auto-generated)**: Includes the following files in the report:

    - `.c` and `.cpp` source files generated by Simulink Coder software.
    - Simulink Coder source files, such as the setup file and supporting files in the build folder.

    This check box is selected by default. Clear it to omit source files.

- **Header files (auto-generated)**: Includes the following files in the report:

    - `.h` and `.hpp` header files generated by Simulink Coder software.
    - Simulink Coder header files in the build folder.

    This check box is selected by default. Clear it to omit source files.

- **Custom files**: Includes custom source files that you specify in the **Code Generation > Custom Code** pane of the Configuration Parameters dialog box. This check box is deselected by default.

Insert Anything into Report?

Yes. Generated code listings.

Class

RptgenRTW.CImportCode

See Also

Code Generation Summary

Look-Up Table

Report on lookup table blocks

Description

The Look-Up Table component reports on the following blocks in the Simulink Lookup Tables library. Some examples of the lookup table blocks include:

- `1-D Lookup Table`
- `n-D Lookup Table`
- `Cosine`
- `Interpolation Using Prelookup`
- `Direct Lookup Table (n-D)`

The Look-Up Table component inserts a figure and/or table into the report. The table contains input and output numeric values. A figure plots these values.

Note: The Look-Up Table component does not display a table or plot for the `Direct Lookup Table (n-D)` block if the block is configured to generate the table during simulation as a block input. Instead, the Look-Up Table displays a note in the report to the effect that the table is generated dynamically during simulation.

Look-Up Table Options

This pane allows you to specify the types of lookup table blocks to include in the report and how they appear. If you select none of the check boxes in this pane, the component does not insert anything into the report.

- The Look-Up Table displays results according to the type of its parent component:
    - `Model Loop`: Includes all lookup tables in the current model.
    - `System Loop`: Includes all lookup tables in the current system.
    - `Block Loop`: If the current block is a lookup table, the reports that block.
    - `Signal Loop`: Includes all lookup tables connected to the current signal.

- If the Look-Up Table does not have any of the looping components as its parent, it includes all lookup tables in all open models.
- **Plot 1-D data**: Plots data from a `1-D Lookup Table` block. Choose the plot type, `Line plot` or `Bar plot`, from the corresponding list. The input data appears on the horizontal or *x*-axis, and the output data appears on the vertical or *y*-axis.

  For more information on line and bar plots, see "2-D and 3-D Plots" in the MATLAB Graphics documentation.
- **Create table for 1-D data**: Creates a table that contains numeric data values from the `1-D Lookup Table` block.
- **Plot 2-D data**: Creates a plot of `2-D Lookup Table` blocks. You can specify whether the data appears as a surface plot or a line plot. The line plot is best for small data sets, and the surface plot for larger tables. For more information on surface and line plots, see "2-D and 3-D Plots" in the MATLAB Graphics documentation.

Note: This option creates a 2-D slice through n-D data.

- **Create table for 2-D data**: Creates a table that contains numeric data values from the `2-D Lookup Table` block.
- **Create table for N-D data**: Creates a table that contains numeric data values from the `n-D Lookup Table` block.

Print Options

- **Image file format**: Specifies the image file format. Select `Automatic HG Format` (the default) to choose automatically the format best suited for the output format that you chose in the Report component. Otherwise, choose an image format that your output viewer can read.

  - `Automatic SL Format` (Uses the Simulink file format selected in the Preferences dialog box)
  - `Bitmap (16m-color)`
  - `Bitmap (256-color)`
  - `Black and white encapsulated PostScript`
  - `Black and white encapsulated PostScript (TIFF)`
  - `Black and white encapsulated PostScript2`

- • Black and white encapsulated PostScript2 (TIFF)
- • Black and white PostScript
- • Black and white PostScript2
- • Color encapsulated PostScript
- • Color encapsulated PostScript (TIFF)
- • Color encapsulated PostScript2
- • Color encapsulated PostScript2 (TIFF)
- • Color PostScript
- • Color PostScript2
- • JPEG high quality image
- • JPEG medium quality image
- • JPEG low quality image
- • PNG 24-bit image
- • TIFF - compressed
- • TIFF - uncompressed
- • Windows metafile

- **Paper orientation**:

  - • Landscape
  - • Portrait
  - • Rotated
  - • Use figure orientation: Uses the orientation for the figure, which you set with the orient command.
  - • Full page image (PDF only): In PDF reports, scales images to fit the full page, minimizes page margins, and maximizes the size of the image by using either a portrait or landscape orientation.

  For more information about paper orientation, see the orient command in the MATLAB documentation.

- **Image size**: Allows you to specify the image size in the report by selecting Use figure PaperPositionMode setting and setting the PaperPositionMode property of the Handle Graphics figure.

- Automatic (same size as on screen):
- Custom: Specifies a custom image size. Set the image size using the **Size** field and **Units** list.

    For more information on paper position mode, see `orient` in the MATLAB documentation.

- **Size**: Allows you to enter the size of the Handle Graphics figure snapshot in the format `wxh` (width times height). This field is active only if you choose `Custom` in the **Image size** list box.

- **Units**: Allows you to enter for the size of the Handle Graphics figure snapshot. This field is active only if you choose `Custom` in the **Image size** list box.

- **Invert hardcopy**: Causes the Handle Graphics `InvertHardcopy` property to invert colors for printing. In other words, this option changes dark colors to light colors and light colors to dark colors. To change colors in your image, choose one of the following options:

    - Automatic: Automatically changes a dark axes colors to light axes colors. If the axes color is a light color, this option does not invert the color.

    - Invert: Changes dark axes colors to light axes colors, and light axes colors to dark axes colors.

    - Don't invert: Does not change the colors in the image that appears on the screen for printing.

    - Use figure's InvertHardcopy setting: Uses the `InvertHardcopy` property set in the Handle Graphics image.

    - Make figure background transparent: Makes the image background transparent.

Display Options

- **Scaling**: Controls size of the image, as displayed in a browser. Making an image larger using this option does not affect the storage size of the image, but the quality of the displayed image may decrease as you increase or decrease the size of the displayed image.

    Generally, to achieve the best and most predictable display results, use the default setting of `Use image size`.

- `Use image size`: Causes the image to appear the same size in the report as on screen (default).
- `Fixed size`: Specifies the number and type of units.
- `Zoom`: Specifies the percentage, maximum size, and units of measure.
- **Size**: Specifies the size of the snapshot in the form w h (width, height) format. This field is active only if you choose `Fixed size` in the **Scaling** selection list.
- **Max size**: Specifies the maximum size of the snapshot in the form w h (width, height). This field is active only if you choose `Zoom` from the **Scaling** selection list.
- **Units**: Allows you to enter units for the size of the snapshot. This field is active only if you choose `Zoom` or `Fixed size` in the **Image size** list box.
- **Alignment**: Only reports in `PDF` or `RTF` format support this property.

  - `Auto`
  - `Right`
  - `Left`
  - `Center`
- **Title**: Enter text to appear above the snapshot.
- **Caption**: Enter text to appear under the snapshot.

Insert Anything into Report?

Yes. Figure and/or table.

Class

`rptgen_sl.csl_blk_lookup`

See Also

`Block Loop`, `Model Loop`, `Signal Loop`, `System Loop`

Machine Loop

Run child components for specified Stateflow machines

Description

This component runs its child components for selected Stateflow machines. The behavior of this component depends on its parent component. If it has no parent, the `Machine Loop` runs its child components for all machines. If it has the `Model Loop` is its parent, it runs its child components for all machines in the model.

Loop Options

Search Stateflow

If selected, searches states that you specify in the field that appears under the check box.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each machine in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select the **Create section for each object in loop** option.

Class

rptgen_sf.csf_machine_loop

See Also

Model Loop

Missing Requirements Block Loop

Apply all child components to blocks that do not have requirements

Description

This component runs its child components for each block in the current system, model, or signal that do not have associated requirements.

For more information on working with looping components, see"Logical and Looping Components".

Report On

This pane describes the type of object on which this component operates.

- **Automatic list from context**: Report on all blocks in the current context that do not have associated requirements. The parent component of the Block Loop component determines its context. If this component does not have the `Model Loop`, `System Loop`, `Signal Loop`, or `Block Loop` as its parent, selecting this option causes this component to report on all blocks in all models that do not have associated requirements.

  - `Model Loop`: Reports on all blocks in the current model with no associated requirements.
  - `System Loop`: Reports on all blocks in the current system with no associated requirements.
  - `Signal Loop`: Reports on all blocks connected to the current signal with no associated requirements.
- **Custom - use block list**: Enables you to specify a list of blocks on which to report. Enter the full path of each block.

Loop Options

Choose block sorting options and reporting options in this pane.

- **Sort blocks**:

Use this option to select how to sort blocks (applied to each level in a model):

- `Alphabetically by block name`: Sorts blocks alphabetically by their names.
- `Alphabetically by system name`: Sorts systems alphabetically. Lists the blocks in each system, but in no particular order.
- `Alphabetically by full Simulink path`: Sorts blocks alphabetically by Simulink path.
- `By block type`: Sorts blocks alphabetically by block type.
- `By block depth`: Sorts blocks by their depth in the model.
- `By layout (left to right)`: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- `By layout (top to bottom)`: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.
- `By traversal order`. Sorts blocks by traversal order.
- `By simulation order`. Sorts blocks by execution order.
- `%<VariableName>`: Inserts the value of a variable from the MATLAB workspace. The `%<>` notation can denote a string or cell array. The following example reports

on the `theta dot` integrator block and the `theta` integrator block in the model `simppend`, using the variable Z={ 'simppend/theta'}:

```
simppend/theta dot
%<Z>
```
The generated report includes information about the following blocks:

- `simppend/theta dot`
- `simppend/theta`

For more information, see `%<VariableName> Notation` on the `Text` component reference page in the MATLAB Report Generator documentation.

- **Search for Simulink property name/property value pairs**: Reports only on Simulink blocks with specified property name/property value pairs that do not have associated requirements.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each block found in the loop.
- **Display the object type in the section title**: Automatically inserts the object type into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each missing requirement in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class Name

`RptgenRMI.NoReqBlockLoop`

See Also

`Block Loop`, `Missing Requirements System Loop`, `Requirements Block Loop`, `Requirements Documents Table`, `Requirements Signal Loop`, `Requirements Summary Table`, `Requirements System Loop`, `Requirements Table`

MATLAB Code Traceability Table

Insert a table that links MATLAB code to requirements

Description

This component inserts a table into the report. The table links MATLAB code to corresponding requirements. This component reports on the currently open `.m` file. Place this component inside a section, paragraph, or table component.

To use this component, your report setup must include `Eval` statements that open a `.m` file or determine the `.m` file that is open. To open a template report that shows an example of these `Eval` statements, at the MATLAB command prompt, enter:

```
setedit([matlabroot '/toolbox/slrequirements/+rmiml/rmiml.rpt'])
```

Find the `Eval` statements in the `if` condition at the beginning of the report setup.

Table Options

Specify information about the table.

- **Table title**: Specify the table title.

  - `No title` — Do not include a table title.
  - `Object name` — Use the name of the `.m` file in the title.
  - `Custom` — Specify your own table title.

Table Columns

Specify the table columns that you want to include in the report. The **Document name**, **Locations within document**, and **Requirement keyword** check boxes correspond to properties on the Requirements Management Interface Link Editor dialog box.

- **Description** — Include the description of the requirement. The description helps you to identify the requirement the table is linking to. Leave this box selected to improve the readability of your table.

- **Document name** — Include the name of the document where the requirement is located.
- **Locations within document** — Include the identifier of a location in the document.
- **Requirement keyword** — Include the requirement keyword.

Insert Anything into Report?

Yes. Table.

Class

```
RptgenRMI.MlReqTable
```

See Also

```
Data Dictionary Traceability Table, Simulink Test Suite Traceability
Table, rmi
```

MATLAB Function

Insert information about MATLAB Function block contents

Description

This component displays tables with information about MATLAB code included in MATLAB Function blocks. You specify which of the following kinds of information to include in the report:

- Function properties — Parameter settings for the MATLAB Function block
- Argument properties — Properties of the function arguments (for example, complexity)
- The function script — MATLAB code of the function
- Function symbol data — Information about the user-defined and (optionally) built-in MATLAB variables and functions invoked by the MATLAB function that computes the block outputs.
- Supporting functions — User-defined functions and, optionally, MATLAB functions that are included in the MATLAB Function block function.

For details about MATLAB Function blocks, see the `MATLAB Function` block reference page.

Use the MATLAB Function component within a section, paragraph, or table.

Note: To view the contents of a MATLAB Function block in a Web viewer, use the Web view feature of the Simulink Report Generator. In the Web view, hover your cursor over the MATLAB Function block. For details, see "Model Web Views".

Function Properties Table

- **Include function properties**: Generates a table with function property information.
- **Table title**: Insert a title for the function properties table.

  - `Automatic`: Use the default title for the table.
  - `Custom`: Use the title that you specify for the table.

- You can change the header text for property and value columns of the function properties table. In the Header column, double-click to change the header text. The Width column indicates the relative width, in relative terms, based on the smallest width you specify. For example, for a three-column table, if the first column width is 1, and the column width of the other two columns is 3, then the second and third columns is three times wider than the first column.
- **Grid lines**: Show grid lines for the table.
- **Spans page width**: Make the table as wide as the page.

Argument Summary Table

- **Include argument summary table**: Generate a table with summary information about the MATLAB Function block function arguments.
- **Table title**: Insert a title for the argument summary table.

  - Automatic: Use the default title for the table.
  - Custom: Use the title that you specify for the table.

- **Argument Summary Table Options**: Specify the property columns to include in the table.

  - To add a property column:

    1 In the table on the right, select a property near where you want to insert the new property column.
    2 From the list of properties to the left of the table, select a property that you want to add to the table.
    3 Click the left-arrow button.
    4 If necessary, use the up or down arrow button to position the new column.

  - To delete a property column, select the property in the table and click the right-arrow button
  - You can change the header text for property and value columns of the table. In the Header column, double-click to change the header text. The Width column indicates the relative width, in relative terms, based on the smallest width you specify. For example, for a three-column table, if the first column width is 1, and the column width of the other two columns is 3, then the second and third columns is three times wider than the first column.

- **Grid lines**: Show grid lines for the table.
- **Spans page width**: Make the table as wide as the page.
- **Column alignment**: Align the text in each column:

  - `Left`
  - `Center`
  - `Right`
  - `Double justified`

Detailed Argument Report

- **Include detailed argument report**: Generate a table with detailed information about the MATLAB Function block function arguments.
- **Argument Property Table Format Options**: Specify the argument property columns to include in the table.

  - **Table title**: Insert a title for the argument properties table.

    - `Automatic`: Use the default title for the table.
    - `Custom`: Use the title that you specify for the table.

  - You can change the header text for property and value columns of the table. In the `Header` column, double-click to change the header text.
  - **Grid lines**: Show grid lines for the table.
  - **Spans page width**: Make the variable table as wide as the page on which the table appears.
- **Include function script**: Include the script for the function.
- **Include function symbol data**: Generate a table that includes information about the user-defined and (optionally) built-in MATLAB variables and functions invoked by the MATLAB function that computes the block outputs.
- **Highlight script syntax**: Use colors to highlight syntax keywords.
- **Include supporting functions**: Include a list of functions invoked directly or indirectly by the function script. If you specify to include supporting functions in the report, also specify whether to include both MATLAB and user-defined functions or just user-defined functions.
- **Supporting Function Table Format Options**:

- **Table title**: Insert a title for the supporting functions table.

  - `Automatic`: Use the default title for the table.
  - `Custom`: Use the title that you specify for the table.

- You can change the header text for property and value columns of the table. In the `Header` column, double-click to change the header text. The `Width` column indicates the relative width, in relative terms, based on the smallest width you specify. For example, for a three-column table, if the first column width is `1`, and the column width of the other two columns is `3`, then the second and third columns is three times wider than the first column.

- **Grid lines**: Show grid lines for the table.

- **Spans page width**: Make the table as wide as the page.

Insert Anything into Report?

Yes. Tables and, optionally, code.

Class

`rptgen_sl.csl_emlfcn`

See Also

`Stateflow Property`

Missing Requirements System Loop

Loop only on systems and subsystems that do not have associated requirements

Description

This component runs its child components for each system or subsystem defined by the parent component that does not have associated requirements. Insert this component as the child of a `Model Loop` component to include systems and subsystems that do not have any associated requirements in the report.

Report On

- **Loop on Systems**:

  - **Select systems automatically**: Reports on all systems in the current context that do not have associated requirements.

    - `Model Loop`: Reports on systems in the current model.
    - `System Loop`: Reports on the current system.
    - `Signal Loop`: Reports on the parent system of the current signal.
    - `Block Loop`: Reports on the parent system of the current block.

    If this component does not have any of these components as its parent, selecting this option reports on all systems in all models that do not have associated requirements.

- **Custom - use system list**: Reports on a list of specified systems. Specify the full path of each system.

- `%<VariableName>`: Inserts the value of a variable from the MATLAB workspace. The `%<>` notation can denote a string or cell array. For more information, see `%<VariableName> Notation` on the `Text` component reference page.

Loop Options

- **Sort Systems**: Specifies how to sort systems.

- `Alphabetically by system name` (default): Sorts systems alphabetically by name.
- `By number of blocks in system`: Sorts systems by number of blocks. The list shows systems by decreasing number of blocks. In other words, it shows the system with the largest number of blocks that do not have requirements appears first in the list.
- `By system depth`: Sorts systems by their depth in the model.
- `By traversal order`: Sorts systems in the traversal order.

- **Search for**: Reports only on blocks with the specified property name/property value pairs. To enable searching, click the check box. In the first row of the property name and property value table, click inside the edit box, delete the existing text, and type the property name and value. To add a row, use the **Add row** button.

  For information about subsystem property names and values, in "Block-Specific Parameters", see the "Ports & Subsystems Library Block Parameters" section.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Number sections by system hierarchy**: Hierarchically numbers sections in the generated report. Requires that **Sort Systems** be set to `By traversal order`.
- **Create link anchor for each object in loop**: Create a link target for each missing requirement in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

`RptgenRMI.NoReqSystemLoop`

See Also

`Block Loop`, `Missing Requirements Block Loop`, `Requirements Block Loop`, `Requirements Documents Table`, `Requirements Signal Loop`, `Requirements Summary Table`, `Requirements System Loop`, `Requirements Table`, `System Loop`

Model Advisor Report

Insert Model Advisor report or link to Model Advisor report for current model

Description

This component inserts a Model Advisor report for the current model into the report if the report is in HTML format. For other report formats, it inserts a link to a Model Advisor report for the current model. For more information about Model Advisor reports, see "Save and View Model Advisor Reports" in the Simulink documentation.

Properties

Use existing report: Includes an existing Model Advisor report in the report. This check box is selected by default. Clearing this option generates a new Model Advisor report.

Insert Anything into Report?

Yes, a Model Advisor report.

Class

rptgen_sl.CModelAdvisor

See Also

Model Change Log

Model Change Log

Construct model history table that displays model revision information

Description

Run this component before you run the `Model Simulation` component. It uses a reported model's `ModifiedHistory` parameter to construct a model history table that displays information about each logged revision to the model. This model history table includes:

- The author of each change
- The model version of the change
- The time and date of the change
- A description of the change

For more information on model history, see "Log Comments History" in the Simulink documentation.

Tip If your model has a long revision history, consider limiting the number of revisions reported.

Table Columns

Choose the information displayed in the model revision table in this section:

- **Author name**: Includes the name of the person who last revised the model.
- **Version**: Includes the version number of the model.
- **Date changed**: Includes the revision date of the model.
- **Description of change**: Includes a description of the revision to the model.

Table Rows

- **Limit displayed revisions to**: Limits the number of revisions that appears in the report.

- **Show revisions since date**: Limits the number of revisions that appears in the report by date. Enter the date in the corresponding text field. This field supports `%<varname>` notation. For example, the default value, `%<datestr(now-14)>`, returns revision history for the last two weeks.

Table Display

Choose how the model revision history table appears in this section.

- **Table title**: Specifies the title of the table.
- **Sort order**: Sorts the table entries from most recent to oldest, or from oldest to most recent.
- **Date format**: Specifies a preferred date format for the date/time stamps in the table.

Insert Anything into Report?

Yes. Table.

Class

`rptgen_sl.csl_mdl_changelog`

See Also

`Model Advisor Report`

Model Configuration Set

Insert active configuration set of a model into a report

Description

This component displays a table with the active configuration set for the model.

For information about configurations sets, see "Manage a Configuration Set".

Display Options

- **Title**: Specifies a title for the table in the generated report.

  - `Automatic`: Generates a title automatically from the parameter.
  - `Custom`: Specifies a custom title.
  - `None`: Uses no title.
- **Show configuration set table grids**: Show grid lines for the table.
- **Make configuration set tables page wide**: Make the table as wide as the page.

Insert Anything into Report?

Yes. Table.

Class

`rptgen_sl.csl_mdl_cfgset`

See Also

`Model Loop`, `System Loop`

Model Loop

Loop on Simulink models and systems, as specified by child components

Description

This component loops on Simulink models and systems, as specified by child components. For example, you can use a Model Loop with a child System Loop to report on the subsystems of the specified system.

Consider making these components children of the Model Loop (although the Model Loop not necessarily required to be the immediate parent of a given component).

For conditional processing based of blocks, you can use the RptgenSL.getReportedBlock function. For more information, see "Loop Context Functions" on page 4-55.

Models to Include

You can add a model to the list by clicking **Add New Model to List**. The following table shows the buttons you can use to move a model up or down in the list, or to add or delete a model.

Button	Action
⬆	Move a model up in the list.
⬇	Move a model down in the list.
✖	Remove a model from the list.
🔧	Add a new model to the list.

Model Options

- **Active**: Includes a given model in the loop. This option is selected by default. Clearing this option omits the model from the loop.

  This option allows you to temporarily omit one or more models from a report.

- **Model name**: Specifies the model name.

  - `Current block diagram`
  - `All open models`
  - `All open libraries`
  - `Block diagrams in current directory`
  - `Custom block diagram`: Selecting this option automatically sets the **Starting system(s)** field `$top` to start in the model root system.
  - `%<VariableName>`: For more information, see `%<VariableName>` Notation on the `Text` component reference page in the MATLAB Report Generator documentation.

- **Traverse model**: Specifies the systems to traverse.

  - `All systems in model`
  - `Selected system(s) only`
  - `Selected system(s) and ancestors`
  - `Selected system(s) and children`

- **Look under masks**: Specifies how to handle masks.

  - `Functional masks only`
  - `No masks`
  - `All masks`
  - `Graphical masks only`

  For more information, see "Block Masks" in the Simulink documentation.

- **Follow library links**: Specifies library links to include.

  - `Do not follow library links`: Library links are treated as blocks.
  - `Include library links`: Library links are treated as subsystems.

- `Include unique library links`: With multiple copies of the same library link in a system, one is treated as a subsystem and the others as blocks.

  For more information, see "Linked Blocks" in the Simulink documentation.

- **Model reference**: Specifies whether to report on models referenced by a Model block. If you want to report on referenced models, then you can control the depth of the model referencing hierarchy and whether to report on model reference variants.

  - `Do not follow model reference blocks`: Do not report on blocks contained in referenced models.
  - `Follow all model reference blocks`: Report on blocks contained in all models that any part of the model hierarchy references.
  - `Follow model reference blocks defined in current model`: Report on blocks in models that the currently selected model references.
  - `<Custom model reference depth>`: Report on blocks in models that your specified level in the model reference hierarchy references.

- **Include all variants**: Report on all model reference variants. To enable this option, set the **Model reference** option to report on blocks in referenced models.

- **Starting system(s)**: Specifies the system in which to start the loop. Available options depend on the value that you select in the **Traverse model** option. Selecting any option other than `All systems in model` for **Traverse model** activates the **Starting system(s)** option.

  If you do not enter a model name in the **Model name** option, then select either `Root model` or `Current` to specify where to start the loop.

  If you specify a model name in the **Model name** option, then the **Starting system(s)** option provides an edit box in which you can enter:

  - The full path of a subsystem or subsystems
  - `$top` to start the loop in the model root system
  - `$current` to start the loop in the currently selected system

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.

- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each model in the loop so that other parts of the report can link to it.

Examples

Generating Reports on Specified Systems and their Subsystems

This example shows how to loop over a specified system and its subsystems in the sample model `sldemo_auto_climate_elec`, which the Simulink software includes.

1   (Optional) To open the `sldemo_auto_climate_elec` model, at the MATLAB command prompt, enter the following command:

`sldemo_auto_climate_elec`

Explore the model to familiarize yourself with its subsystems.

2   Open the Report Explorer.

3   Create a report setup file by clicking **File** > **New**.

4   Save the report setup file by clicking **File** > **Save As**. Give it the name `sldemo_auto_report`.

5   Add a `Chapter/Subsection` component to the report setup file to include information about model subsystems:

   a   In the Library pane in the middle, double-click `Chapter/Subsection` to add it to the report setup file.

   b   For **Title**, choose `Custom`. In the title field, enter `Description of subsystems`.

6   Add a `Model Loop` as a child of the `Chapter/Subsection` component. This loops over the `ClimateControlSystem` system and its subsystems in the `sldemo_auto_climate_elec` model:

   a   In the Library pane in the middle, double-click `Model Loop` to add it to the report setup file. By default, the Report Explorer adds that component as a child of the `Chapter/Subsection` component.

   b   In the Model Loop properties pane, from the **Model name** selection list, select `<Custom block diagram>` .

7-65

c In the **Model name** field, delete the text `<Custom block diagram>`, and then enter `sldemo_auto_climate_elec.slx`. Click any component in the report setup file to add this model to the **Models to include** list.

d In the **Traverse model** selection list, select `Selected system(s) and children`.

e In the **Look under masks** selection list, select `All masks`.

f In the **Model reference** selection list, select `Do not follow model reference blocks`.

g In the **Starting system(s)** field, enter `sldemo_auto_climate_elec/ClimateControlSystem`. Because you selected `Selected system(s) and children` for **Traverse model**, the `Model Loop` loops over `sldemo_auto_climate_elec/ClimateControlSystem` and its subsystems.

h Under **Section Options**, select the **Create section for each object in loop** check box. Selecting this option creates separate sections in the generated report for each model over which the component loops.

The `Model Loop` properties pane looks as follows.
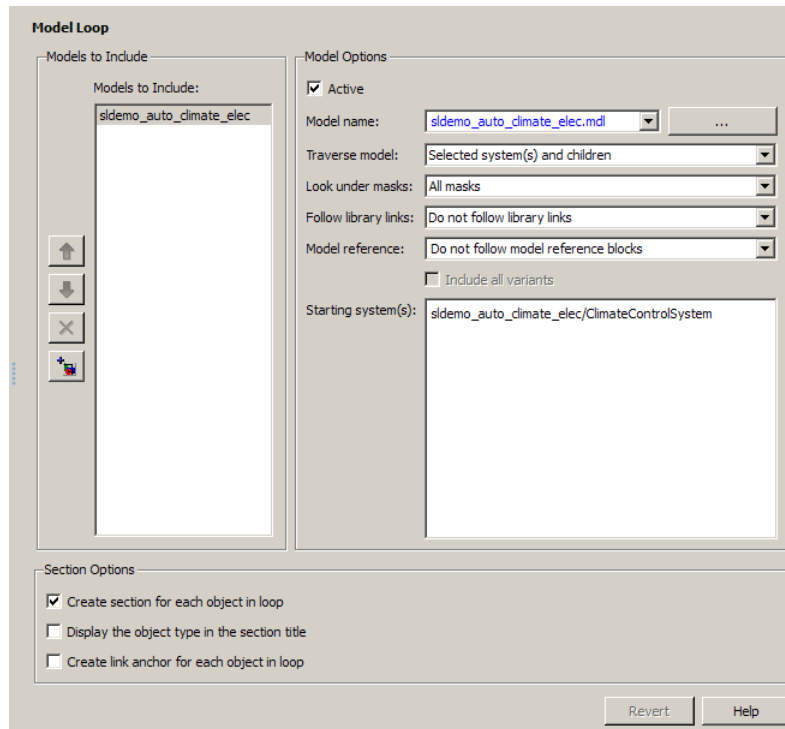
7   Save the report by clicking **File** > **Save**.

8   Add a `System Loop` as a child of the `Model Loop` component.

   a   In the Library pane in the middle, double-click `System Loop` to add it to the report setup file. By default, Model Explorer adds this component as a child of the `Model Loop` component.

   b   In the `System Loop` properties pane, under **Section Options**, select the **Create section for each object in loop** check box. Selecting this option creates a section in the generated report for each subsystem on which the component loops. Accept the default values for all other fields.

9   Add a `System Snapshot` component as a child of the `System Loop` component. This step creates snapshots of all the subsystems of `ClimateControlSystem` in the generated report. In the Library pane in the middle, double-click `System Snapshot`. By default, Model Explorer adds this component as a child of the `System Loop` component.

10 Save the report.

The report setup file hierarchy now looks as follows.
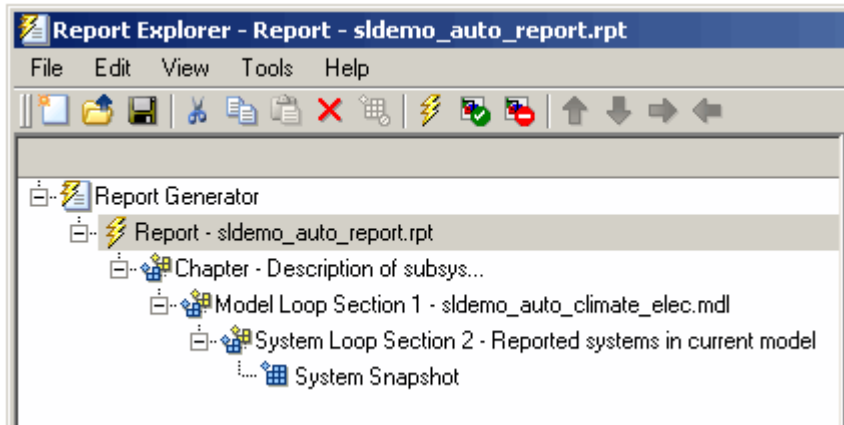
11 Run the report by clicking **File** > **Report**.

The report loops on the system `ClimateControlSystem` of the
`sldemo_auto_climate_elec` model and all of its subsystems, as shown in the
following Message List.
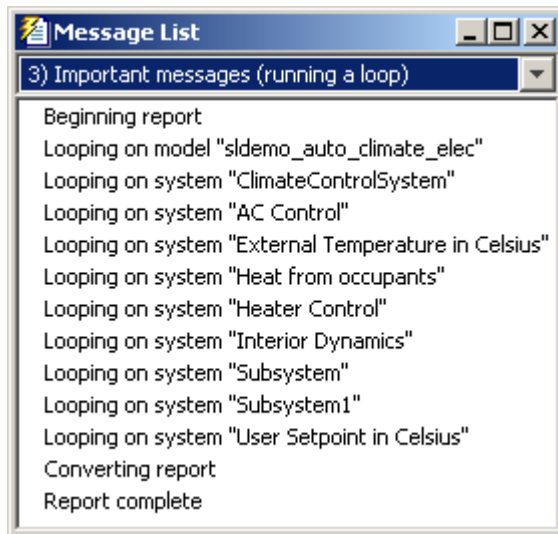
Below is an excerpt from the generated report.

Chapter 1. Description of subsystems

Table of Contents

sldemo_auto_climate_elec

AC Control
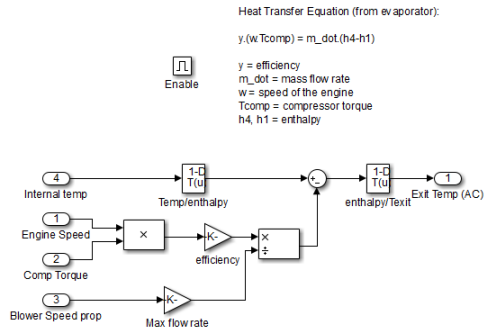
ClimateControlSystem
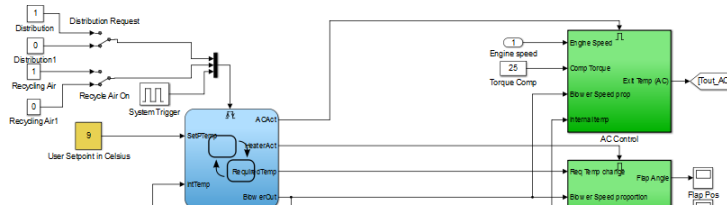
Temporarily Omitting a Model from a Loop

This example shows how to use the `Model Loop` **Active** check box to temporarily omit a model from the loop. This example uses the report setup file that you created in Generating Reports on Specified Systems and their Subsystems, `sldemo_auto_report.rpt`, and the model `f14`, which the Simulink software includes.

1   In the Report Explorer, click **File** > **Open**, and then open `sldemo_auto_report.rpt` by double-clicking it.

2   In the Outline pane on the left, click `Model Loop Section 1 -`
`sldemo_auto_climate_elec`.

3
In the `Model Loop` properties pane, click the ![button icon] button to add a model to the
Models to include list.

4   In the `Model Loop` properties pane, from the **Model name** selection list, select
`<Custom block diagram>`.

5   In the **Model name** field, delete the text `<Custom block diagram>` and enter
`f14.mdl`.

The `Model Loop` properties pane now looks as follows.
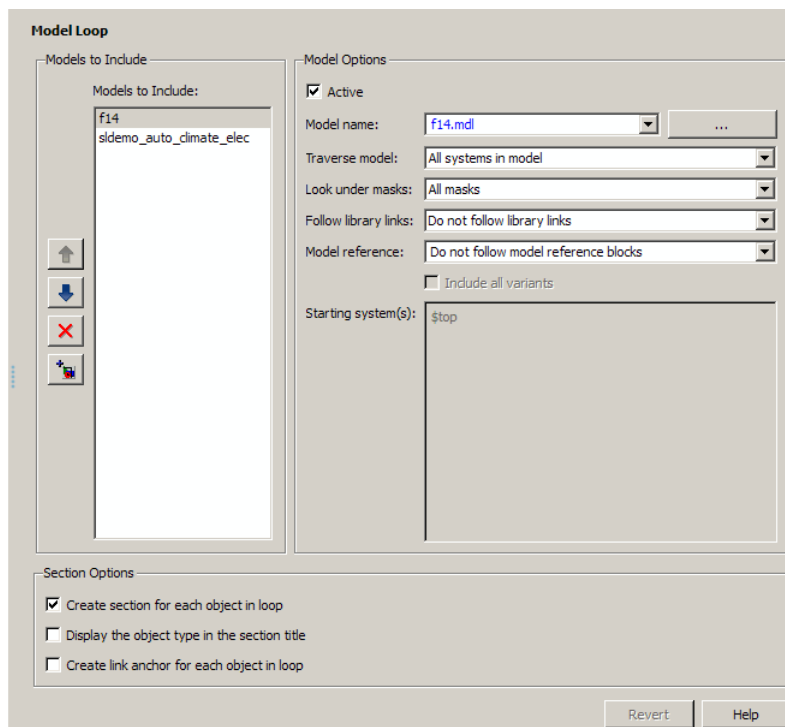
6   Save the report setup file.

7   Generate the report.

The report generation process loops over the specified systems in the `f14` and `sldemo_auto_climate_elec` models, as shown in the following message box.
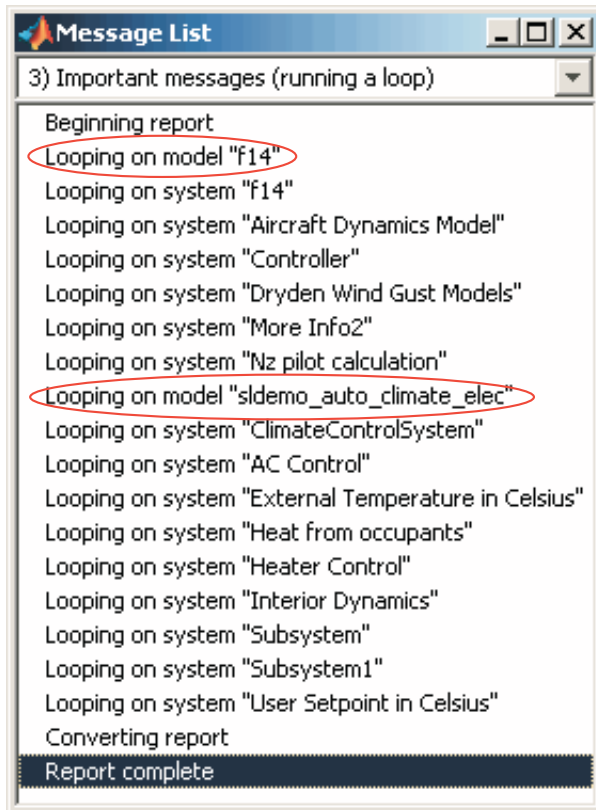
Below is an excerpt from the generated report.

Chapter 1. Description of subsystems

Table of Contents

f14

Aircraft Dynamics Model

Controller

8 In the **Models to include** list, click f14 to select it.

9 Clear the **Active** check box to omit f14 model information from the generated report.

10 Rerun the report.

The report now includes information only on the sldemo_auto_climate_elec model, as shown at the end of the previous example, Generating Reports on Specified Systems and their Subsystems.

11 To reactivate the f14 model, in the Model Loop **Models to include** list, select the f14 model and then select the **Active** check box.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

rptgen_sl.csl_mdl_loop

See Also

Block Loop, System Loop

Model Simulation

Run current model with specified simulation parameters

Description

This component runs the current model using specified simulation parameters. Ensure that this component has the Model Loop component as its parent.

For more information on simulation parameters, see "Configure Simulation Conditions" in the Simulink documentation.

I/O Parameters

Use model's workspace I/O variable names

Use the names of the parameters specified in the Simulation Parameters dialog box.

The following options are available if you do not select the **Use model's workspace I/O variable names** option:

- **Time** : Specifies a new variable name for the Time parameter.
- **States**: Specifies a new variable name for the States parameter.
- **Output**: Specifies a new variable name for the Output parameter.

Timespan

Use model's timespan values: Use the model's Start time and Stop time values, as specified in the **Solver** tab in the Simulation Parameters dialog box.

The following options are available if you do not select the **Use model's timespan values** option:

- **Start**: Specifies a simulation starting time.
- **Stop**: Specifies a simulation ending time.

Note: If you set the stop time of your model to `inf` (infinity) in Simulink or on this component attribute page, Simulink Report Generator terminates the model simulation after 60 seconds. Terminating the report prevents the report generation process from entering an infinite loop.

Simulation Options

- **Compile model before simulation**: Compiles the model before simulating, preserving scope content. Select this option if:

  - You use Simulink Coder Summary properties.
  - You sort systems or blocks by simulation order.
  - You use scope snapshots.

- **Simulation status messages**: Displays simulation status messages, or inserts them into the report.

  - `Display to command line`: Sends messages to a command-line window.
  - `Display to Report Generator Message List`: Sends messages to the Simulink Report Generator message window.
  - `Insert into report`: Includes messages in the report.

- **Simulation parameters**: Specifies simulation parameters.

Insert Anything into Report?

No.

Class

rptgen_sl.csl_mdl_sim

See Also

Model Loop

Object Loop

Run child components for Stateflow objects, and then insert table into report

Description

This component runs its child components for each Stateflow object and inserts a table into the generated report.

For conditional processing of Stateflow objects, you can use the `RptgenSF.getReportedObject` function. For more information, see "Loop Context Functions" on page 4-55.

Object Types

- **Report on "Data" objects**: Includes Stateflow data objects in the loop.
- **Report on "Event" objects**: Includes Stateflow event objects in the loop.
- **Report on "Transition" objects**: Includes Stateflow transition objects in the loop.
- **Report on "Junction" objects**: Includes Stateflow junction objects in the loop.
- **Report on "Target" objects**: Includes Stateflow target objects in the loop.
- **Report on "Annotation" objects**: Includes Stateflow note objects in the loop.

Loop Options

- **Report depth**: Specifies the level at which to loop:

  - `Local children only` (Default). Reports only on children one level down.
  - `All objects`. Reports on all Stateflow objects.
- **Skip autogenerated charts under truth tables**: Excludes autogenerated charts under truth tables from the report.
- **Remove objects which do not contain more information than a snapshot**: Excludes objects that contain only a snapshot.
- **Search Stateflow**: Reports on Stateflow charts with specified property name/ property value pairs.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Automatically inserts the object type into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each Stateflow object in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

```
rptgen_sf.csf_obj_loop
```

See Also

```
Stateflow Filter, Stateflow Hierarchy, Stateflow Hierarchy Loop,
Simulink Function System Loop
```

Requirements Block Loop

Apply child components to blocks with requirements

Description

This component applies its child components to blocks with associated requirements.

Report On

- **Automatic list from context**: If selected, this option reports on all blocks in the current context. The parent of the `Requirements Block Loop` component determines its context.

  - `Model Loop`: Reports on all blocks with requirements in the current model.
  - `System Loop`: Reports on all blocks with requirements in the current system.
  - `Signal Loop`: Reports on all blocks with requirements connected to the current signal.

  If the `Requirements Block Loop` does not have the `Model Loop`, `System Loop`, `Signal Loop`, or `Block Loop` component as its parent, it reports on all blocks in all models.

- **Custom - use block list**: Reports on a list of blocks with specified requirements. Enter the full paths of each block into this field.

Loop Options

- **Sort blocks**

  Specify how to sort blocks (applied to each level in a model):

  - `Alphabetically by block name`: Sorts blocks alphabetically by name.
  - `Alphabetically by system name`: Sorts systems and subsystems alphabetically by name. (Blocks in each system do not appear in alphabetical order).

- **Alphabetically by full Simulink path**: Sorts blocks alphabetically by Simulink path.

- **By block type**: Sorts blocks alphabetically by block type.

- **By block depth**: Sorts blocks by their depth in the model.

- **By layout (left to right)**: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- **By layout (top to bottom)**: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- **By traversal order**: Sorts blocks by traversal order.

- **By simulation order**: Sorts blocks by execution order.

- **Search for Simulink property name/property value pairs**: Reports on Simulink blocks with specified property name/property value pairs that have associated requirements.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each block found in the loop that has associated requirements.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each requirement in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

RptgenRMI.CBlockLoop

See Also

Missing Requirements Block Loop, Missing Requirements System Loop, Model Loop, Requirements Block Loop, Requirements Documents Table, Requirements Signal Loop, Requirements Summary Table, Requirements System Loop, Requirements Table

Requirements Documents Table

Insert table of linked requirements documents

Description

This component creates a table that lists all requirements documents linked to model objects.

Table Options

- **Show documents linked to**

    - `Simulink and Stateflow objects`: Inserts requirements documents linked to both Simulink and Stateflow objects in the model.
    - `Simulink objects`: Inserts requirements documents linked only to Simulink objects in the model.
    - `Stateflow objects`: Inserts requirements documents linked only to Stateflow objects in the model.

- **Table title**: Specifies a title for the table.

    - `No title`
    - `Model name` (Default)
    - `Custom`

Table Columns

- **Replace document paths with links**: Inserts links to requirements documents when possible.

- **When replacing with links, note absolute vs. relative file path**: Indicates absolute or relative file paths when including links to requirements documents.

- **Include document modification time**: Includes the document modification information.

- **Count # references to each document**: Includes a count of the number of references to the requirements document in the model.

Document References

- **Replace file names with document IDs in the main body of the report**: Includes shortened IDs to identify requirements documents to simplify the requirements documents table.
- **Retrieve full module path for DOORS links (requires login)**: This option applies only to DOORS® requirements. Append the DOORS module ID to the module path in the DOORS database if the module information is not stored with the model.

Insert Anything into Report?

Yes. Table.

Class

RptgenRMI.ReqDocTable

See Also

, Requirements Summary Table, Requirements Table

Requirements Signal Loop

Apply all child components to signal groups with requirements

Description

The Requirements Signal Loop component applies all child components to signal groups that have requirements in Signal Builder blocks.

Properties

- **Create link anchor for each object in loop**: Create a link target for each requirement in the loop so that other parts of the report can link to it.
- **Display the object type in the section title**: Inserts the object name with requirements into the section title.
- **Create section for each object in loop**: Creates a hyperlink to each object with requirements in the loop.
- **Section Type**: Specifies the section type to insert. If you choose `Automatic`, the Simulink Report Generator software determines the appropriate section type:

  - `Book`
  - `Chapter`
  - `Section 1`
  - `Section 2`
  - `Section 3`
  - `Section 4`
  - `Section 5`
  - `Simple Section`
  - `Automatic`

Report On

Loops on signal groups in systems:

- `Collect all Signal Builders`: Processes all Signal Builder blocks, looking for signal groups with requirements.
- `Custom - use list`: Processes all subsystems in the user-defined list. If a subsystem on the list does not have requirements, the Simulink Report Generator software does not include it in the report.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

`RptgenRMI.CSystemLoop`

See Also

`Missing Requirements Block Loop`, `Missing Requirements System Loop`, `Requirements Block Loop`, `Requirements Documents Table`, `Requirements Summary Table`, `Requirements System Loop`, `Requirements Table`, `Signal Loop`

Requirements Summary Table

Properties of blocks, systems, or Stateflow objects with associated requirements

Description

This component displays properties of blocks, systems, or Stateflow objects with associated requirements.

Object Type

Choose the object type to display in the generated report.

- `Block` (Default)
- `System`
- `Stateflow`

The selected object type affects the options available in the **Property Columns** pane.

Table Title

Specify a table title in the generated report.

- `Automatic`: Generates a title automatically from the parameter.
- `Custom`: Specifies a custom title.

Property Columns

- Object properties to include in the Requirements Summary Table appear in a list.

    - To add a property:
        1. Select the appropriate property level in the text box on the left.
        2. In the text box on the right, select the property that you want to add and click **Add**.
    - To delete a property, select the property name and click **Delete**.

%<SplitDialogParameters> is a unique property that you can specify for Requirements Summary Tables where the object type is Block. This property generates multiple summary tables, grouped by block type. Each Summary Table group contains the dialog box parameters for that block.

Some entries in the list of available properties (such as Depth) are "virtual" properties that you cannot access using the get_param command. The properties used for property/value filtering in the block and System Loop components must be retrievable by the get_param. Therefore, you cannot configure your Requirements Summary Table to report on all blocks of Depth == 2.

- **Remove empty columns**: Removes empty columns from the table.
- **Transpose table**: Changes the summary table rows into columns in the generated report, putting the property names in the first column and the values in the other columns.

Object Rows

- **Insert anchor for each row**: Inserts an anchor for each row in the Requirements Summary Table.
- **Report On**

    - Automatic list from context: Reports on all blocks in the current context. The parent of this component determines its context.
    - Custom - use block list: Reports on a list of blocks that you specify, and enters the block names in the corresponding field. Specify the full path of each block.

Loop Options

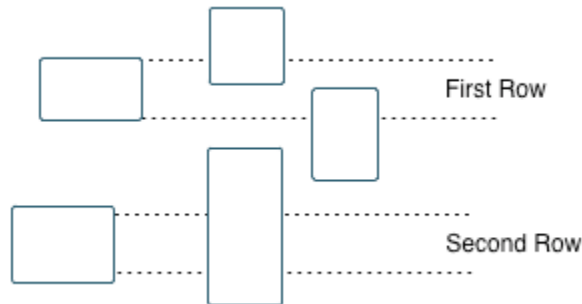Choose block sorting options and reporting options in this pane.

- **Sort blocks**: Use this option to select how to sort blocks (applied to each level in a model):

    - Alphabetically by block name: Sorts blocks alphabetically by name.
    - Alphabetically by system name. Sorts systems alphabetically. Lists blocks in each system, but in no particular order.

- **Alphabetically by full Simulink path**: Sorts blocks alphabetically by Simulink path.

- **By block type**: Sorts blocks alphabetically by block type.

- **By block depth**: Sorts blocks by their depth in the model.

- **By layout (left to right)**: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- **By layout (top to bottom)**: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- **By traversal order**: Sorts blocks by traversal order.

- **By simulation order**: Sorts blocks by execution order.

- **Search for Simulink property name/property value pairs**: Reports on blocks with specified property name/property value pairs.

Insert Anything into Report?

Yes. Table.

Class

`RptgenRMI.CSummaryTable`

See Also

`Block Loop`, `Missing Requirements Block Loop`, `Missing Requirements System Loop`, `Requirements Block Loop`, `Requirements Documents Table`, `Requirements Signal Loop`, `Requirements System Loop`, `Requirements Table`

Requirements System Loop

Apply child components to systems with requirements

Description

This component applies its child components to systems with associated requirements.

Report On

- **Loop on systems**

    - `Select systems automatically`: If selected, this option reports on all systems in the current context. The parent of the component determines the context of this setting:

        - `Model Loop`: Reports on systems in the current model.
        - `System Loop`: Reports on the current system.
        - `Signal Loop`: Reports on the parent system of the current signal.
        - `Block Loop`: Reports on the parent system of the current block.

        If the `Requirement System Loop` does not have any of these components as its parent, selecting this option reports on all systems with requirements in all models.

    - `Custom - use system list`: Reports on a list of specified systems. Enter the full path of each system.

Loop Options

- **Sort Systems**:

    - `Alphabetically by system name` (default): Sorts systems alphabetically by name.
    - `By number of blocks in system`: Sorts systems by the number of blocks in the system. The list displays systems by decreasing number of blocks; the system with the largest number of blocks appears first in the list.

- By `system depth`: Sorts systems by their depth in the model.
  - By `traversal order`: Sorts systems in the traversal order .
- **Search for**: Reports on Simulink blocks with specified property name/property value pairs.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Number sections by system hierarchy**: Numbers sections in the generated report hierarchically. Requires that **Sort Systems** be set to `By traversal order`.
- **Create link anchor for each object in loop**: Create a link target for each requirement in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

RptgenRMI.CSystemLoop

See Also

Missing Requirements Block Loop, Missing Requirements System Loop, Requirements Block Loop, Requirements Documents Table, Requirements Signal Loop, Requirements Summary Table, Requirements Table, System Loop

Requirements Table

Requirements links for current context

Description

This component creates a table that contains information from the Simulink Verification and Validation software. Objects can have multiple requirements. Each requirement is a row in the table.

Table Options

- **Show requirements for current**: Specifies the object type to display.

  - `Simulink object`
  - `Stateflow object`
- **Table title**: Specifies a title for the table.

  - `No title`
  - `Object name` (Default)
  - `Custom`

Table Columns

- **Description**: Includes the object description in the table.
- **Document name**: Includes the report name in the table.
- **Locations within document**: Includes the locations of the object within the document in the table.
- **Requirement keyword**: Includes the requirement keyword for the object in the table.

Insert Anything into Report?

Yes. Table.

Class

RptgenRMI.CReqTable

See Also

Missing Requirements Block Loop, Missing Requirements System Loop, Requirements Block Loop, Requirements Documents Table, Requirements Signal Loop, Requirements Summary Table, Requirements System Loop, Stateflow Automatic Table, Stateflow Name

Scope Snapshot

Insert images of scopes and XY graphs

Description

This component inserts images of scopes and XY graphs. Examples of blocks for which this component inserts snapshots include:

- Scope (and Floating Scope) blocks and the XY Graph block (Simulink)
- 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)

If the model has not been simulated, scopes are empty. For more information, see the `Model Simulation` component reference page.

The parent component of the Scope Snapshot determines its behavior.

- `Model Loop` or `no Simulink looping component`: Includes all XY graphs and scopes in the current model.
- `System Loop`: Includes all XY graphs and scopes in the current system.
- `Block Loop`: Includes the current block when it is an XY graph or scope.
- `Signal Loop`: Includes all XY graphs and scopes connected to the current signal.

  If the Scope Snapshot does not have any of the Simulink looping components as its parent, it includes all XY graphs and scopes in all open models.

Scope Options

- **Report on closed scopes**: Takes a snapshot of all scopes in context. This option forces closed scopes to open when the report is generating.
- **Autoscale time axis**: Scales the Simulink scope time axis to include the entire log.

Print Options

- **Image file format**: Specifies the image file format (for example, JPEG, TIFF, etc.). Select `Automatic HG Format` (the default) to choose the format best suited for the specified output format automatically. Otherwise, choose an image format that your output viewer can read.

  - `Automatic HG Format` (uses the Simulink file format selected in the Preferences dialog box)
  - `Bitmap (16m-color)`
  - `Bitmap (256-color)`
  - `Black and white encapsulated PostScript`
  - `Black and white encapsulated PostScript (TIFF)`
  - `Black and white encapsulated PostScript2`
  - `Black and white encapsulated PostScript2 (TIFF)`
  - `Black and white PostScript`
  - `Black and white PostScript2`
  - `Color encapsulated PostScript`
  - `Color encapsulated PostScript (TIFF)`
  - `Color encapsulated PostScript2`
  - `Color encapsulated PostScript2 (TIFF)`
  - `Color PostScript`
  - `Color PostScript2`
  - `JPEG high quality image`
  - `JPEG medium quality image`
  - `JPEG low quality image`
  - `PNG 24-bit image`
  - `TIFF - compressed`
  - `TIFF - uncompressed`
  - `Windows metafile`

- **Paper orientation**:

- Landscape
- Portrait
- Rotated
- `Use figure orientation`: Uses the orientation for the figure, which you set with the `orient` command.
- `Full page image (PDF only)`: In PDF reports, scales images to fit the full page, minimizes page margins, and maximizes the size of the image by using either a portrait or landscape orientation.

For more information about paper orientation, see the `orient` reference page in the MATLAB documentation.

- **Image size**: Specifies the size of the Handle Graphics figure snapshot in the form `[w h]` (width, height). In the units text box, select one of the following options:

  - Inches
  - Centimeters
  - Points
  - Normalized

- **Invert hardcopy**: Inverts colors for printing; changes dark colors to light colors and light colors to dark colors.

  - `Automatic`: Automatically changes dark axes colors to light axes colors. If the axes color is a light color, this option does not invert the color.
  - `Invert`: Changes dark axes colors to light axes colors and light axes colors to dark axes colors.
  - `Don't invert`: Does not change the colors in the image on the screen for printing.
  - `Use figure's InvertHardcopy setting`: Uses the `InvertHardcopy` property set in the Handle Graphics image.
  - `Make figure background transparent`: Makes the image background transparent.

Display Options

- **Scaling**: Controls size of the image, as displayed in a browser. Making an image larger using this option does not affect the storage size of the image, but the quality of the displayed image may decrease as you increase or decrease the size of the displayed image.

  Generally, to achieve the best and most predictable display results, use the default setting of Use image size.

  - Use image size: Causes the image to appear the same size in the report as on screen (default).
  - Fixed size: Specifies the number and type of units.
  - Zoom: Specifies the percentage, maximum size, and units of measure.
- **Size**: Specifies the size of the snapshot in the form w h (width, height). This field is active only if you choose Fixed size from the **Scaling** selection list.
- **Max size**: Specifies the maximum size of the snapshot in the form w h (width, height). This field is active only if you choose Zoom from the **Scaling** selection list.
- **Units**: Specifies the units for the size of the snapshot. This field is active only if you choose Zoom or Fixed size in the **Image size** list box.
- **Alignment**: Only reports in PDF or RTF format support this property.

  - Auto
  - Right
  - Left
  - Center
- **Title**: Specifies a title for the snapshot figure.

  - Block name: Uses the block name as the title.
  - Full Simulink path name: Uses the Simulink path as the title.
  - Custom: Specifies a custom title.
- **Caption**: Select or enter a short text description for the snapshot figure.

  - No caption
  - Automatic (use block description). Uses the Simulink block description as the caption.

- Custom. Specifies a short text description for the snapshot figure.

Insert Anything into Report?

Yes. Image.

Class

rptgen_sl.csl_blk_scope

See Also

Block Loop, Model Loop, Signal Loop, System Loop

Signal Loop

Run child components for each signal contained in current system, model, or block

Description

The Signal Loop component runs its child components for each signal contained in the current system, model, or block. The parent component determines the behavior of this component.

- `Model Loop`: Loops on all signals in the current model.
- `System Loop`: Loops on all signals in the current system. Choose not to report on the following types of signals by clearing the corresponding option in the **Section options** area:

  - `System input signals`
  - `System output signals`
  - `System internal signals`
- `Signal Loop`: Loops on the current signal.
- `Block Loop` : Loops on all signals connected to the current block. Choose not to report on the following types of signals by clearing the corresponding option in the **Section options** area:

  - `Block input signals`
  - `Block output signals`
- If the Signal Loop does not have a looping component as its parent, it loops on all signals in all models. Choose not to report on the following types of signals by clearing the corresponding option in the **Section options** area:

  - `Block input signals`
  - `Block output signals`
  - `System input signals`
  - `System output signals`
  - `System internal signals`

For conditional processing of signals, you can use the `RptgenSL.getReportedSignal` function. For more information, see "Loop Context Functions" on page 4-55.

Select Signals

- **Include block input signals**: Loops on signals that feed into blocks. This option is valid only when the parent component of this component is a `Block Loop`.

- **Include block output signals**: Loops on signals that leave the block. This option is valid only when the parent component of this component is a `Block Loop`.

- **Include system input signals**: Loops on signals coming from inports. This option is valid only when the parent component of this component is a `System Loop`.

- **Include system internal signals**: Loops on system internal signals. This option is valid only when the parent component of this component is a `System Loop`.

- **Include system output signals**: Loops on signals going to outports. This option is valid only when the parent component of this component is a `System Loop`.

- **Sort signals**: Specifies how to sort signals:

  - `Alphabetically by signal name`: Sorts signals alphabetically by name.

  - `Alphabetically by signal name (exclude empty)`: Sorts signals alphabetically by name.

  - `Alphabetically by system name`: Sorts alphabetically by parent system names. Lists signals in each system, but in no particular order.

  - `By signal depth`: Sorts signals by their depth in the model.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.

- **Display the object type in the section title**: Automatically inserts the object type into the section title in the generated report.

- **Create link anchor for each object in loop**: Create a link target for each signal in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

rptgen_sl.csl_sig_loop

See Also

Block Loop, Model Loop, System Loop

Simulink Automatic Table

Insert two-column table with information on selected model, system, signal, or block

Description

This component inserts a two-column table that contains details for the selected model, system, signal, or block into a generated report.

Options

- **Show current**: Modeling object to specify properties for.

    - `Automatic`: Uses the context of the parent loop.
    - `Model`
    - `System`
    - `Block`
    - `Annotation`
- **Properties list**: Specifies whether to have Report Explorer select properties automatically or to list the properties to report on.

    - `Determine properties automatically`: Let the Report Explorer automatically select the properties to report.

Modeling Component Selected in the Show current Field	Listed Properties
`Model`	`Description`
`System`	`Description`
`Block`	Block parameter dialog box prompt properties
`Annotation`	`Text`
`Signal`	`Description`

    - `Show properties`: Specify a list of properties to report. Enter the names of object properties that you want the report to include for the modeling object you specified

in the **Show current** field. Use this option to display properties that the Report Explorer does not include automatically.

Property names often differ from the Simulink dialog box prompts. Refer to the Simulink documentation to determine property names for blocks, signals, and other modeling objects. You can also use the MATLAB `get` command to determine the property names of an object. For example, to determine the property names of the block currently selected in a model, enter the following at the MATLAB command line:

```
get(get_param(gcb,'Handle'))
```

- **Show full path name**: Displays the full path of the selected Simulink model.
- **Display property names as prompts**: Displays property names as prompts in the generated report. The report includes the dialog box string instead of the underlying code property.

Display Options

- **Table title**: Displays a table title in the generated report.

  - `Name`: Automatically generates a title from the parameter.
  - `Custom`: Specifies a custom title.
  - `No title`: Does not include a title.

- **Header row**: Select a header row for the table in the generated report.

  - `No header`: Includes no header row.
  - `Type and Name`: Includes a header row with columns for name and object type.
  - `Custom`: Includes a custom header.

- **Don't display empty values**: Excludes empty parameters in the generated report.

Insert Anything into Report?

Yes. Table.

Class

rptgen_sl.csl_auto_table

See Also

Block Loop, Model Loop, Signal Loop, System Loop

Simulink Data Dictionary

Report Simulink data dictionary information

Description

This component reports on the data dictionary currently active in the data dictionary loop specified by the Data Dictionary Loop component. Include this component as a child of a Simulink Data Dictionary Loop component.

Presentation Format

The report for a data dictionary includes a table that summarizes the properties of each variable in the dictionary. The report also includes a dictionary details section that fully reports the properties and value of each variable in the dictionary. If you use a conversion template to generate the report, you can specify template-defined styles for the summary table title and the summary table.

To use a conversion template, in the Report Options dialog box, set **File format** to one of the `from template` options, for example, `Direct PDF (from template)`.

- **Table title style name**: Specifies the style to use for the data dictionary table title. To specify the default style name `rgTableTitle`, which the default conversion template defines, use `Auto`. To specify a custom style defined in a custom template that you use with this report, select `Specify`.

- **Table style name**: Specifies the style to use for the data dictionary table. To specify the default table style name `rgUnruledTable`, which is the default conversion template defines, use `Auto`. To specify a custom style defined in a custom template that you use with this report, select `Specify`.

Options

You can specify whether to include dictionaries referenced by a dictionary and how to present the referenced information.

- **Include referenced data dictionaries**: Includes information from the data dictionaries that the dictionary currently active in the data dictionary loop specified by the Data Dictionary Loop component references. The referenced information displays at the end of the table for the referencing data dictionary, unless you select **Make separate table for each referenced dictionary**.

- **Make separate table for each referenced dictionary**: If you select **Include referenced data dictionaries**, display a table for each referenced data dictionary.

- **Include referenced dictionaries list**: If you select **Include referenced data dictionaries**, following the referencing data dictionary summary table, include a list of the referenced data dictionaries.

Sections to Report

You can specify the data dictionary sections to include data for.

- **Design Data** (default): Include information from the Design Data section of the current data dictionary.

- **Configuration**: Include information from the Configuration section of the current data dictionary.

- **Other Data**: Include information from the Other Data section of the current data dictionary.

Fields to Report

The current dictionary summary table lists properties of the variables that it contains. The table always includes the variable name and value. In addition, it optionally includes these properties:

- **Data type**
- **Last modified**
- **Last modified by**
- **Status**
- **Referenced dictionary that contains data**

Example

Suppose that you configure an HTML report with the Simulink Data Dictionary Loop component.
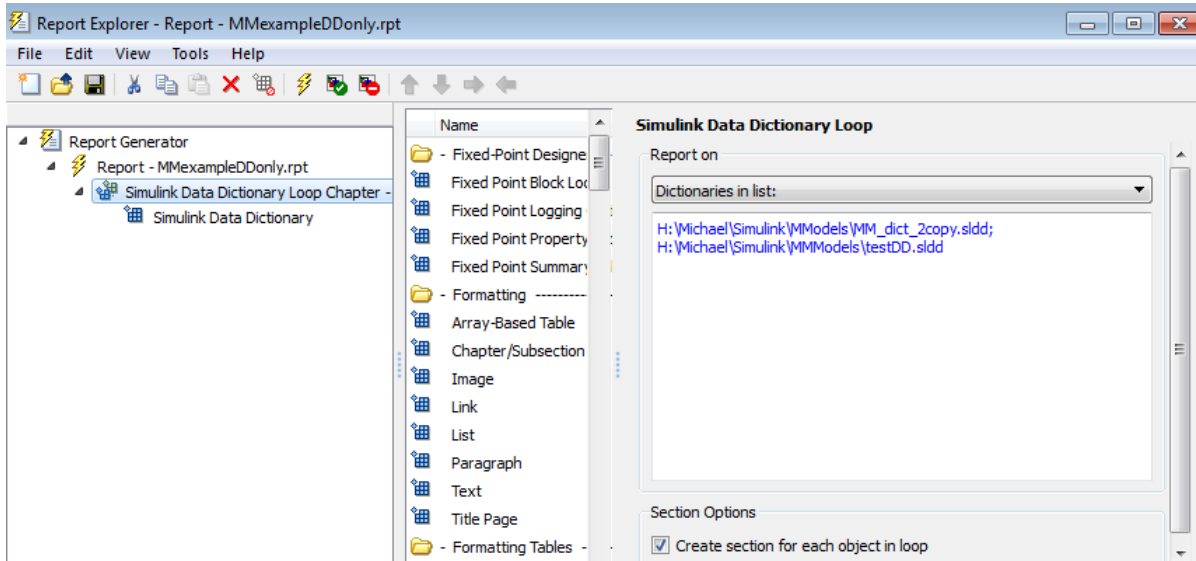
Then you configure the Simulink Data Dictionary component.
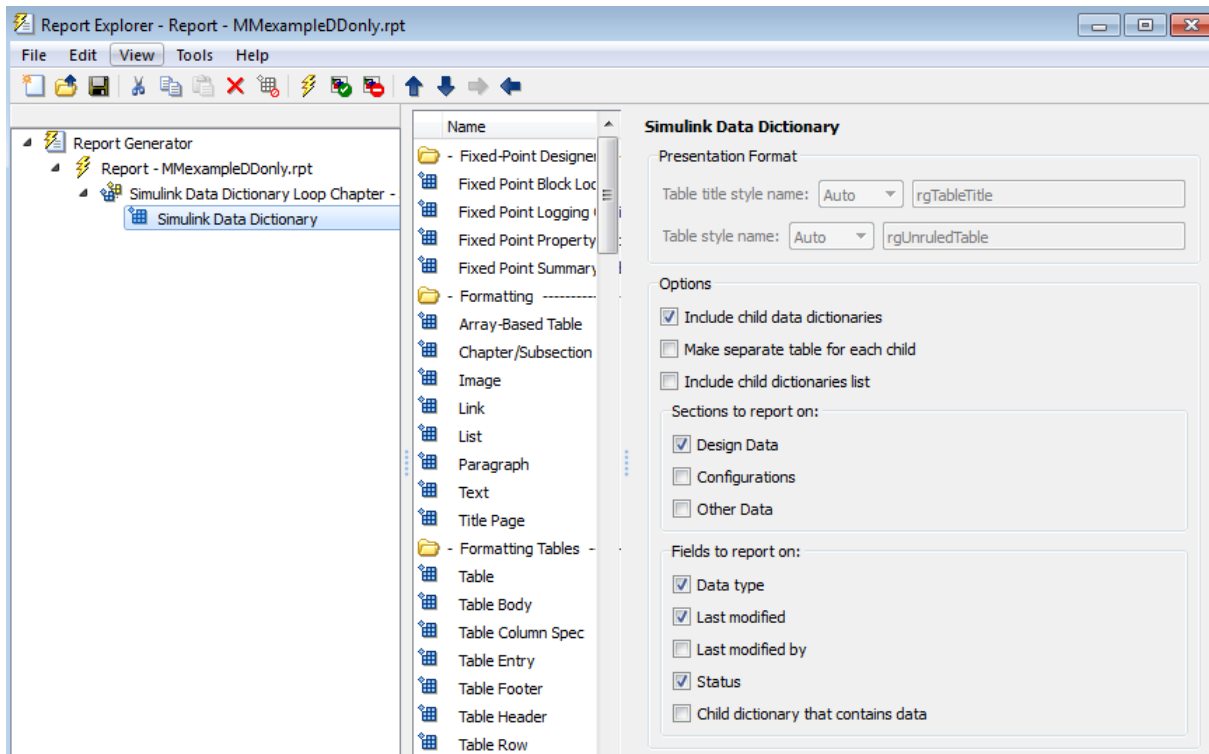
Here is the resulting report.

Table of Contents

Chapter 1. H:\Michael\Simulink\MMmodels\MM_dict_2copy.sldd

MM_Entry1

- Name: MM_Entry1
- Value: 27
- class: double
- LastModified: 2015-Jan-20 01:15:09.619512
- Status: Unchanged

MM_d2_Entry1

- Name: MM_d2_Entry1
- Value: 27
- class: double
- LastModified: 2015-Jan-30 23:30:23.182407
- Status: Modified

MM_d2_phone_number

- Name: MM_d2_phone_number
- Value: 508-647-8103
- class: char
- LastModified: 2015-Jan-30 23:30:05.414130
- Status: Modified

MM_phone_number

- Name: MM_phone_number
- Value: 508-647-8103
- class: char
- LastModified: 2015-Jan-20 01:15:09.782512
- Status: Unchanged

Chapter 2. H:\Michael\Simulink\MMmodels\testDD.sldd

testVar

- Name: testVar
- Value: testVarVal
- class: char
- LastModified: 2015-Jan-30 23:37:08.139822
- Status: New

Class

rptgen_sl.csl_data_dictionary

See Also

Simulink Data Dictionary Loop

Simulink Data Dictionary Loop

Run Simulink Data Dictionary child component for each Simulink data dictionary in specified context

Description

This component runs the Simulink Data Dictionary child component for each Simulink data dictionary in the specified context. You can specify whether to have each data dictionary in the loop.

Report on

Specify the data dictionaries to report on.

- `Dictionaries in MATLAB path`: Report on all data dictionaries on the MATLAB path. If you select **Include child data dictionaries**, then also reports on child data dictionaries whose parent is on the MATLAB path.
- `Dictionaries in list`: Report on all data dictionaries that you specify in the text box. Enter data dictionary names, separated by either a comma or semicolon. You can use multiple lines. If you do not specify the full path to a data dictionary, the loop includes that data dictionary only if the dictionary is on the MATLAB path.

Use a Summary Table component to show annotation objects in reports. Each Summary Table component creates a single table with each reported annotation on a single row of the table.

Section Options

Create section for each object in loop: Create a separate chapter for each data dictionary.

Example

Suppose you have an HTML report with the Simulink Data Dictionary Loop component configured like this:
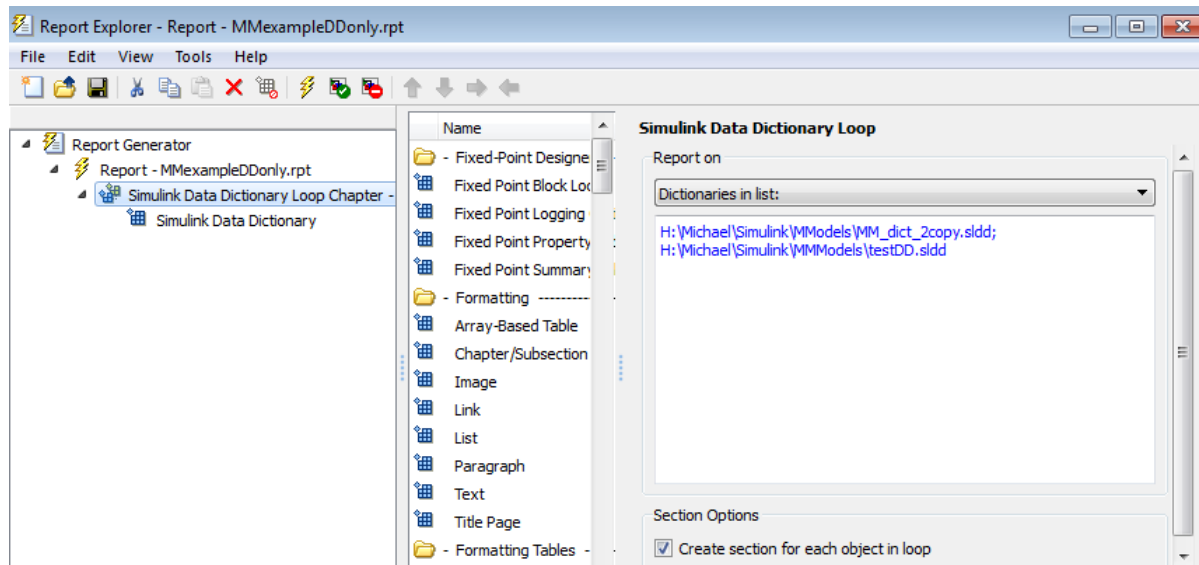
Then you configure the Simulink Data Dictionary component like this:
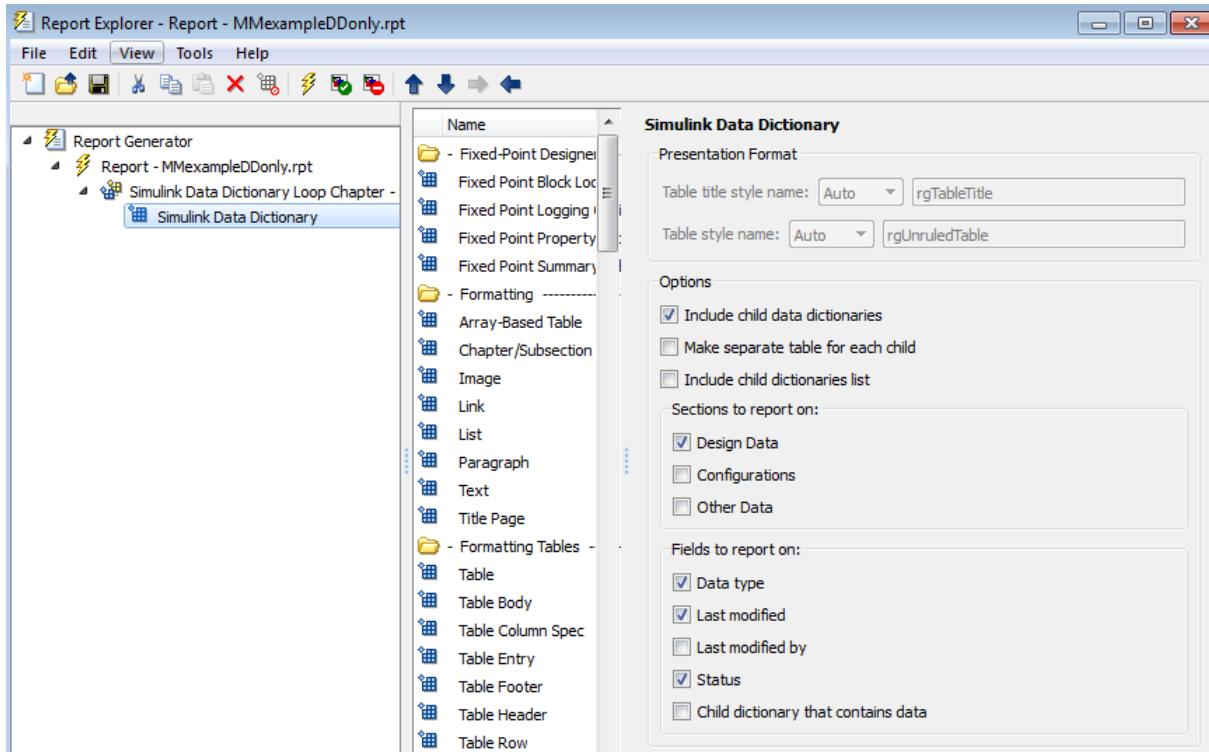
The resulting report looks like this:

Table of Contents

Chapter 1. H:\Michael\Simulink\MMmodels\MM_dict_2copy.sldd

MM_Entry1

- Name: MM_Entry1
- Value: 27
- class: double
- LastModified: 2015-Jan-20 01:15:09.619512
- Status: Unchanged

MM_d2_Entry1

- Name: MM_d2_Entry1
- Value: 27
- class: double
- LastModified: 2015-Jan-30 23:30:23.182407
- Status: Modified

MM_d2_phone_number

- Name: MM_d2_phone_number
- Value: 508-647-8103
- class: char
- LastModified: 2015-Jan-30 23:30:05.414130
- Status: Modified

MM_phone_number

- Name: MM_phone_number
- Value: 508-647-8103
- class: char
- LastModified: 2015-Jan-20 01:15:09.782512
- Status: Unchanged

Chapter 2. H:\Michael\Simulink\MMmodels\testDD.sldd

testVar

- Name: testVar
- Value: testVarVal
- class: char
- LastModified: 2015-Jan-30 23:37:08.139822
- Status: New

Class

```
rptgen_sl.csl_data_dict_loop
```

See Also

```
Simulink Data Dictionary
```

Simulink Dialog Snapshot

Insert snapshots of Simulink editor dialog boxes

Description

This component takes snapshots of Simulink editor dialog boxes. You use it to display the current settings associated with an object or document the appearance of your custom mask dialog boxes.

The parent component of this component determines the behavior of this component.

- `Block Loop`: Documents the dialog box of the current reported block.
- `System Loop`: Documents the dialog box of the current reported system.

Format

- **Image file format**: Specifies the format for the snapshot image file. The **automatic** format chooses `BMP` format for PDF files, and `PNG` for other formats.
- **Show all tabs**: Automatically generates images for all the tabs for the dialog box. If you clear this check box, Simulink Report Generator creates an image of only the first tab.

Display Options

- **Scaling**: Controls size of the image, as displayed in a browser. Making an image larger using this option does not affect the storage size of the image, but the quality of the displayed image may decrease as you increase or decrease the size of the displayed image.

  Generally, to achieve the best and most predictable display results, use the default setting of `Use image size`.

  - `Use image size`: Causes the image to appear the same size in the report as on screen (default).
  - `Fixed size`: Specifies the number and type of units.

- Zoom: Specifies the percentage, maximum size, and units of measure.
- **Size**: Specifies the size of the snapshot in the format w h (width, height). This field is active only if you choose Fixed size from the **Scaling** selection list.
- **Max size**: Specifies the maximum size of the snapshot in the format w h (width, height). This field is active only if you choose Zoom from the **Scaling** selection list.
- **Units**: Specifies the units for the size of the snapshot. This field is active only if you choose Zoom or Fixed size in the **Image size** list box.
- **Alignment**: Only reports in PDF or RTF format support this property.

  - Auto
  - Right
  - Left
  - Center
- **Title**: Specifies text to appear above the snapshot.
- **Caption**: Specifies text to appear under the snapshot.

Insert Anything into Report?

Yes. Snapshot.

Class

rptgen_sl.CDialogSnapshot

See Also

Block Loop, System Loop

Simulink Function System Loop

Report on Simulink functions specified in a Stateflow loop

Description

This component loops over the Simulink systems that implement a Stateflow Simulink Function object, including the function's parent system, subsystems, and optionally the systems that implement Simulink Functions nested in this function. This component must be a descendant of a `State Loop` component that is descendent of a `Chart Loop` component. This component executes when the current object in the state loop is a Simulink Function. For example, this structure creates a snapshot of each Simulink Function in a chart followed by snapshots of the systems that implement the function:

- Chart Loop - All chart blocks in all models
  - State Loop - All objects
    - Stateflow Snapshot
  - Simulink Function System Loop - All systems in all models
    - System Snapshot
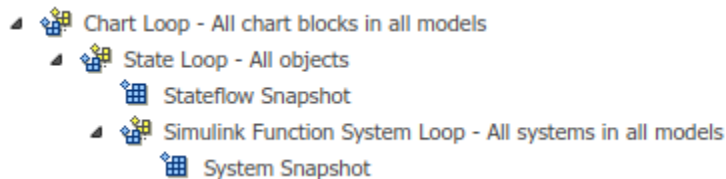
Report On

Include subsystems in nested Simulink functions: Specifies whether to include subsystems in nested Simulink functions. By default, this option is enabled.

Loop Options

- **Sort Systems**: Specifies how to sort systems.

  - `Alphabetically by system name` (default): Sorts systems alphabetically by name.
  - `By number of blocks in system`: Sorts systems by number of blocks. The list shows systems by decreasing number of blocks; that is, the system with the largest number of blocks appears first in the list.
  - `By system depth`: Sorts systems by their depth in the model.

- By `traversal order`: Sorts systems in traversal order.
- **Search for**: Reports only on Subsystem blocks with the specified property name/ property value pairs. To enable searching, click the check box. In the first row of the property name and property value table, click inside the edit box, delete the existing text, and type the property name and value.

  For information about subsystem property names and values, in "Block-Specific Parameters", see the "Ports & Subsystems Library Block Parameters" section.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Number sections by system hierarchy**: Hierarchically numbers sections in the generated report. Requires that **Sort Systems** be set to `By traversal order`.
- **Create link anchor for each object in loop**: Create a link target for each Simulink Function system in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

rptgen_sl.csl_sys_loop

See Also

`Object Loop`, `State Loop`, `Chart Loop`, `System Loop`, `Block Loop`, `Model Loop`, `Signal Loop`

Simulink Functions and Variables

Create table that displays workspace variables and MATLAB functions used by reported blocks in Simulink models

Description

This component creates a table that displays workspace variables and MATLAB functions used by blocks in a Simulink model. The `Model Loop` component specifies the current model and systems in which the blocks appear. For example, suppose a Simulink `Gain` block has a string `cos(x)` instead of a number. The Simulink software looks for a variable `x` in the workspace and uses the `cos` function.

Functions

- **Include table of functions**: Includes a table of Simulink functions in the generated report.
- **Table Title**: Specifies a title for the table in the generated report:

    - `Automatic`: Generates a title automatically from the parameter.
    - `Custom`: Specifies a custom title.

- **Parent block**: Includes a column in the table that includes the name of the block, which contains the reported variable.
- **Calling string**: Includes the MATLAB code that calls the reported variable.
- **Include fixed-point functions (sfix, ufix, ...)**: Includes Fixed-Point Designer functions in the report.

Variables

- **Include table of variables**: Includes a table of Simulink variables in the generated report.
- **Table title**: Specifies a title for the table in the generated report.

    - `Automatic`: Generates a title automatically from the parameter.
    - `Custom`: Specifies a custom title.

7-120

- **Include Workspace I/O parameters**: Reports on variables that contain parameters with time vectors and state matrices. Set these parameters in the **Workspace I/O** pane in the Simulation Parameters dialog box in a Simulink model.

  In the following table, if any of the entries in the first column are `on`, the component looks for the variable listed in the second column. If the component finds the variable in the workspace, it includes it in the report.

Parameter name	Variable name
`LoadExternalInput`	`ExternalInput`
`SaveTime`	`TimeSaveName`
`SaveState`	`StateSaveName`
`SaveOutput`	`OutputSaveName`
`LoadInitialState`	`InitialState`
`SaveFinalState`	`FinalStateName`

- **Parent block**: Includes the name of the block that contains the reported variable.
- **Calling string**: Includes the MATLAB code that calls the reported variable.
- **Size of variable**: Includes the size of the reported variable.
- **Class of variable**: Includes the variable class to which the reported variable belongs.
- **Memory size**: Includes the amount of memory in bytes that the reported variable needs.
- **Value in workspace**: Includes the value of the reported variable.

  Large arrays may appear as `[MxN CLASS]`. For example, if you have a 300-by-200 double array, it appears in the report as `[300x200 DOUBLE]`.

- **Storage class**: Include the storage class of the reported variable.

  The title of this column is **Storage Class**. This option looks at the model's `TunableVars` property to see if any of the model variables specify their storage class. If you specify the storage class, `TunableVarsStorageClass` and `TunableVarsTypeQualifier` appear in a table column in the model variables table.

  The column entries are `TunableVarsStorageClass` (`TunableVarsTypeQualifier`) when `TunableVarsTypeQualifier` is not empty. If `TunableVarsTypeQualifier` is empty, the column entry is `TunableVarsStorageClass`.

Values for `TunableVarsStorageClass` include:

- `Exported Global`
- `Auto`
- `ImportedExtern`
- `ImportedExtern Pointer`

• **Data object properties**: For variables that are `Simulink.Parameter` data objects, includes the values of the object properties that you list in the edit box.

Example

This table is an example of a table created by the `Model Variables` component. This Property Table reports on the variables in the `Controller` in the `F14` model.

Variable Name	Parent Blocks	Calling String	Value
Ka	`f14/Controller/Gain3`	Ka	`0.677`
Kf	`f14/Controller/Gain`	Kf	`-1.746`
Ki	`f14/Controller/Proportional plus integral compensator`	[Ki]	`-3.864`
Kq	`f14/Controller/Gain2`	Kq	`0.8156`

Insert Anything into Report?

Yes. Table.

Class

`rptgen_sl.csl_obj_fun_var`

See Also

`Block Loop`, `Model Loop`, `Signal Loop`, `System Loop`

Simulink Library Information

Insert table that lists library links in the current model, system, or block

Description

This component inserts a table that lists library links in the current model, system, or block.

Table Columns

- **Block**: Includes the Simulink block name in the generated table.
- **Library**: Includes the Simulink library root name in the generated table.
- **Reference block**: Includes the Simulink reference block name in the generated table.
- **Link status**: Includes the link status in the generated table.

Display Options

- **Title**: Specifies a title for the generated report.
- **Sort table by**:

    - `Block`: Sorts the table by block name.
    - `Library`: Sorts the table by library name.
    - `Reference Block`: Sorts the table by reference block name.
    - `Link Status`: Sorts the table by link status.
- **Merge repeated rows**: Merges sorted rows in the generated table.

Example

The following table sorts based on `Reference Block` column. The Report Explorer uses the `aero_guidance` model with **Merge repeated rows** deselected to generate the table.

Block	Library	Reference Block	Status
Equations of Motion (Body Axes)	Aerospace	Equations of Motion (Body Axes)	resolved
Incidence & Airspeed	Aerospace	Incidence & Airspeed	resolved
Fin Actuator	Aerospace	2nd Order Nonlinear Actuator	resolved
3DoF Animation	Aerospace	3DoF Animation	resolved
Atmosphere	Aerospace	Atmosphere model	resolved
Cm	Simulink	Interpolation (n-D) using PreLookup	resolved
Cx	Simulink	Interpolation (n-D) using PreLookup	resolved
Cz	Simulink	Interpolation (n-D) using PreLookup	resolved
Kg	Simulink	Interpolation (n-D) using PreLookup	resolved
Ki	Simulink	Interpolation (n-D) using PreLookup	resolved
Alpha Index	Simulink	PreLookup Index Search	resolved
Mach Index	Simulink	PreLookup Index Search	resolved
Mach Index	Simulink	PreLookup Index Search	resolved
|Alpha| Index	Simulink	PreLookup Index Search	resolved

When you select **Merge repeated rows**, the generated table collapses rows in the `Block` column. Each row in the `Reference Block` column is unique, as shown in the following table.

Block	Library	Reference Block	Status
Equations of Motion (Body Axes)	Aerospace	Equations of Motion (Body Axes)	resolved

Block	Library	Reference Block	Status
Incidence & Airspeed	Aerospace	Incidence & Airspeed	resolved
Fin Actuator	Aerospace	2nd Order Nonlinear Actuator	resolved
3DoF Animation	Aerospace	3DoF Animation	resolved
Atmosphere	Aerospace	Atmosphere model	resolved
Cm Cx Cz Kg Ki	Simulink	Interpolation (n-D) using PreLookup	resolved
Alpha Index Mach Index Mach Index |Alpha| Index	Simulink	PreLookup Index Search	resolved

Insert Anything Into Report?

Yes. Table.

Class

rptgen_sl.CLibinfo

See Also

Block Loop, Model Loop, System Loop

Simulink Linking Anchor

Designate locations to which links point

Description

This component designates a location to which links point. Use the `Model Loop`, `System Loop`, `Block Loop`, or `Signal Loop` component as the parent component for this component.

Properties

- **Insert text**: Specifies text to appear after the linking anchor.
- **Link from current**: Sets the current model, system, block, or signal as the linking anchor.

  - `Automatic`: Automatically selects the appropriate model, system, block, or signal as a linking anchor. If the `Model Loop` component is the parent component, the linking anchor is set on the current model. Similarly, if the `Block Loop` or `Signal Loop` is the parent component, the linking anchor is inserted for the current system, block, or signal, respectively.
  - `Model`: Sets the linking anchor to the current model.
  - `System`: Sets the linking anchor to the current system.
  - `Block`: Sets the linking anchor to the current block.
  - `Annotation`: Sets the linking anchor to the current annotation.
  - `Signal`: Sets the linking anchor to the current signal.

  Note: Use only one anchor per report each object. For more information, see the `Simulink Summary Table` component reference page.

Insert Anything into Report?

Yes. A link, and possibly text, depending on attribute choices.

Class

rptgen_sl.csl_obj_anchor

See Also

Block Loop, Model Loop, Signal Loop, System Loop

Simulink Name

Insert name of a Simulink model, system, block, or signal into report

Description

This component inserts the name of a Simulink model, system, block, or signal into the report.

Using this component as the first child component of a `Chapter/Subsection` component allows the current Simulink model, system block, or signal name to be the chapter or section title.

Properties

- **Object type**

    - `Automatic`: Automatically selects the appropriate model, system, block, or signal name as the Simulink object name to include in the report. If the Model Loop component is the parent component, the object name is the current model name. If the System Loop, Block Loop, or Signal Loop is the parent component, then the object name is the name of the current system, block, or signal, respectively.
    - `Model`: Includes the current model name in the report.
    - `System`: Includes the current system name in the report.
    - `Block`: Includes the current block name in the report.
    - `Signal`: Includes the current signal name in the report. If the signal name is empty, the signal `<handle>`, which is a unique numerical identifier to that signal, appears in the report.
    - `Annotation`: Includes the current annotation name in the report.

- **Display name as**: Display the Simulink object name in the report.

    - `Name`: For example, `f14`
    - `Type Name`: For example, `Model f14`
    - `Type - Name`: For example, `Model - f14`
    - `Type: Name`: For example, `Model: f14`

- **Show full path name**: Displays the full path of a system or block. Choosing this option for a block causes the Simulink block name to appear as `<Model Name>/<System Name>/<Block Name>`.

Note: This option is not available for models or signals.

Insert Anything into Report?

Yes. Text.

Class

`rptgen_sl.csl_obj_name`

See Also

`Chapter/Subsection`

Simulink Property

Insert property name/property value pair for current Simulink model, system, block, or signal

Description

This component inserts a single property name/property value pair for the current Simulink model, system, block, or signal.

Simulink Object and Parameter

- **Object type**: Specifies the Simulink object type to include in the report.

  - System
  - Model
  - Block
  - Signal
  - Annotation

- **System parameter name**: Specifies a Simulink parameter name to include in the generated report:

  - If you select Model for **Object type**, this option appears as **Model parameter name**.
  - If you select Block for **Object type**, this option appears as **Block parameter name**.
  - If you select Signal for **Object type**, this option appears as **Signal parameter name**.

Display Options

- **Title**: Choose a title to display in the generated report:

  - Automatic: Uses the parameter name as the title.

- Custom: Specifies a custom title.
- None: Uses no title.

- **Size limit**: Limits the width of the display in the generated report. Units are in pixels. The size limit for a given table is equal to the hypotenuse of the table width and height [sqrt(w^2+h^2)]. The size limit for text is the number of characters squared. If you exceed the size limit, the variable appears in condensed form. Setting a size limit of zero always causes the variable to display, regardless of its size

- **Display as**: Specifies a display style.

  - Auto table/paragraph: Displays as a table or paragraph based on the information.
  - Table: Displays as a table.
  - Paragraph: Displays as a text paragraph.
  - Inline text: Displays as inline, which fits in line with the surrounding text.

- **Ignore if value is empty**: If the parameter is empty, the report excludes the parameter from the generated report.

Insert Anything into Report?

Yes. Text.

Class

rptgen_sl.csl_property

See Also

Block Loop, Model Loop, System Loop

Simulink Property Table

Insert table that reports on model-level property name/property value pairs

Description

This component inserts a table that reports on model-level property name/property value pairs.

Properties

Select Object: Choose the object for the Property Table in the generated report.

- `Model`
- `System`
- `Block`
- `Signal`
- `Annotation`

For more information about selecting object types in Property Table components, see "Select Object Types".

Table

Select a preset table, which is already formatted and set up, in the preset table list in the upper-left corner of the attributes page.

- **Preset table**: Specifies the type of the object property table.

  - `Default`
  - `Simulation parameters`
  - `Version information`
  - `Simulink Coder information`

- Summary (req. Simulink Coder)
- Blank 4x4

To apply a preset table, select the table and click **Apply**.

- **Split property/value cells**: Split property name/property value pairs into separate cells. For the property name and property value to appear in adjacent horizontal cells, select the **Split property/value cells** check box. In this case, the table is in split mode, so there is only one property name/property value pair per cell. If there is more than one name/property pair in a cell, only the first pair appears in the report. All subsequent pairs are ignored.

  For the property name and property value to appear together in one cell, clear the **Split property/value cells** check box. That setting specifies nonsplit mode. Nonsplit mode supports more than one property name/property value pair and text per cell.

  Before switching from nonsplit mode to split mode, make sure that there is only one property name/property value pair per table cell. If there is more than one property name/property value pair or text per cell, only the first property name/property value pair appears in the report. The report omits subsequent pairs and text.

- **Display outer border**: Displays the outer border of the table in the generated report.

- **Table Cells**: Specifies table properties to modify. The selection in this pane affects available fields in the **Cell Properties** pane.

Cell Properties

The options in this pane depend on the object selected in the **Table Cells** pane. If you select %<Name> Information, only **Contents** and **Show** appear. If you select any other object in the **Table Cells** pane, **Lower border** and **Right border** display.

- **Contents**: Enables you to change the contents of the table cell selected in the **Table Cells** pane.

- **Show as**: Specifies the format for the contents of the table cell.

  - Value
  - Property Value
  - PROPERTY Value

- - Property: Value
  - PROPERTY: Value
  - Property - Value
  - PROPERTY - Value
- **Alignment**: Specifies the alignment of the contents of the selected table cell in the **Table Cells** pane.

  - Left
  - Center
  - Right
  - Double justified
- **Lower border**: Displays the lower border of the table in the generated report.
- **Right border**: Displays the right border of the table in the generated report.

Creating Custom Tables

To create a custom table, edit a preset table, such as the Blank 4x4 table. Add and delete rows and add properties. To open the Edit Table dialog box, click **Edit**.

For more information on creating custom property tables, see "Property Table Components".

If the Simulink Coder software is not installed, Summary (req Simulink Coder) does not appear in this list. If you are using a report setup file that contains a summary property, the property name appears in the report, but the property value does not.

Example

The following report displays information on the F14 model using the **Simulation Parameters** preset table.

Solver ode45	*ZeroCross* on	*StartTime* 0.0 *StopTime* 60
RelTol 1e-4	*AbsTol* 1e-6	*Refine* 1
InitialStep auto	*FixedStep* auto	*MaxStep* auto
LimitMaxRows off	*MaxRows* 1000	*Decimation* 1

Insert Anything into Report?

Yes. Table.

Class

rptgen_sl.csl_prop_table

See Also

Model Loop, Signal Loop, System Loop

Simulink Sample Time

Insert title of Simulink sample time into report

Description

This component inserts a title for a Simulink sample time into the report.

Properties

- **Table Options**
  - **Title**: Specifies a title for the table in the generated report.
  - **Grid lines**: Show grid lines for the table.

Insert Anything into Report?

Yes. Table.

Class

rptgen_sl.CSampleTime

See Also

Chapter/Subsection

Simulink Summary Table

Properties or parameters of specified Simulink models, systems, blocks, or signals in table

Description

This component displays properties or parameters of selected Simulink models, systems, blocks, or signals in a table.

Object type

Choose the object type to display in the generated report.

- `Block` (Default)
- `Model`
- `System`
- `Signal`
- `Annotation`

The selected object type affects the options available in the **Property Columns** pane.

Table title

Choose a title to appear in the generated report:

- `Automatic`: Automatically generates a title from the parameter.
- `Custom`: Specifies a custom title.

Property Columns

This pane displays object properties to include in the Summary Table in the generated report.

- To add a property:

> **1** select the appropriate property level in the text box on the left.
>
> **2** In the text box on the right, select the property that you want to add and click **Add**.

- To delete a property, select the property name and click **Delete**.

`%<SplitDialogParameters>` is a unique property for Simulink Summary Tables, where the object type is `Block`. This property generates multiple summary tables, organized by block type. Each Summary Table group contains the dialog box parameters for that block.

Some entries in the list of available properties (such as `Depth`) are "virtual" properties that you cannot access using the `get_param` command. The properties used for property/value filtering in the block and System Loop components must be retrievable by the `get_param`. Therefore, you cannot configure your Summary Table to report on all blocks of `Depth == 2`.

You can create multiple values for a property in a Simulink Summary Table. For example, to report on blocks of type `Inport`, `Outport` and `Constant`:

1 Check the **Search for Simulink property name/property value pairs** box.

2 Make sure that you set **Property Name** to `BlockType`.

3 Type the following text into the **Property Value** field:

`\<(Inport|Outport|Constant)\>`

Remove empty columns: Removes empty columns from the table.

Transpose table: Changes the summary table rows into columns in the generated report, putting the property names in the first column and the values in the other columns.

Object Rows

- **Insert anchor for each row**: Inserts an anchor for each row in the summary table.

- **Report On**:

  - `Automatic list from context`: Reports on all blocks in the current context, as set by the parent component.

- `Custom - use block list`: Reports on a list of specified blocks. Specify the full path of each block.

Loop Options

- **Sort blocks**

    - `Alphabetically by block name`: Sorts blocks alphabetically by name.
    - `Alphabetically by system name`: Sorts systems alphabetically by name. Lists blocks in each system, but in no particular order.
    - `Alphabetically by full Simulink path`: Sorts blocks alphabetically by Simulink path.
    - `By block type`: Sorts blocks alphabetically by block type.
    - `By block depth`: Sorts blocks by their depth in the model.
    - `By layout (left to right)`: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

    - `By layout (top to bottom)`: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor

block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- By `traversal order`: Sorts blocks by traversal order.
- By `simulation order`: Sorts blocks by execution order.
- `%<VariableName>`: Inserts the value of a variable from the MATLAB workspace. The `%<>` notation can denote a string or cell array. For more information, see `%<VariableName> Notation` on the `Text` component reference page.

- **Search for Simulink property name/property value pairs**: Reports on blocks with specified property name/property value pairs.

Example

Specify the following options to generate a Summary Table in a report for on the model `F14`:

- Sort on systems by system depth.
- Include the system parameters `Name` and `Block` in the table.

The following table appears in the report.

Name	Blocks
f14	u, Actuator Model, Aircraft Dynamics Model, Angle of Attack, Controller, Dryden Wind Gust Models, Gain, Gain1, Gain2, Gain5, More Info, More Info1, Nz pilot calculation, Pilot, Pilot G force Scope, Stick Input, Sum, Sum1, alpha (rad), Nz Pilot (g)
Aircraft Dynamics Model	Elevator Deflection d (deg), Vertical Gust wGust (ft/sec), Rotary Gust qGust (rad/sec), Gain3, Gain4, Gain5, Gain6, Sum1, Sum2, Transfer Fcn.1, Transfer Fcn.2, Vertical Velocity w (ft/s), Pitch Rate q (rad/s)
Controller	Stick Input (in), alpha (rad), q (rad/s), Alpha-sensor Low-pass Filter, Gain, Gain2, Gain3, Pitch Rate Lead Filter, Proportional plus integral compensator, Stick Prefilter, Sum, Sum1, Sum2, Elevator Command (deg)
Dryden Wind Gust Models	Band-Limited White Noise, Q-gust model, W-gust model, Wg, Qg
More Info	None

Name	Blocks
More Info1	None
Nz pilot calculation	w, q, Constant, Derivative, Derivative1, Gain1, Gain2, Product, Sum1, Pilot g force (g)

Insert Anything into Report?

Yes. Table.

Class

rptgen_sl.csl_summ_table

See Also

Block Loop, Model Loop, Signal Loop, System LoopSimulink Function System Loop

Simulink Test Suite Traceability Table

Insert a table that links a Simulink test suite to requirements

Description

This component inserts a table into the report. The table links a Simulink test suite to corresponding requirements. This component reports on the currently open Simulink test suite. Place this component inside a section, paragraph, or table component.

To use this component, your report setup must include `Eval` statements that open a Simulink test suite or determine the test suite that is open.

Table Options

Specify information about the table this component inserts.

- **Table title**: Specify the table title.

  - `No title` — Do not include a table title.
  - `Object name` — Use the name of the Simulink test suite in the title.
  - `Custom` — Specify your own table title.

Table Columns

Specify the table columns that you want to include in the report. The **Document name**, **Locations within document**, and **Requirement keyword** check boxes correspond to properties on the Requirements Management Interface Link Editor dialog box.

- **Description** — Include the description of the requirement. The description helps you to identify the requirement the table is linking to. Leave this box selected to improve the readability of your table.
- **Document name** — Include the name of the document where the requirement is located.
- **Locations within document** — Include the identifier of a location in the document.

- **Requirement keyword** — Include the requirement keyword.

Insert Anything into Report?

Yes. Table.

Class

```
RptgenRMI.TMReqTable
```

See Also

`Data Dictionary Traceability Table`, `MATLAB Code Traceability Table`, `rmi`

Simulink Workspace Variable

Report on workspace variables used in model, in loop generated by Simulink Workspace Variable Loop component

Description

This component provides information about those workspace variables that the Simulink model uses, in a loop generated by a `Simulink Workspace Variable Loop` component. Your report setup must include `Simulink Workspace Variable` component as a child of a `Simulink Workspace Variable Loop` component.

The report includes the name and value each variable. Optionally, you can include the following information for each variable:

- Variable source (MATLAB workspace, model workspace, or data dictionary)
- Blocks that use the variable

For variables that are Simulink data objects (for example, a `Simulink.Parameter` object), the report includes the properties of the object. You can filter out properties to streamline the report.

Use a `Simulink Workspace Variable Loop` component as a parent for a `Simulink Workspace Variable` component. In the Report Options dialog box, select **Compile model to report on compiled information**.
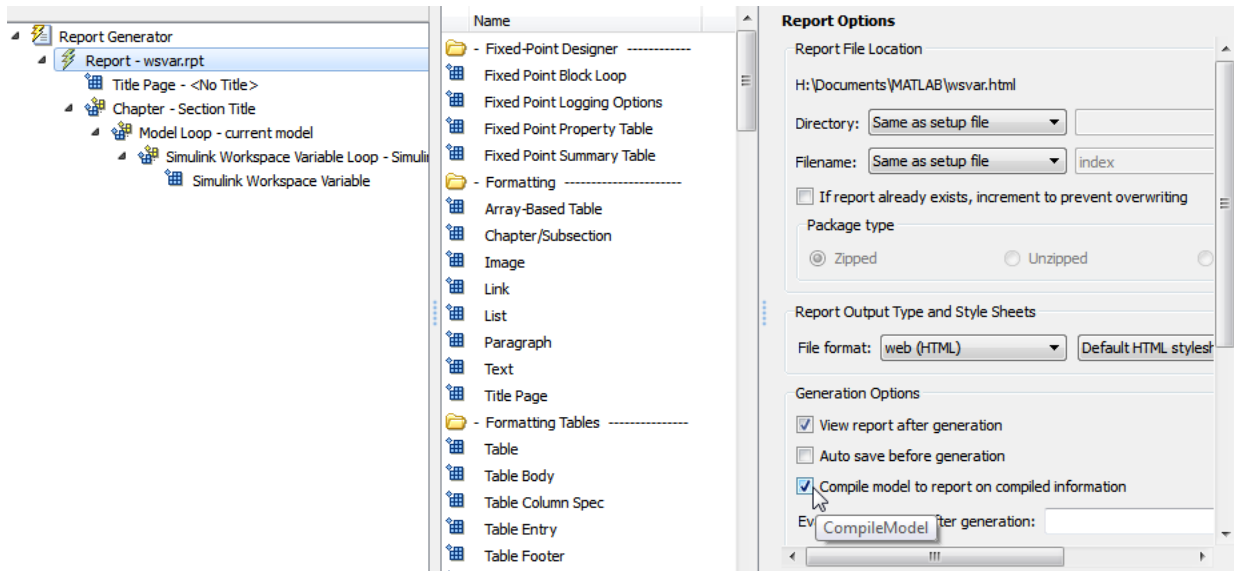
Options

The following options specify additional information that the report can include about each variable:

- **Show workspace**: Report the source of each variable — MATLAB workspace, model workspace, or data dictionary.

- **Show blocks that use variable**: Report the blocks that use each variable.

For variables whose values are Simulink data objects, you can filter the properties to include in the report, using one of the following approaches:

- Use the **Filter Properties** area of the dialog box to specify a standard filter.

  The standard filter options apply to all variables whose values are instances of the class or classes that you specify. For example, you can use a standard filter to filter out the `Description` property for all variables used by the model whose values use a `Simulink.Parameter` object.

- Select the **Use custom property filter** option and write MATLAB code for filtering.

Writing custom filtering code allows you to do kinds of filtering that the standard filter does not perform. Some common examples of custom filters that you might want to create are filters that filter out:

- A property for some, but not all, instances of a class
- Properties that match a regular expression

The **Filter Properties** area of the dialog box, where you specify a standard filter, has these fields.

- **Class name (* for all classes)**: Specify the class of the variables for which you want to filter out specific properties. You can specify one class at a time, or enter an asterisk (*) to specify all classes. After you enter the class name, move the cursor outside of the edit box.
- **Available Properties**: If the class that you entered in **Class name (* for all classes)** is on the MATLAB path, then this list displays the properties of that class.
- **Filtered Properties**: Displays the properties to filter out. Use the right-arrow button to add to the **Filtered Properties** list the properties that you selected in the **Available Properties** list.
- If the class that you enter is *not* on the MATLAB path, then a **Comma-separated list of properties to be filtered** edit box appears. Enter the names of properties to use for filtering.
- **Convert to Custom**: Generate custom MATLAB code that implements your **Filter Properties** standard filter settings.

Note: Selecting the **Convert to Custom** button overwrites any existing MATLAB custom filtering code for this component.

To create and apply custom filtering MATLAB code, select the **Use custom property filter** check box. Selecting this check box opens an edit box where you define a MATLAB function for filtering properties. The edit box includes a sample function (commented out) that you can use as a starting point for your filtering function. Use the `isFiltered` variable for the output of your function. For example:

- To filter out the `Owner` and `testProp` properties, in the edit box enter:

```
isFiltered = strcmpi(propertyName, 'Owner')||...
strcmpi(propertyName, 'testProp');
```

- To filter out all properties *except* for the `CoderInfo` property, in the edit box, enter:

```
isFiltered = ~strcmpi(propertyName, 'CoderInfo');
```

If you clear the **Use custom property filter** check box, Simulink Report Generator saves your custom MATLAB filtering code, but does not use that code to filter properties.

Insert Anything into Report?

Yes. List.

Class

```
rptgen_sl.csl_ws_variable
```

See Also

`Simulink Workspace Variable Loop`, `Bus`, `Simulink Functions and Variables`

Simulink Workspace Variable Loop

Generates a model variable loop

Description

This component generates a model variable loop used by the `Simulink Workspace Variable` component to report on those workspace variables that the Simulink model uses.

You can limit the variables included in the loop to those that match property name and value pairs that you specify. If you want to report on model variables, your report setup file must include this component as a child of a `Model Loop` component and must include a `Simulink Workspace Variable` component as its child. Also, in the Report Options dialog box, select **Compile model to report on compiled information**. For example:
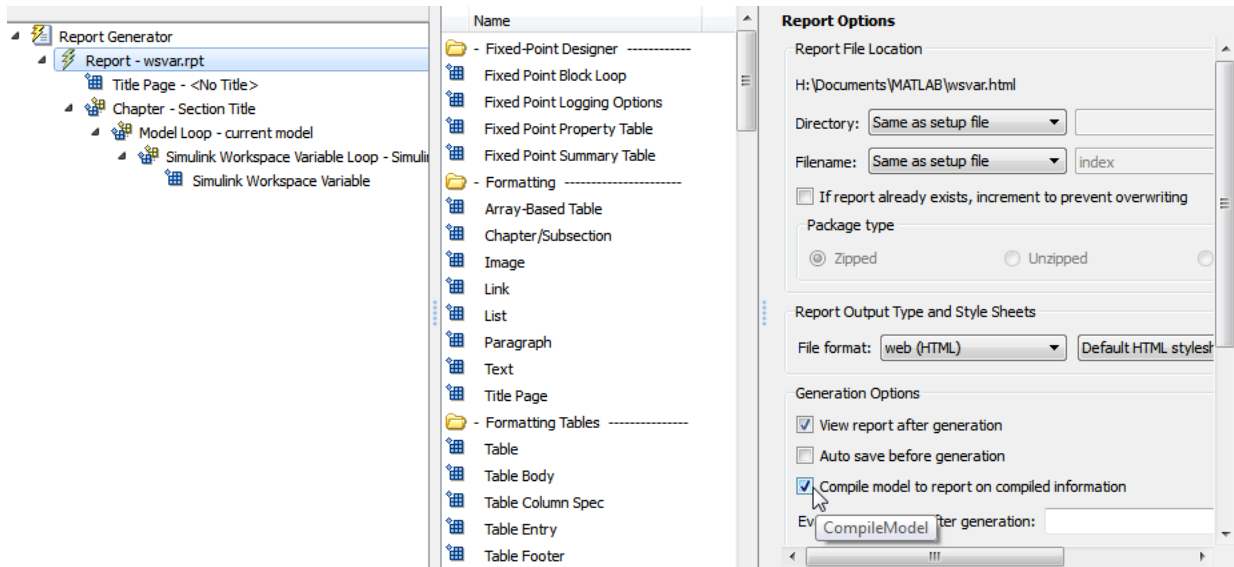
Loop Options

- **Sort**

    - `Alphabetically by text`: Sort variables alphabetically by name.
    - `By data type`: Sort variables alphabetically by data type.
- **Search for Simulink property name/property value pairs**: Reports on variables with specified property name/property value pairs.

Section Options

- **Create section for each object in loop**: Creates a separate section in the output for each variable.

    - If you specify to create a section for each variable, you can select the **Display the object type in the section title** to insert a variable name in each section title.
- **Create link anchor for each object in loop**: Create a link target for each workspace variable in the loop so that other parts of the report can link to it.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

rptgen_sl.csl_ws_var_loop

See Also

Simulink Workspace Variable, Bus, Simulink Functions and Variables

State Loop

Run child components for all states in current context

Description

This component runs its children for all states in its context. The parent component of this component determines the context.

- `Model Loop`: Includes all states in the models.
- `System Loop`: Includes all states in the systems.
- `Machine Loop`: Includes all states in the machines.
- `Chart Loop`: Includes all states in the charts.
- `State Loop`: Includes all states in the current state.

For conditional processing based on states, you can use the `RptgenSF.getReportedState` function. For more information, see "Loop Context Functions" on page 4-55.

State Types

- **Include "and" and "or" states**: Includes AND and OR states in the loop.
- **Include "box" states**: Includes "box" states in the loop.
- **Include functions**: Includes "function" states in the loop.
- **Include truth tables**: Includes truth tables in the loop.
- **Include MATLAB functions**: Includes MATLAB functions in the loop.

Loop Options

- **Report depth**: Specifies the level on which to loop.

  - `Local children only`
  - `All objects`

- **Skip autogenerated charts under truth table**: Keeps autogenerated state objects under truth tables from appearing in the report.
- **Search Stateflow**: Indicates specific states to include in the loop.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target for each state in the loop so that other parts of the report can link to it. For example, the image created by a `Stateflow Snapshot` component can link to the chart information only if you select this check box.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

`rptgen_sf.csf_state_loop`

See Also

`Chart Loop`, `Machine Loop`, `Model Loop`, `State Loop`, `System Loop`, `Simulink Function System Loop`

State Transition Matrix

Inserts state transition matrix contents into report

Description

This component inserts the contents of state transition matrices into a report. A state transition matrix is an alternative view of a state transition table. In the state transition matrix, you can easily see how the state transition table reacts to each condition and event.

Options

- **Title**

  - `No title` (default): Report uses no title for the state transition matrix.
  - `Use Stateflow name`: For the title in the report, uses the names of the State Transition Table blocks from which the state transition matrices are generated.
  - `Custom`: In the text field, specify a custom name for the state transition matrix.
- **Display condition actions on matrix**: Include the state transition matrix condition actions. A condition action is an action that executes as soon as a condition evaluates to true. The condition action is part of a transition label.

Insert Anything into Report?

Yes, inserts state transition matrices and optionally, condition actions.

Class

`rptgen_sf_csf_statetransitionmatrix`

See Also

`State Transition Table`

State Transition Table

Inserts state transition tables into report

Description

This component inserts the state transition tables into a report. A state transition table is an alternative way of expressing sequential modal logic. Instead of drawing states and transitions graphically in a Stateflow® chart, you express the modal logic in tabular format.

Options

- **Title**

    - `No title` (default): Report uses no title for the state transition table.
    - `Use Stateflow name`: Uses the name of the State Transition Table block as the title.
    - `Custom`: In the text field, specify a custom name for the state transition table.

Insert Anything into Report?

Yes, inserts state transition table.

Class

`rptgen_sf_csf_statetransitiontable`

See Also

`State Transition Matrix`

Stateflow Automatic Table

Insert table with properties of current Stateflow object

Description

This component inserts a table that contains the properties of the current Stateflow object. Parents of this component can be:

- `Machine Loop`
- `State Loop`
- `Chart Loop`
- `Graphics Object Loop`

Display Options

- **Table title**: Specifies a title for the table in the generated report.

  - `No title`: Includes no title.
  - `Custom`: Includes a custom title.
  - `Name` (default): Uses an object name as the title.

    - `Object name`
    - `Object name with Stateflow path`
    - `Object name with Simulink and Stateflow path`

- **Header row**: Selects a header row for the table in the generated report.

  - `No header`: Includes no header row.
  - `Type and Name`: Includes a header row with columns for name and object type. When selected, this option creates a header row for the table with object name and type.
  - `Custom`: Includes a custom header.

- **Don't display empty values**: Excludes empty values from the generated report.

Insert Anything into Report?

Yes. Table.

Class

rptgen_sf.csf_auto_table

See Also

Chart Loop, Graphics Object Loop, Machine Loop, State Loop

Stateflow Count

Count number of Stateflow objects in current context

Description

This component counts the number of Stateflow objects in the current context.

Properties

- **Search depth**: Specifies the search depth for the count.

    - `Immediate children only` (default): Searches only children one level under the Stateflow object.
    - `All descendants`: Searches all children of the Stateflow object.
- **Sort results**: Specifies the sort method for the count results.

    - `Numerically decreasing by object count` (Default)
    - `Alphabetically increasing by object type`
- **Include a list of objects in table**: Inserts a column containing the counted objects.
- **Show total count**: Displays a total of counted objects.

Insert Anything into Report?

Yes. Table.

Class

rptgen_sf.csf_count

See Also

State Loop

Stateflow Dialog Snapshot

Insert snapshots of Stateflow editor dialog boxes

Description

This component reports on the current reported Stateflow dialog box object, depending on its context. If this component is the child of a `State Loop`, for example, the report includes information about the dialog box of the current State. Display the current settings associated with an object or document the appearance of your custom mask dialog boxes.

Format

- **Image file format**: Specifies the format for the snapshot image file. The `Automatic` format uses `BMP` format for PDF files and `PNG` for other formats.
- **Show all tabs**: Automatically generates images for all the tabs for the dialog box. If you clear this check box, the Simulink Report Generator software creates an image of only the first tab.

Display Options

- **Scaling**: Controls size of the image, as displayed in a browser. Making an image larger using this option does not affect the storage size of the image, but the quality of the displayed image may decrease as you increase or decrease the size of the displayed image.

  Generally, to achieve the best and most predictable display results, use the default setting of `Use image size`.

  - `Use image size`: Causes the image to appear the same size in the report as on screen (default).
  - `Fixed size`: Specifies the number and type of units.
  - `Zoom`: Specifies the percentage, maximum size, and units of measure.
- **Size**: Specifies the size of the snapshot in the form `w h` (width, height). This field is active only if you choose `Fixed size` in the **Scaling** selection list.

- **Max size**: Specifies the maximum size of the snapshot in the form w h (width, height). This field is active only if you choose Zoom in the **Scaling** selection list.
- **Units**: Specifies the units for the size of the snapshot. This field is active only if you choose Zoom or Fixed size in the **Image size** list box.
- **Alignment**: Aligns your snapshot. Only reports in PDF or RTF format support this property.

  - Auto
  - Right
  - Center
  - Left

- **Title**: Specifies text to appear above the snapshot.
- **Caption**: Specifies text to appear under the snapshot.

Insert Anything into Report?

Yes. Snapshot.

Class

rptgen_sl.Cdialog boxesnapshot

See Also

State Loop

Stateflow Filter

Run child components only if current object type matches specified object type

Description

This component runs its children only if the current object type, as set by its parent `Stateflow Hierarchy Loop`, matches the selected object type.

Properties

- **Object type**: Specifies the Stateflow object type to include in the report.
- **Run only if Stateflow object has at least the following number of Stateflow children**: Specifies a minimum number of children that a Stateflow object must have to include in the report.
- **Automatically insert linking anchor**: Inserts a linking anchor before the reported object. If an anchor for this object exists, this option does not create a second anchor.

Insert Anything into Report?

No.

Class

rptgen_sf.csf_obj_filter

See Also

Stateflow Hierarchy Loop

Stateflow Hierarchy

Provide visual representation of the hierarchy of a Stateflow object

Description

This component inserts a tree that shows the hierarchy of a given Stateflow object.

Tree Options

- **Construct tree from**: Specifies the object to use for the tree representation.

  - `Current object`
  - `Root of current object`: Starts reporting from the top of the hierarchy.
- **Emphasize current object in tree**: Highlights the current object in the tree representation.
- **Show number of parents**: Specifies the number of parents to include in the tree representation.
- **Show siblings**: Displays siblings in the tree representation.
- **Show children to depth**: Specifies the depth of children to display for each object in the tree representation.

Children

- **Show junctions**: Specifies the level of junction detail to display in the generated report.

  - `All`
  - `Non-redundant`
  - `None`
- **Show transitions**: Specifies the level of transition detail to display in the generated report.

  - `All`
  - `Labeled or non-redundant`

- Non-redundant
- Labeled
- None
- **Skip autogenerated charts under truth tables**: Excludes autogenerated charts under truth tables.

List Formatting

- **List style**:

  - Bulleted list
  - Numbered list: Allows you to specify numbering options in the **Numbering style** section.

    - **Numbering style**: Allows you to specify a numbering style. This setting supports only the RTF/DOC report format.

      - 1,2,3,4...
      - a,b,c,d...
      - A,B,C,D...
      - i,ii,iii,iv...
      - I,II,III,IV...

    To show the parent number in each list entry, select Show parent number in nested list (1.1.a). To show only the current number or letter, select Show only current list value (a).

Insert Anything into Report?

Yes. Tree graphic.

Class

rptgen_sf.csf_hier

See Also

```
Stateflow Hierarchy Loop
```

Stateflow Hierarchy Loop

Run child components on Stateflow object hierarchy

Description

This component runs its child components on the Stateflow object hierarchy.

Loop Options

- **Minimum legible font size**: Specifies the minimum font size to use in the report. The default font size, **8**, is the smallest recommended font size.
- **Skip autogenerated charts under truth tables**; Excludes autogenerated charts under truth tables in the report.
- **Search Stateflow**: Reports on Stateflow charts with specified property name/ property value pairs.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Create link anchor for each object in loop**: Create a link target on each object in the loop so that other parts of the report can link to it.

Insert Anything into Report?

No.

Class

rptgen_sf.csf_hier_loop

See Also

```
Stateflow Hierarchy
```

Stateflow Linking Anchor

Designate locations to which links point

Description

This component designates a location to which other links point. The linking anchor is set to the current object, as defined by the parent component.

This component must have the `Chart Loop`, `State Loop`, `Machine Loop`, or `Stateflow Filter` component as its parent.

Properties

Insert text: Specifies text to appear after the linking anchor.

Insert Anything into Report?

Yes. A link, and possibly text, depending on attribute choices.

Class

`rptgen_sf.csf_obj_anchor`

See Also

`Chart Loop`, `Machine Loop`, `State Loop`, `Stateflow Filter`,

Stateflow Name

Insert into report name of Stateflow object specified by parent component

Description

This component inserts the name of the Stateflow object, as defined by its parent component, into the report. This component must have the State Loop, Chart Loop, or Stateflow Filter component as its parent.

Using this component as the first child component of a Chapter/Subsection component allows the current Stateflow object name to be the chapter or section title.

Properties

- **Display name as**: Displays the Stateflow object name in the report.

    - Name: For example, Object
    - Type Name: For example, Object <ObjectName>
    - Type - Name: For example, Object - <ObjectName>
    - Type: Name: For example, Object: <ObjectName>
- **Display name as**: Specifies the level of detail with which the Stateflow object name displays the report.

    - Object name
    - Object name with Stateflow path
    - Object name with Simulink and Stateflow path

Insert Anything into Report?

Yes. Text.

Class

rptgen_sf.csf_obj_name

See Also

Chapter/Subsection, Chart Loop, State Loop, Stateflow Filter

Stateflow Property

Insert into report table, text, or paragraph with information on selected Stateflow object property

Description

This component inserts a table, text, or paragraph that contains details of the selected Stateflow object property.

Property to Display

Property name: Specifies the Stateflow property name to display.

Display Options

- **Title**: Specifies a title to display in the generated report.

    - `Automatic`: Uses the parameter name as the title.
    - `Custom`: Specifies a custom title.
    - `None`: Specifies no title.

- **Size limit**: Specifies the width of the display in the generated report. Units are in pixels. The size limit for a given table is the hypotenuse of the width and height of the table, `sqrt(w^2+h^2)`. The size limit for text is the number of characters squared. If you exceed the size limit, the variable appears in condensed form.

    Setting a size limit of `0` always displays the variable in long form, regardless of its size.

- **Display as**: Specifies a display style from the menu.

    - `Auto table/paragraph` (default): Displays as a table or paragraph based on the information.
    - `Table`: Displays as a table.
    - `Paragraph`: Displays as a text paragraph.

- `Inline text`: Displays in line with the surrounding text.
- **Ignore if value is empty**: Excludes empty parameters from the generated report.

Insert Anything into Report?

Yes. Text, paragraph, or table.

Class

rptgen_sf.csf_property

See Also

Paragraph, Table, Text, Stateflow Name

Stateflow Property Table

Insert into report property-value table for Stateflow object

Description

This component inserts a property-value table for a Stateflow object into the report. Use the `Stateflow Filter` component as the parent of this component.

For more information on working with Property Table components, see "Property Table Components".

Table

Select a preset table, which is already formatted and set up, in the preset table list in the upper-left corner of the attributes page.

- **preset table**: Specifies a type of table to display the object property table.

  - `Default`
  - `Machine`
  - `Chart`
  - `State`
  - `Truth table`
  - `EM function`
  - `Data`
  - `Event`
  - `Junction`

  To apply a preset table, select the table and click **Apply**.

- **Split property/value cells**: Splits property name/property value pairs into separate cells.

  - For the property name and property value to appear in adjacent horizontal cells, select the **Split property/value cells** check box. In this case, the table is in split

mode, there is only one property name/property value pair per cell. If there is more than one name/property pair in a cell, only the first pair appears in the report. The report ignores all subsequent pairs.

- For the property name and property value to appear together in one cell, clear the **Split property/value cells** check box. This setting is nonsplit mode. Nonsplit mode supports more than one property name/property value pair and text.

- Before switching from nonsplit mode to split mode, make sure that there is only one property name/property value pair per table cell. When there is more than one property name/property value pair or any text in a given cell, only the first property name/property value pair appears in the report. The report omits subsequent pairs and text.

- **Display outer border**: Displays the outer border of the table in the generated report.

- **Table Cells**: Specifies table properties to modify. The selection in this pane affects the available fields in the **Cell Properties** pane.

Cell Properties

The options in the **Title Properties** pane depend on the object selected in the **Table Cells** pane. If you select %<Name>, only **Contents** and **Show** appear. If you select any other object in the **Table Cells** pane, **Lower border** and **Right border** appear.

- **Contents**: Modifies the contents of the table cell selected in the **Table Cells** pane.

- **Alignment**: Justifies the contents of the selected table cell in the **Table Cells** pane.

  - Left
  - Center
  - Right
  - Double justified

- **Show As**: Specifies the format for the contents of the table cell.

  - Value
  - Property Value
  - PROPERTY Value
  - Property: Value

7-171

- • `PROPERTY: Value`
- • `Property - Value`
- • `PROPERTY - Value`
- • **Lower border**: Displays the lower border of the table in the generated report.
- • **Right border**: Displays the right border of the table in the generated report.

Creating Custom Tables

You can edit a preset table, such as the `Blank 4x4` table, to create a custom table. Add and delete rows and add properties. To open the Edit Table dialog box, click **Edit**.

For details about creating custom property tables, see "Property Table Components".

Insert Anything into Report?

Yes. Table.

Class

`rptgen_sf.csf_prop_table`

See Also

`Stateflow Filter`

Stateflow Snapshot

Insert into report snapshot of Stateflow object

Description

This component inserts a snapshot (screen capture) of a Stateflow object, defined by the `Stateflow Filter` parent component, into a report.

This component executes only if the selected object in the `Stateflow Filter` component is a graphical object, such as `Chart`, `State`, `Transition`, and `Frame`.

For HTML and `Direct PDF (from template)` output, the state charts in the resulting image can link to associated report information. To enable this linking, on the `Chart Loop` component that this component is a descendent of, select the **Create link anchor for each object in loop** check box. For `Direct PDF (from template)` output, you also need to set the output format to `Automatic SF format` or `Scalable Vector Graphics`.

Snapshot

- **Format**: Specifies the image file format. Select `Automatic SF Format` (default) to choose the format best suited for the specified report output format automatically. Otherwise, choose an image format that your output viewer can read.

  - `Automatic SF Format` — Select this option, or `Scalable Vector Graphics`, with `Direct PDF (from template)` report output format if you are linking the snapshot to related report content.
  - `Bitmap`
  - `JPEG high quality image`
  - `JPEG low quality image`
  - `JPEG medium quality image`
  - `PNG 24—bit image`
  - `Scalable Vector Graphics`

- **Paper orientation**:

- `Portrait`
- `Landscape`
- `Rotated`
- `Largest dimension vertical`: Positions the image so that its largest dimension is vertical.
- `Use Chart PaperOrientation setting`: Uses the paper orientation setting for the chart. Use the Simulink `PaperOrientation` parameter to specify the orientation.
- `Full page image (PDF only)`: In PDF reports, scales images to fit the full page, minimizes page margins, and maximizes the size of the image by using either a portrait or landscape orientation.

For more information about paper orientation, see the `orient` command in the MATLAB documentation.

- **Image sizing**:

  - `Shrink image to minimum font size specified in Stateflow Hierarchy Loop`: Resizes the image so that the text label font size is the minimum font size.
  - `Fixed` and `Zoom`: Specifies the size of the image.
- **Scaling**: Specifies the percentage of the image size to which to scale it.
- **Maximum size**: Specifies the maximum size for the snapshot in the generated report in the selected units. Use `[width, height]` format. In the units text box, select `Inches`, `Centimeters`, `Points`, or `Normalized`.
- **Use printframe**: Inserts a frame around your image. Use the default frame or create a custom one.
- **Use printframe paper settings**: Uses the dimensions and parameters as set by the specified **printframe** to size your image. If you choose this option, all other options (except for **Image file format**) become inactive.

Properties

- **Include callouts to describe visible objects**: Displays descriptive callouts for visible objects.

- **Insert anchors for transitions and junctions**: Inserts anchors for transitions and junctions into the report.

  - None
  - Redundant children only
  - All

- **Run only if Stateflow object has at least the following number of children**: Specifies the minimum number of children that the current Stateflow object must have to include in the report. This option is inactive unless the selected object in the parent `Stateflow Filter` component is a graphical object.

  Tip This option allows you to exclude certain images to decrease the size of the report for large models.

Display Options

- **Scaling**:

  - Use image size: Uses the image size that you specify in the snapshot option.
  - Zoom and Fixed size: Allows you to specify the size of the image.

- **Size**: Specifies a size in inches for your image. The default is 7-by-9.

- **Max size**: Specifies the maximum size of the snapshot in the format w h (width, height). This field is active only if you choose Zoom from the **Scaling** selection list.

- **Units**: Specifies the units for the size of the snapshot. This field is active only if you choose Zoom or Fixed size in the **Image size** list box.

- **Alignment**: Only reports in PDF or RTF format support this property.

  - Auto
  - Right
  - Center
  - Left

- **Image title**:

  - None(Default).
  - Object name: Uses the object name as the title.

- `Full Stateflow name`: Specifies the Stateflow path and the name of the object.
  - `Full Simulink + Stateflow name`: Specifies the Simulink path and name of the object.
  - `Custom`: Enter a different title.
- **Caption**: Specifies a caption for your image.

  - `None`(Default).
  - `Custom`: Specifies a custom caption.
  - `Description`: Sets the caption to the value of the object `Description` property.

Insert Anything into Report?

Yes. Image.

Class

`rptgen_sf.csf_obj_snap`

Class

`rptgen_sf.csf_prop_table`

See Also

`Stateflow Filter`

Stateflow Summary Table

Table of properties or parameters of specified Stateflow object

Description

This component displays a table of properties or parameters of specified Stateflow objects. It can have the following parents:

- Any Stateflow looping component
- Any Simulink looping component (`Model Loop`, `System Loop`, `Block Loop`, or `Signal Loop`)

Properties

- **Object type**: Specifies the object type to display in the generated report. This value affects the options available in the **Property Columns** pane.
- **Table title**: Specifies a title for the Summary Table in the generated report.

  - `Automatic`: Generates a title automatically from the parameter.
  - `Custom`: Specifies a custom title.

Property Columns

- **Property columns**: Displays the object properties to include in the Summary Table in the generated report.

  - To add a property:

    - Select the appropriate property level in the text box.
    - In the context list under the text box, select the property that you want to add and click **Add**.
  - To delete a property, select the property name and press the **Delete** key.

  Some entries in the list of available properties (such as `Depth`) are "virtual" properties that you cannot access using the `get_param` command. The properties

used for property/value filtering in the block and System Loop components must be retrievable by the `get_param`. Therefore, you cannot configure your Summary Table to report on all blocks of `Depth == 2`.

- **Remove empty columns**: Removes empty columns from the Summary Table in the generated report.
- **Transpose table**: Changes the summary table rows into columns in the generated report, putting the property names in the first column and the values in the other columns.

Object Rows

Insert anchor for each row: Inserts an anchor for each row in the summary table.

Report On

- **Automatic list from context**: Reports on all blocks in the current context, as set by the parent component.
- **Custom - use block list**: Reports on a specified list of blocks. Specify the full path of each block.

Loop Options

Choose block sorting options and reporting options in this pane.

- **Sort blocks**: Specifies how to sort blocks (applied to each level in a model):

  - `Alphabetically by block name`: Sorts blocks alphabetically by name.
  - `Alphabetically by system name`: Sorts systems alphabetically by name. Lists blocks in each system, but in no particular order.
  - `Alphabetically by full Simulink path`: Sorts blocks alphabetically by Simulink path.
  - `By block type`: Sorts blocks alphabetically by block type.
  - `By block depth`: Sorts blocks by their depth in the model.
  - `By layout (left to right)`: Sorts blocks by their location in the model layout, by *rows*. The block appearing the furthest toward the left top corner of the

model is the anchor for the row. The row contains all other blocks that overlap the horizontal area defined by the top and bottom edges of the anchor block. The other rows use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- By `layout (top to bottom)`: Sorts blocks by their location in the model layout, by *columns*. The block appearing the furthest toward the left top corner of the model is the anchor for the column. The column contains all other blocks that overlap the vertical area defined by the left and right edges of the anchor block. The other columns use the same algorithm, using as the anchor the next unreported block nearest the left top of the model.

- By `traversal order`: Sorts blocks by traversal order.

- By `simulation order`: Sorts blocks by execution order.

- **Search for Simulink property name/property value pairs**: Reports on blocks with specified property name/property value pairs.

- **Search Stateflow**: Reports on Stateflow charts with specified property name/property value pairs.

Insert Anything into Report?

Yes. Table.

Class

rptgen_sf.csf_summ_table

See Also

`Block Loop`, `Chart Loop`, `Model Loop`, `Object Loop`, `Signal Loop`, `State Loop`, `Stateflow Hierarchy Loop`, `System Loop`

System Filter

Run child components if current system meets specified conditions

Description

This component runs its child components if the current system meets the conditions that you specify with this component.

Properties

- **Report only if system has at least N blocks**: Specifies the minimum number of blocks that the system must include for any of the child components to run. If you enter 0, child components run regardless of the number of blocks in the system.
- **Report only if system has at least N subsystems**: Specifies the minimum number of subsystems that the system must include for the child components to run. If you enter 0, child components run regardless of the number of subsystems in the system.
- **Report only if system mask type is**: Specifies which masks to include in the generated report.

    - Either masked or unmasked
    - Masked
    - Unmasked

- **Custom filtering MATLAB code**: Specifies custom MATLAB filtering code that the System Filter applies when determining which systems and subsystems to report on in a System Loop component. The edit box includes a sample function (commented out) that you can use as a starting point for your own filtering function. Use the `isFiltered` variable for the output of your function. For example, to filter out systems and subsystems whose names start with `engine`, enter:

    ```
    isFiltered = strncmpi( currentSystem, 'engine', 6);
    ```

Insert Anything into Report?

No.

Class

rptgen_sf.csf_obj_filter

See Also

System Loop

System Hierarchy

Create nested list that shows hierarchy of specified system

Description

This component creates a nested list that shows the hierarchy of a specified system. The list can display all systems in a model, or the parents and children of the current system.

Starting System

- **Build list from**: Specifies the system or model from which to build the list.

    - `Current system`
    - `Current model`

- **Emphasize current system**: Highlights the current system or model in the generated report.

Display Systems

- **Show number of parents**: Specifies the number of parents to list.
- **Display peers of current system**: Shows the peers of the current system in the generated report.
- **Show children to depth**: Specifies the depth of children to list.

List Formatting

- **List style**:

    - `Bulleted list`
    - `Numbered list`: Allows you to select numbering options in the **Numbering style** section.

- **Numbering style**: Allows you to select a numbering style in the selection list, by setting **List style** to Numbered List. Only the RTF/DOC report format supports this option.

  - 1,2,3,4,...
  - a,b,c,d,...
  - A,B,C,D,...
  - i,ii,iii,iv,...
  - I,II,III,IV,...

Insert Anything into Report?

Yes. List.

Class

rptgen_sl.csl_sys_list

See Also

Model Loop, System Loop

System Loop

Specify systems and subsystems on which to loop, as defined by parent component

Description

This component runs its child components for each system defined by the parent component. For example, to include systems and subsystems within a given model in the report, you can include this component as the child of a `Model Loop` component.

For conditional processing systems, you can use the `RptgenSL.getReportedSystem` function. For more information, see "Loop Context Functions" on page 4-55.

Report On

- **Loop on Systems**:

  - **Select systems automatically**: Reports on all systems in the current context as set by the parent component.

    - `Model Loop`: Reports on systems in the current model.
    - `System Loop`: Reports on the current system.
    - `Signal Loop`: Reports on the parent system of the current signal.
    - `Block Loop`: Reports on the parent system of the current block.

    If this component does not have any of these components as its parent, selecting this option reports on all systems in all models.

- **Custom - use system list**: Reports on a list of specified systems. Specify the full path of each system.

- `%<VariableName>`: Inserts the value of a variable from the MATLAB workspace. The `%<>` notation can denote a string or cell array. For more information, see `%<VariableName> Notation` on the `Text` component reference page.

- **Include subsystems in Simulink functions**: Specifies whether to include subsystems in Simulink functions. By default, this option is enabled.

Loop Options

- **Sort Systems**: Specifies how to sort systems.

  - `Alphabetically by system name` (default): Sorts systems alphabetically by name.
  - `By number of blocks in system`: Sorts systems by number of blocks. The list shows systems by decreasing number of blocks; that is, the system with the largest number of blocks appears first in the list.
  - `By system depth`: Sorts systems by their depth in the model.
  - `By traversal order`: Sorts systems in traversal order.

- **Search for**: Reports only on blocks with the specified property name/property value pairs. To enable searching, click the check box. In the first row of the property name and property value table, click inside the edit box, delete the existing text, and type the property name and value. To add a row, use the **Add row** button.

  For information about subsystem property names and values, in "Block-Specific Parameters", see the "Ports & Subsystems Library Block Parameters" section.

Section Options

- **Create section for each object in loop**: Inserts a section in the generated report for each object found in the loop.
- **Display the object type in the section title**: Inserts the object type automatically into the section title in the generated report.
- **Number sections by system hierarchy**: Hierarchically numbers sections in the generated report. Requires that **Sort Systems** be set to `By traversal order`.
- **Create link anchor for each object in loop**: Create a link target for each system in the loop so that other parts of the report can link to it. For example, the image created by a `System Snapshot` component can link to the subsystem section only if you select this check box.

Examples

For an example of how to use this component with a `Model Loop` as its parent, see `Model Loop`.

Insert Anything into Report?

Yes, inserts a section if you select **Create section for each object in loop** and a link target if you select **Create link anchor for each object in loop**.

Class

rptgen_sl.csl_sys_loop

See Also

Block Loop, Model Loop, Signal Loop, System Loop

System Snapshot

Insert snapshot of the current system into report

Description

This component inserts a snapshot (screen capture) of the current system into a report. A `System Snapshot` must be a child or descendant of a `System Loop` component.

For HTML and `Direct PDF (from template)` output, the blocks and subsystems in the resulting image can link to associated report information. For example, a block in the report can link to a block property table. A subsystem can link to the subsystem block diagram or to the block properties.

To enable this linking, select the **Create link anchor for each object in loop** check box on the appropriate loop component:

- For blocks and masked subsystems in this system, select the check box on the `Block Loop` component that reports on the system's blocks.
- To provide links to the sections that report on the unmasked subsystems of this system, select the check box on the `System Loop` component.

For `Direct PDF (from template)` output, you also need to set the output format to `Automatic SL format` or `Scalable Vector Graphics`.

Snapshot Options

- **Format**: Specifies the image file format. Select `Automatic SL Format` (the default) to choose the format best suited for the specified report output format automatically. Otherwise, choose an image format that your output viewer can read.

  - `Automatic SL Format` — Select this option. or `Scalable Vector Graphics`, with `Direct PDF (from template)` report output format if you are linking the snapshot to related report content.
  - `Bitmap`
  - `JPEG high quality image`
  - `JPEG low quality image`

- `JPEG medium quality image`
- `PNG 24–bit image`
- `Scalable Vector Graphics`

- **Orientation**:

  - `Largest dimension vertical`: Positions the image so that its largest dimension is vertical.

  - `Landscape`

  - `Portrait`

  - `Use system orientation`: Uses the paper orientation setting for the system. Use the Simulink `PaperOrientation` parameter to specify the orientation.

  - `Full page image (PDF only)`: In PDF reports, scales images to fit the full page, minimizes page margins, and maximizes the size of the image by using either a portrait or landscape orientation.

- **Scaling**: Controls the size of the image in the image file.

  - `Automatic` (default): Automatically scales the image to output dimensions.

  - `Custom`: Specifies image size.

  - `Zoom`: Enlarges or reduces the image size to the percent that you specify. Use **Max Size** to specify the maximum size other than the default for the image.

Note: Selecting the **Use printframe** deactivates the `Custom` and `Zoom` options and automatically scales the image to the print frame size.

Properties Options

- **Include callouts to describe visible objects**: Displays descriptive callouts for visible objects

- **Use printframe**: Prints a frame around the image. You can use the default frame, `rptdefaultframe.fig`, or use the Frame Editor to build a custom frame. For more information, see the `frameedit` function in Simulink documentation.

  The default frame is five inches wide and four inches high. It includes the name of the system and the model folder. This frame is optimized for use with a `portrait` paper

orientation. The Flight Control Model in the `f14` Simulink model appears here with the default Simulink Report Generator frame option.
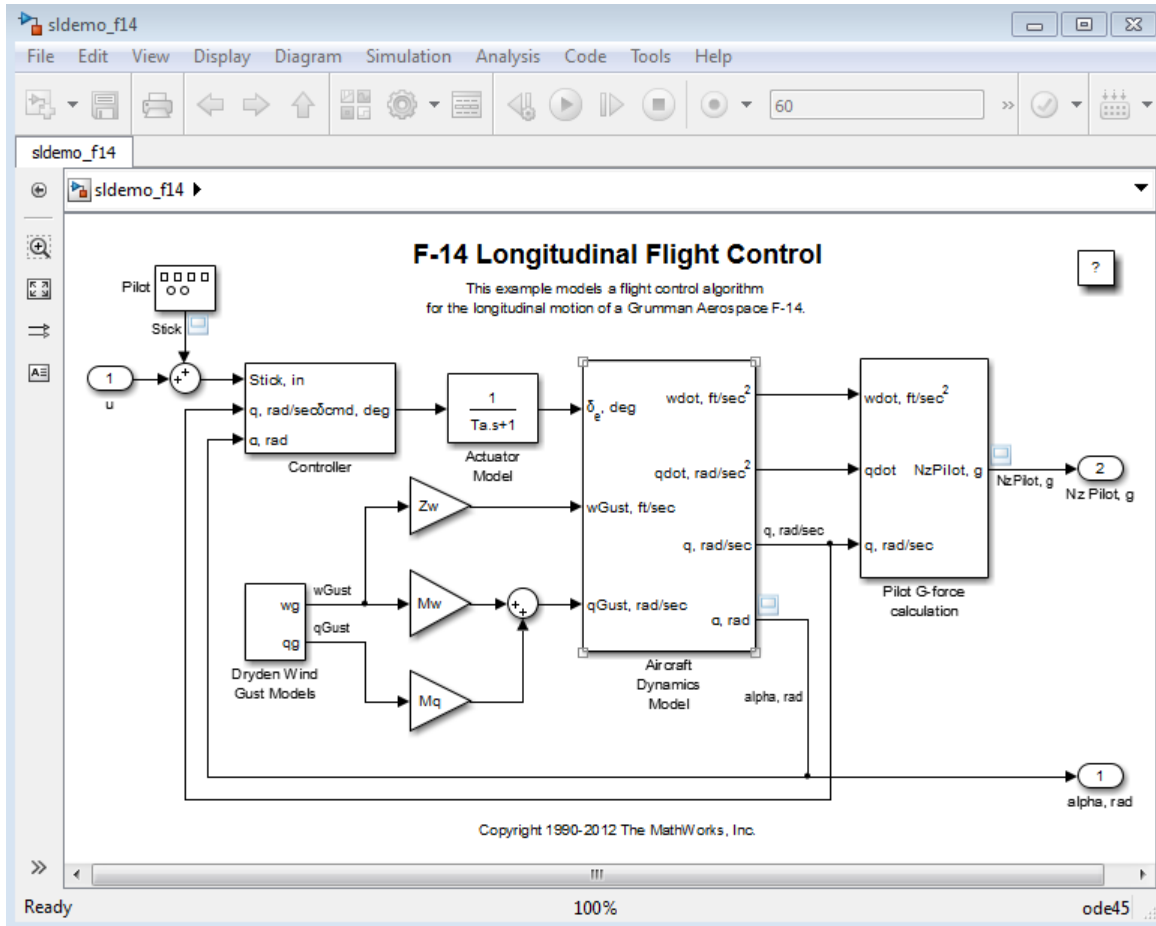
Display Options

To access the display options, click the **Advanced** button.

- **Scaling**: Controls size of the image, as displayed in a browser. Making an image larger using this option does not affect the storage size of the image, but the quality

of the displayed image may decrease as you increase or decrease the size of the displayed image.

Generally, to achieve the best and most predictable display results, use the default setting of `Use image size`.

- `Use image size`: Causes the image to appear the same size in the report as on screen (default).
- `Fixed size`: Specifies the number and type of units.
- `Zoom`: Specifies the percentage, maximum size, and units of measure.

- **Size**: Specifies the size of the snapshot in a browser, using the format `w h` (width, height). This field is active only if you choose `Fixed size` in the **Scaling** selection list.
- **Max size**: Specifies the maximum size of the snapshot in a browser, using the format `w h` (width, height). This field is active only if you choose `Zoom` in the **Scaling** selection list.
- **Units**: Specifies the units for the size of the snapshot in a browser. This field is active only if you choose `Fixed size` in the **Image size** list box.
- **Alignment**: Only reports in `PDF` or `RTF` format support this property.

  - `Auto`
  - `Right`
  - `Left`
  - `Center`

- **Image title**:

  - `None` (Default)
  - `System name`: Uses the system name as the image name.
  - `Full system name`: Uses the system name, with path information, as the image name.
  - `Custom`: Specifies a custom title.

- **Caption**:

  - `None` (Default)
  - `Description (use system description)`
  - `Custom`: Specifies a custom caption.

Insert Anything into Report?

Yes. Image.

Class

rptgen_sl.csl_sys_snap

See Also

System Loop, Block Loop

Test Sequence

Capture Test Sequence block information

Description

This component captures information about Simulink Test™ Test Sequence blocks. The report includes the test sequence using the tabular series of steps from the Test Sequence block.

Test Sequence Block Section Title

Title to use for the Test Sequence block section:

- `Use Test Sequence name` (default): Use the name of the Test Sequence block as the section title.
- `Custom`: Specify a custom section title in the text box.

Step Content

Select the step content to include in the report.

- **Include description, action, and transition table** (default): Include all the step data in the report, i.e., step descriptions, action statements, transition table, and when condition.
- **Include description only**: Include only the description of each step in the report.
- **Include action and transition table only**: Include the action statements, transition table, and when condition in the report.
- **None**: Do not include the description, action, or transition table in the report.
- **Include requirements**: Include links to requirements that are attached to steps in the Test Sequence block.

Insert Anything into Report?

Yes. Test Sequence block name or specified title and optional step information.

Class

```
rptgen_stm.cstm_testseq
```

To Workspace Plot

Capture plot figure created in the MATLAB workspace

Description

This component captures a plot figure created in the MATLAB workspace, and then inserts one or both of the following into the report:

- A table that includes input and output numeric values.
- A figure that plots the values included in the table.

Print Options

- **Image file format**: Specifies the image file format (for example, `JPEG` or `TIFF`) from this list. Select `Automatic HG Format` (the default) to choose the format best suited for the specified report output format automatically. Otherwise, choose an image format that your output viewer can read. Other options are:

  - `Automatic HG Format` (Uses the file format selected in the Preferences dialog box)
  - `Bitmap (16m-color)`
  - `Bitmap (256-color)`
  - `Black and white encapsulated PostScript`
  - `Black and white encapsulated PostScript (TIFF)`
  - `Black and white encapsulated PostScript2`
  - `Black and white encapsulated PostScript2 (TIFF)`
  - `Black and white PostScript`
  - `Black and white PostScript2`
  - `Color encapsulated PostScript`
  - `Color encapsulated PostScript (TIFF)`
  - `Color encapsulated PostScript2`
  - `Color encapsulated PostScript2 (TIFF)`

- Color PostScript
- Color PostScript2
- JPEG high quality image
- JPEG medium quality image
- JPEG low quality image
- PNG 24-bit image
- TIFF - compressed
- TIFF - uncompressed
- Windows metafile

- **Paper orientation**:

  - Landscape
  - Portrait
  - Rotated
  - Use figure orientation: Uses the orientation for the figure, which you set with the orient command.
  - Full page image (PDF only): In PDF reports, scales images to fit the full page, minimizes page margins, and maximizes the size of the image by using either a portrait or landscape orientation.

  For more information about paper orientation, see the orient command in the MATLAB documentation.

- **Image size**:

  - Use figure PaperPositionMode setting: Uses the PaperPositionMode property of the Handle Graphics figure to set the image size in the report. For more information about paper position mode, see the orient command in the MATLAB documentation.

  - Automatic (same size as on screen): Sets the image in the report to the same size as it appears on the screen.

  - Custom: Specifies a custom image size. Specify the image size in the size and units fields.

- **Size**: Specifies the size of the Handle Graphics figure snapshot in the format `wxh` (width times height). This field is active only if you choose `Custom` in the **Image size** list box.

- **Units**: Specifies units for the size of the Handle Graphics figure snapshot. This field is active only if you choose `Set image size` in the **Custom** list box.

- **Invert hardcopy**: Uses the Handle Graphics figures `InvertHardcopy` property, which inverts colors for printing; it changes dark colors to light colors, and light colors to dark colors.

  - `Automatic`: Automatically changes dark axes colors to light axes colors. If the axes color is a light color, it is unchanged.

  - `Invert`: Changes dark axes colors to light axes colors, and light axes colors to dark axes colors.

  - `Don't invert`: Retains image colors displayed on screen in the printed report.

  - `Use figure's InvertHardcopy setting`: Uses the `InvertHardcopy` property set in the Handle Graphics image.

  - `Make figure background transparent`: Makes the image background transparent.

Display Options

- **Scaling**: Controls size of the image, as displayed in a browser. Making an image larger using this option does not affect the storage size of the image, but the quality of the displayed image may decrease as you increase or decrease the size of the displayed image.

  Generally, to achieve the best and most predictable display results, use the default setting of `Use image size`.

  - `Use image size`: Causes the image to appear the same size in the report as on screen (default).

  - `Fixed size`: Specifies the number and type of units.

  - `Zoom`: Specifies the percentage, maximum size, and units of measure.

- **Size**: Specifies the size of the snapshot in the format `w h` (width, height). This field is active only if you choose `Fixed size` in the **Scaling** list.

- **Max size**: Specifies the maximum size of the snapshot in the format w h (width, height). This field is active only if you choose Zoom from the **Scaling** list.
- **Units**: Specifies units for the size of the snapshot. This field is active only if you choose Zoom or Fixed size in the **Image size** list box.
- **Alignment**: Only reports in PDF or RTF format support this property.

  - Auto
  - Right
  - Left
  - Center
- **Title**: Specifies text to appear above the snapshot.
- **Caption**: Specifies text to appear under the snapshot.

Insert Anything into Report?

Yes. Figure.

Class

rptgen_sl.csl_blk_toworkspace

See Also

Figure Loop

Truth Table

Report on truth tables in Simulink and Stateflow models

Description

The Truth Table component reports on truth tables in Simulink and Stateflow models. It displays both the condition table and the action table. The parent component of the Truth Table determines its behavior.

- `Model Loop`: Reports on all truth tables in the current model.
- `System Loop`: Reports on all truth tables in the current system.
- `Block Loop` : Reports on all truth tables in the current block.
- `Signal Loop`: Reports on all truth tables in the current signal.

Title

Title: Specifies a title for the truth table.

- `No title`
- `Use Stateflow name`
- `Custom`

Condition Table

Specify display parameters for the condition table.

- **Show header**: Displays the column headers in the table.
- **Show number**: Displays the condition number column in the table.
- **Show condition**: Displays the condition column in the table.
- **Show description**: Displays the description column in the table.
- **Wrap if column count**: Specifies how many columns to display before creating a table continuation. If the specified number is greater than the number of columns that can appear on the page, some columns do not appear in the report.

Action Table

- **Show header**: Displays the column headers in the table.
- **Show number**: Displays the condition number column in the table.
- **Show condition**: Displays the condition column in the table.
- **Show description**: Displays the description column in the table. If you do not select this option, no action table appears in the report.

Insert Anything into Report?

Yes. Table.

Class

`rptgen_sf.csf_truthtable`

See Also

`Block Loop`, `Model Loop`, `Signal Loop`, `System Loop`

Functions – Alphabetical List

slwebview

Export Simulink models to Web views

Syntax

```
slwebview
filename = slwebview(system_name)
filename = slwebview(system_name,Name,Value)
```

Description

slwebview starts the Web View dialog box in the Report Explorer.

filename = slwebview(system_name) exports the subsystem system_name and its child systems to the file filename.

filename = slwebview(system_name,Name,Value) provides additional options specified by one or more Name,Value pairs.

Examples

Export Web View for a Subsystem and Systems that Contain that Subsystem

Open the fuel rate controller subsystem.

```
open_system('fuelsys')
```

Export to a Web view the fuel rate controller subsystem and the system that contains it. Do not export the subsystems that it contains. This example assumes the current folder is the H: drive.

```
fuelsys_web_view = slwebview...
('fuelsys/fuel rate controller','SearchScope','CurrentAndAbove')

fuelsys_web_view =
```

```
H:\fuel_rate_controller\webview.html
```

The Web view displays in the system browser.
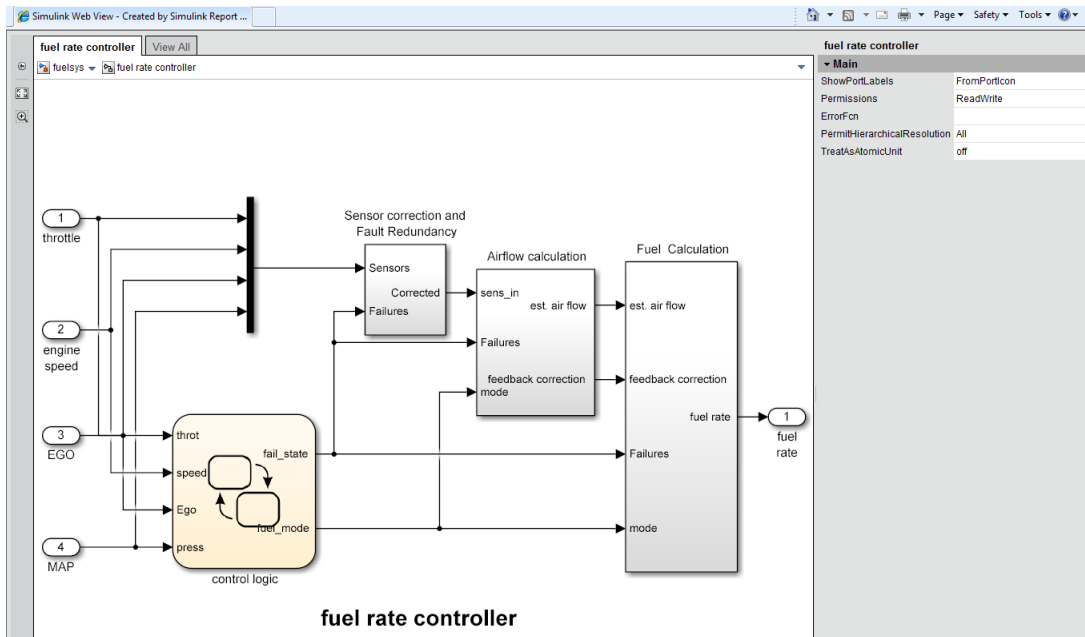
Export Web View with Access to Referenced Models

Open the `sldemo_mdlref_depgraph` model.

```
open_system('sldemo_mdlref_depgraph')
```
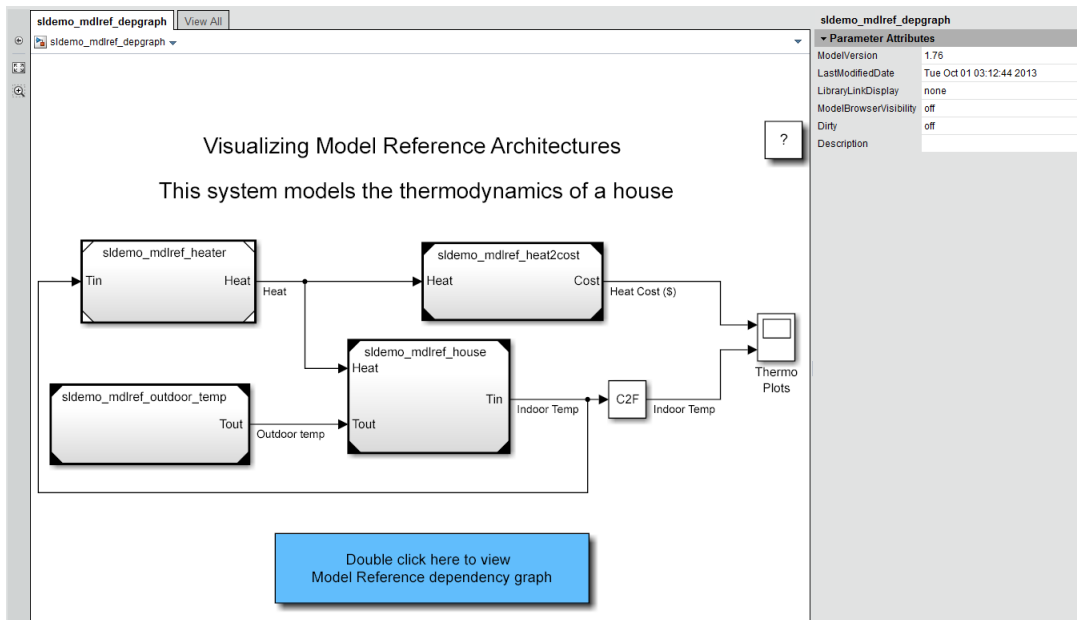
Export to a Web view the `sldemo_mdlref_depgraph` model and allow access to the models it references.

```
depgraph_web_view = slwebview...
('sldemo_mdlref_depgraph','FollowModelReference','on')

depgraph_web_view =

H:\sldemo_mdlref_depgraph\webview.html
```

The Web view displays in the system browser. In the Web view, you can open the models referenced by the Model blocks.

Click a Model block to see its properties. Double-click a Model block to display the referenced model.

- "Create and Use a Web View" on page 5-14

Input Arguments

`system_name` — The system to export to a Web view file
string containing the path to the system | handle to a subsystem or block diagram | handle to a chart or subchart

Exports the specified system or subsystem and its child systems to a Web view file. By default, child systems of the `system_name` system are also exported. Use the `SearchScope` name-value pair to export other systems, in relation to `system_name`.

Name-Value Pair Arguments

Specify optional comma-separated pairs of `Name,Value` arguments. `Name` is the argument name and `Value` is the corresponding value. `Name` must appear inside single

quotes (' '). You can specify several name and value pair arguments in any order as
`Name1,Value1,...,NameN,ValueN`.

Example: `htmlFileName = slwebview(gcs,'LookUnderMasks','all',...`
`'FollowLinks','on')` Export to a Web view all layers of the model hierarchy to which
the current system belongs, including the ability to interact with library links and masks.

`'SearchScope'` — Systems to export, relative to the `system_name` system
`'CurrentAndBelow'` (default) | `'Current'` | `'CurrentAndAbove'` | `'All'`

`'CurrentAndBelow'` exports the Simulink system or the Stateflow chart specified by
`system_name` and all systems or charts that it contains.

`'Current'` exports only the Simulink system or the Stateflow chart specified by
`system_name`.

`'CurrentAndAbove'` exports the Simulink system or the Stateflow chart specified by
the `system_name` and all systems or charts that contain it.

`'All'` exports all Simulink systems or Stateflow charts in the model that contains the
system or chart specified by `system_name`.

Data Types: `char`

`'LookUnderMasks'` — Specifies whether to export the ability to interact with masked blocks
`'none'` (default) | `'all'`

`'none'` does not export masked blocks in the Web view. Masked blocks are included in
the exported systems, but you cannot access the contents of the masked blocks.

`'all'` exports all masked blocks.

Data Types: `char`

`'FollowLinks'` — Specifies whether to follow links into library blocks
`'off'` (default) | `'on'`

`'off'` does not allow you to follow links into library blocks in a Web view.

`'on'` allows you to follow links into library blocks in a Web view.

Data Types: `char`

`'FollowModelReference'` — Specifies whether to access referenced models in a Web view
`'off'` (default) | `'on'`

'off' does not allow you to access referenced models in a Web view.

'on' allows you to access referenced models in a Web view.

Data Types: char

'ViewFile' — Specifies whether to display the Web view in a Web browser when you export the Web view
true (default) | false

true displays the Web view in a Web browser when you export the Web view.

false does not display the Web view in a Web browser when you export the Web view.

Data Types: logical

'ShowProgressBar' — Specifies whether to display the status bar when you export a Web view
true (default) | false

true displays the status bar when you export a Web view.

false does not display the status bar when you export a Web view.

Data Types: logical

Output Arguments

filename — The name of the HTML file for displaying the Web view
string

Reports the name of the HTML file for displaying the Web view. Exporting a Web view creates the supporting files, in a folder.

More About

Tips

A Web view is an interactive rendition of a model that you can view in a Web browser. You can navigate a Web view hierarchically to examine specific subsystems and to see properties of blocks and signals.

You can use Web views to share models with people who do not have Simulink installed.
Web views require a Web browser that supports Scalable Vector Graphics (SVG).

Introduced in R2006a