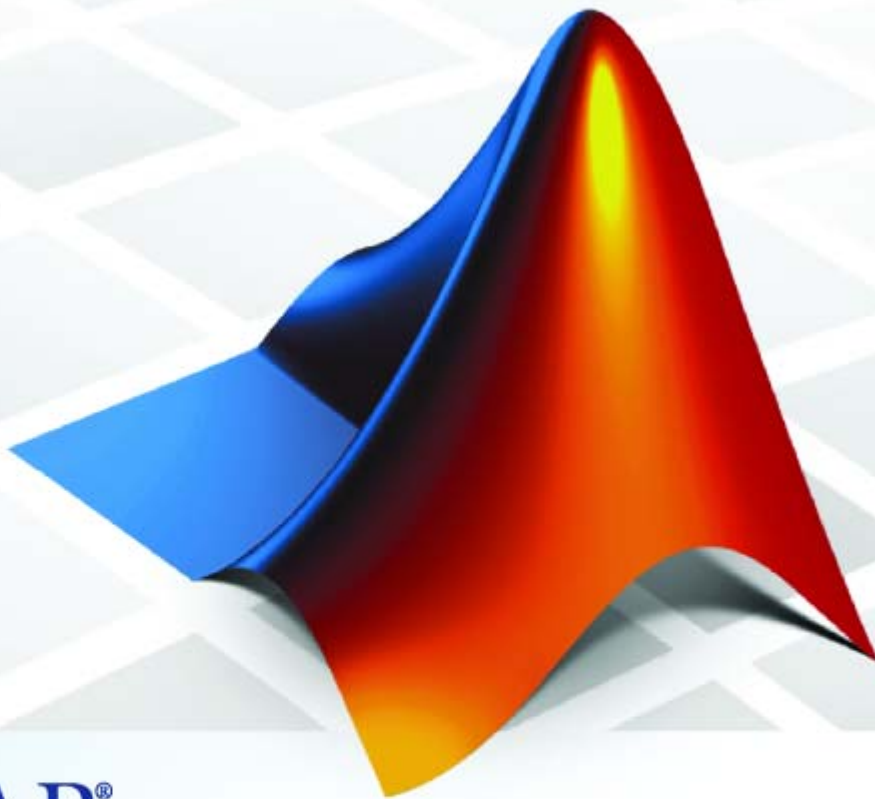


MATLAB® Report Generator 3

User's Guide



MATLAB®

How to Contact The MathWorks



www.mathworks.com
comp.soft-sys.matlab
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



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

For contact information about worldwide offices, see the MathWorks Web site.

MATLAB® Report Generator User's Guide

© COPYRIGHT 1999–2007 by The MathWorks, Inc.

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

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

Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, SimBiology, SimHydraulics, SimEvents, and xPC TargetBox are registered trademarks and The MathWorks, the L-shaped membrane logo, Embedded MATLAB, and PolySpace are trademarks of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Patents

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

Revision History

January 1999	First printing	New (Release 11)
December 2000	Second printing	Revised (Release 12)
June 2004	Third printing	Revised for Version 2.02 (Release 14)
August 2004	Online only	Revised for Version 2.1
October 2004	Online only	Revised for Version 2.1.1 (Release 14SP1)
December 2004	Online only	Revised for Version 2.2 (Release 14SP1+)
April 2005	Online only	Revised for Version 2.2.1 (Release 14SP2+)
September 2005	Online only	Revised for Version 2.3.1 (Release 14SP3)
March 2006	Online only	Revised for Version 3.0 (Release 2006a)
September 2006	Online only	Revised for Version 3.1 (Release 2006b)
March 2007	Fourth printing	Revised for Version 3.2 (Release 2007a)
September 2007	Fifth printing	Revised for Version 3.2.1 (Release 2007b) This publication was previously for MATLAB and Simulink. It is now for MATLAB only.

Getting Started

1

What Is MATLAB Report Generator?	1-2
MATLAB Report Generator Workflow	1-3
How MATLAB Report Generator Interacts with MATLAB	1-4
Supported Report Formats	1-6

Creating Reports in MATLAB

2

About This Tutorial	2-2
Setting Report Options in the Report Template	2-3
Adding Components to the Report Template	2-8
About Components	2-8
Setting Initial Values	2-10
Creating a Title Page	2-13
Adding the First Chapter	2-16
Adding Components to the First Chapter	2-17
Creating the Magic Squares and Their Images	2-24
Generating the Report	2-41

3

Setting Report Output Options	3-2
Setting Preferences	3-2
Setting the Report File Directory	3-3
Setting the Report File Name	3-4
Setting the Report Format	3-4
Viewing and Printing Reports	3-8
Converting English Strings to Other Languages	3-9
Autosaving Report Templates	3-9
Regenerating Images	3-9
Including a Report Description	3-10
Converting XML Documents	3-11
Why Convert XML Documents?	3-11
Converting XML Documents Using the Report Explorer ..	3-11
Converting XML Documents Using the Command Line ..	3-13
Editing XML Source Files	3-14
Creating Log Files	3-15
Generating M-code from Report Templates	3-16
Working with Legacy Report Templates	3-17
What Are Legacy Report Templates?	3-17
Enabling Legacy MATLAB Report Generator Interfaces ..	3-17
Using Legacy MATLAB Report Generator Interfaces	3-18
Troubleshooting	3-19
Managing MATLAB Report Generator Memory Usage ...	3-19
Displaying HTML Reports on UNIX	3-20

Working with Components

4

About Components	4-2
-------------------------------	------------

Working with Looping Components	4-3
About Looping Components	4-3
Editing the Figure Loop Tutorial Report Template	4-3
Creating Handle Graphics Figures	4-3
Figure Properties	4-5
Editing Figure Loop Components	4-6
Working with Property Table Components	4-9
About Property Table Components	4-9
Opening the Example Report Template	4-11
Examining the Property Table Output	4-11
Selecting Object Types	4-12
Displaying Property Name/Property Value Pairs	4-12
Editing Table Titles	4-16
Entering Text into Table Cells	4-16
Adding, Replacing, and Deleting Properties in Tables	4-17
Displaying or Hiding Cell Borders	4-18
Adding and Deleting Columns and Rows	4-19
Resizing Columns	4-19
Zooming and Scrolling	4-19
Selecting Tables	4-20
Working with Summary Table Components	4-21
About Summary Table Components	4-21
Opening the Example Report Template	4-23
Selecting Object Types	4-23
Adding and Removing Properties	4-23
Setting Relative Column Widths	4-24
Setting Object Row Options	4-24

Creating Custom Components

5

About Custom Components	5-2
Component Creation Process	5-3
Defining Components	5-6
Required Component Data	5-6

Specifying the Location of Component Files	5-6
Setting Component Display Options	5-7
Specifying Component Properties	5-9
Modifying Existing Components	5-13
Building Components	5-13
Rebuilding Existing Components	5-13
Removing a Component	5-13
Defining Component Tasks	5-15
About Component Customization	5-15
Required Customization: Specifying Format and Content of Report Output	5-16
Changing a Component's Outline String in the Report Explorer Hierarchy	5-18
Modifying the Appearance of Properties Dialog Boxes	5-19
Specifying Additional Component Properties	5-19
Examples	5-22
Fetching Securities Data and Displaying It in a Table	5-22
Displaying Securities Data in Two Tables	5-27

Creating Custom Stylesheets

6

About Stylesheets	6-2
Built-in vs. Custom Stylesheets	6-2
Customizing Stylesheets Using Data Items	6-3
Working with Stylesheets	6-5
Using the Report Explorer to Edit Stylesheets	6-5
Creating New Stylesheets	6-8
Saving Stylesheets	6-8
Deleting Stylesheets	6-9
Editing Stylesheet Data Items	6-10
Categories of Data Items in Built-in Stylesheets	6-10
Editing Data Items in Simple vs. Advanced Edit Mode	6-14
Working with Data Items	6-15

Using Stylesheet Cells to Manage PDF Report Header and Footer Content	6-21
About Stylesheet Cells and Cell Groups	6-21
Working with Headers and Footers	6-22
Using Templates to Add Content to Headers and Footers ..	6-24
Inserting Graphics Files	6-25
Modifying Fonts and Other Properties	6-26
Examples	6-27
Numbering Pages in a Report	6-27
Adding Graphics to Headers in PDF Reports	6-28
Changing Font Size, Page Orientation, and Paper Type of a Generated Report	6-33
Editing Font Size as a Derived Value in XML	6-36

MATLAB Components — By Category

7

Formatting	7-2
Handle Graphics	7-4
Logical and Flow	7-6
MATLAB	7-7
MATLAB Report Generator	7-8

MATLAB Components — Alphabetical List

8

Examples

A

Working with Components	A-2
Customizing Components	A-2
Customizing Stylesheets	A-2

Index

Getting Started

What Is MATLAB Report Generator? (p. 1-2)	Describes tasks you can perform using MATLAB® Report Generator
MATLAB Report Generator Workflow (p. 1-3)	Illustrates a typical workflow for MATLAB Report Generator
How MATLAB Report Generator Interacts with MATLAB (p. 1-4)	Describes how MATLAB Report Generator and MATLAB® interact to create reports
Supported Report Formats (p. 1-6)	Document formats that MATLAB Report Generator supports

What Is MATLAB Report Generator?

You can use MATLAB Report Generator to:

- Document tasks you perform in MATLAB such as analyzing and visualizing data and developing algorithms
- Create a report template that can be reused and that reflects your own styles and standards
- Keep your documentation and specifications up to date with your workflow
- Create user manuals that accurately capture your application capabilities and specifications

MATLAB Report Generator creates documentation (called *reports*) about MATLAB data and workflow. You can customize report templates to include:

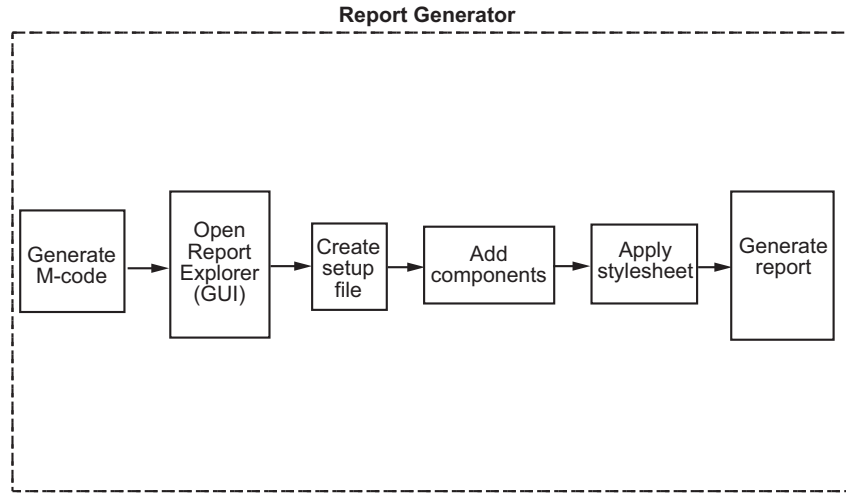
- MATLAB code and its output
- Workspace data
- Graphics created in MATLAB
- Logical components like IF, THEN, ELSE, and WHILE that allow conditional report generation
- Loops that perform a series of operations on multiple objects
- User-defined text, styles, and components

MATLAB Report Generator Workflow

MATLAB Report Generator interacts with MATLAB to give you access to report generation capabilities. Most people use a workflow similar to the one described in this section. Basic workflow steps are:

- 1** Open the Report Explorer, the graphical user interface (GUI) that helps you design and lay out your report by entering report at the MATLAB command line.
- 2** Create a new report template. For details about report templates, see Chapter 2, “Creating Reports in MATLAB”.
- 3** Add existing components to the report template or create your own custom components. For details on using components, see Chapter 4, “Working with Components”.
- 4** Choose an existing stylesheet or create a new stylesheet to apply styles and standards to the report. For details on stylesheets and attributes, see Chapter 6, “Creating Custom Stylesheets”.
- 5** Generate the report.

The following figure illustrates a typical workflow for MATLAB Report Generator.



How MATLAB Report Generator Interacts with MATLAB

MATLAB Report Generator interacts with MATLAB to create reports. You can access the Report Explorer GUI from the MATLAB command line.

The following table describes these interactions in detail.

User Interface	Interaction with MATLAB Report Generator	Purpose
Report Explorer	<p>The Report Explorer is the graphical user interface (GUI) for MATLAB Report Generator that allows you to:</p> <ul style="list-style-type: none"> • Create and modify report templates • Apply stylesheets to the report format • Choose output file format • Generate reports 	<p>Use the Report Explorer to edit existing report templates, components, stylesheets, and attributes, or to customize your own.</p>
MATLAB command line	<p>Enter commands at the MATLAB command line to:</p> <ul style="list-style-type: none"> • Start the Report Explorer • Create and modify report template files • Apply stylesheets • Specify output formats for reports • Generate reports 	<p>The following MATLAB commands work with MATLAB Report Generator:</p> <ul style="list-style-type: none"> • <code>report</code> — Start the Report Explorer. • <code>setedit</code> — Edit a report template with the Report Explorer. • <code>rptconvert</code> — Convert a MATLAB Report Generator source file to the desired output format. • <code>rptlist</code> — List <code>.rpt</code> files in the current path.

Supported Report Formats

When MATLAB Report Generator first creates a report, it generates a DocBook XML source file. You can customize this XML as needed. For more information on how to customize DocBook XML, see the OASIS DocBook TC Web page.

Next, MATLAB Report Generator converts the XML source to one of these user-specified report formats:

- Rich Text Format (RTF)
- Hypertext Markup Language (HTML)
- Microsoft Word (.doc)
- Adobe Acrobat PDF

Note PDF reports only support bitmap (.bmp) and jpeg (.jpg) images.

Creating Reports in MATLAB

About This Tutorial (p. 2-2)	Describes MATLAB Report Generator features you use in this tutorial
Setting Report Options in the Report Template (p. 2-3)	How to create a report template and specify options for the format and appearance of the generated report
Adding Components to the Report Template (p. 2-8)	How to build a report by adding components to its template
Generating the Report (p. 2-41)	How to generate a report after defining its components

About This Tutorial

MATLAB Report Generator tasks demonstrated in this tutorial include the following:

- Evaluating MATLAB expressions
- Formatting reports with multiple chapters
- Running loops and flow control
- Handling errors
- Inserting workspace data
- Inserting images created by MATLAB

This tutorial takes you through the steps required to design a report template file and generate a report from it in MATLAB.

The report template that you design creates several magic squares (A magic square is a matrix whose columns, rows, and diagonal all add up to the same number) using M-code, and then displays them in the generated report as matrices or images, depending on their size.

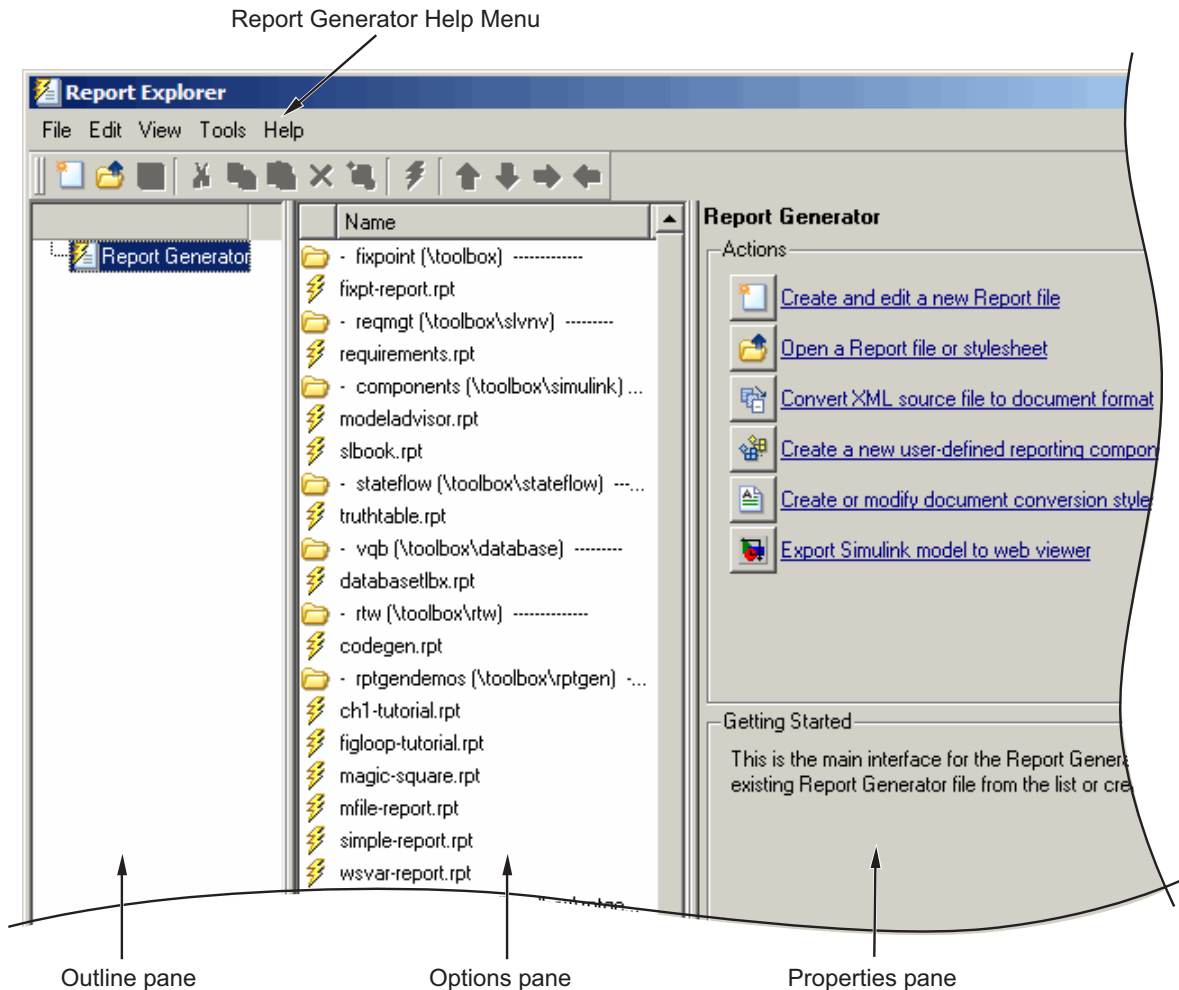
For more information on magic squares, see “Matrices and Magic Squares” in the MATLAB documentation.

Note You do not need to know MATLAB to use this tutorial, but it might be helpful for understanding the M-code that executes during report generation.

Setting Report Options in the Report Template

To create and configure a report template:

- 1 Start MATLAB.
- 2 Type report in the MATLAB Command Window. The Report Explorer opens.



Tip If the Report Explorer window opens with only two panes, one of the panes is hidden. The vertical boundaries can be moved to reveal any hidden pane.

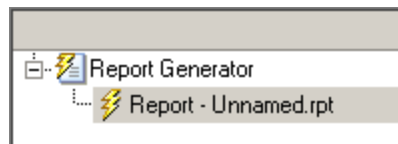
The Report Explorer has three panes:

- The *Outline pane* on the left shows the hierarchy of components in currently opened reports. Report components can reside within other report components, creating parent, child, and sibling relationships.
- The *Options pane* in the middle lists the options available in the context of the Outline pane. If no report is open, the Options pane lists available reports. When a report is open, the Options pane lists components available to insert into the report template. When a stylesheet is open, the Options pane lists available attributes.
- If no report is open, the *Properties pane* on the right displays tasks the Report Explorer can perform. If a report is open, the Properties pane displays the properties for the element that is currently selected in the Options pane.

These panes can be made wider or narrower by dragging the vertical boundaries between the panes.

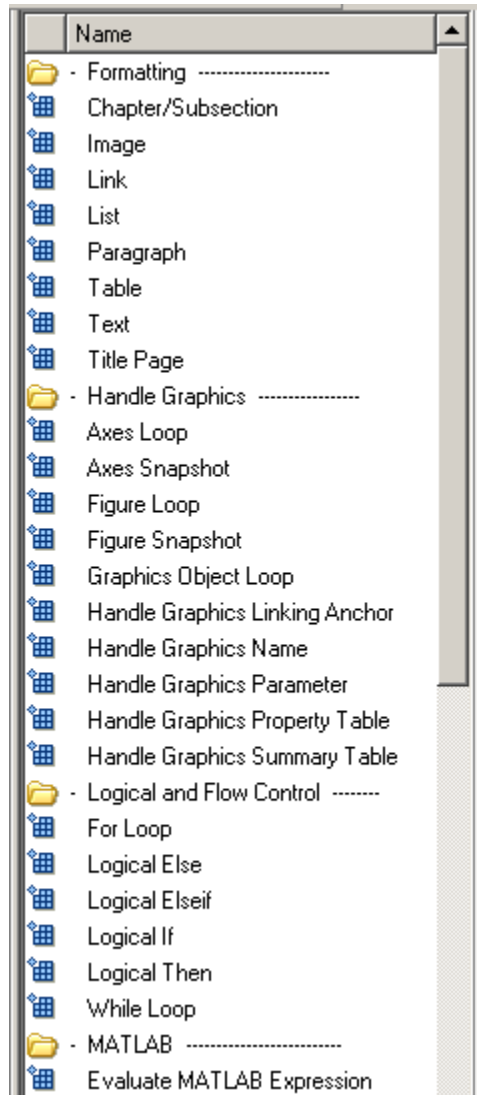
- 3 Select **File > New** to create a new report. Your new report will be created with the default name `Unnamed.rpt`.

The three Report Explorer panes have the following contents.



Outline Pane

The Outline pane initially contains the top level of the report.



Options Pane

The Options pane lists by category components that are available for adding to reports. In this figure, the visible component categories are Formatting, Handle Graphics®, Logical and Flow Control, and MATLAB. For details

about report components, see Chapter 7, “MATLAB Components — By Category”.

The folder icon indicates a component category. The blue square icon indicates a component.

The screenshot shows the 'Report Options' dialog box, which is organized into several sections:

- Report File Location:** Contains a text field with the path 'C:\WINNT\Unnamed.htm' and a 'View...' button. Below it are two rows of controls: 'Directory:' with a dropdown menu set to 'Same as setup file' and a browse button (...); 'Filename:' with a dropdown menu set to 'Same as setup file' and a text field containing 'index', with a browse button (...).
- Checkboxes:** An unchecked checkbox labeled 'If report already exists, increment to prevent overwriting'.
- Report Format and Stylesheet:** Contains two dropdown menus: 'File format:' set to 'web (HTML)' and 'Default HTML stylesheet'.
- Generation Options:** Contains two checkboxes: 'View report after generation' (checked) and 'Auto save before generation' (unchecked). Below them is a text field labeled 'Evaluate this string after generation:'.
- Report description:** A large text area containing the text 'A report'.

Properties Pane

After you create a new report, the Properties pane initially displays properties for the report as a whole.

4 In the Properties pane on the right:

- a To save the report in the current working directory, select Present Working Directory from the **Directory** list.
- b Set **File format** to web (HTML) to create the report as an HTML file.
- c In the **Report description** text box, replace the existing text with the following text.

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

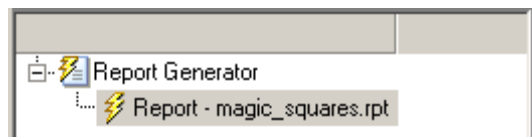
This report creates a series of magic squares and displays them as images.

A magic square is a matrix in which the columns, rows, and diagonal all add up to the same number.

Note When you change a Properties pane field, the background changes to a different color (the default is a cream color), meaning that there are unapplied changes to that field. As soon as you perform any action with another component, the changes are applied and the background color becomes white again.

- 5 Save your report. Select **File > Save As** and name your report's report template magic_squares.rpt.

The new file name appears in the Outline pane on the left.



Adding Components to the Report Template

In this section...
“About Components” on page 2-8
“Setting Initial Values” on page 2-10
“Creating a Title Page” on page 2-13
“Adding the First Chapter” on page 2-16
“Adding Components to the First Chapter” on page 2-17
“Creating the Magic Squares and Their Images” on page 2-24

About Components

Report components specify what information you want to include in the report. The following figure shows a sample page from the report you create in this tutorial and what components are used to produce it.

Note Report components added to the report template must not be deactivated in order for the report to be correctly generated.

Title Page component

Magic Squares

Columns, Rows, Diagonals: Everyone is Equal

Albrecht Durer

Copyright © 1998 The MathWorks

25-Oct-2006 10:37:18

Abstract

An introduction to Magic Squares and their meaning.

Table of Contents

- [1. Magic Squares Explained](#)
- [2. Magic Square Size 4](#)
- [3. Magic Square Size 8](#)
- [4. Magic Square Size 16](#)
- [5. Magic Square Size 32](#)

Chapter component

Chapter 1. Magic Squares Explained

Text component

MAGIC Magic square.
 MAGIC(N) is an N-by-N matrix constructed from the integers 1 through N^2 with equal row, column, and diagonal sums.
 Produces valid magic squares for all $N > 0$ except $N = 2$.

Text component

The German artist Albrecht Durer (1471-1528) created many woodcuts and prints with religious and scientific symbolism. One of his most famous works, Melancholia I, explores the depressed state of mind which opposes inspiration and expression. Renaissance astrologers believed that the Jupiter magic square (shown in the upper right portion of the image) could aid in the cure of melancholy. The engraving's date (1514) can be found in the lower row of numbers in the square.

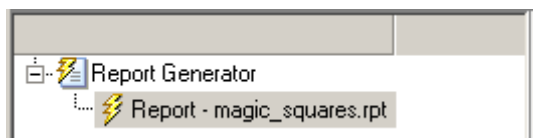
Figure Snapshot component



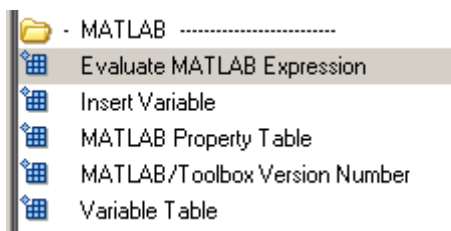
Setting Initial Values

To specify the sizes of the magic squares to be included in the report, use the Evaluate MATLAB Expression component.

- 1 In the Outline pane on the left, select the top level of the report template.



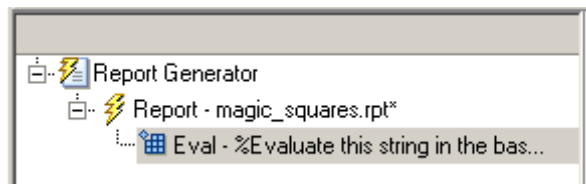
- 2 In the Options pane in the middle, under the MATLAB category, select Evaluate MATLAB Expression.



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

You cannot edit the component information in the Properties pane until you have added the component to the report.

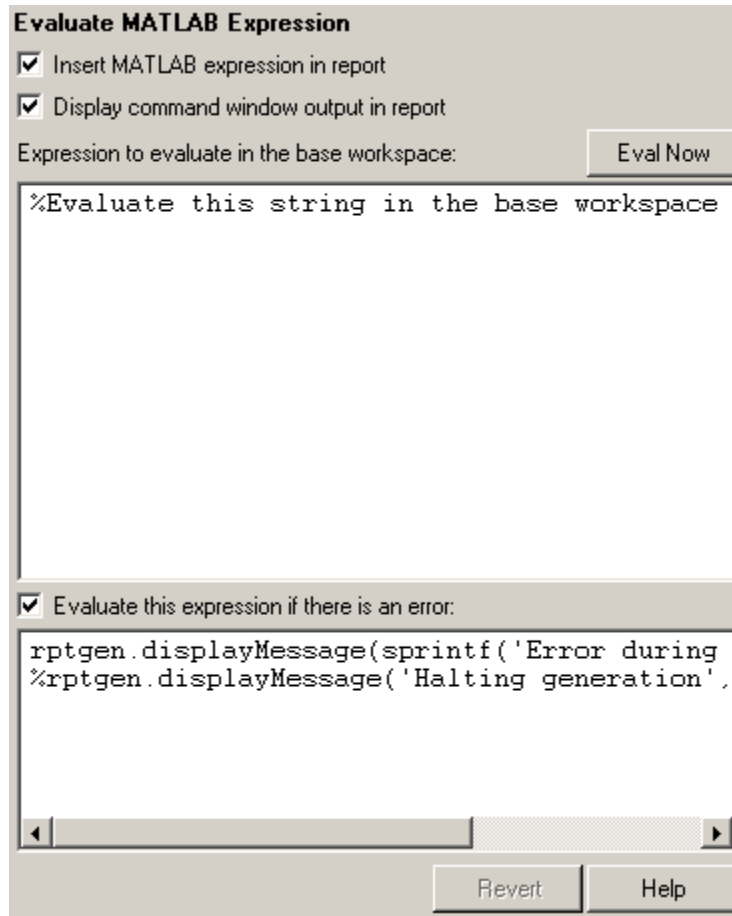
In the Outline pane on the left, the Eval component is listed under the magic_squares report.



The icon in the upper left corner of the Eval component indicates that this component cannot have child components. By default, any components

you add with the Eval component selected are added as siblings to this component.

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



- 4 Clear the **Insert MATLAB expression in report** and **Display command window output in report** check boxes. You do not want to include the MATLAB code details or its output in this report.

- 5** In the **Expression to evaluate in the base workspace** text box, replace the existing text with the following M-code.

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

```
%This M-code sets up two variables
%that define how the report runs.
%magicSizeVector is a list of MxM
%Magic Square sizes to insert into
%the report. Note that magic
%squares cannot be 2x2.
```

```
magicSizeVector=[4 8 16 32];
```

```
%largestDisplayedArray sets the
%limit of array size that will be
%inserted into the report with the
%Insert Variable component.
```

```
largestDisplayedArray=15;
```

- 6** In the **Evaluate this expression if there is an error** text box, replace the existing text with the following text.

```
disp(['Error during eval: ', lasterr])
```

This causes an error to display if the M-code fails.

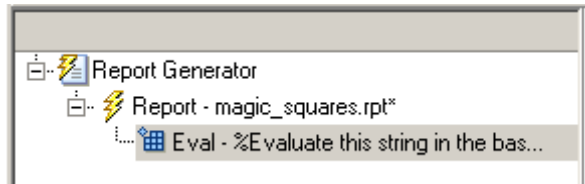
Tip Click the **Eval Now** button at the top right corner of the Report Explorer window to execute the commands that you specified in your MATLAB expression immediately. This is an easy way to check if your commands are correct and will not result in problems at run time.

- 7** Save the report by clicking **File > Save**.

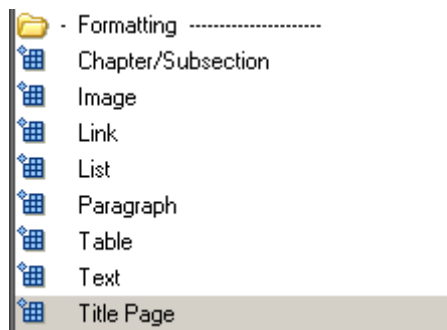
Creating a Title Page

To create a title page for the report, use the Title Page component.

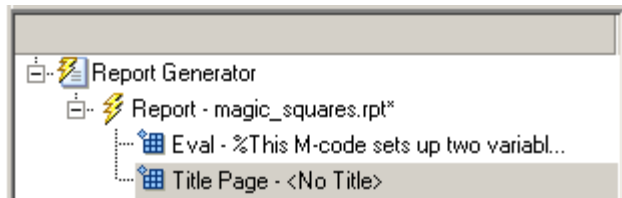
- 1 In the Outline pane on the left, select the Eval component.



- 2 In the Options pane in the middle, under the Formatting category, double-click Title Page to add the component to the report.



Because the Eval component's icon indicates that this component cannot have children, the Title Page component is added as a sibling of the Eval component. Likewise, the Title Page component cannot have children either.



- 3 In the Properties pane on the right:

- a In the **Title** text box, enter Magic Squares.
- b In the **Subtitle** text box, enter Columns, Rows, Diagonals: Everyone is Equal.
- c Under **Options**, you do not need to enter an image file name, but make sure the list is set to Custom author.

The screenshot shows the 'Options' dialog box. It has a title bar 'Options'. Below the title bar, there are several fields:

- 'Image file name:' followed by an empty text box.
- 'Custom author:' followed by a dropdown menu showing 'Custom author:' and an empty text box to its right.
- 'Include report creation date:' with a checked checkbox and a dropdown menu showing 'dd-mmm-yyyy HH:MM:SS (25-Oct-2006 10:19:54)'. The dropdown menu is open, showing the current selection.
- 'Include copyright holder and year:' with an unchecked checkbox and two empty text boxes.

- d In the field to the right of the Custom author field, enter Albrecht Durer.

Albrecht Dürer created the etching that contains a magic square. Your final report includes an image of that etching.

- e Select the **Include copyright holder and year** check box.
- f In the next text box, enter The MathWorks.
- g In the second text box, enter 1998.
- h In the **Abstract** text box, enter the following text:

An introduction to Magic Squares and their meaning.

The Properties pane on the right should look as follows.

Title Page

Title

Title:

Subtitle:

Options

Image file name:

Custom author:

Include report creation date:

Include copyright holder and year:

Abstract

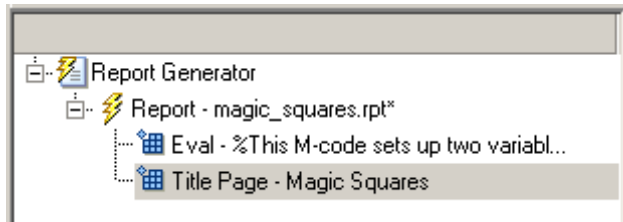
Legal notice

4 Save the report.

Adding the First Chapter

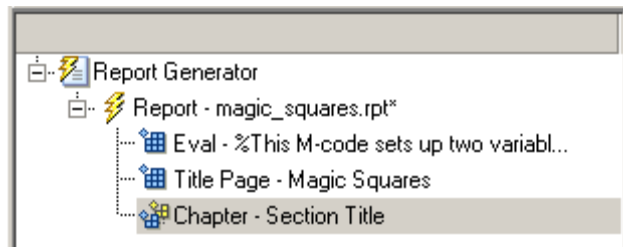
Add a chapter to the report by using the Chapter/Subsection component.

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



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

The Outline pane should look as follows.

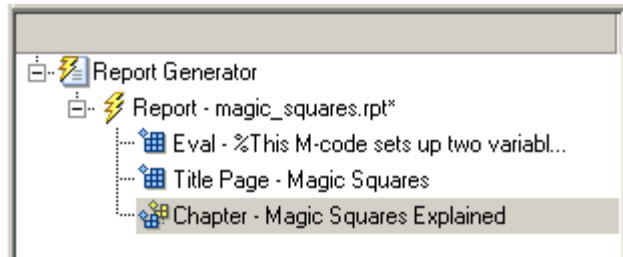


The Eval, Title Page, and Chapter components are all child components of the report's top level, but siblings of one another.

The Chapter component can have child components. The next section explains how to add child components to this Chapter component.

- 3 In the Properties pane on the right, enter Magic Squares Explained for the custom chapter title.

The Outline pane on the left changes to reflect the chapter title.

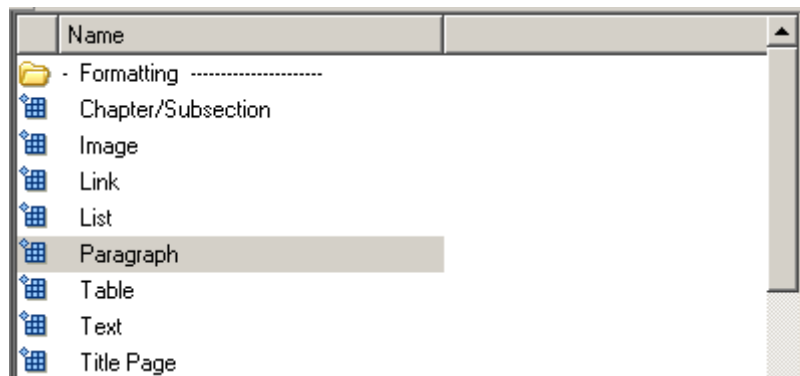


4 Save the report.

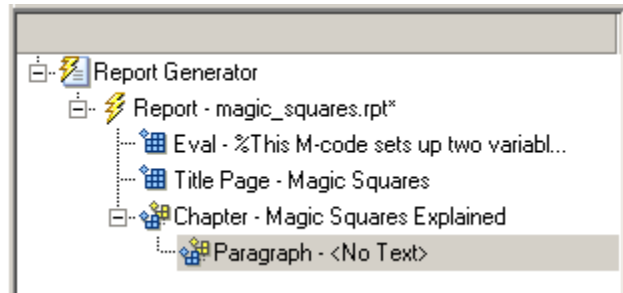
Adding Components to the First Chapter

Add introductory text to the first chapter using the Paragraph and Text components.

- 1** In the Outline pane on the left, select the Chapter component.
- 2** In the Options pane in the middle, under the Formatting category, double-click Paragraph.



In the Outline pane on the left, the new component is added as a child to the Chapter component.

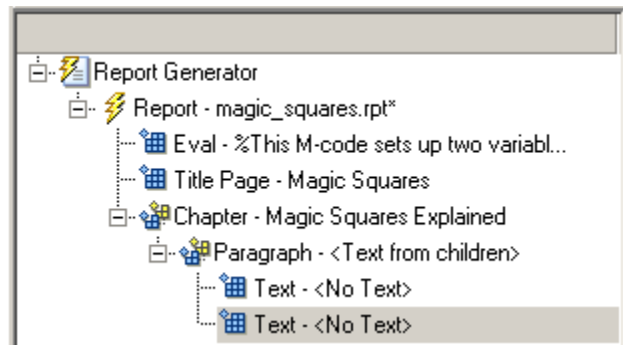


- 3 By default, the Paragraph component inherits its text from its child components. Add two Text components.

Note The text component must have the Paragraph component as its parent.

- 4 In the Options pane in the middle, under the Formatting category, double-click Text.
- 5 Double-click Text again to add a second component.

The Outline pane should look as follows.



- 6 In the Outline pane on the left, select the first Text component.
- 7 In the **Text to include in report** text box, enter `%<help('magic')>`.

The % sign and angle brackets <> indicate to MATLAB Report Generator that this is MATLAB code to be evaluated. The command `help('magic')` displays information about the MATLAB function `magic`.

- 8** In the Outline pane on the left, select the second Text component.
- 9** In the **Text to include in report** text box, enter the following text.

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

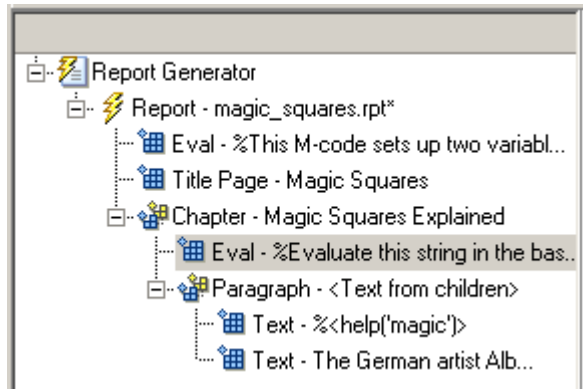
```
The German artist Albrecht Durer (1471-1528)
created many woodcuts and prints with religious
and scientific symbolism. One of his most famous
works, Melancholia I, explores the depressed state
of mind that opposes inspiration and expression.
Renaissance astrologers believed that the Jupiter magic
square (shown in the upper right portion of the image)
could aid in the cure of melancholy. The engraving's
date (1514) can be found in the lower row of numbers
in the square.
```

- 10** Save the report.

The next step includes an image of the etching shown at the beginning of this section.

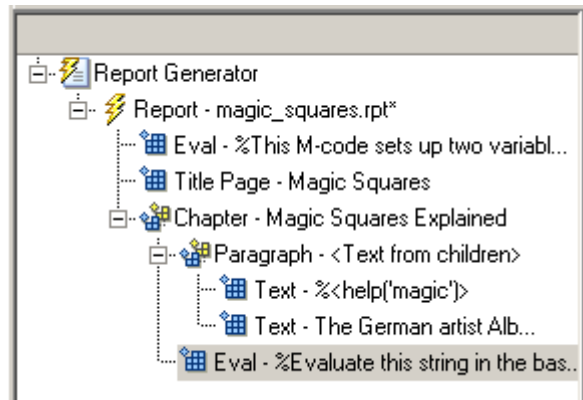
- 11** In the Outline pane on the left, select the Chapter component.
- 12** In the Options pane in the middle, under the MATLAB category, double-click Evaluate MATLAB Expression.

The new component becomes a child of the Chapter component.



You need to move the Eval component below the Paragraph component so that the image follows the introductory text.

- 13 Click the **down** arrow on the toolbar to move the Eval component below the Paragraph component.



- 14 With the Eval component still selected, do the following in the Properties pane on the right:
 - a Clear the **Insert MATLAB expression in report** and **Display command window output in report** check boxes. You do not want to include the code or its output in the report.
 - b In the **Expression to evaluate in the base workspace** text box, replace the existing text with the following M-code.

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

```
%This loads a self-portrait of Albrecht
%Durer, a German artist. There is a
%magic square in the upper right corner
%of the image.

durerData=load('durer.mat','-mat');
figure('Units','Pixels',...
'Position',[200 200 size(durerData.X,2)*.5 size(durerData.X,1)*.5 ]);

image(durerData.X);
colormap(durerData.map);
axis('image');
set(gca,...
    'Xtick',[],...
    'Ytick',[],...
    'Units','normal',...
    'Position',[0 0 1 1]);

clear durerData
```

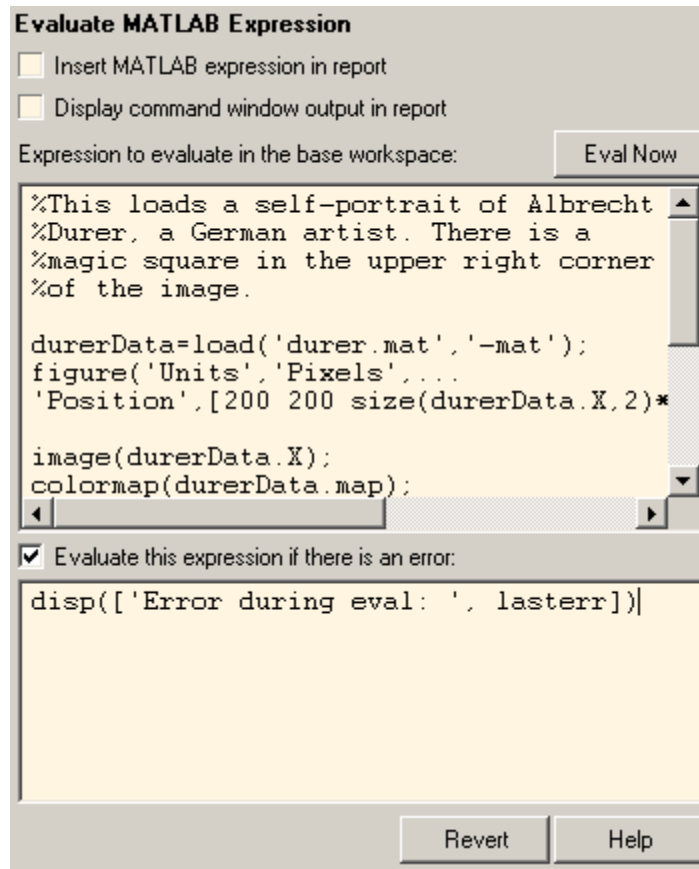
This M-code displays the Dürer etching in a MATLAB figure window.

- c In the **Evaluate expression if there is an error** text box, replace the existing text with the following text:

```
disp(['Error during eval: ', lasterr])
```

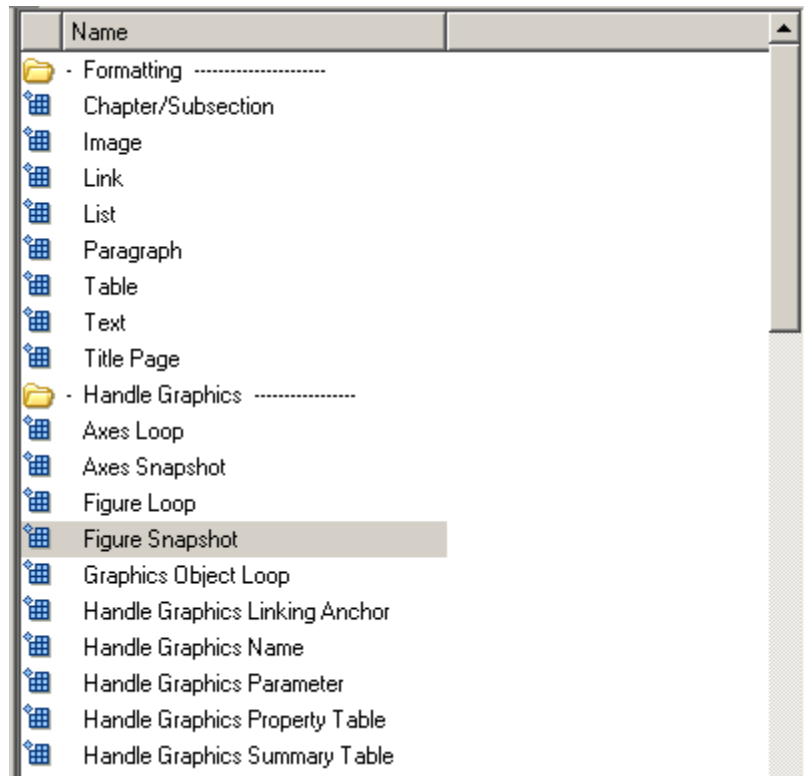
This code executes if an error occurs while loading the Dürer etching.

The Properties pane on the right should look as follows.



Now that the image of the Dürer etching is loaded in the MATLAB workspace, include it in the report using the Figure Snapshot component.

- 15** In the Outline pane on the left, select the Eval component.
- 16** In the Options pane in the middle, under the Handle Graphics category, double-click Figure Snapshot.



17 In the Properties pane on the right:

- a** In the **Paper** orientation list, select Portrait.
- b** In the **Invert hardcopy** list, select Don't invert.

Selecting this option specifies not to change the image's on-screen colors for printing.

Delete the image after it has been added to the report.

18 In the Outline pane on the left, select the Figure Snapshot component.

19 In the Options pane in the middle, under the MATLAB category, double-click Evaluate MATLAB Expression.

20 In the Properties pane on the right:

- a** Clear the **Insert MATLAB expression in report** and **Display command window output in report** check boxes. You do not want to include the code or its output in the report.
- b** In the **Expression to evaluate in the base workspace** text box, replace the existing text with the following text:

```
%This command deletes the Durer image  
delete(gcf);
```

The `delete(gcf)` command deletes the current image in the MATLAB workspace, in this case, the Dürer etching.

- c** In the **Evaluate expression if there is an error** text box, replace the existing text with the following text:

```
disp(['Error during eval: ', lasterr])
```

This code executes if an error occurs while deleting the Dürer etching.

- 21** Save the report.

The contents of the first chapter are now complete.

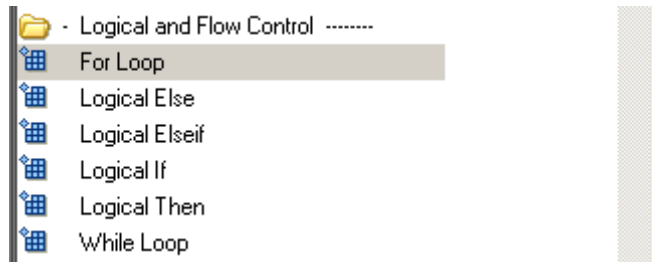
Creating the Magic Squares and Their Images

Use components to create several magic squares and insert the contents of the squares, or representative images, into the report.

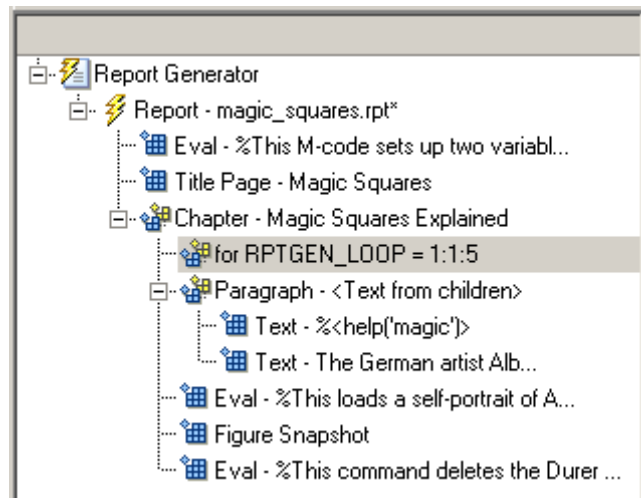
Creating the For Loop

Each square has its own chapter. A For Loop component performs tasks for each square.

- 1** In the Outline pane on the left, select the Chapter component.
- 2** In the Options pane in the middle, under the Logical and Flow Control category, double-click For Loop.

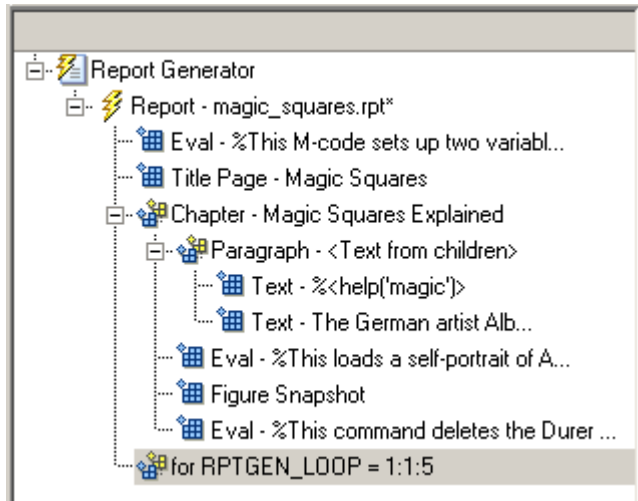


The Outline pane on the left should look as follows.



This component is added inside the Chapter component. However, the magic squares should be processed *after* the first chapter, so the for component should be a sibling of the Chapter component, not a child.

- 3 In the Outline pane on the left, select the for component.
- 4 Click the **left** arrow to make the for component a sibling, not a child, of the Chapter component.



5 In the Properties pane on the right:

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

```
length(magicSizeVector)
```

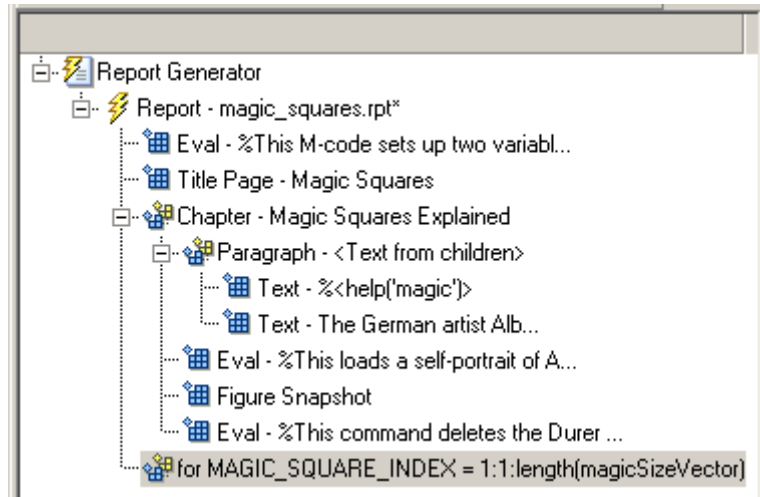
This is the length of the vector that contains the various sizes for the magic square matrices.

- b In the **Variable name** text box, replace the existing text with the following text:

```
MAGIC_SQUARE_INDEX
```

This variable will act as a loop index.

The Outline pane on the left should look as follows.



6 Save the report.

Adding a Chapter for Each Square

Add a chapter for each square processed using the Chapter/Subsection component.

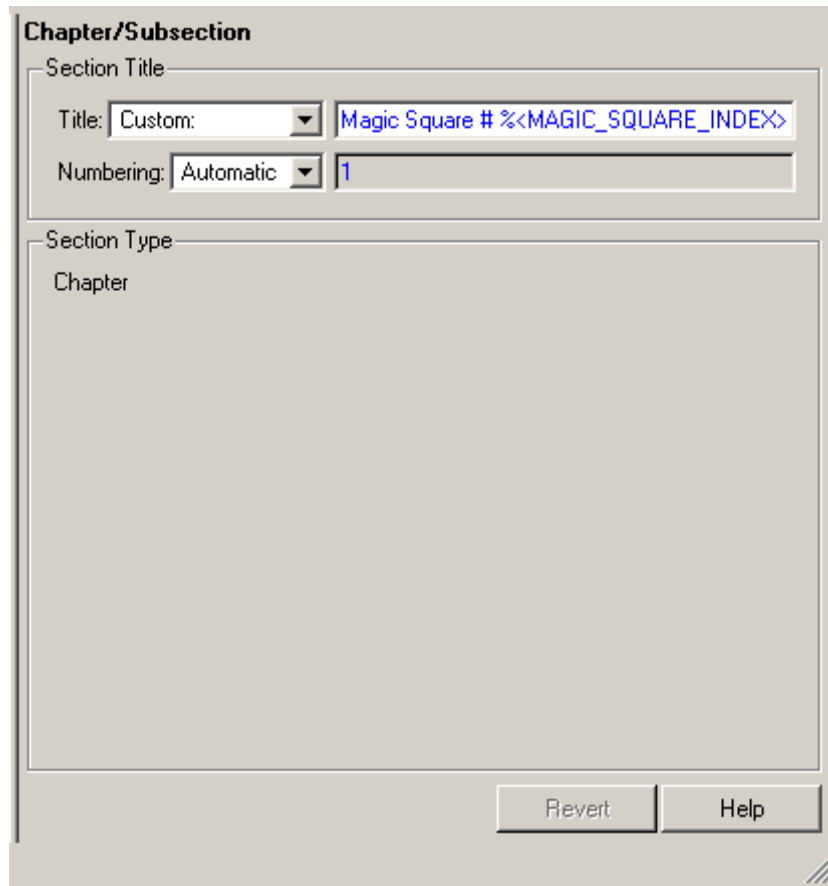
- 1 In the Outline pane on the left, select the for component.
- 2 In the Options pane in the middle, under the Formatting category, double-click Chapter/Subsection.

It becomes a child of the for component.

- 3 In the Properties pane on the right, select Custom from the **Title** list and enter the following for the chapter title:

```
Magic Square # %<MAGIC_SQUARE_INDEX>
```

The Properties pane should look as follows.



4 Save the report.

Determining the Matrix Size

You need to extract the size of each magic square matrix from `magicSizeVector` using an Evaluate MATLAB Expression component.

- 1** In the Outline pane on the left, select the Chapter component.
- 2** In the Options pane in the middle, under the MATLAB category, double-click Evaluate MATLAB Expression.

- 3 In the Properties pane on the right:
 - 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, replace the existing text with the following text:

```
magic_Square_Size=magicSizeVector(MAGIC_SQUARE_INDEX);
```

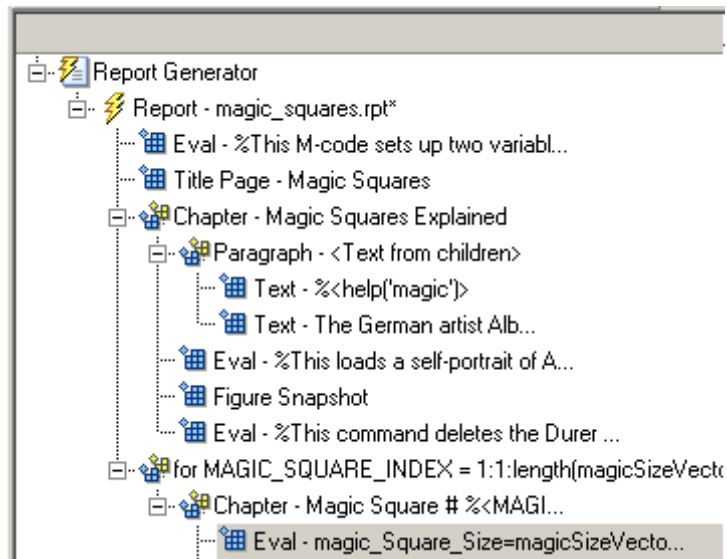
This command extracts the next size for the magic square from the vector of sizes initialized in the first Eval component of the report. The variable `magic_Square_Size` represents the size of the current magic square being processed.

- c In the **Evaluate expression if there is an error** text box, replace the existing text with the following:

```
disp(['Error during eval: ', lasterr])
```

This code executes if an error occurs while attempting to extract a value from `magicSizeVector`.

The Outline pane on the left should look as follows.



- 4 Save the report.

Inserting the Magic Square Size into the Report

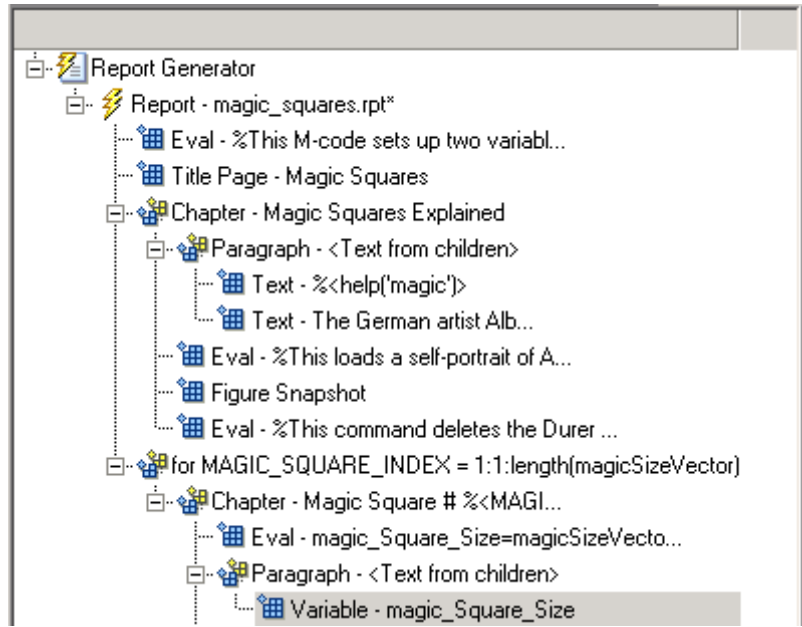
Insert the size of the magic square into the report using the Paragraph and Insert Variable components.

- 1 In the Outline pane on the left, select the Eval component.
- 2 In the Options pane in the middle, under the Formatting category, double-click Paragraph.

Do not change the properties. The variable that contains the size of the magic square will go in this paragraph.

- 3 In the Outline pane on the left, select the Paragraph component.
- 4 In the Options pane in the middle, under the MATLAB category, double-click Insert Variable.
- 5 In the Properties pane on the right:
 - a In the **Variable name** text box, enter `magic_Square_Size`.
 - b In the **Display as** list, select `Inline text`.

The Outline pane on the left should look as follows.



6 Save the report.

Displaying the Magic Square

To create the magic square and display the associated matrix or image, use the Evaluate MATLAB Expression component.

- 1** In the Outline pane on the left, select the Paragraph component.
- 2** In the Options pane in the middle, under the MATLAB category, double-click Evaluate MATLAB Expression.

You need to make this component a sibling of the Paragraph component, not a child.

- 3** In the Outline pane on the left, select the Eval component.
- 4** Click the left arrow on the toolbar to make the Eval component a sibling of the previous Paragraph component.
- 5** In the Properties pane on the right:

- 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, replace the existing text with the following M-code.

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

```
%This m-script produces a magic
%square of size magic_Square_Size
%and creates an image of that square.

mySquare=magic(magic_Square_Size);
clf
imagesc(mySquare);
title(sprintf('Magic Square N=%i',magic_Square_Size))
set(gca,'Ydir','normal');
axis equal;
axis tight;
```

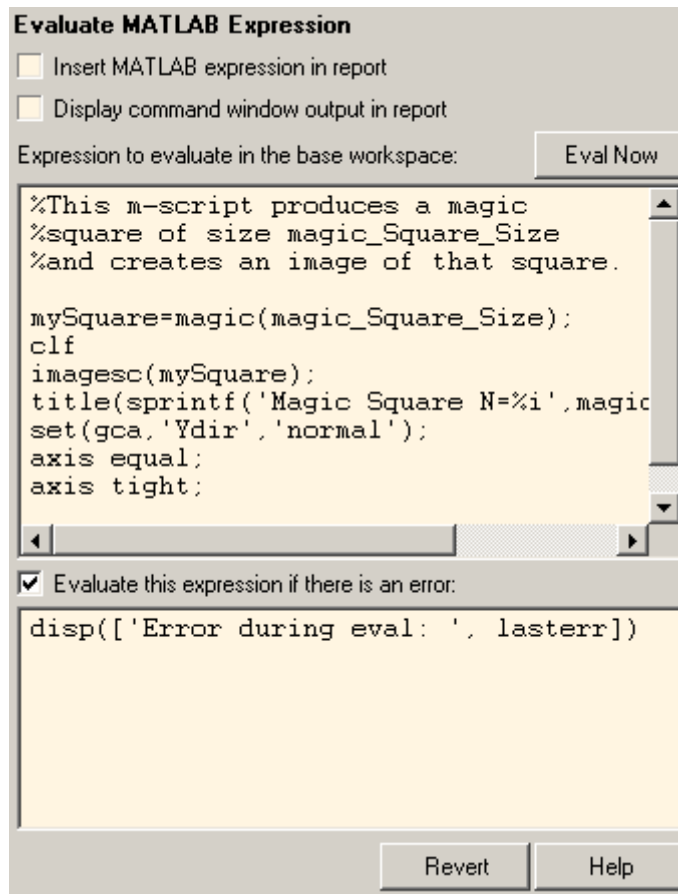
This code creates a magic square matrix `mySquare` of size `magic_Square_Size`, and opens an image of that matrix in the MATLAB Figure window.

- c In the **Evaluate expression if there is an error** text box, replace the existing text with the following:

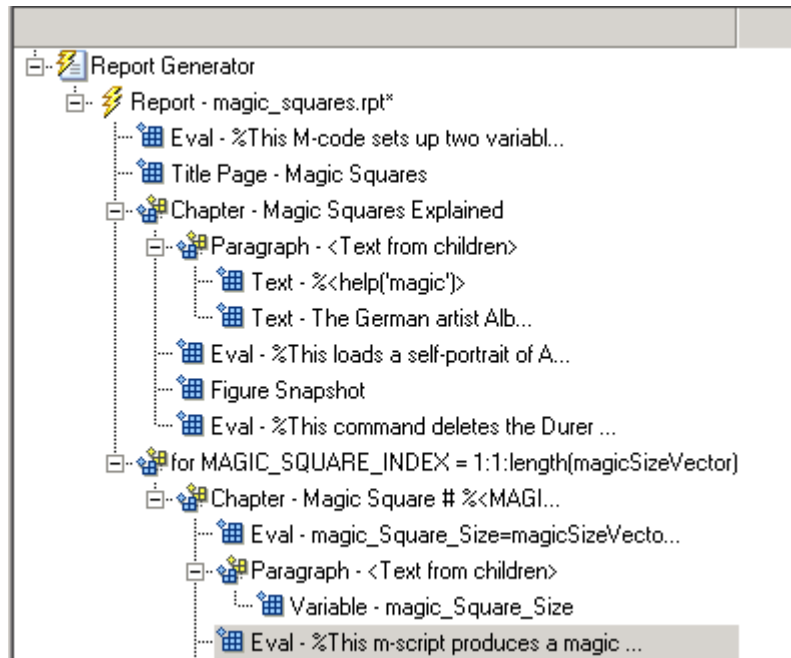
```
disp(['Error during eval: ', lasterr])
```

This code executes if an error occurs while creating and displaying the magic square.

The Properties pane on the right should look as follows.



- 6** Save the report.
- 7** In the Outline pane on the left, select the Eval component.



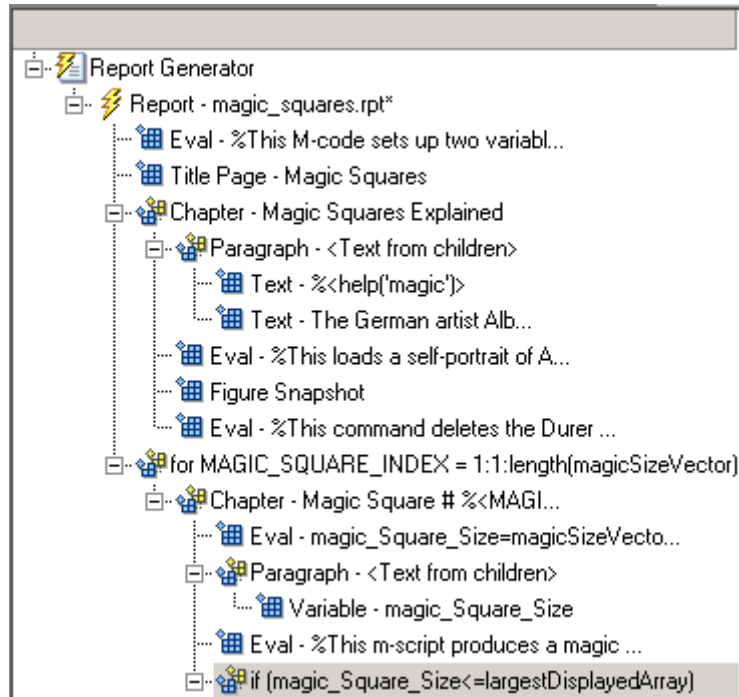
- 8 On the Options pane in the middle, under the Logical and Flow Control category, double-click Logical If.
- 9 On the Properties pane on the right, in the **Test Expression** text box, replace the existing text with the following text:

```
magic_Square_Size<=largestDisplayedArray
```

This command tests if the current matrix size (`magic_Square_Size`) is less than or equal to the value assigned in the first Eval component of the report (`largestDisplayedArray=15`).

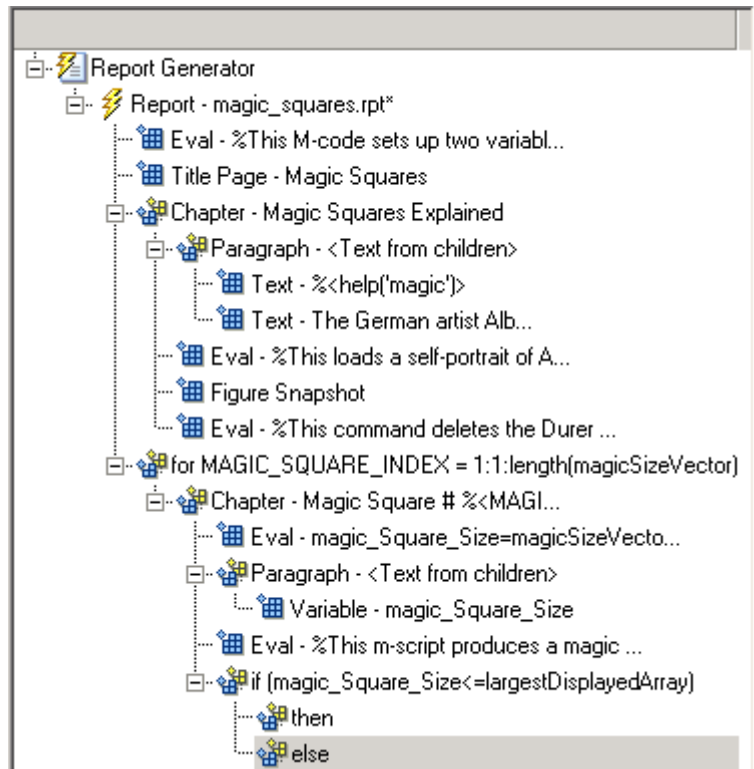
To process the result of this Logical If component, create two child components—Logical Then and Logical Else. If `magic_Square_Size` is less than or equal to 15, the matrix variable is inserted into the report. If `magic_Square_Size` is greater than 15, the matrix image is inserted into the report.

- 10 On the Outline pane on the left, select the if component.



- 11** On the Options pane in the middle, under Logical and Flow Control, double-click Logical Else.
- 12** On the Outline pane on the left, select the if component again.
- 13** On the Options pane in the middle, under Logical and Flow Control, double-click Logical Then.

The then component is added above the else component.



- 14 In the Outline pane on the left, select the then component.
- 15 In the Options pane in the middle, under the MATLAB category, double-click Insert Variable.
- 16 In the Properties pane on the right:
 - a In the **Variable name** text box, enter mySquare, which is the variable that contains the magic square of the specified size.
 - b In the **Title** list, select None.
 - c In the **Size Limit** text box, enter 0.

The Properties pane on the right should look as follows.

Insert Variable

Source

Variable name:

Variable location:

Warning: "mySquare" not found in workspace.

Display Options

Title:

Size limit:

Display as:

Ignore if value is empty

This Variable component displays the magic square matrix, stored in the variable mySquare.

- 17** In the Outline pane on the left, select the else component.
- 18** In the Options pane in the middle, under the Handle Graphics category, double-click Figure Loop.
Do not change its properties.
- 19** In the Outline pane on the left, select the Figure Loop component.
- 20** In the Options pane in the middle, under the Handle Graphics category, double-click Figure Snapshot.
- 21** In the Properties pane on the right:
 - a** In the **Paper orientation** list, select Portrait.

- b** In the **Image size** list, select Custom.
- c** Below the **Image size** list, enter [5 4] for the custom image size.
- d** In the **Invert hardcopy** list, select Invert.

This option changes dark axes colors to light axes colors, and vice versa.

The Properties pane on the right should look as follows.

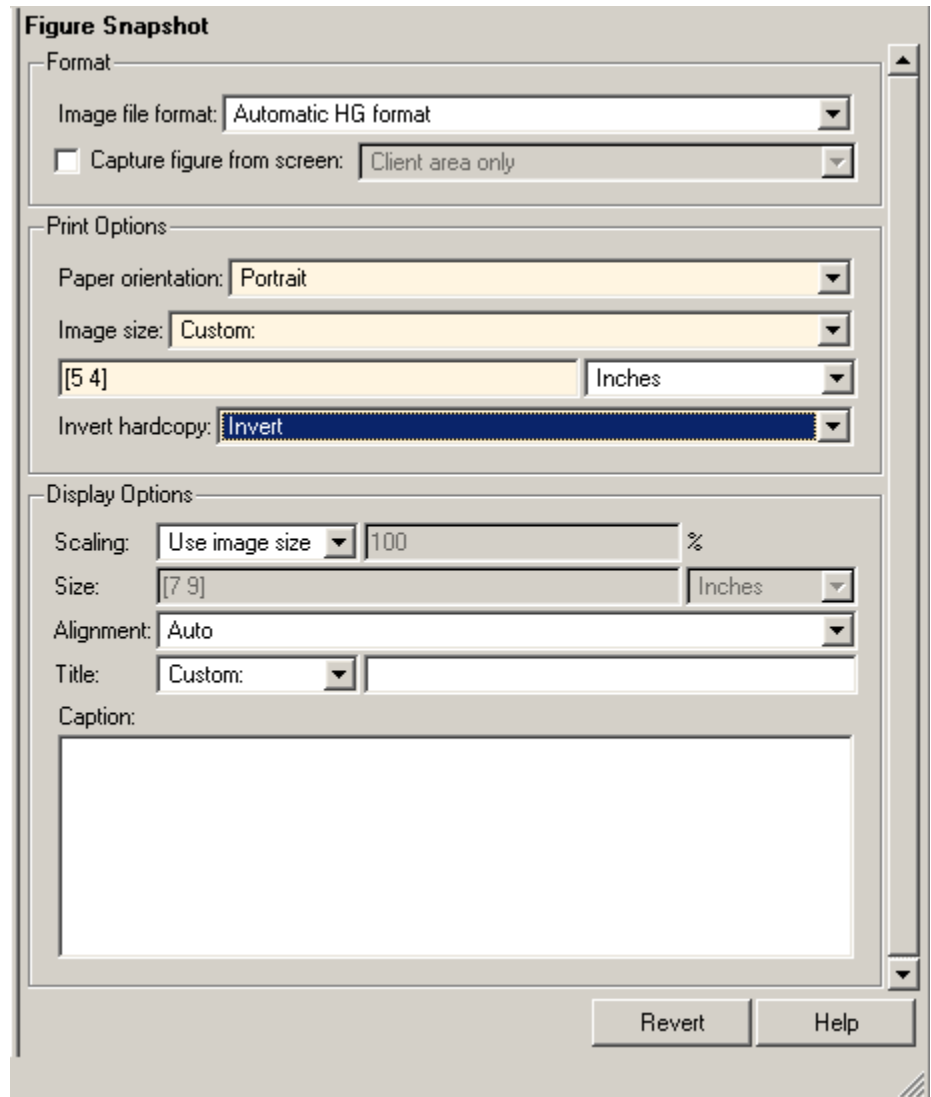


Figure Snapshot

Format

Image file format: Automatic HG format

Capture figure from screen: Client area only

Print Options

Paper orientation: Portrait

Image size: Custom:

[5 4] Inches

Invert hardcopy: Invert

Display Options

Scaling: Use image size 100 %

Size: [7 9] Inches

Alignment: Auto

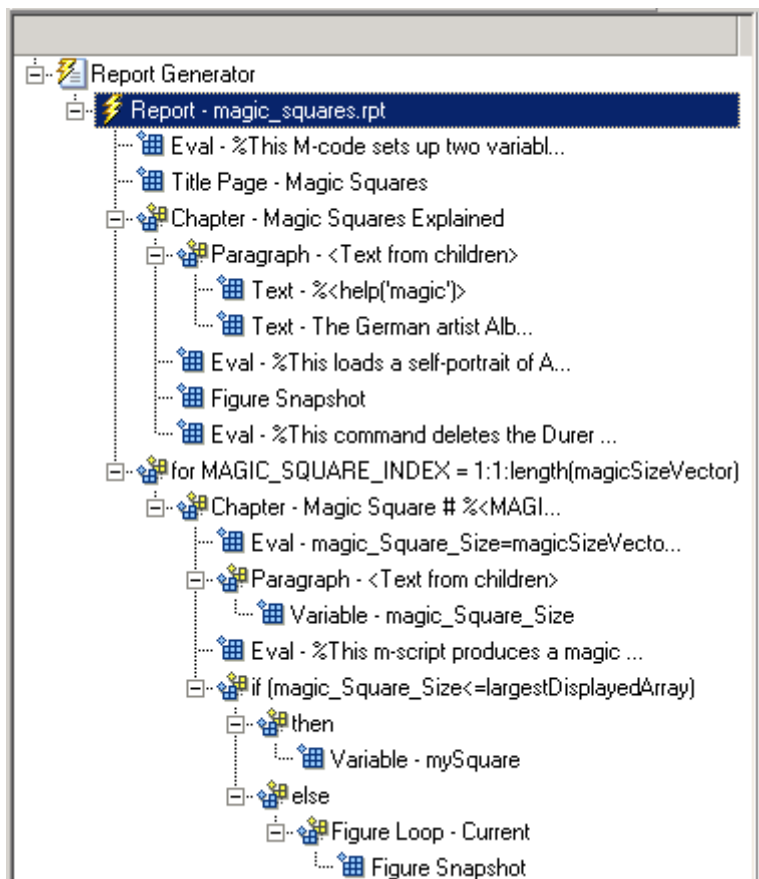
Title: Custom:

Caption:

Revert Help

The dialog box is titled "Figure Snapshot" and is divided into three main sections: "Format", "Print Options", and "Display Options". The "Format" section includes a dropdown for "Image file format" set to "Automatic HG format" and a checkbox for "Capture figure from screen" which is unchecked, with a dropdown set to "Client area only". The "Print Options" section includes a dropdown for "Paper orientation" set to "Portrait", a dropdown for "Image size" set to "Custom:", a text input field containing "[5 4]" and a dropdown for units set to "Inches", and a dropdown for "Invert hardcopy" set to "Invert". The "Display Options" section includes a "Scaling" dropdown set to "Use image size" with a text input field containing "100" and a percentage symbol, a "Size" dropdown set to "[7 9]" and a units dropdown set to "Inches", an "Alignment" dropdown set to "Auto", a "Title" dropdown set to "Custom:" followed by an empty text input field, and a "Caption" label above a large empty text area. At the bottom right, there are two buttons: "Revert" and "Help".

The Outline pane on the left should look like the following.

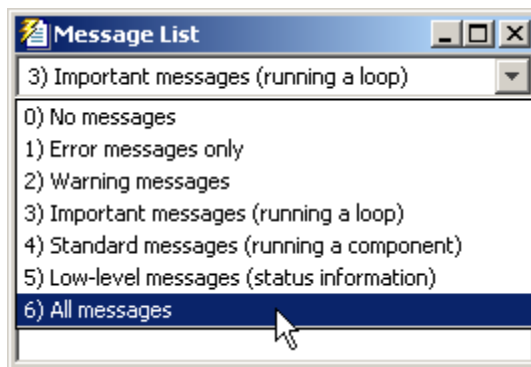


Generating the Report

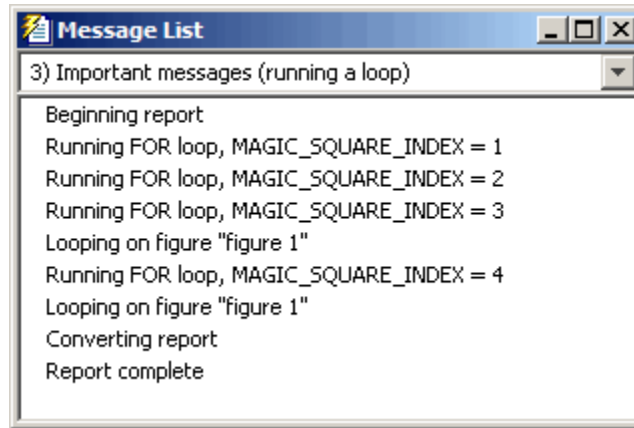
Now the report contains all the components it needs.

Click the **Report** icon on the toolbar to generate the report. You will see the following displayed on your screen.

- A Message List window appears, displaying informational and error messages as MATLAB Report Generator is processing the report. While the report is being generated, specify the level of detail you would like the Message List window to display. Options range from 0 (least detail) to 6 (most detail). Click the list located under the title bar of the Message List window to choose an option, as shown in the figure below.

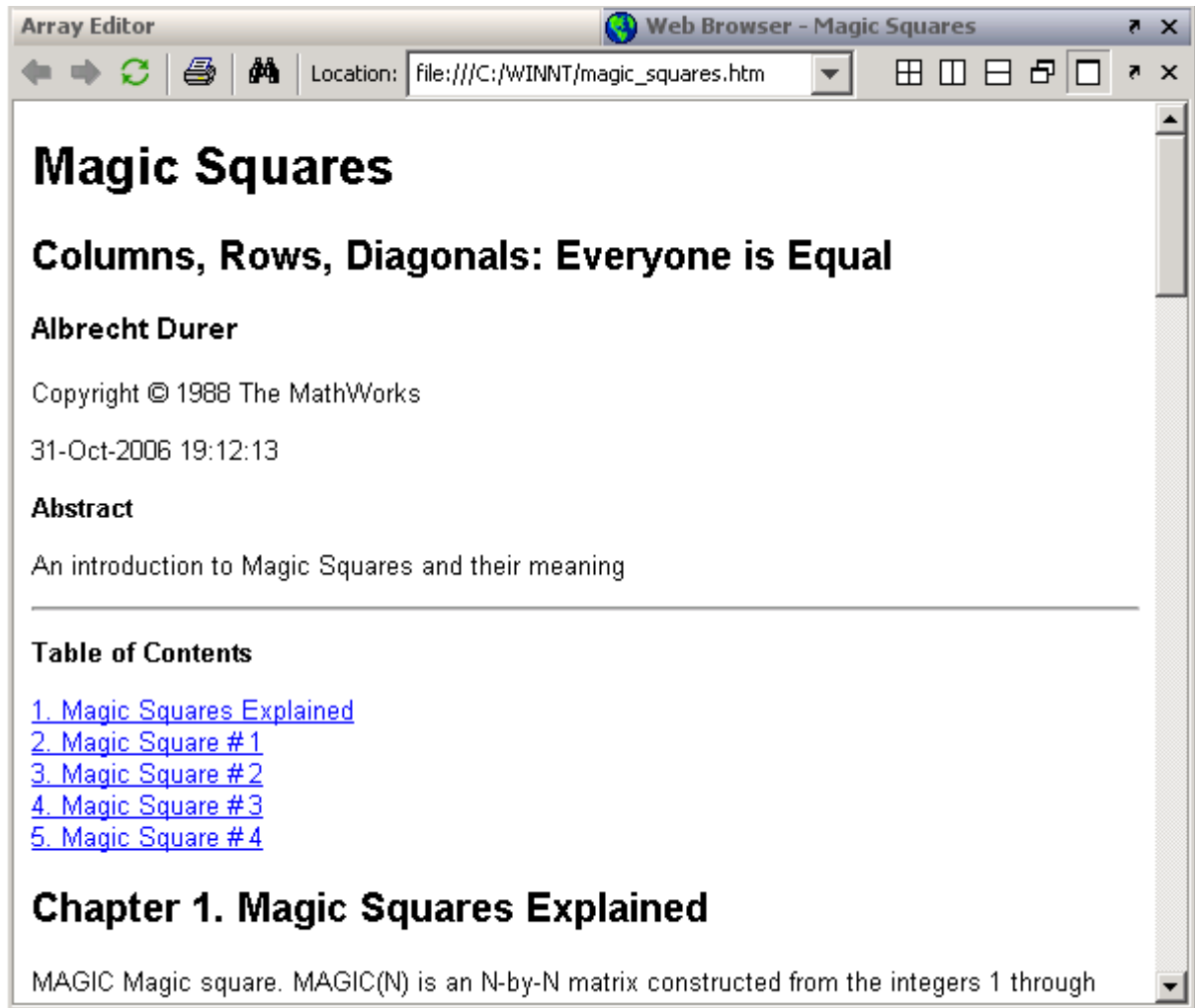


Message level 3 (Important messages) is used for the remainder of this example.



- An image of the etching appears briefly.
- Images of two magic square images of sizes 16 and 32 appear briefly.
- In the Outline pane on the left of your Report Explorer window, each component of the report template is highlighted as it is executed.

At the beginning of this tutorial you specified HTML as the output format of this report. When MATLAB Report Generator finishes processing, the MATLAB Web browser opens and displays the report's HTML file.



The screenshot shows a web browser window titled "Web Browser - Magic Squares". The address bar displays the file path "file:///C:/WINNT/magic_squares.htm". The page content includes a main heading "Magic Squares", a sub-heading "Columns, Rows, Diagonals: Everyone is Equal", and the author "Albrecht Durer". It also contains copyright information, a date, an abstract, a table of contents with five links, and the start of "Chapter 1. Magic Squares Explained".

Array Editor | **Web Browser - Magic Squares**

Location: file:///C:/WINNT/magic_squares.htm

Magic Squares

Columns, Rows, Diagonals: Everyone is Equal

Albrecht Durer

Copyright © 1988 The MathWorks
31-Oct-2006 19:12:13

Abstract

An introduction to Magic Squares and their meaning

Table of Contents

- [1. Magic Squares Explained](#)
- [2. Magic Square #1](#)
- [3. Magic Square #2](#)
- [4. Magic Square #3](#)
- [5. Magic Square #4](#)

Chapter 1. Magic Squares Explained

MAGIC Magic square. MAGIC(N) is an N-by-N matrix constructed from the integers 1 through

Generating Reports

Setting Report Output Options (p. 3-2)	How to set options for generated reports
Converting XML Documents (p. 3-11)	How to convert XML source files to generate a given report in a different format
Creating Log Files (p. 3-15)	How to create log files for report generation
Generating M-code from Report Templates (p. 3-16)	How to generate M-code versions of report templates
Working with Legacy Report Templates (p. 3-17)	How to work with report templates created in a previous version of MATLAB Report Generator
Troubleshooting (p. 3-19)	How to troubleshoot report generation issues

Setting Report Output Options

In this section...
“Setting Preferences” on page 3-2
“Setting the Report File Directory” on page 3-3
“Setting the Report File Name” on page 3-4
“Setting the Report Format” on page 3-4
“Viewing and Printing Reports” on page 3-8
“Converting English Strings to Other Languages” on page 3-9
“Autosaving Report Templates” on page 3-9
“Regenerating Images” on page 3-9
“Including a Report Description” on page 3-10

Setting Preferences

Specify report output settings using the **Preferences** pane. To display this pane, click **File > Preferences**. Options available in the Preferences pane are as follows.

Option	Purpose
Format ID	Specify the report output format. Choices include Acrobat (PDF), Rich Text format, and Word document. The default is Web (HTML).
Extension	Automatically sets the file extension of the report file according to the value specified for Format ID. The default is html.
Simulink Images	Specify the format for Simulink® images that are included in the report. The default is PNG 24-bit image.
Stateflow Images	Specify the format for Stateflow® images that are included in the report. The default is PNG 24-bit image.

Option	Purpose
HG Images	Specify the format for Handle Graphics® images that are included in the report. The default is PNG 24-bit image.
View command	Specify the MATLAB command you want to use to view the report.
Visible in Report Explorer	Deselect this check box to make the current output format unavailable in the Report Explorer Interface. For example, if your specified report format is Word document and you deselect this check box, then the Word document format is no longer available for reports created using the Report Explorer.
Use version 1.x environment	Choose this option to use previous versions of MATLAB Report Generator interfaces. For more information, see “Using Legacy MATLAB Report Generator Interfaces” on page 3-18.
Animate Report Explorer when generating reports	Select this check box if you want MATLAB Report Generator to animate components in the Outline pane as the report is generated. This box is checked by default.

Setting the Report File Directory

Choose the appropriate directory for the report file according to the table below. Make sure you have write privileges for that directory.

Directory	Option
The same directory as the report template	Same as report template
The current working directory	Present working directory

Directory	Option
Temporary directory	Temporary directory
Another directory	Custom. Specify a directory in the Custom text box by using the Browse button (...) to select from a list of directories.

Use %<VariableName> notation to specify a directory in the **Custom** text box. For more details about this notation, see %<VariableName> Notation on the Text component reference page.

Setting the Report File Name

Choose the appropriate file name for the report file according to the table below.

Images are placed in a directory with the same name as the report file. For example, testreport.html images are placed in a directory named testreport_files.

File Name	Option
The same file name as the report template	Same as report template (default)
A file name different from the report template name	Custom. Enter the name of the report in the text box.

Use %<VariableName> notation to specify a file name in the **Custom** text box. For more details about this notation, see %<VariableName> Notation on the Text component reference page.

Setting the Report Format

Choose the report output format in the **File format** text box, according to how you want to view the report. For example, if you want to use Microsoft Word, you should choose Word document or Rich Text Format.

Report Format and Stylesheet

File format: Rich Text Format Very Large Type Print

Generation Options

View report after generation

Auto save before generation

Evaluate this string after generation:

Each output format has a default stylesheet. Specify the stylesheet in the text box next to the **File format** text box.

The table below shows which output format to use with different viewers or software, as well as which stylesheets are available for use with each output format.

Viewer	Format	Purpose	Stylesheet Options
Adobe Acrobat Reader	Acrobat (PDF)	Produces a PDF that can be viewed with Adobe Acrobat Reader	PDF (see “PDF Stylesheets” on page 3-6)
Web browser	Web (HTML) (default)	Use for publishing on the World Wide Web	Web (see “Web Stylesheets” on page 3-7)
Word processor	Rich Text Format (RTF) or Word Document	Compatible with most word-processing packages, including Microsoft Word	Print (see “RTF (DSSSL Print) and Word Stylesheets” on page 3-8)
DocBook	DocBook (XML)	Produces a report in DocBook format	N/A

Tip To create and use customized styles, see Chapter 6, “Creating Custom Stylesheets”.

PDF Stylesheets

PDF Stylesheet	Description
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
Logo stylesheet for PDF	Lets you specify a logo, such as your company logo, in the header
Verbose Print	Lets you specify advanced print options

Web Stylesheets

Web Stylesheet	Description
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 numbered

RTF (DSSSL Print) and Word Stylesheets

RTF or Word Stylesheet	Description
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

Note Some Web and print stylesheets include an automatically generated list of titles. The list of titles includes a list of table titles and a list of figures with titles.

Viewing and Printing Reports

Use one of the following options to view your report after generation:

- To view the report automatically, select the **View report after generation** check box in the **Generation Options** section in the Properties pane on the right. When report generation is complete, the viewer associated with the report output format displays the report.
- To view the report from the MATLAB command line, use the `rptviewfile` command, specifying the name of the report file.

- To view the report manually, browse to the location specified in the **Report File Location** section in the Properties pane on the right and open the file.

If you want to print your report, select the **Print** option from the viewer.

Converting English Strings to Other Languages

MATLAB Report Generator and Simulink® Report Generator Versions 2.0 and later use the system language settings through the Java interface; therefore, they should use the language specified on your system.

Alternatively, you can change the language directly in Java from the MATLAB command line. In the following example, the language is set to Italian:

```
java.util.Locale.setDefault(java.util.Locale.ITALY)
```

Alternatively, set the preferred language directly in your `.rpt` file:

- 1 Right-click the **Report** component and select **Send to Workspace**.

This displays the properties of the report, which are stored in the variable *ans*. Access the report's Language property from the command line through this variable. By default, Language is set to auto, which indicates that the system's default language is in use.

- 2 Override the default value of Language by setting this property to your desired language; for example, `en` for English or `it` for Italian.

Autosaving Report Templates

To automatically save the report template before you generate a report, select **Auto save before generation** from the **Generation Options** section of the Properties pane on the right.

Regenerating Images

By default, the **Regenerate Simulink and Stateflow Images** option in the **Generation Options** section of the Properties pane on the right is not selected. This means that previously generated images are not regenerated each time you generate a report, which results in faster report generation. If you make changes that affect a generated image, select this option.

Including a Report Description

Use the **Report Description** field in the Properties pane on the right to record notes and comments about your report template. This text appears in the Properties pane on the right when you select a report template in the Outline pane on the left.

Converting XML Documents

In this section...

“Why Convert XML Documents?” on page 3-11

“Converting XML Documents Using the Report Explorer” on page 3-11

“Converting XML Documents Using the Command Line” on page 3-13

“Editing XML Source Files” on page 3-14

Why Convert XML Documents?

You can generate a report in a different output file format without regenerating it by using either the Report Explorer File Converter or the `rptconvert` command. These utilities convert DocBook XML source files created by MATLAB Report Generator into formatted documents such as HTML, RTF, or PDF.

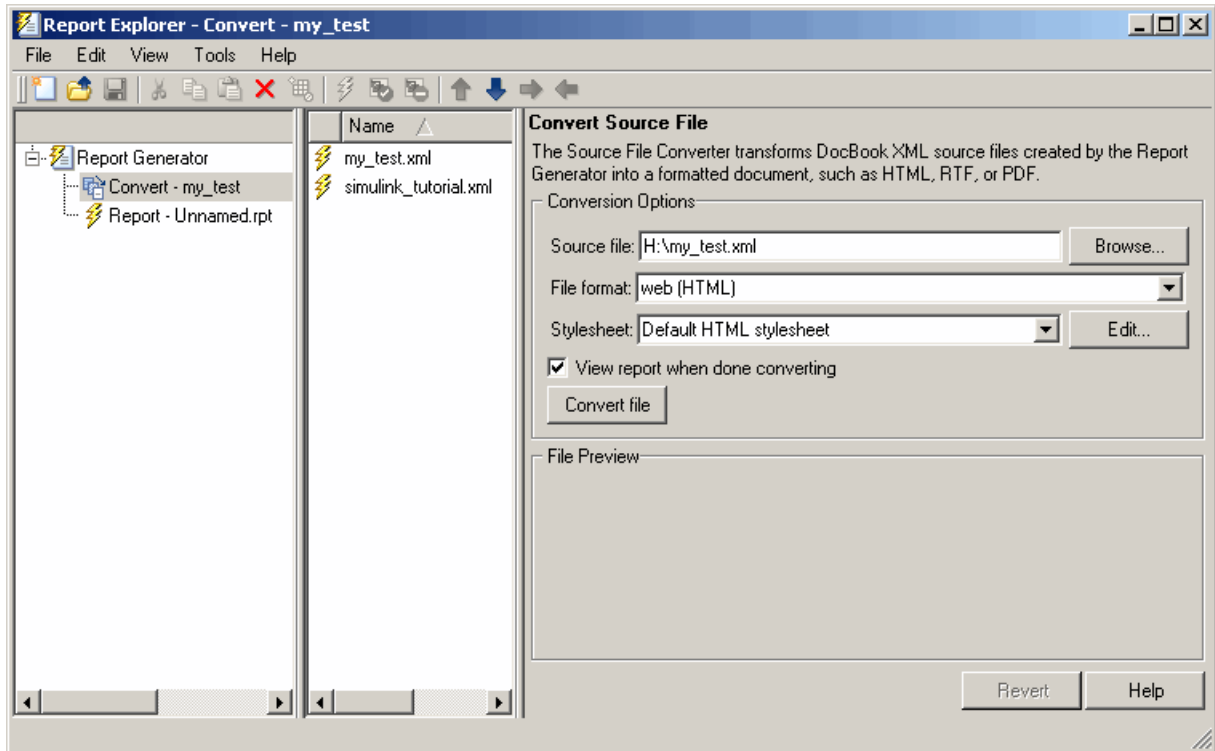
Note MATLAB Report Generator can only convert XML source files created by the latest version of MATLAB Report Generator.

Converting XML Documents Using the Report Explorer

To open the **Convert** Properties pane:

- 1 In the Report Explorer, select **Tools > Convert source file**.

The Convert Source File Properties pane appears. All XML files in your current directory are listed in the Options pane in the middle.



2 Select your XML source file by using one of the following methods:

- Click **Browse** in the Properties pane on the right to browse to the location of your XML source.
- Double-click a file name in the Options pane in the middle. This automatically enters the file name into the **Source file** field in the Properties pane.

3 Select your output format and stylesheet as follows:

- a** In the **File format** text box, select an output format.
- b** In the **Stylesheet** text box, select a stylesheet. The stylesheet choice depends on the selected output format. You can use a predefined stylesheet or one that you have customized.

For more information about available formats and predefined stylesheets, see “Setting the Report Format” on page 3-4.

For more information about customizing stylesheets, see Chapter 6, “Creating Custom Stylesheets”.

- 4 Use the **View Report when done converting** check box to indicate whether or not you want to view the report after it has been converted.
- 5 To begin the conversion, click **Convert file**.

Converting XML Documents Using the Command Line

To convert files outside the Report Explorer interface, use `rptconvert` from the command line with varying levels of definition. Options are as follows:

- Run the command `<ReportName>=rptconvert(' <DocBookfilename> ')`, where `<ReportName>` is the name of your report and `DocBookfilename` is the name of the DocBook file (with or without the `.xml` extension). MATLAB Report Generator uses the default format (HTML) and stylesheet (Single-Page Web).
- Run the command `<ReportName>=rptconvert(' <DocBookfilename> ',format)`, where the valid format arguments are:
 - Web(HTML)
 - Rich Text Format
 - PDF
 - Word Document

Enter the format name in single quotation marks, for example, 'PDF'. MATLAB Report Generator uses the default stylesheet for the selected option.

- Run the command `<ReportName>=rptconvert(' <DocBookfilename> ',format,stylesheet)`

For a discussion of valid stylesheet options for each output format, see “Setting the Report Format” on page 3-4.

For a list of valid stylesheet arguments, enter the following command in the MATLAB Command Window:

```
rptconvert #'STYLESHEETLIST'
```

Editing XML Source Files

Before you send a source file to the converter, edit it as text in the Report Explorer:

- 1** In the Outline pane on the left, open the File Converter.
- 2** Right-click **MATLAB Report Generator** and select **Convert source file**.
- 3** In the Options pane in the middle, select the source file that you want to edit.
- 4** In the Properties pane on the right, click **Edit as text**.
- 5** Use the MATLAB Editor to edit and save the text.

Creating Log Files

In MATLAB Report Generator, a log file describes a report template's report generation settings and components. A log file can be used for many purposes, including:

- As a debugging tool
- As a reference to a report template
- To share information about a report template through e-mail

A log file includes the following information:

- Report template outline
- Components and their attributes
- Generation status messages currently displayed in the **Generation Status** tab
- All MATLAB licenses

To generate a log file, click **File > Log File**. An HTML version of the log file with the name <report_template_file_name_log>.html is saved in the same directory as the report template.

Generating M-code from Report Templates

Using MATLAB Report Generator, generate M-code versions of report templates in the form of an M-file (*.m). An M-file of a report template is useful for a variety of purposes, including generating reports and modifying report templates programmatically.

To generate an M-file from MATLAB Report Generator, load a report template and click **File > Generate M-File**. After the M-file is generated, it opens in the MATLAB Editor. The M-file is saved in the same location as the report template.

Working with Legacy Report Templates

In this section...

“What Are Legacy Report Templates?” on page 3-17

“Enabling Legacy MATLAB Report Generator Interfaces” on page 3-17

“Using Legacy MATLAB Report Generator Interfaces” on page 3-18

What Are Legacy Report Templates?

Legacy report templates are files that you created in a previous version of MATLAB Report Generator. You can load legacy report templates in the current version of the Report Explorer, or use previous versions of MATLAB Report Generator interfaces to work with legacy report templates.

Enabling Legacy MATLAB Report Generator Interfaces

Enable the previous versions of interfaces at the MATLAB command line:

```
RptgenML.v1mode(true)
```

MATLAB responds with:

```
ans =
```

```
1
```

A logical 1 indicates that Version 1 mode is on. To turn Version 1 mode off, enter:

```
RptgenML.v1mode(false)
```

MATLAB responds with:

```
ans =
```

```
0
```

A logical 0 indicates that Version 1 mode is off.

Using Legacy MATLAB Report Generator Interfaces

When Version 1 mode is enabled, you can use previous versions of MATLAB Report Generator interfaces. MATLAB functions related to MATLAB Report Generator also load the old interfaces.

When using the previous versions of MATLAB Report Generator interfaces, the following restrictions apply:

- You cannot use a report template created in the latest version of MATLAB Report Generator with the old interfaces. If you save a legacy report template in the latest version of MATLAB Report Generator, you will not be able to work with the same report template in the old interface.
- You cannot use the old interfaces with the latest version of MATLAB Report Generator. While the Setup File Editor can coexist with the Report Explorer, MathWorks does not recommend using the interfaces together.

Troubleshooting

In this section...

“Managing MATLAB Report Generator Memory Usage” on page 3-19

“Displaying HTML Reports on UNIX” on page 3-20

Managing MATLAB Report Generator Memory Usage

By default, MATLAB sets a limit of 100 MB on the amount of memory the Java Virtual Machine (JVM) can allocate. The memory MATLAB Report Generator uses to build a document must fit within this 100 MB limit. If you are having trouble processing large reports, you might need to increase the amount of memory MATLAB Report Generator can allocate. The following sections describe how to do this.

Running MATLAB Without a Desktop

One way to increase the amount of JVM memory available to MATLAB Report Generator is to run MATLAB with `-nodesktop` mode enabled.

Note This option is available on UNIX platforms only.

Increasing the MATLAB JVM Memory Allocation Limit

Increase the amount of JVM memory available by increasing the MATLAB JVM memory allocation limit.

- 1 Create a file named `java.opts`. In this file, include the `-Xmx` option, specifying the amount of memory you want to give the JVM. For example, to increase the JVM memory allocation limit to 128 MB, use the following syntax in the `java.opts` file:

```
-Xmx128m
```

Caution To avoid virtual memory thrashing, never set the `-Xmx` option to more than 66% of available physical RAM.

- 2 Put the `java.opts` file into the appropriate directory. Where you put the file depends on what operating system you are running.
 - On UNIX systems, put the `java.opts` file in a directory where you intend to start MATLAB, and navigate to that directory before starting MATLAB.
 - On Windows systems:
 - a Put the `java.opts` file in a directory where you intend to start MATLAB.
 - b Create a shortcut to MATLAB.
 - c Right-click the shortcut and select **Properties**.
 - d In the Properties dialog box, specify the MATLAB startup directory for the name of the directory in which you created the `java.opts` file.
 - On Mac systems, put the `java.opts` file into the `/Applications/matlab` directory.

Displaying HTML Reports on UNIX

HTML reports may not automatically display on some UNIX platforms. To work around this issue, configure MATLAB Report Generator to launch an external browser:

- 1 In the Report Explorer, click **File > Preferences**.
- 2 Enter the following text in the **View command** field:

```
web(rptgen.file2urn('%file name'), '-browser')
```

Where *file name* is the name of your report template file.

You may also need to modify the `docopt.m` file to specify which Web browser you want MATLAB to use. For more details on how to do this, see the `docopt` function in the MATLAB documentation.

Working with Components

About Components (p. 4-2)

Explains what components are and how to use them

Working with Looping Components (p. 4-3)

Example of how to use the Figure Loop component to loop on Handle Graphics® figures

Working with Property Table Components (p. 4-9)

Example of how to use Property Table components to display property name/property value pairs in tables in generated reports

Working with Summary Table Components (p. 4-21)

Example of how to use Summary Table components to display information about specified objects in tables in generated reports

About Components

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 template. You use components to customize the appearance and output of reports.

The following sections provide examples of how to use different types of components.

Working with Looping Components

In this section...

“About Looping Components” on page 4-3

“Editing the Figure Loop Tutorial Report Template” on page 4-3

“Creating Handle Graphics Figures” on page 4-3

“Figure Properties” on page 4-5

“Editing Figure Loop Components” on page 4-6

About Looping Components

A *looping component* runs its child components a specified number of times. MATLAB Report Generator offers numerous looping components, including logical loops and Handle Graphics® loops. Most loops behave similarly; the Figure Loop used in this example is representative of many types of loops.

The Figure Loop component runs its child components several times. In each iteration, it applies its child components to Handle Graphics figures. The `figloop-tutorial` report template, which is provided with MATLAB Report Generator, creates a report that documents multiple Handle Graphics figures.

For more information on Handle Graphics concepts, see “Handle Graphics” in the Getting Started with MATLAB documentation.

Editing the Figure Loop Tutorial Report Template

To edit the figure loop tutorial report template, enter the following command at the MATLAB command prompt:

```
setedit figloop-tutorial
```

Creating Handle Graphics Figures

There are three important Handle Graphics concepts with which you need to be familiar before proceeding with this tutorial:

- **Invisibility.**

Invisible figures do not appear on screen. They exist, but they are hidden; that is, the `visible` property of these figures is set to `'off'`. To make a given figure visible, set its `visible` property to `'on'`.

Some of the figures used in this example are invisible. To see this, enter the command `figloopfigures` at the MATLAB command line. Figures 1, 3, and 5 appear, but Figures 2 and 4 do not because they are invisible.

- **The `HandleVisibility` parameter.**

The `HandleVisibility` parameter of a Handle Graphics figure allows you to control the availability of the object's handle from the command line and from within callback routines in MATLAB. Valid values for this parameter are `'on'` and `'off'`.

To list all figures currently in your workspace whose `HandleVisibility` parameter is `'on'`, enter the following command at the MATLAB command line:

```
get(0, 'children')
```

MATLAB returns:

```
ans =  
    5  
    1  
    2
```

This indicates that `HandleVisibility` is `'on'` for figures 1, 2, and 5.

To set `HandleVisibility` to `'off'`, enter the following command at the MATLAB command line:

```
set(handle, 'HandleVisibility', 'off')
```

Where the handle of a figure is the figure number.

- **Tags.**

You can set *tags*, or labels, on Handle Graphics figures. Tags do not appear on their associated figures. By default, figure tags are empty.

To see a given figure's tag, enter the following at the MATLAB command line:

```
get(handle, 'tag')
```

Where the handle of a figure is the figure number.

For example, to see whether Membrane Data has a tag, enter the following at the MATLAB command line:

```
get(1, 'tag')
```

MATLAB returns:

```
ans =
    membrane
```

This indicates that figure 1 has a tag called membrane.

Figure Properties

The following table shows the values for the properties of the figures used in this tutorial.

Name	Handle	Visible	HandleVisibility	Tag
Membrane Data	1	on	on	membrane
Invisible Membrane Data	2	off	on	membrane
An Application	3	on	off	app
An Invisible Application	4	off	off	app
Peaks Data	5	on	on	peaks

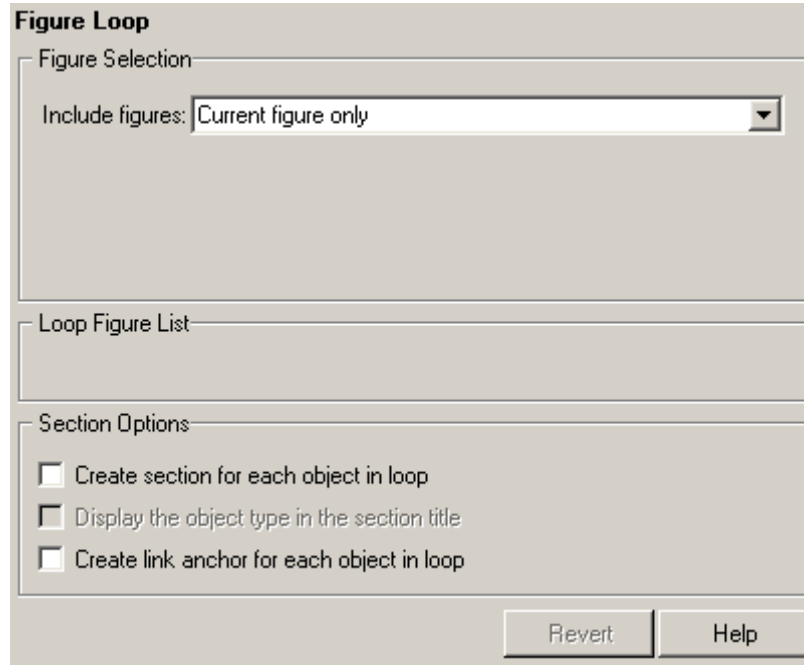
To display the Handle Graphics figures, enter the following at the MATLAB command line:

```
figloopfigures
```

The figures Membrane Data, An Application, and Peaks Data appear on the screen. Invisible Membrane Data and An Invisible Application do not because they are invisible.

Editing Figure Loop Components

Select the Figure Loop component from the Outline pane on the left. The Properties pane for the Figure Loop component appears.



Looping on the Current Figure

To include only the current figure in the report, select **Current figure only** from the **Include figures** selection list. In this case, “current figure” refers to the figure that is current when the report is generated. This may not be the same figure you selected as the current figure in the Report Explorer before report generation. For example, if the report generation process creates figures in your report, the last figure created with `HandleVisibility` set to 'on' is the current figure.

Looping on Visible Figures

To include snapshots of all visible figures in your report, select `Visible` figures in the **Include figures** selection list. This option inserts a snapshot and Property Table for all figures that are currently open and visible.

Including only Figures with `HandleVisibility='on'`.

- 1 Select the **Data figures only (Exclude applications)** option to exclude figures from the loop whose `HandleVisibility` parameter is 'off'.
- 2 Click the **Report** button to generate the report.

The report displays the Membrane Data and Peaks Data figures.

Including All Figures.

- 1 Clear the **Data figures only (Exclude applications)** option.
- 2 Click **Report** to generate the report.

The report now displays all figures, regardless of the value of their `HandleVisibility` parameter.

Looping on Figures with Tags

To include figures specified tags in the report:

- 1 Select the `All figures with tags` option in the **Include figures** selection list.
- 2 Specify a tag in the text box.
- 3 Click **Report** to generate the report.

The report now displays all figures with the specified tag, regardless of the value of their `HandleVisibility` parameter.

Modifying Section Options

In a MATLAB Report Generator loop, a *section* refers to a space in the generated report in which information, including text, images, and tables,

appears. You can alter the appearance of sections in each loop appear in the report by using the options in the Figure Loop component's **Section Options** pane.

Create Section for Each Object in Loop. When this option is selected, the loop automatically creates an individual section for each object found in the loop and uses the object title as the section title. This option is useful when a loop does not contain a Chapter/Subsection component that organizes the loop results.

Display the Object Type in the Section Title. This option is enabled only when **Create section for each object in loop** is selected. It prefaces section titles with object titles in the generated report.

- 1 Clear this option and generate the figloop-tutorial report.

The list of figures produced by the loop looks like this:

```
Membrane Data
Invisible Membrane Data
An Application
An Invisible Application
```

- 2 Now select this option and regenerate the report.

The list of figures now looks like this:

```
Figure - Membrane Data
Figure - Invisible Membrane Data
Figure - An Application
Figure - An Invisible Application
```

Create a Link Anchor for Each Object in Loop. Select this option to create a hyperlink to the object in the generated report.

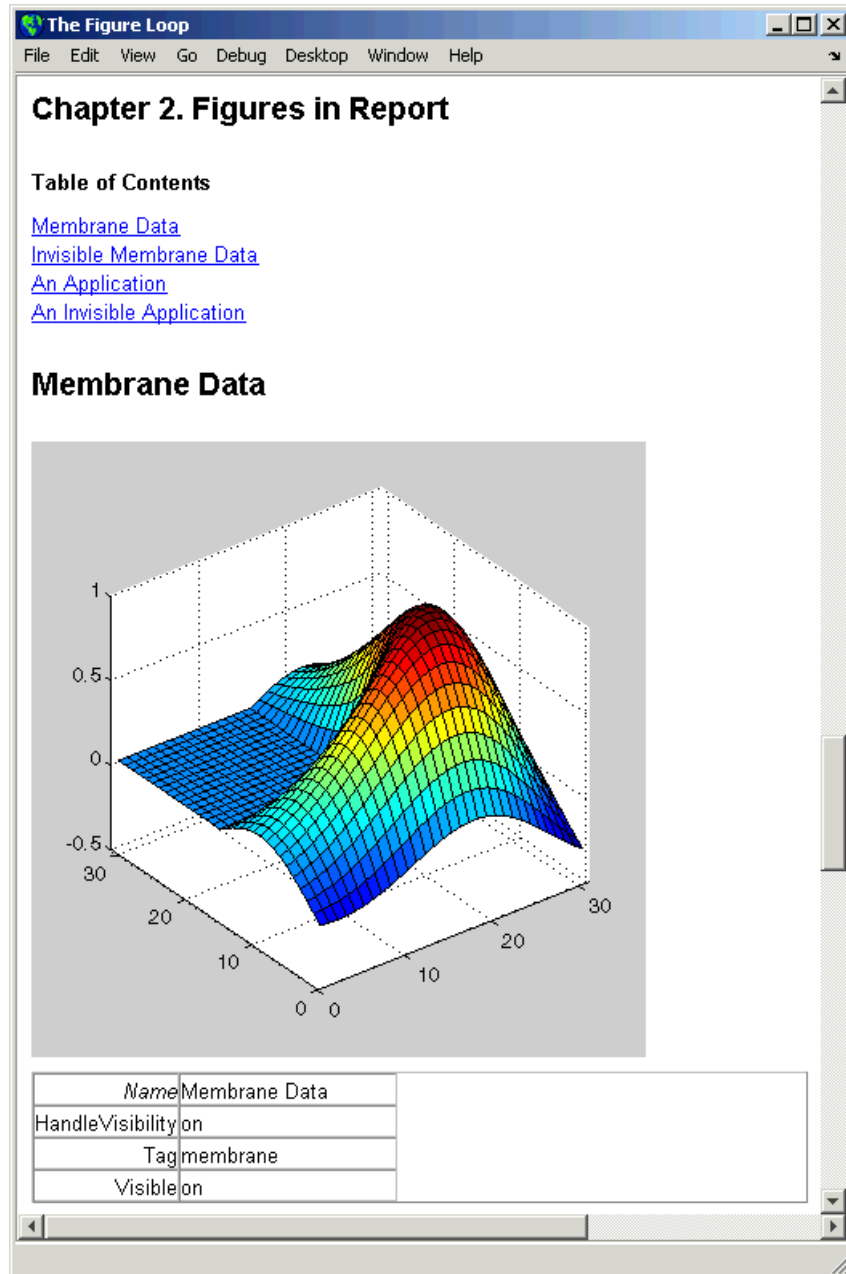
Working with Property Table Components

In this section...

- “About Property Table Components” on page 4-9
- “Opening the Example Report Template” on page 4-11
- “Examining the Property Table Output” on page 4-11
- “Selecting Object Types” on page 4-12
- “Displaying Property Name/Property Value Pairs” on page 4-12
- “Editing Table Titles” on page 4-16
- “Entering Text into Table Cells” on page 4-16
- “Adding, Replacing, and Deleting Properties in Tables” on page 4-17
- “Displaying or Hiding Cell Borders” on page 4-18
- “Adding and Deleting Columns and Rows” on page 4-19
- “Resizing Columns” on page 4-19
- “Zooming and Scrolling” on page 4-19
- “Selecting Tables” on page 4-20

About Property Table Components

Property Table components display property name/property value pairs for objects in tables, as shown in the example table below from the `figloop-tutorial` report.



MATLAB Report Generator offers many types of Property Table components, including:

- MATLAB Property Table
- Simulink Property Table
- Stateflow Property Table

Property Table components all behave similarly. The Handle Graphics Property Table component used in this example is representative of all Property Table components.

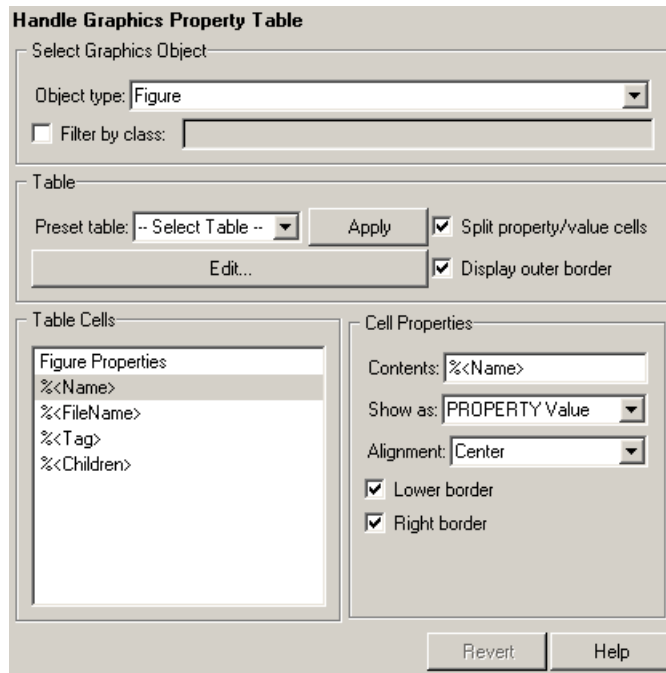
Opening the Example Report Template

This example uses the `figloop-tutorial` report template. To open the figure loop tutorial report template, enter the following at the MATLAB command line:

```
setedit figloop-tutorial
```

Examining the Property Table Output

Property pages for all Property Table components are similar in form. The following figure shows the Properties pane for the Handle Graphics Property Table component. To modify table settings, click **Edit**.



Selecting Object Types

Property Table components offer multiple object types on which to report. For example, the Handle Graphics Property table lets you report on a figure, an axes object, or a Handle Graphics object.

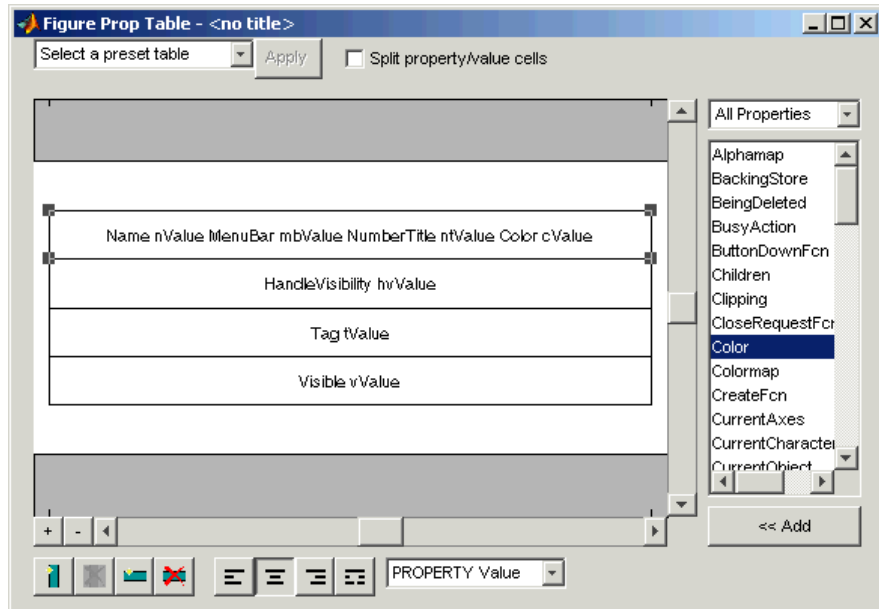
You can select a different object type on which to report in the **Object type** list in the component's Properties pane.

Displaying Property Name/Property Value Pairs

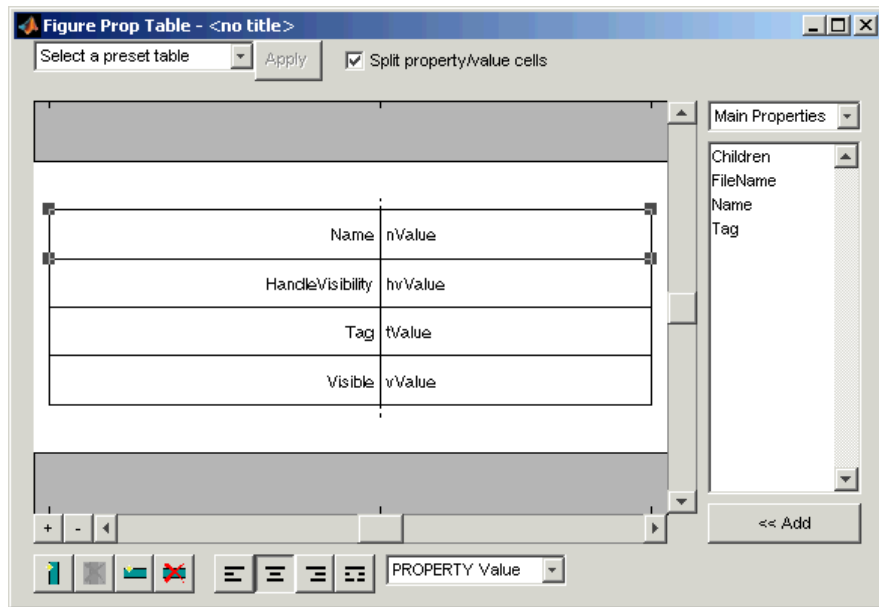
Split Property/Value Cells

In the Handle Graphics Property Table component's Properties pane, click **Edit**.

- 1 For a given property name and property value to appear together in one cell of a table, clear the **Split property/value cells** check box. The table is now in *nonsplit mode*. Nonsplit mode supports more than one property name/property value pair per cell and text.

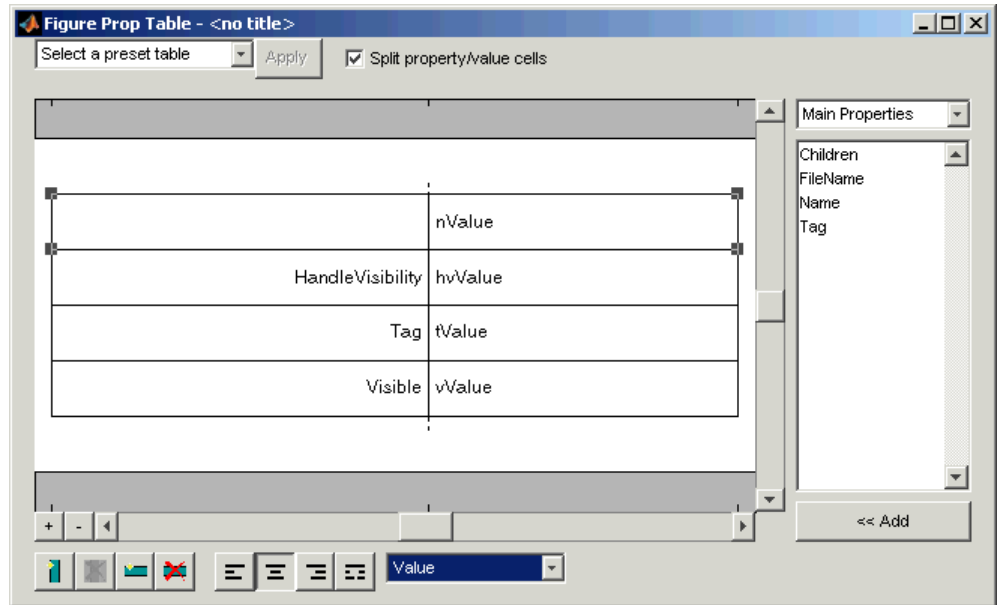


- 2 For the property name and property value to appear in adjacent horizontal cells in the table, select the **Split property/value cells** check box. The table is now in *split mode*. Split mode supports only one property name/property value pair per cell. If more than one property pair is included in a cell, only the first pair appears in the report; all subsequent pairs are ignored.



Display Options

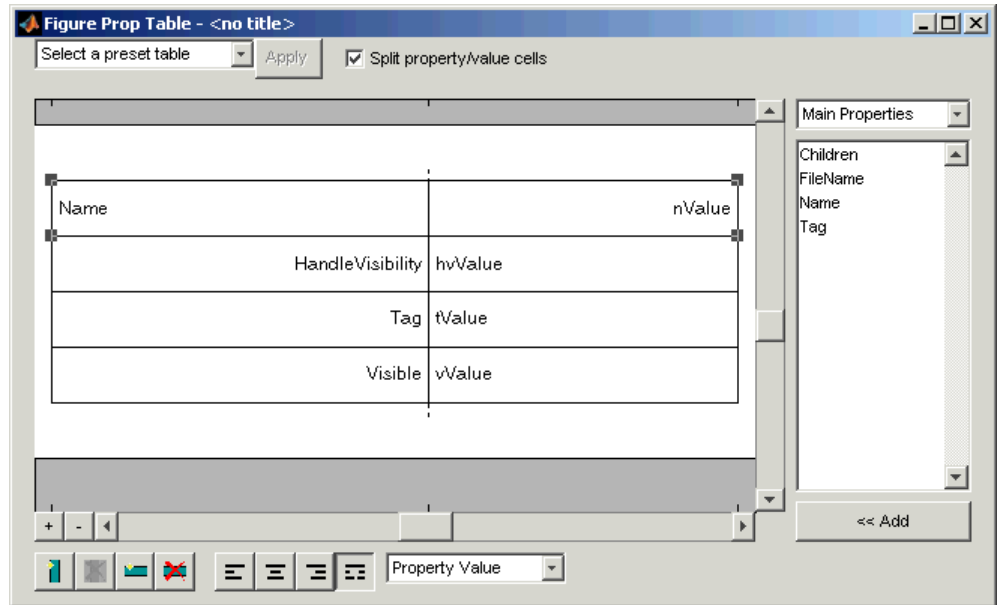
Property name/property value pairs can display in cells in several ways. To specify how a given property name/property value pair should appear in a cell, select that field in the table and choose Value from the **Display As** selection list under the Figure Properties box. In the selected table row, only the value appears.



Formatting Options

To specify alignment for text in a given cell, use the four justification buttons in the toolbar below the Figure Properties box

In this example, the selected table row is double justified. To change justification options, right-click a table row and make the changes in its context menu.



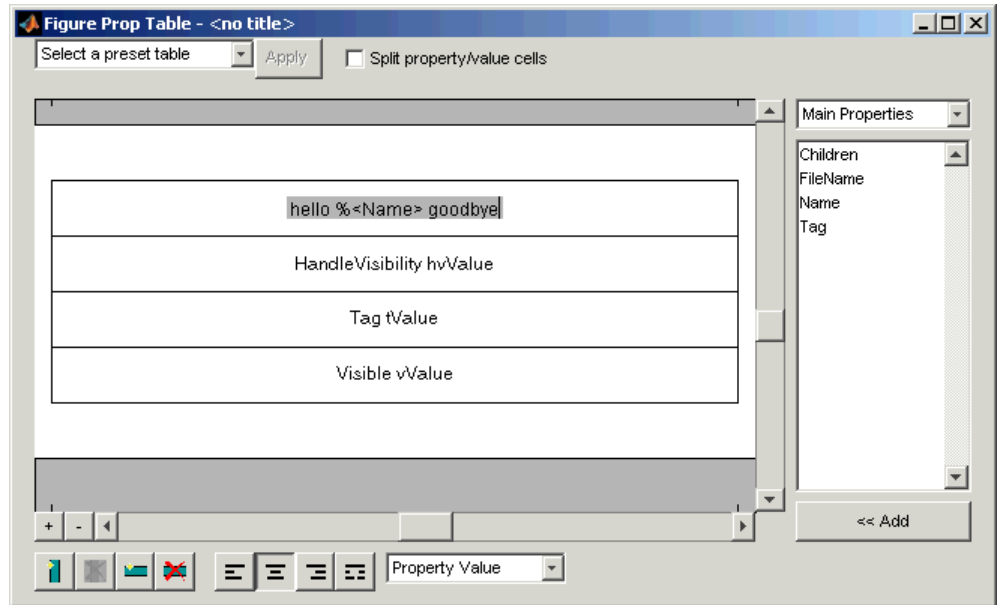
Editing Table Titles

Table titles can contain properties and text. By default, the title of a table is the same as the value of the %<Name> property. You can modify this property to modify the table title.

Note Table titles are always in nonsplit mode.

Entering Text into Table Cells

To enter text into a table cell, double-click the cell. A gray box appears with the label for the cell property.



If you type text outside the angle brackets, the text appears as is in the report. Text inside the table brackets must specify a valid property name. If you enter an invalid property name, the property name appears in the report without a property value.

Note For the text to be visible, the table must be in nonsplit mode.

Adding, Replacing, and Deleting Properties in Tables

Adding Table Properties

To add a Handle Graphics property to a table cell:

- 1 In the Figure Prop Table window, add a new table row to the current table.
 - a Select a table row directly below the row to which to add a new property.
 - b Click **Add Row Above Current Cell**.

A new row is added above the current row.

- 2 Add the property to the new table row.
 - a Select the new table row.
 - b In the Properties Type selection menu at the upper right of the window, select a property type.
 - c In the **Properties** menu, select the property you want to add.
 - d Click the << **Add** button, or double-click the property name. The property appears in the table row.

Alternatively, if you know the name of the property you want to add, enter the property name directly into the cell as described in “Entering Text into Table Cells” on page 4-16. For information about adding new table rows, see “Adding and Deleting Columns and Rows” on page 4-19.

Replacing Table Properties

To replace a property in a cell of a table in split mode, follow the instructions in “Adding Table Properties” on page 4-17.

Note You cannot use these steps to delete a property in a cell when the table is in nonsplit mode, because cells can have more than one property name/property value pair.

Deleting Table Properties

Delete a property by backspacing over it or using the **Delete** key.

Displaying or Hiding Cell Borders





To toggle cell borders on and off:

- 1 Place your cursor in a cell and right-click to invoke its context menu.
- 2 Choose **Cell borders > Top, Bottom, Right, or Left** to toggle the specified border on or off.

Adding and Deleting Columns and Rows

To add or delete a column or row, select a cell and then click one of the buttons described in the following table.

Note You cannot delete a row or column when it is the only row or column in the table.

Button	Action
	Add column (added to the left of the selected column)
	Delete selected column
	Add row (added above the selected row)
	Delete selected row



Resizing Columns

To resize the width of a column, click and drag its vertical borders as needed.

Zooming and Scrolling

Zooming

While you are editing the table, you can zoom in and out of the table with the zoom buttons, which are located to the left of the horizontal scroll bar.

Button	Action
	Zoom in
	Zoom out

Scrolling

You can scroll vertically and horizontally using the table's scroll bars.

Selecting Tables

To display property name/property value pairs, you can select a preset table or use a custom table.

- A preset table is built-in and formatted. You can select a preset table in the preset table selection list in the upper left of the Figure Prop Table window. To apply a preset table, select the table and click **Apply**.
- To create a custom table, select a preset table and modify it to fit your needs by adding and/or deleting rows and properties. You may want to start with the **Blank 4x4** preset table.

Note You cannot save a custom table as a preset table. If you try to do so, all the changes to the custom table are lost.

Working with Summary Table Components

In this section...
“About Summary Table Components” on page 4-21
“Opening the Example Report Template” on page 4-23
“Selecting Object Types” on page 4-23
“Adding and Removing Properties” on page 4-23
“Setting Relative Column Widths” on page 4-24
“Setting Object Row Options” on page 4-24

About Summary Table Components

Summary Table components insert tables that include specified properties for objects into generated reports. Tables contain one object per row, with each object property displayed in a column, as shown below in a Summary Table in the `figloop-tutorial` report.

Table 1.1. Figure Properties

Name	Tag	Visible	HandleVisibility
<u>Membrane</u> <u>Data</u>	membrane	on	on
<u>Invisible</u> <u>Membrane</u> <u>Data</u>	membrane	off	on
<u>An</u> <u>Application</u>	app	on	off
<u>An</u> <u>Invisible</u> <u>Application</u>	app	off	off
Peaks Data	peaks	on	on

MATLAB and Simulink Report Generator offer many types of Summary Table components, including:

- Handle Graphics Summary Table
- Simulink Summary Table
- Stateflow Summary Table

Summary Table components all behave similarly. The Handle Graphics Summary Table component used in this example representative of all Summary Table components.

Opening the Example Report Template

This example uses the `figloop-tutorial` report template. To open the figure loop tutorial report template, enter the following at the MATLAB command line:

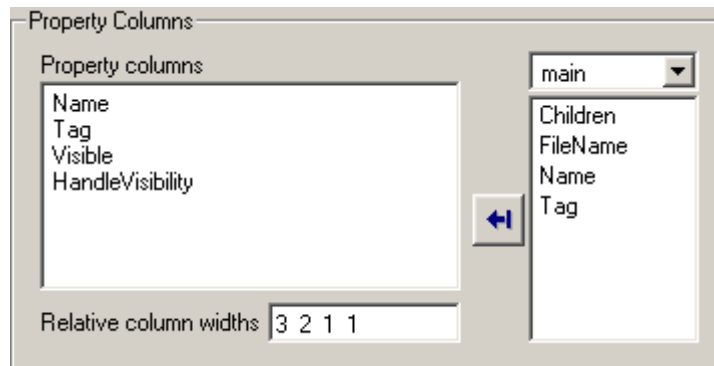
```
setedit figloop-tutorial
```

Selecting Object Types

You can use the **Object type** selection list to choose Handle Graphics object types for the Summary Table, including blocks, signals, systems, and models. The `figloop-tutorial` reports on figure objects.

Adding and Removing Properties

You can select object properties to be displayed in the Summary Table from the **Property Columns** pane. To add a property to the summary table, select the property category from the property category box to the right of the **Property columns** field. Selecting different property categories causes the field below the box to display the selected category's properties. The following figure shows `main` as the selected category.



- To add a property:
 - e** Select the category from the property category box.
 - f** Select a property in the properties list.

g Click the Add property  button.
The property appears in the **Property columns** field.

- To remove a property from the table:
 - Select the property in the **Property columns** field.
 - Press **Delete**.
The property name is removed from the **Property columns** field.

Note After making changes in the Report Explorer interface, you must click **Apply** for the changes to take effect.

You can define your own properties by entering their names into the **Property columns** field. You must use valid Report Generator variable notation. For details, see %<VariableName> Notation on the Text component reference page.

Setting Relative Column Widths

Use the **Width** field to apply a relative column width to the Summary Table columns in the generated report. If omitted, column widths are determined automatically.

Setting Object Row Options

You can use the **Object Rows** pane to set options for table rows, including anchor, filtering, and sorting options. Select **Insert anchor for each row** to place an anchor in each table row in the report. Use the **Include figures** list to specify what objects to include in the Summary Table.

Summary Table components in figloop-tutorial report on figure objects. For more information on options for these figure objects, see the following sections:

- “Looping on the Current Figure” on page 4-6
- “Looping on Visible Figures” on page 4-7
- “Looping on Figures with Tags” on page 4-7

Creating Custom Components

About Custom Components (p. 5-2)	Why create custom components?
Component Creation Process (p. 5-3)	Overview of steps required to create components
Defining Components (p. 5-6)	How to define, modify, and remove custom components
Defining Component Tasks (p. 5-15)	Explains how to customize your component to perform specified tasks
Examples (p. 5-22)	Examples of how to create custom components to display securities data in generated reports

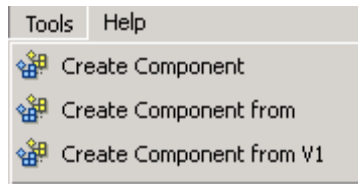
About Custom Components

In most cases, the components provided with MATLAB Report Generator should be more than adequate for your needs. You can, however, create custom components if you want to generate a report that cannot be created by the standard MATLAB Report Generator components. For example, you can create a component that inserts a corporate logo into your report, or a component that plots data.

Component Creation Process

To create a component:

- 1 Open the Report Explorer.
- 2 Select one of the component creation choices from the **Tools** menu:

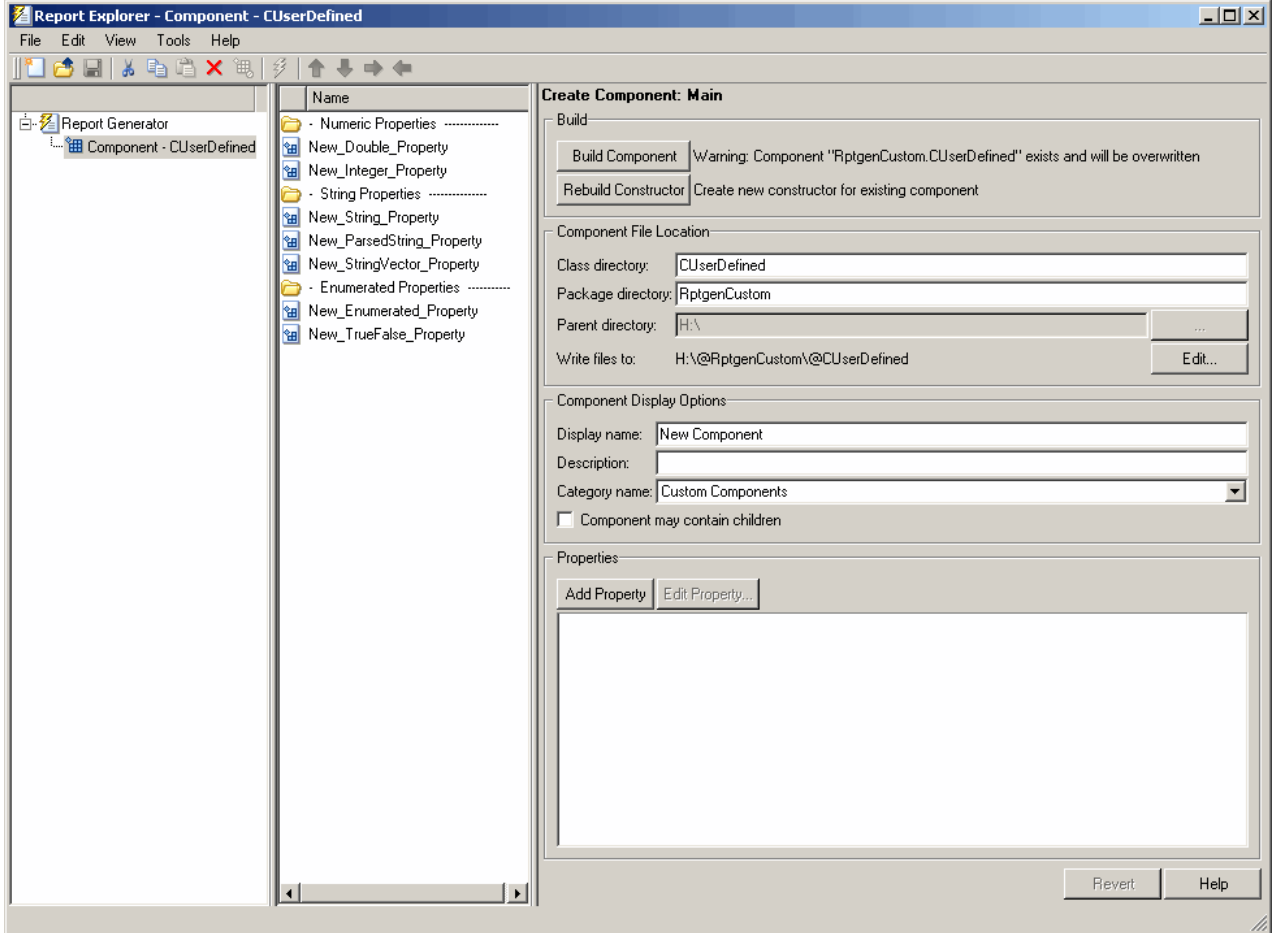


- To create a new component, select **Create Component**.
- To create a new component from an existing component, select **Create Component from**.
- To create a component from an existing version 1 component, select **Create Component from V1**.

Tip You can also create a new component by clicking the **Create a new user-defined reporting component** link in the Report Explorer Properties pane on the right.

The Report Explorer displays as follows:

- The Outline pane on the left displays the structure of components you create.
- The Options pane in the middle lists properties you add to components.
- The Properties pane on the right specifies the behavior of component properties.



- 3 Specify properties of the component in the Properties pane of the Report Explorer. For more information, see “Defining Components” on page 5-6.
- 4 Specify tasks you want the component to perform by editing the M-files that comprise the framework of the component. For more information, see “Defining Component Tasks” on page 5-15.
- 5 Build the component. For more information, see “Building Components” on page 5-13.

After you build the custom component, you can use it to specify options for your generated report in the report template.

Note You must restart MATLAB before using a newly created or rebuilt component.

Defining Components

In this section...
“Required Component Data” on page 5-6
“Specifying the Location of Component Files” on page 5-6
“Setting Component Display Options” on page 5-7
“Specifying Component Properties” on page 5-9
“Modifying Existing Components” on page 5-13
“Building Components” on page 5-13
“Rebuilding Existing Components” on page 5-13
“Removing a Component” on page 5-13

Required Component Data

When creating a component, you need to specify the following information:

- 1** The path where you want to put the directory that contains all files for the component. For information on how to specify this directory, see “Specifying the Location of Component Files” on page 5-6.
- 2** Properties of the component. For more information, see “Specifying Component Properties” on page 5-9.
- 3** Display options for the component, including its display name, category, and description. For more information, see “Setting Component Display Options” on page 5-7.

Specifying the Location of Component Files

You can create components that perform similar functions and group them together in *Package Directories*. Each package directory must have a *Parent Directory* that is in the MATLAB path. When you build a new component, MATLAB Report Generator creates files that make up the component. These files are stored in the directory structure `<parent>/@package_name/@class_name`.

Specify these directories in the following fields in the **Component File Location** area of the Properties pane:

- 1 Class Directory Field.** Specify a class name for your component. The build process creates a directory with the name you specify and places the component's files in it. The class directory name must be unique for each component in the package. By convention, component class names begin with an uppercase or lowercase letter c; for example, `cUserDefinedComponent`.
- 2 Package Directory Field.** Specify the directory in which to store files for groups of components you create. Files for each component are stored in a subdirectory with the name you entered into **Class Directory Field**.
- 3 Parent Directory Field.** Specify this directory when you create a package for the first time. This directory is the parent directory of the Package Directory.

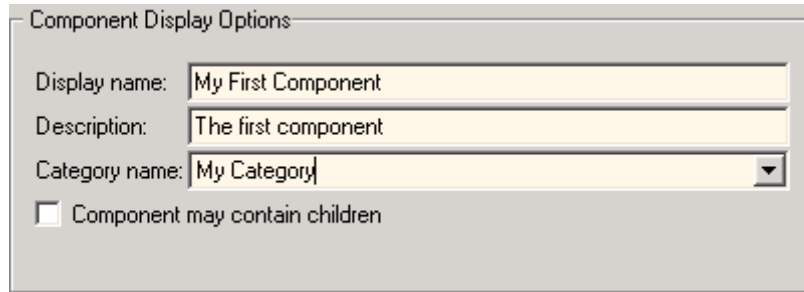
Setting Component Display Options

You can specify how you want your component to appear in the Report Explorer by entering data in the **Component Display Options** area of the Properties pane. Enter the following information:

- 1 Display Name.** Specify a display name for the component to appear in the list of components for its associated category. Component categories and display names appear in the Options pane in the middle of the Report Explorer.

For information on specifying component categories, see step 3, **Category Name**.

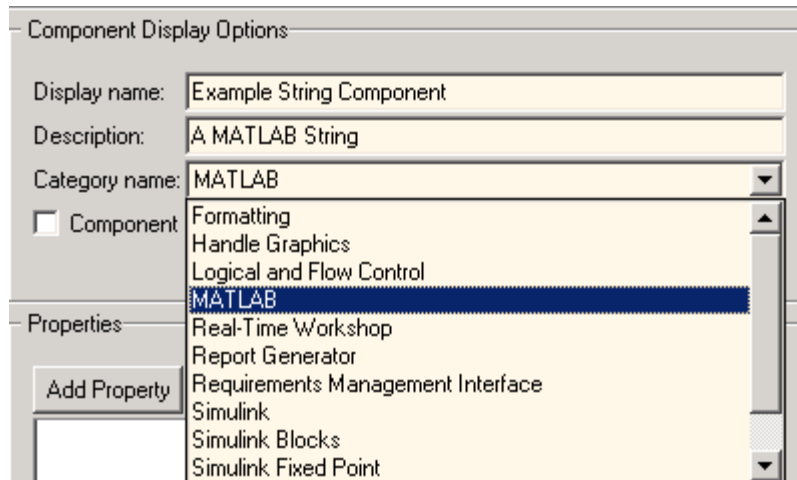
The following example shows how to create a component called My First Component in a category called My Category.



2 Description. Enter a description for the component. This description appears when you click the component name or category name in the Options pane in the middle of the Report Explorer. Make the description informative, but brief.

3 Category Name. Specify the category of components to which the new component belongs. The component appears under this category in the Options pane in the middle of the Report Explorer.

Predefined choices appear in the **Category name** list. Select an existing component category from this list.



To create a new component category, type the name for the category into the **Category name** field. This category name is added to the list of available categories in the Report Explorer.



Category name: My category

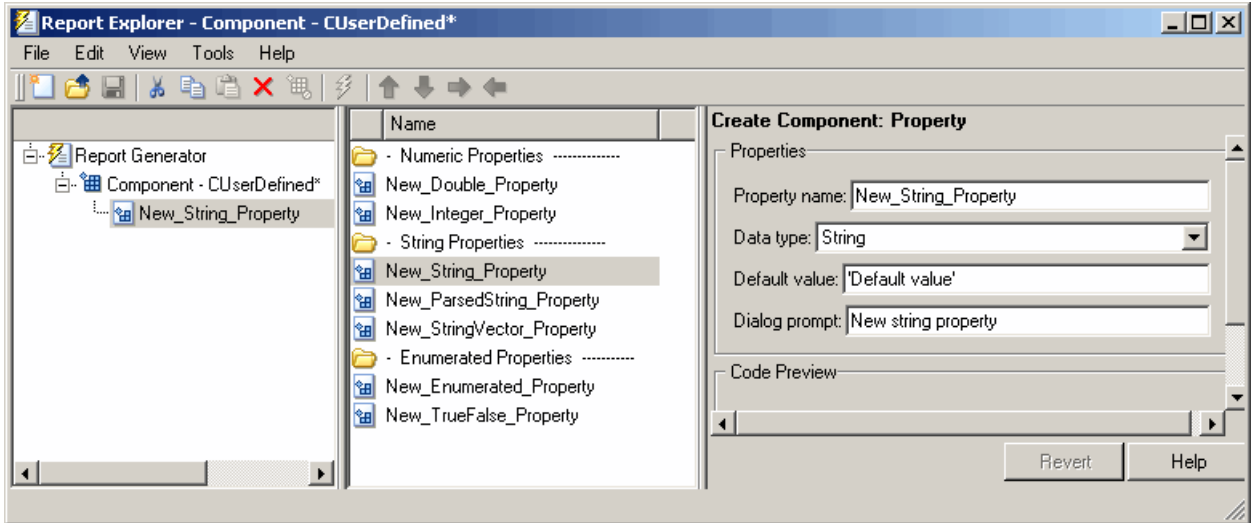
Component may contain children

4 child components.

Select the **Component may contain children** check box if you want the component to have child components. Child components appear under the component in the Report Explorer hierarchy. During report generation, the component runs all child components and includes their output in the report.

Specifying Component Properties

Component properties determine how a component behaves and what information it inserts into a report. Double-clicking a property name for a component shown in the Outline pane on the left in the Report Explorer causes the Properties pane on the right to display the current values for the property. For example, the following figure displays the property values for `New_String_Property`.

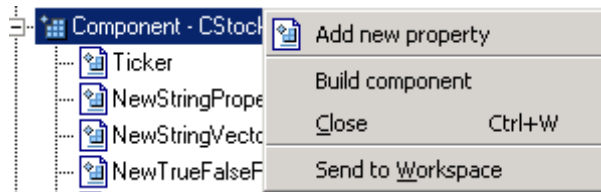


Adding Properties to Components

You add properties to a component from the properties list. Each property has a default value that you can modify as needed.

There are several ways to add properties to components:

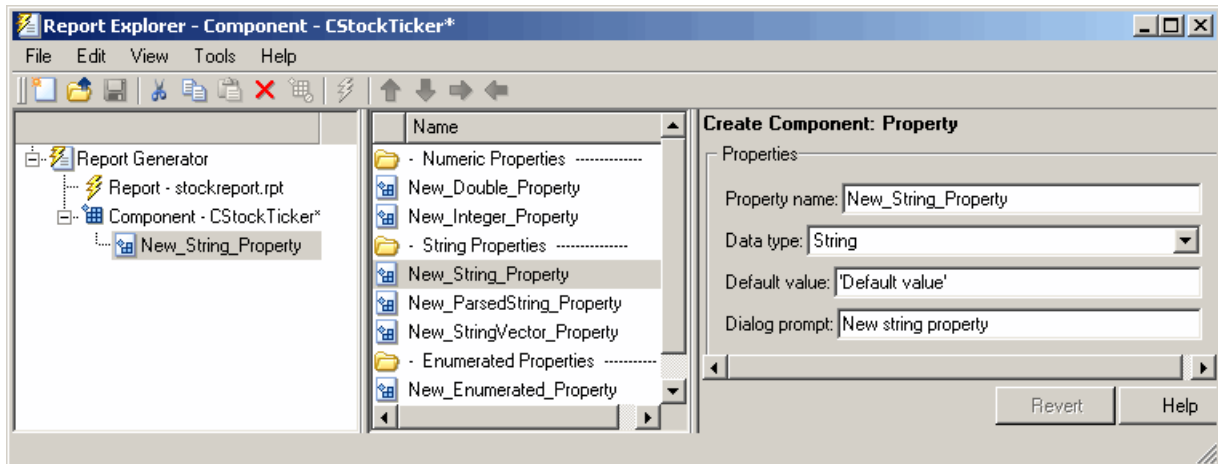
- 1 Right-click the name of the component to which you want to add properties in the Outline pane on the left. Select **Add new property** from its context menu.



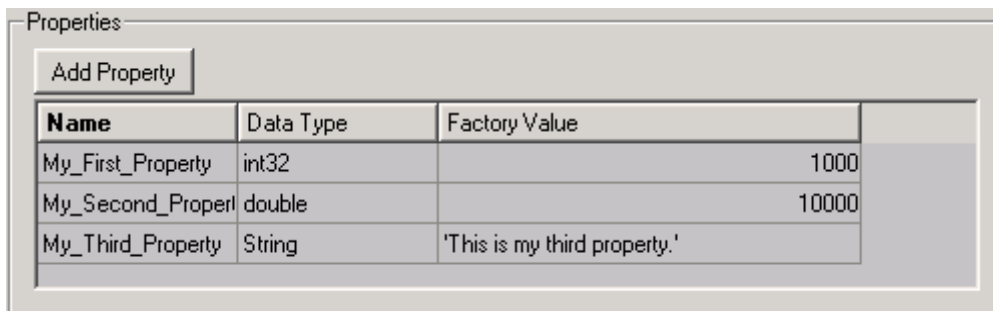
- 2 Right-click the name of a predefined property in the Options pane in the middle. From the context menu, select **Add property**.



- 3 Left-click the name of a property in the Options pane and drag it on top of the name of a component in the Outline pane on the left.
- 4 Double-click the property name in the Options pane in the middle. The property is added to the component and property values appear in the Properties pane on the right.



- 5 Click the **Add Property** button on the Properties pane on the right.



Specifying Component Properties

1 Property Name. Create a name for the new property. A property name must be a valid MATLAB variable name, and must be unique within a component.

2 Data Type. Specify the property's data type. Options are:

- Double
- Enumeration
- Integer
- String
- String Vector
- %<Parsed String>

Use this data type to include the value of a variable in the MATLAB workspace in a component. For more information about this data type, see “%<VariableName> Notation” on page 8-71 on the Text reference page.

- True/False

For more details about data types in MATLAB, see “Data Types” in the MATLAB Programming documentation.

3 Default Value. Set a default value for the property. The default value needs to be compatible with the data type. If incompatibilities exist between the default value and the data type, the component may not build.

4 Dialog Prompt. This text appears next to the widget on the component's dialog box. It indicates what the property does and what effect it has on report generation.

Note When the component is built, a colon is appended to your entry in the **Dialog prompt** field. Your entry appears in the Properties pane with the colon appended.

Modifying Existing Components

The Report Explorer modifiable components allow you to derive a new component from an existing component. To do this, double-click the name of the component you want to modify. The Report Explorer appears, showing values and properties previously specified for this component.

Building Components

After you have entered all data required for defining the component, you build it by clicking the **Build Component** button. The build process creates all files needed for the component and stores them in the specified directory. For more information about specifying where components are stored, see “Specifying the Location of Component Files” on page 5-6.

Note Existing files in this location are overwritten.

Rebuilding Existing Components

When you need to add, remove, or change properties of an existing component, use the **Rebuild Constructor** button. This button becomes active only after you have previously created a component using the **Build Component** button. To activate the **Rebuild Constructor** button, specify the **Package name** and **Class name** for an existing component in the **Component File Location** area of the Properties pane.

If you select a component using **Tools > Create component from**, the component's fields are filled in automatically and the button becomes active.

After you have finished modifying the component, click the **Rebuild Constructor** button to rebuild the component. Writable files in the component's directory location are not overwritten.

Removing a Component

To remove a component:

- 1** Delete its class directory, `<root>/@package_name/@class_name`. If the component you want to remove is the only component in the package, delete the entire package.
- 2** Edit `<root>/@package_name/rptcomps2.xml` to remove the XML element that registers the component.

Defining Component Tasks

In this section...

“About Component Customization” on page 5-15

“Required Customization: Specifying Format and Content of Report Output” on page 5-16

“Changing a Component’s Outline String in the Report Explorer Hierarchy” on page 5-18

“Modifying the Appearance of Properties Dialog Boxes” on page 5-19

“Specifying Additional Component Properties” on page 5-19

About Component Customization

Building a component creates M-files in your MATLAB workspace. You can specify tasks you want your component to perform by editing these M-files.

Note You *must* specify the format and content of your report output by editing `execute.m`. This file is called during report generation to invoke your component’s tasks. You can optionally specify additional component properties and behavior by editing other M-files.

For more information, see the following sections:

- “Required Customization: Specifying Format and Content of Report Output” on page 5-16
- “Changing a Component’s Outline String in the Report Explorer Hierarchy” on page 5-18
- “Modifying the Appearance of Properties Dialog Boxes” on page 5-19
- “Specifying Additional Component Properties” on page 5-19

Required Customization: Specifying Format and Content of Report Output

After you build a new component, you must edit the `execute.m` file to specify the format and content of your report output.

The `execute` command has the following syntax:

```
out = execute(thisComp, parentDoc)
```

Where:

- `thisComp` is a handle to the component that you are running.
- `parentDoc` is a handle to the document that you are generating.
- `out` is a Document Object Model (DOM) node or string to be added to the report.

For information on manipulating DOM nodes, see `xmlwrite` in the MATLAB documentation.

One or more default lines of code within the `execute.m` file show each property for the component. Here is an example of a component property line within an `execute.m` file:

```
pstring = thisComp.NewStringProperty; % New string property;
```

The following sections describe how to edit `execute.m` to create additional report elements.

Creating Tables

To create a table, replace the Source property value with the name of a cell array or structure:

```
out = execute(rptgen.cfr_table(...  
    'Source', tableSrc,...  
    'numHeaderRows',1,...  
    'TableTitle', 'Example Title'),...  
    parentDoc);
```

For more information, enter `help(rptgen.cfr_table)` at the MATLAB command line.

Creating Lists

To create a list, replace the Source property value with the name of a cell vector:

```
out = execute(rptgen.cfr_list(...
    'Source', listSrc,...
    'ListStyle','orderedlist',...
    'ListTitle','Example List'),...
    parentDoc);
```

For more information, enter `help(rptgen.cfr_list)` at the MATLAB command line.

Creating Text

To create text, replace the ParaText property value with a text string:

```
out = execute(rptgen.cfr_paragraph(...
    'ParaText', paraSrc,...
    parentDoc);
```

For more information, enter `help(rptgen.cfr_paragraph)` at the MATLAB command line.

Creating Figures

To create figures, specify a figure in the FigureHandle property value.

```
figSrc =(gcf;
out = execute(rptgen_hg.chg_fig_snap(...
    'FigureHandle', figSrc,...
    'Title', '',...
    'isResizeFigure', 'manual',...
    'PrintSize', [6 4],...
    'PrintUnits', 'inches'),...
    parentDoc);
```

For more information, enter `help(rptgen_hg.chg_fig_snap)` at the MATLAB command line.

Running Child Components

The following code runs child components. The first line calls `execute.m` for child components. The second line appends the results of running the child components to the report:

```
childOut = thisComp.runChildren(parentDoc);  
out = parentDoc.createDocumentFragment(out, childOut);
```

Changing a Component's Outline String in the Report Explorer Hierarchy

To change the string used to describe the component in the Report Explorer hierarchy, edit the `getOutlineString` M-file. By default, `getoutlinestring` returns the display name of the component. The `getOutlineString` command has the following syntax:

```
olstring = getOutlineString(thisComp)
```

Where:

- `thisComp` is the component whose description you are specifying.
- `olstring` is a single-line string that displays information about the component. It can contain a maximum of 32 characters.

Customize the string to include additional information about the component, such as information about its properties. In the following example, the `truncatestring` function converts input data into a single-line string. The second argument is the return value if the data is empty. The third argument is the maximum allowed size of the resulting string.

```
cInfo = '';  
pstring = rptgen.truncateString(thisComp.string, '<empty>', 16);
```

Use a dash (-) as a separator between the name and additional component information, as follows:

```
if ~isempty(cInfo)
    olstring = [olstring, '- ', cInfo];
end
```

Modifying the Appearance of Properties Dialog Boxes

You can edit the `getdialog boxeschema` file to control most aspects of dialog box layout, including:

- Creation and placement of widgets
- Organization of widgets into panes
- Creation of the top-level display within which panes reside

The syntax of the `getdialog boxeschema` command is:

```
dlgstruct = getdialog boxeschema(thisComp, name)
```

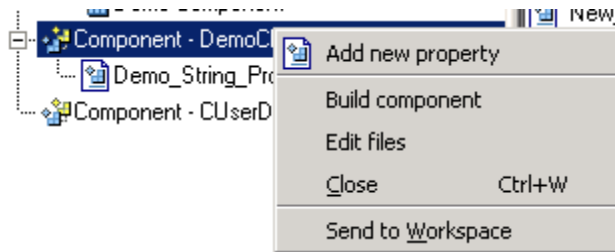
Where:

- `thisComp` is the instance of the component being edited.
- `name` is a string that can be passed to `getdialog boxeschema` to build a specific type of pane. Usually, `name` is empty in the Report Explorer.

Note The MathWorks does not recommend modifying fields that are not explicitly mentioned in this file. These fields are subject to change in future releases.

Specifying Additional Component Properties

You can edit additional M-files to further customize your component. To access these files, right-click the component in the Outline pane on the left in the Report Explorer and select **Edit files** from its context menu.



For more information, see the following sections:

- “Specifying Whether Components Can Have Children Components” on page 5-20
- “Modifying a Component’s Description” on page 5-20
- “Changing a Component’s Display Name” on page 5-21
- “Changing a Component’s Category Name” on page 5-21
- “Registering Components” on page 5-21
- “Displaying Component Help in the MATLAB Help Browser” on page 5-21

Specifying Whether Components Can Have Children Components

To specify whether a component can have children, edit `getParentable.m`. This command returns the value `true` or `false`. For example, if you no longer want your component to have child components, modify the value within the code as follows:

```
p = false;
```

Modifying a Component’s Description

The description in `getDescription.m` is the same value as the **Description** field in the Report Explorer. The following example shows how to edit the `compDesc` string in this file to change a component’s description to A demonstration component:

```
compDesc = 'A demonstration component';
```

Changing a Component's Display Name

The display name in `getName.m` is the same value as the **Display name** field in the Report Explorer. The following example shows how to edit the `compName` string in this file to change a component's display name to Demo Component:

```
compName = 'Demo Component';
```

Changing a Component's Category Name

The category name in `getType.m` is the same value as the **Category name** field in the Report Explorer. The following example shows how to edit the `compCategory` string in this file to change a component's category name to Custom Components:

```
compCategory = 'Custom Components';
```

Registering Components

You can register components in the Report Explorer using `rptcomps2.xml`. This file also helps build the list of available components.

The content of this file must be consistent with the values in the `getName.m` and `getType.m` files. If you have changed values in either of these files, you must also change their values in `rptcomps2.xml`. You must restart MATLAB for the Report Explorer to display the new information.

Displaying Component Help in the MATLAB Help Browser

The `viewHelp.m` file displays a help file for the component within the MATLAB Help browser. To display the help file, highlight the name of the component in the Report Explorer and click **Help**.

Examples

In this section...
“Fetching Securities Data and Displaying It in a Table” on page 5-22
“Displaying Securities Data in Two Tables” on page 5-27

Note These examples require Datafeed Toolbox.

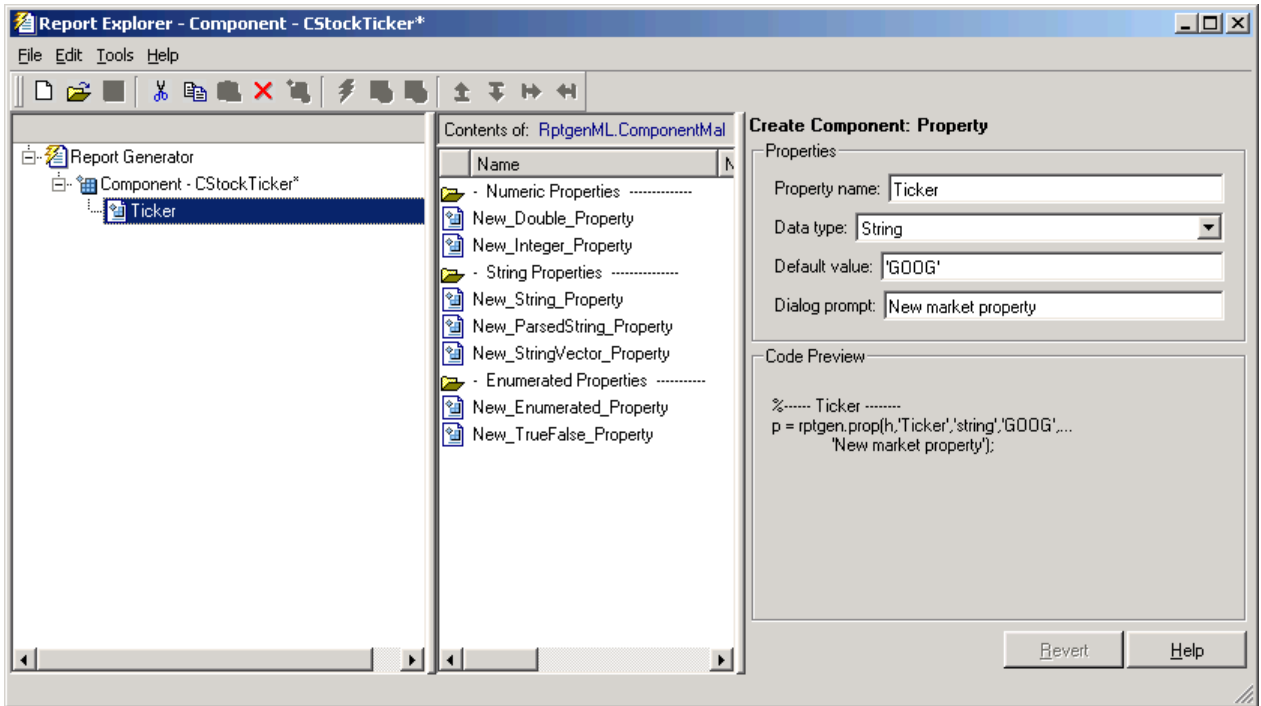
Fetching Securities Data and Displaying It in a Table

This example shows how to create a component that fetches securities data and displays it in a report as a table.

- 1** Create a component named `Equity Values` in the class directory named `CStockTicker`.
- 2** Give the component one string property named `Ticker`, and specify its attributes.
 - a** In the Options pane of the Report Explorer, double-click **New_String_Property**.
 - b** For **Property name**, specify a valid MATLAB variable name.
 - c** Specify the property’s data type. In this case, `Ticker` is a string value, which is the default data type.
 - d** Specify the property’s default value.

Because this example fetches ticker values for the security Google, set the **Default value** to `'GOOG'`. (The single quotation marks are required to make this a string value.)

Your specified settings appear in the **Code Preview** pane.



- 3 To build the new component, click the **Build** button in the Report Explorer. The Equity Values component now appears in the Options pane in the middle of the Report Explorer.



- 4 Edit the component's `execute.m` file to retrieve stock market data and display it in a table in the generated report.
- In the `@CStockTicker` directory, open `execute.m`.
 - Enter the following text into `execute.m`.

```
function out=execute(thisComp,parentDoc,varargin)
```

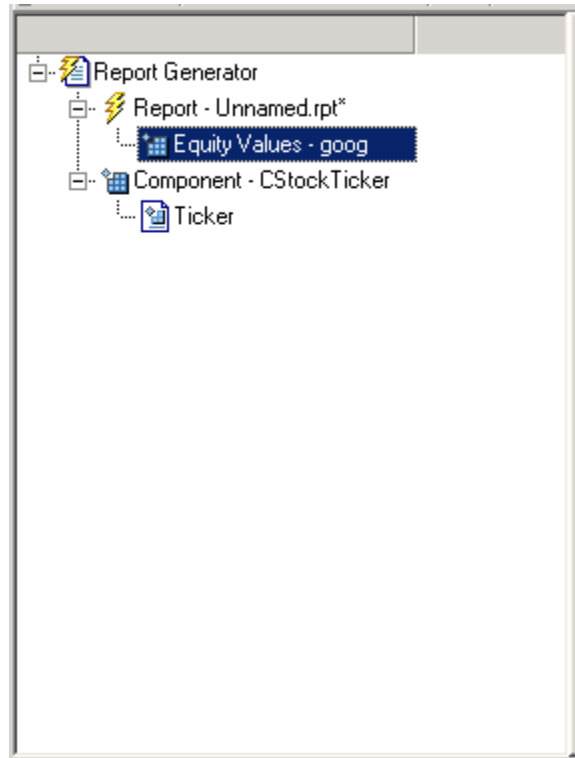
```
stockQuote = fetch(GOOG, thisComp.Ticker);
stockQuote.Date = datestr(stockQuote.Date,1);
stockQuote.Time = datestr(stockQuote.Time,13);
out = execute(rptgen.cfr_table(...
    'Source', stockQuote,...
    'numHeaderRows', 0,...
    'TableTitle', 'Stock Market Pricing Data'),...
parentDoc);
```

- 5** Append the security symbol, goog, to the component name. Enter the following text into `getOutlineString.m`.

```
function olstring=getOutlineString(thisComp)

    olstring = [getName(thisComp),' - ',thisComp.Ticker];
```

The component name now appears as `Equity Values goog`.



- 6 Modify the `getdialog boxeschema.m` file to change the appearance of the Properties pane. Enter the following text into this file to display the last quoted price for the security in the Properties pane.

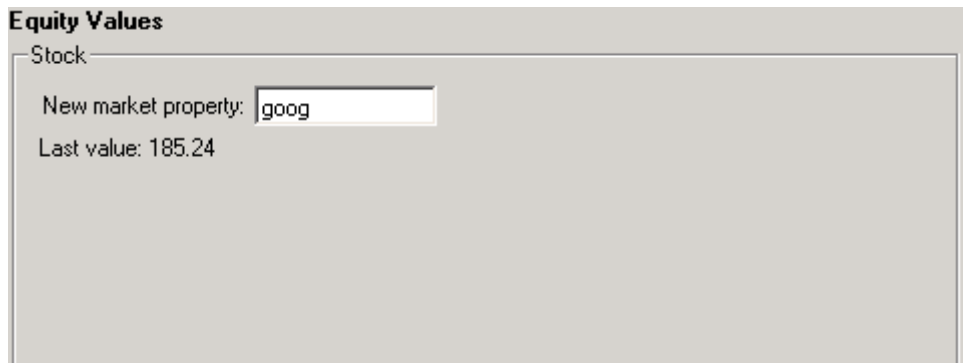
```
function dlgStruct = getdialog boxeschema(thisComp, name)

try
    currQuote = fetch(yahoo, thisComp.Ticker);
    quoteStr = sprintf('Last value: %g', currQuote.Last);
catch
    quoteStr = sprintf('Warning: ...
    "%s" is not a valid symbol.', thisComp.Ticker);
end

dlgStruct = thisComp.dlgMain(name,...
```

```
thisComp.dlgContainer({
  thisComp.dlgWidget('Ticker',...
    'DialogRefresh',true,...
    'RowSpan',[1 1],'ColSpan',[1 1]);
  thisComp.dlgText(quoteStr,...
    'RowSpan',[2 2],'ColSpan',[1 1]);
}, 'Stock',...
  'LayoutGrid',[3 2],...
  'RowStretch',[0 0 1],...
  'ColStretch',[0 1]));
```

The Properties pane for the component, Equity Values, now looks as follows.



- 7 Click **File > Report** to generate the report. The following output appears in the report.

Table 1. Stock Market Pricing Data

Symbol	GOOG
Last	185.25
Date	15-Nov-2004
Time	15:20:00
Change	3.25
Open	180.45
High	188.32
Low	178.75
Volume	10651060

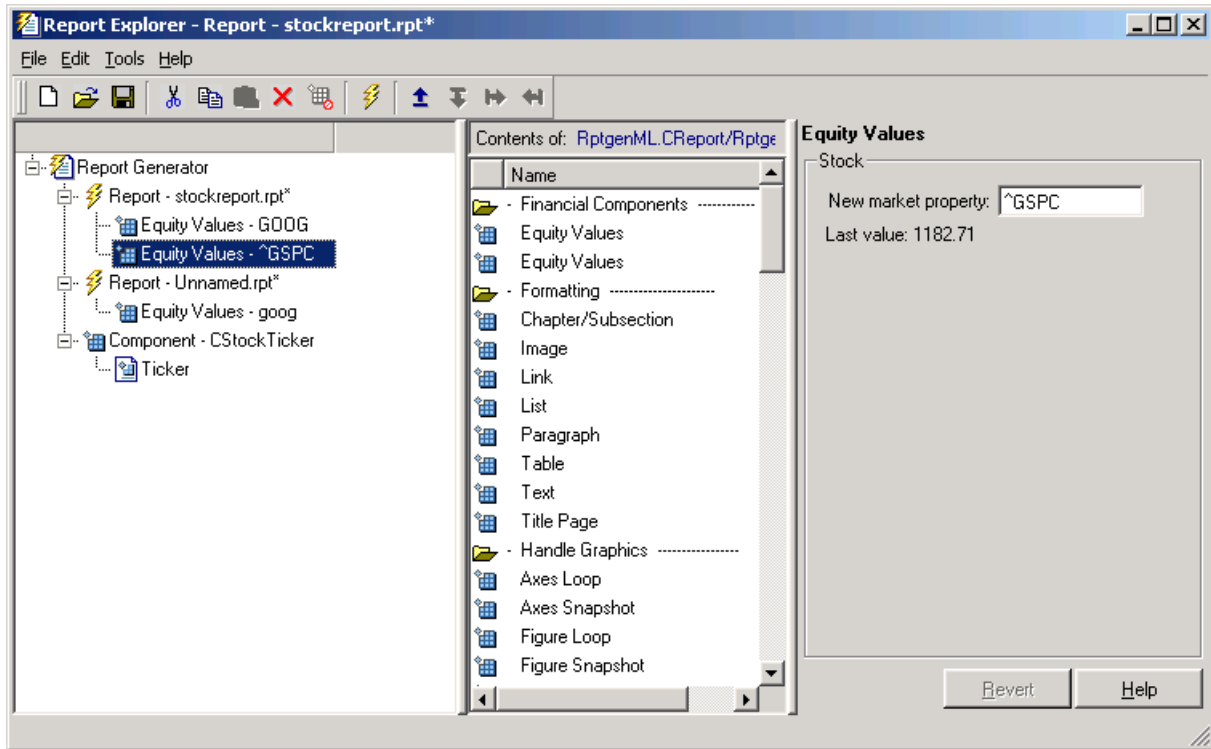
Displaying Securities Data in Two Tables

This example, which shows how to use multiple properties within a component, expands upon “Fetching Securities Data and Displaying It in a Table” on page 5-22.

- 1 Create a new report template and save it as `stockreport.rpt`. Add two Equity Values components to the template.



- 2 Edit the entry in the **New marker property** field to change the ticker property of the second component to '^GSPC' (S&P 500 index).



3 Run the report.

The report displays two tables of data, one for Google and another for the S&P 500 index.

Table 1. Stock Market Pricing Data

Symbol	GOOG
Last	185.25
Date	15-Nov-2004
Time	15:20:00
Change	3.25
Open	180.45
High	188.32
Low	178.75
Volume	10651060

Table 2. Stock Market Pricing Data

Symbol	^GSPC
Last	1183.13
Date	15-Nov-2004
Time	15:35:00
Change	-1.04
Open	1183.81
High	1184.48
Low	1179.85
Volume	1503245440

Creating Custom Stylesheets

About Stylesheets (p. 6-2)

Specify report output formats using default and customized stylesheets

Working with Stylesheets (p. 6-5)

How to create, edit, save, and delete custom stylesheets

Editing Stylesheet Data Items (p. 6-10)

How to customize styles by editing data items

Using Stylesheet Cells to Manage PDF Report Header and Footer Content (p. 6-21)

How to use stylesheet cells to manage headers and footers in reports

Examples (p. 6-27)

Examples of how to modify stylesheets to change the appearance and formats of generated reports

About Stylesheets

In this section...

“Built-in vs. Custom Stylesheets” on page 6-2

“Customizing Stylesheets Using Data Items” on page 6-3

Built-in vs. Custom Stylesheets

Stylesheets specify formatting and display settings for reports. MATLAB Report Generator uses stylesheets to convert reports from DocBook XML format to a format that you specify. If you want to generate the given report in a different format than initially specified, you can convert the XML document using a different or modified stylesheet.

The following table lists report output formats and their default stylesheets.

Report Format	Default Stylesheet
HTML	Uses stylesheets for either single- or multiple-page documents
PDF	Formatting Object (FO) stylesheet
RTF, Word	Document Style Semantics and Specification Language (DSSSL) stylesheet

The following table shows a list of properties for the built-in stylesheets.

Properties of Stylesheets

Name	Description
Description	A description of the stylesheet.

Properties of Stylesheets (Continued)

Name	Description
Display name	The stylesheet name that appears in the Options pane.
Transform type	<p>The process that MATLAB Report Generator uses to generate reports that use a specified stylesheet. Supported types are:</p> <ul style="list-style-type: none"> • HTML • FO (Formatting Object) for PDF reports • DSSSL (Document Style Semantics and Specification Language) for RTF and Word reports <hr/> <p>Note This field is not editable.</p> <hr/>

In most cases, the stylesheets provided with MATLAB Report Generator should be more than adequate for your needs. However, you may want to modify the built-in stylesheets to meet special requirements. For example, suppose one of the built-in stylesheets meets your requirements, but you want to change the page orientation. You can edit the built-in stylesheet to your specifications or create a new stylesheet.

Customizing Stylesheets Using Data Items

Each built-in stylesheet includes editable styles, also called *data items*, that are grouped together into categories. These data items specify styles that the file converter uses for a given report. You can edit these data items to customize stylesheets for your reports.

Data items can be of different types, some of which may require different editing methods. For more information about editing data items, see “Editing Stylesheet Data Items” on page 6-10.

Tip See the **Help** area at the bottom of the Properties pane on the right for a description of a specific data item that you are editing.

Working with Stylesheets

In this section...

“Using the Report Explorer to Edit Stylesheets” on page 6-5

“Creating New Stylesheets” on page 6-8

“Saving Stylesheets” on page 6-8

“Deleting Stylesheets” on page 6-9

Using the Report Explorer to Edit Stylesheets

To edit a stylesheet:

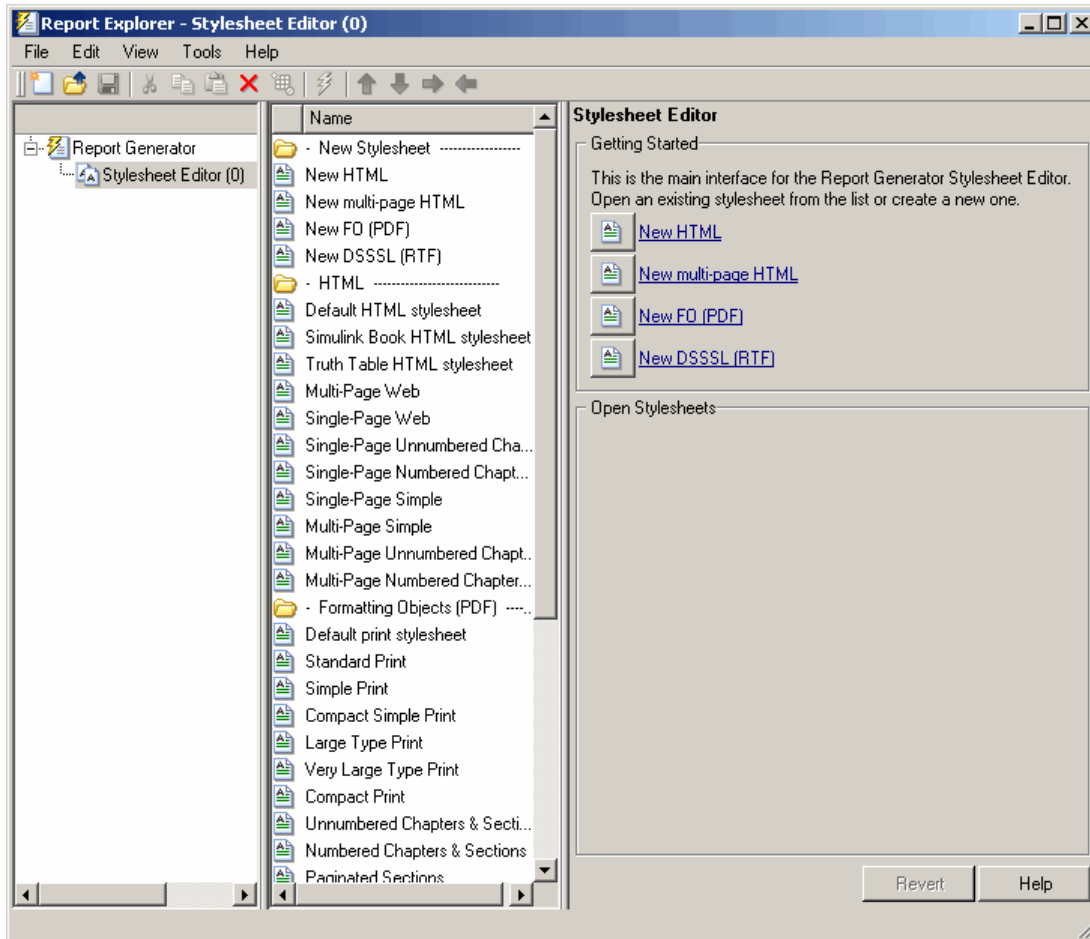
- 1 In Report Explorer, select a report template in the Outline pane on the left.
- 2 From the menu bar, click **Tools > Edit Stylesheet**.

The Report Explorer displays as follows.

- The Outline pane on the left displays the structure of stylesheets you create.
- The Options pane in the middle lists stylesheets available for customizing.

Tip Double-click a category to collapse it. Double-click it again to expand it.

- The Properties pane on the right shows properties of stylesheets, such as name and description.



You can use the Report Explorer to work with stylesheets as follows.

Task	Pane to Use	Action
Create a new stylesheet	Properties	Click the link that corresponds to the kind of stylesheet you want to create
Open an existing stylesheet	Properties	Click the name of the stylesheet, which should appear in the Open Stylesheets area
Select a stylesheet to use for converting an XML source file	Options	Select a stylesheet by clicking on it
View a list of customized styles in a stylesheet	Outline	Expand any open stylesheet
View a list of styles in a stylesheet	Outline or Options	Double-click the stylesheet
View a list of stylesheets available for editing in a given category	Options	Double-click the folder that corresponds to the kind of output you want (that is, HTML, PDF, RTF, or Word)
View open stylesheets	Outline	Expand the Stylesheet Editor item in Report Explorer
Change the name or description of the current stylesheet	Properties	Edit the text in the Display Name or Description field.
Convert an XML source file using the current stylesheet	Properties	Click Send to Source File Converter in the Properties pane.
Edit style data that is already customized	Properties	Click the style data item, which appears in the Stylesheet Customizations area
Open a style data item for editing or viewing	Options	Double-click the data item that you want to edit.
View a list of style data that is already customized	Outline	Expand the stylesheet

Creating New Stylesheets

To create a new stylesheet:

- 1 Open the Report Explorer.
- 2 From the menu bar, click **Tools > Edit Stylesheet**.
- 3 In the Properties pane on the right, choose the built-in stylesheet for the format with which you want to work. Options are:
 - **New HTML**. Creates a new stylesheet for HTML reports.
 - **New multi-page HTML**. Creates a new stylesheet for HTML reports with more than one page.
 - **New FO (PDF)**. Creates a new stylesheet for PDF reports.
 - **New DSSSL (RTF)**. Creates a new stylesheet for RTF reports.

The new stylesheet appears in the Outline pane on the left.

- 4 In the Properties pane on the right, modify the properties for the stylesheet as needed. Add data items to the new stylesheet:
 - a Drag the data item you want to add from the Options pane in the middle to the stylesheet in the Outline pane on the left.
 - b In the Properties pane on the right, edit the data items for the selected style. For more information on how to do this, see “Editing Stylesheet Data Items” on page 6-10
- 5 Save the stylesheet. For information about how to save a stylesheet, see “Saving Stylesheets” on page 6-8.

Saving Stylesheets

You must save a stylesheet before you can use it to convert a source file or associate it with a report. To use the Report Explorer to save a stylesheet:

- 1 Select the stylesheet that you want to save in the Outline pane on the left.
- 2 Select **File > Save As** from the menu bar and specify a new name for the stylesheet (to avoid overwriting built-in stylesheets). You must save the

file in a directory in your MATLAB path for the stylesheet to appear in the Report Explorer. The file name must be unique in the MATLAB path.

By convention, MATLAB Report Generator stylesheets have .rgs as their filename extension.

Deleting Stylesheets

To use the Report Explorer to delete a stylesheet that you created:

- 1 Select the stylesheet that you want to delete in the Outline pane on the left.
- 2 Click the stylesheet to delete from the Options pane in the middle.
- 3 Click **Delete stylesheet** in the stylesheet's Properties pane on the right.

You must restart MATLAB for deleted stylesheets to disappear from the Options pane.

Note You cannot delete built-in stylesheets.

Editing Stylesheet Data Items

In this section...
“Categories of Data Items in Built-in Stylesheets” on page 6-10
“Editing Data Items in Simple vs. Advanced Edit Mode” on page 6-14
“Working with Data Items” on page 6-15

Categories of Data Items in Built-in Stylesheets

You can edit data items in built-in stylesheets to customize them. Data items are grouped into *categories*, according to function. The following tables list the categories and data items for each type of stylesheet provided with MATLAB Report Generator.

Categories of Styles in PDF (FO) Stylesheets

Category	Description of Data Items in Category
Automatic labeling	Options for enumeration of parts of the report, such as chapters and sections
Callouts	Options and specifications related to callouts, such as defaults, use of graphics, size, path, fonts, characters, and extensions
Font Families	Specification of defaults for body text, copyright, quotes, symbols, dingbats, monospace, sans serif, and titles
Graphics	Specification of default width and options related to scaling attributes
Lists	Specification of spacing related to lists and list items
Meta/*Info	Options related to year ranges

Categories of Styles in PDF (FO) Stylesheets (Continued)

Category	Description of Data Items in Category
Miscellaneous	Options and specifications for placement of titles, comments, variable lists, block quotations, ulinks, hyphenations of URLs, verbatim environment display, use of SVG, table footnote numbers, superscript, and subscript
Pagination and General Styles	Specifications of page orientation, margins, double-sided, paper type, hyphenation, line height, columns, master font, draft mode, watermark, blank pages, rules for headers and footers, and content of headers and footers
Properties	Specification and options related to figure titles, monospace properties, verbatim text, section titles, and levels of sections
Stylesheet Extensions	Line numbering and table columns extensions
Table of Contents (TOC)/List of Tables (LOT)/Index Generation	Specifications for layout of TOC, depth of sections, indentation, and margins
Tables	Specifications for size of tables and their borders

Categories of Styles in HTML Stylesheets

Category of Style	Description of Data Items in Category
Automatic labeling	Options for enumeration of parts of the report, such as chapters and sections
Callouts	Options and specifications related to callouts, such as defaults, use of graphics, size, path, fonts, characters, and extensions
Chunking	Options related to using an explicit TOC for chunking, depth of section chunks, navigational graphics, and display of titles in headers and footers

Categories of Styles in HTML Stylesheets (Continued)

Category of Style	Description of Data Items in Category
Extensions	Line numbering, graphic size, and table columns extensions
Graphics	Specification of default width and depth, use of HTML embed for SVG, viewports, and options related to scaling attributes
HTML	Specifications related to dynamically served HTML, base and head elements, type of stylesheet, css, propagation of styles, longdesc, validation, cleanup, draft mode, watermark, and generation of abstract
Linking	Specification of Mailto URL and target for ulinks
Meta/*Info	Options related to year ranges
Miscellaneous	Options and specifications for comments, verbatim environment pixels, em space, use of SVG, and table footnote numbers
Table of Contents (TOC)/List of Tables (LOT)/Index Generation	Specifications for layout of TOC, depth of sections, indentation, and margins
Tables	Specifications for size of tables, table cell spacing and padding, and borders
XSLT Processing	Options related to header and footer navigation and rules

Categories of Styles in RTF (DSSSL) Stylesheets

Category of Style	Description of Data Items in Category
Admonitions	Options and path for admonition graphics
Backends	Options for Tex, MIF, and RTF back-end usage

Categories of Styles in RTF (DSSSL) Stylesheets (Continued)

Category of Style	Description of Data Items in Category
Bibliographies	Options related to checking citations; suppressing, enumerating, and using titles of entries
Fonts	Specifications for font family and size to use for some elements
Footnotes	Options for ulinks as footnotes and page location
Graphics	Specifications for file extensions, file names, and loading library database
Indents	Specifications for hanging indents, first paragraphs, and start of blocks
Labeling	Enumeration of sections and other elements
Miscellaneous	Options for floating formal objects, punctuation for run-in heads and honorifics, bold for first use of term, minimum leading between lines, and automatic hyphenation
OLinks	Using an extension for finding outline information
Object Rules	Specifications for placement and width of rules
Paper/Page Characteristics	Specifications for paper type, page numbers, width of pages, margins, and columns; heading-levels, sides; and writing mode (such as left-to-right)
Quadding	Specifications for justifying paragraphs
RefEntries and Functions	Options related to generation and display of reference entries and synopses for functions
Running heads	Options for generating and displaying running heads of chapters
Table of Contents (TOC)/List of Tables (LOT)	Options to produce or display TOC for sets, books, parts, references, articles. Options to display TOC on title page

Categories of Styles in RTF (DSSSL) Stylesheets (Continued)

Category of Style	Description of Data Items in Category
Tables	Specification of width in simple list
Titlepages	Options to produce and display title pages for sets, books, parts, references, articles. Options for author's name and ordering elements
VariableLists	Options and specifications for term length and formatting
Verbatim Environment	Specifications for width, enumeration, size, indentation, line frequency, and callouts
Vertical Spacing	Specifications for space between lines and paragraphs

Editing Data Items in Simple vs. Advanced Edit Mode

- To edit a data item in *simple edit mode* means that you edit a simple string that corresponds to the data in the stylesheet. This string appears in the field to the right of the **Value** label. For some values, use a selection list to change the value instead of typing in text.
- To edit a data item in *advanced edit mode* means that you directly edit the XML code.

Note This section gives instructions for simple edit mode, except where explicitly specified otherwise.

The user interface is in simple edit mode when the data item appears in a pane labeled **Value**. It is in advanced edit mode when the data item appears in a pane labeled **Value (XML)**. To switch from simple to advanced edit mode, click **Edit as XML**.

Edit values for most data items in PDF and HTML stylesheets in either simple edit mode or advanced edit mode. Edit values for RTF stylesheets

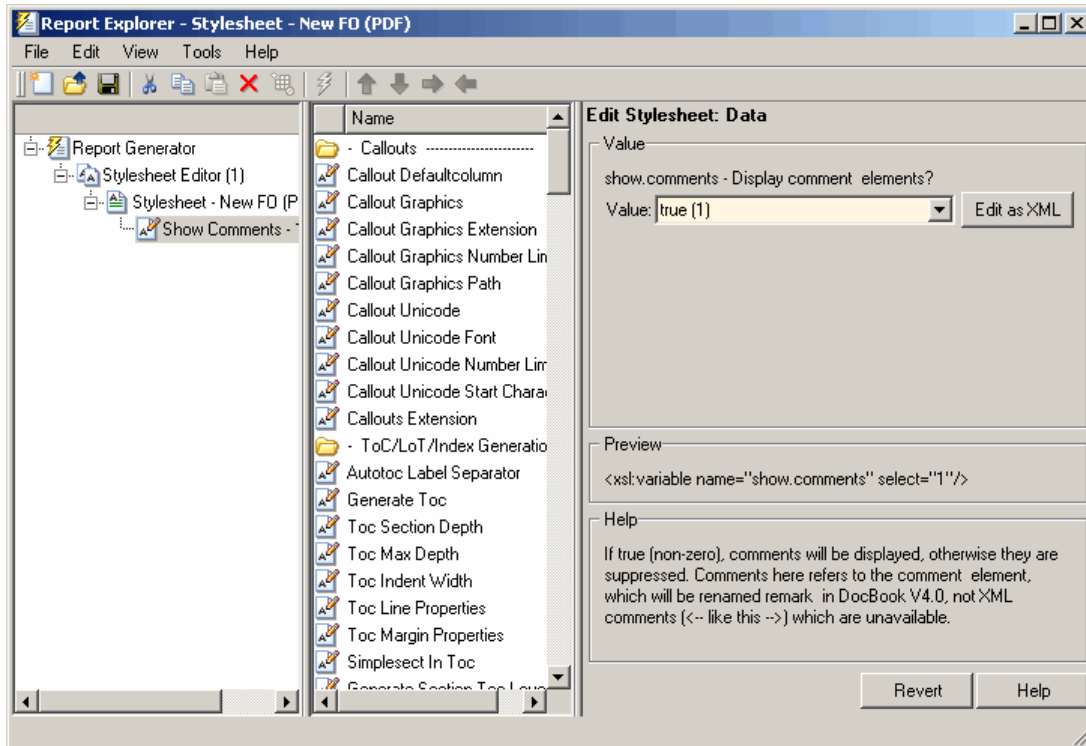
in simple edit mode only. Data items in RTF stylesheets do not support advanced edit mode.

Note To modify content for headers and footers you edit *stylesheet cells*, which do not appear in either simple or advanced mode. For more information, see “Using Stylesheet Cells to Manage PDF Report Header and Footer Content” on page 6-21.

Working with Data Items

Select a stylesheet from the Options pane in the middle of the Report Explorer. The Outline pane on the left shows the name of the current style data item inside its stylesheet. The Options pane in the middle shows a list of available stylesheet data items. The Properties pane on the right displays **Stylesheet Editor: Data**. It also includes the following information:

- The value of the data item is in a pane labeled **Value** in simple edit mode or **Value (XML)** in advanced edit mode.
- To the right of the value is the **Edit as XML** toggle button.
- The **Preview** pane includes a partial view of the stylesheet that specifies the data item. The data in this pane is not editable.
- The **Help** pane contains information about the data item. This information is not editable.



Editing Boolean and Enumerated Values

In the example below, the value of the Show Comments data item is of type Boolean and the current value is specified as `true (1)`. To change this value, use the menu list for the value field and select a different value; in this case, the only other possible value is `false (2)`.

Editing Strings

For the values of some data items, the Report Explorer displays text in the **Value** field. This text is not an XML expression; edit it in the **Value** field. The text does not have to be specified as an XML expression, but you have the option of typing XML characters.

Editing XML Expressions

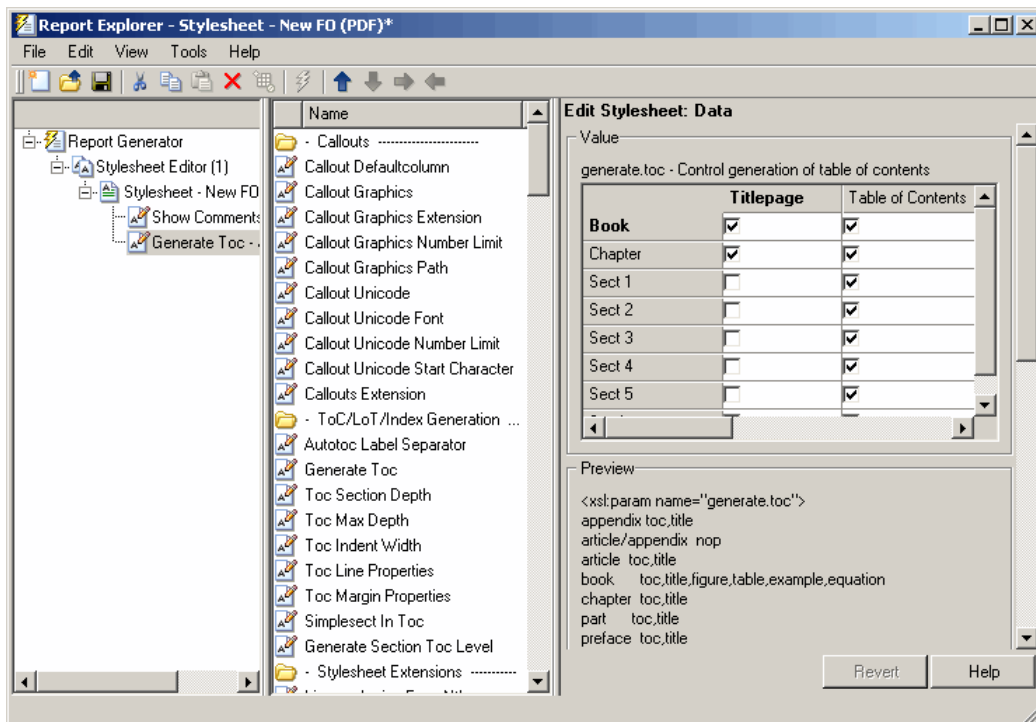
To make complex changes to a stylesheet, consider using Advanced edit mode; that is, edit XML expressions directly in the **Value (XML)** pane. If this pane is not displayed, click **Edit as XML** to switch to advanced edit mode.

Make sure that you enter valid XML. Invalid XML values generate an error, which appears at the top of the Properties pane.

Modifying TOC Properties

To change values for generation of the report's table of contents (TOC), select the appropriate values from a matrix of check boxes.

The figure below shows the values for the **Generate Toc** data item on the **PDF** stylesheet. Select the check boxes to control the values that appear in the report's title page and table of contents.



Modifying Title Placement Properties

The **Title Placement** data items, which are in the **Miscellaneous** category, control the position of titles for figures and tables.

Selecting one of these data items for editing causes the Properties pane on the right to display possible values in a menu list. Specify whether you want the title to appear before or after a given figure or table.

Modifying Attributes

An *attribute* is a data item that specifies information for an XML element. An attribute must be a child of an *attribute set*. For more information, see “Editing Attribute Sets” on page 6-18.

Note The information in the **Help** area of the Properties pane of an attribute describes the set to which the attribute belongs.

Editing Attribute Sets

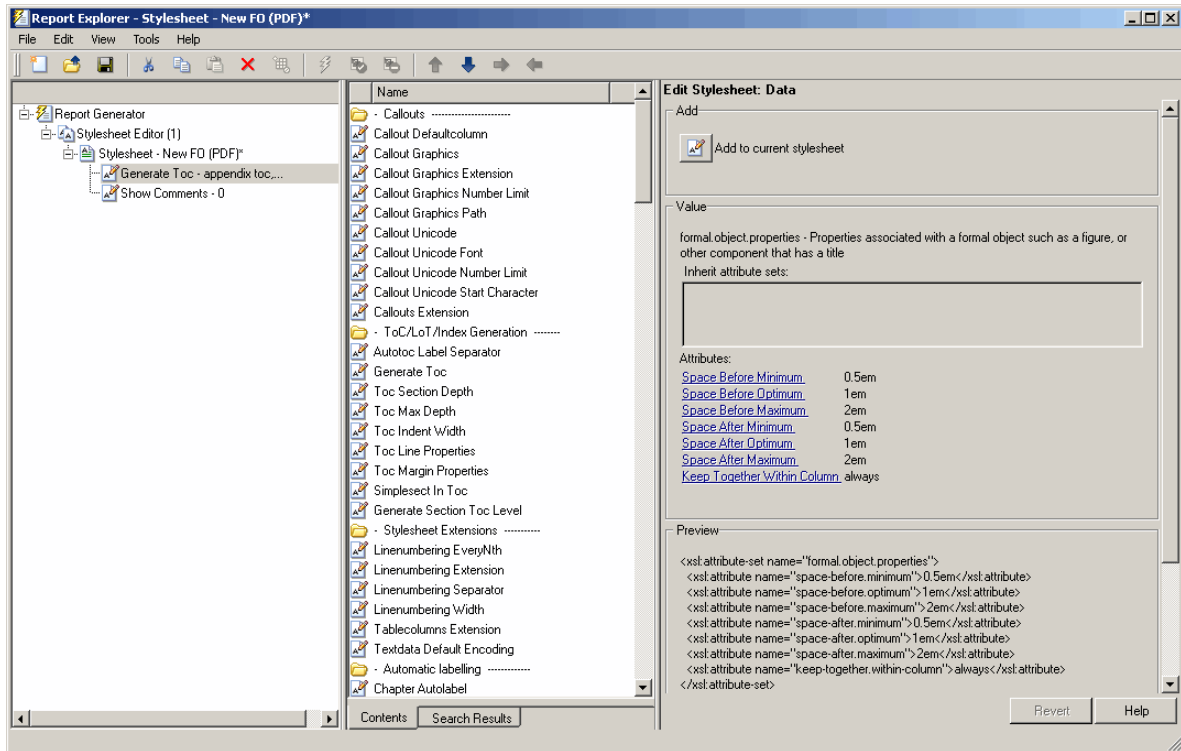
An *attribute set* consists of a group of attributes. Selecting an attribute set in the Outline pane on the left causes the Properties pane to list the attributes that belong to that set.

To edit a specific attribute, expand the attribute set in the Outline pane and select the attribute you want to edit.

To edit the attribute set, type text in the **Inherit attribute sets** area of the Properties pane.

An example of an attribute set is **Formal Object Properties**, a data item in the **Property Sets** category of the default print stylesheet for PDF documents.

Here is an example of the Report Explorer showing the **Formal Object Properties** attribute set in the Properties pane.



Editing Varpair Values

Data items in RTF stylesheets are represented as varpair data items, which are name/value pairs of information. RTF stylesheets are the only type of stylesheet that includes varpair data items.

Edit varpair data items as strings or as Boolean values. Boolean values are represented as true (#t) and false (#f).

Note You cannot edit RTF stylesheet data items as XML.

Note Because varpair data is sometimes represented in stylesheets as DSSSL rather than XML, the code that appears in the **Preview** pane of the Properties pane on the right looks different from the code associated with other kinds of stylesheets supported by MATLAB Report Generator.

Deleting Data Items

To delete a customized data item:

- 1 Right-click the data item in the Outline pane on the left.
- 2 Select **Delete**.

Using Stylesheet Cells to Manage PDF Report Header and Footer Content

In this section...

“About Stylesheet Cells and Cell Groups” on page 6-21

“Working with Headers and Footers” on page 6-22

“Using Templates to Add Content to Headers and Footers” on page 6-24

“Inserting Graphics Files” on page 6-25

“Modifying Fonts and Other Properties” on page 6-26

About Stylesheet Cells and Cell Groups

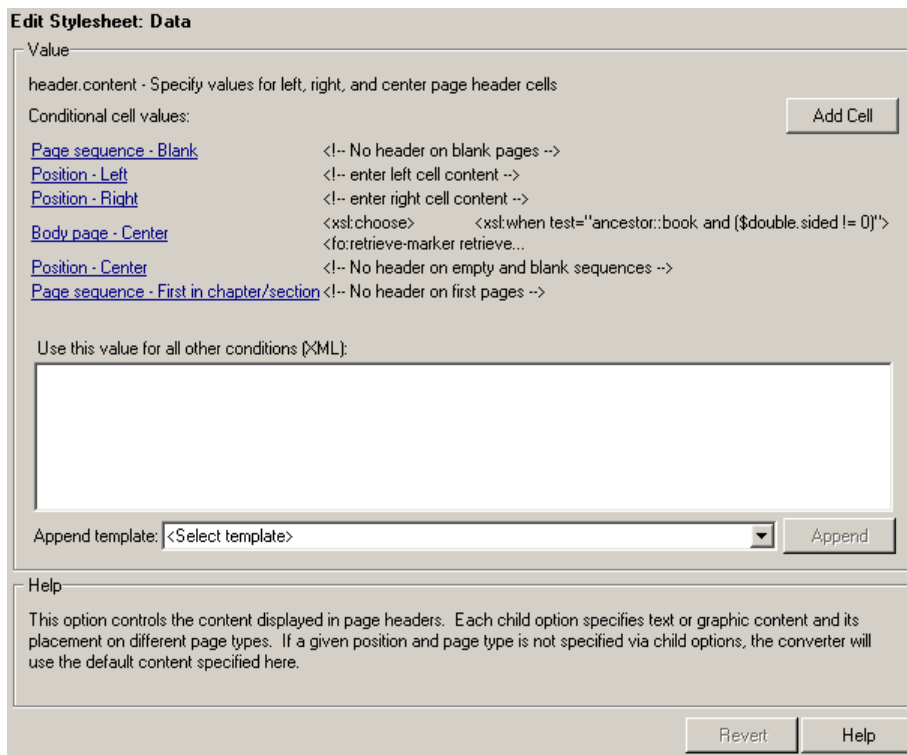
Use *stylesheet cells* to specify content of headers and footers in PDF reports.

MATLAB Report Generator defines a page as 6 *cells*. These cells correspond to the left, right, and center of the page’s header and the left, right, and center of the page’s footer.

A *cell group* consists of one or more stylesheet cells. Two cell groups are available for PDF reports: **Header Content** and **Footer Content**.

The Properties pane for each cell in a cell group lists the current stylesheet cell definitions for the group in a two-column list of **Conditional cell values**. The first column displays the name of a *condition*, and the second column displays the *content* and default value to be inserted into a report if the specified condition is met.

For example, the stylesheet cell `Page sequence - Blank` specifies the content for a blank page; by default the content is set to empty. Similarly, `Cell - Right Side` specifies the content for the right side of the header on every page.



There are many combinations of conditions and values that you can use to customize content of headers and footers. MATLAB Report Generator provides several predefined conditions that are frequently used. These predefined cells are listed in the Properties panes for the Header Content and for Footer Content cell groups.

Working with Headers and Footers

Adding Content that Satisfies Specified Conditions

You can use the Properties pane of a stylesheet cell to specify content that satisfies specified conditions. The Properties pane for a stylesheet cell includes the following.

Label	Definition	Description
Condition	Condition that must be satisfied for content to be included	This is a selection list of frequently used and predefined conditions. Select a condition and click Edit to view or change a condition's XML code
Value (XML)	Content to be included if the condition is met	Modify or create XML code for header or footer content
Append Template	Name of the template you use to add content	Templates containing XML code that you can use to add content. For more information, see "Using Templates to Add Content to Headers and Footers" on page 6-24.

When the File Converter processes a page, it evaluates settings that are relevant to each of the six cells on the page and adds content accordingly. If there are no conditions in effect for a given cell, the File Converter uses the default values for the cell group.

Possible conditions and their values as coded in XML are shown in the following table.

Name of Condition	Possible Values for the Condition	Sample XML Code
\$position	right center left	\$position='right' \$position='center' \$position='left'
\$sequence	odd even first blank	\$sequence=odd \$sequence=even \$sequence=first \$sequence=blank
\$double-sided	0 1	\$double-sided=0 \$double-sided=1
\$pageclass	\$titlepage \$lot \$body	\$pageclass=\$titlepage \$pageclass=\$lot \$pageclass=\$body

Use standard logical operators (such as = , != , and, or) and nested expressions (characters between parentheses are an expression within an expression) to specify *complex conditions*. You can use complex conditions to set the position of headers and footers on pages and to specify other settings, such as whether the content should appear on a title page.

Using Templates to Add Content to Headers and Footers

Templates are available for adding the following items to headers and footers:

- Text
- Author names
- Page numbers
- Titles for chapters and sections

- Chapter numbering
- Draft information
- Comments
- Graphics

Templates used by the File Converter are Extensible Style Language Transformations (XSLT), which is a language for transforming XML documents into other XML documents. For details about XSLT, see the Web site for the World Wide Web Consortium (W3C).

To use a template to specify content for a header or footer:

- 1** In the **Append template** list, select the type of content you want to add.
- 2** Click **Append**.

The Properties pane on the right displays default content for the type you select. Edit the XML code to change the default content.

For example, to specify text as the content:

- 1** Select **Text** from the **Append template** list.
- 2** Click **Append**.
- 3** The default value for `xsl:text` is `Confidential`. Edit the value as needed.

Inserting Graphics Files

To add graphics files to headers or footers, you must specify the name of the file in the stylesheet cell that belongs to **Header Content** or **Footer Content**, as well as other pagination properties to allow room for the graphic to appear properly. Select and edit the values of the `Region Before Extent` and `Region After Extent` data items in the **Pagination and General Styles** folder of the **Options** pane for PDF formatting.

For an example of adding a graphic file to a header, see “Adding Graphics to Headers in PDF Reports” on page 6-28.

Note PDF reports support only bitmap images (.bmp files) in headers and footers.

Modifying Fonts and Other Properties

You cannot use stylesheet cells to modify the font family or other such properties of headers and footers. To specify the style of the content in headers and footers, use the **Header Content Properties** and **Footer Content Properties** attribute sets.

Each of these attribute sets is a pagination style data item for PDF stylesheets. Modify an attribute in the set, such as the **Font Family** attribute in **Header Content Properties**, by selecting the attribute in the Outline pane on the left.

For an example of modifying font size and other properties of a PDF report, see “Changing Font Size, Page Orientation, and Paper Type of a Generated Report” on page 6-33.

Examples

In this section...
“Numbering Pages in a Report” on page 6-27
“Adding Graphics to Headers in PDF Reports” on page 6-28
“Changing Font Size, Page Orientation, and Paper Type of a Generated Report” on page 6-33
“Editing Font Size as a Derived Value in XML” on page 6-36

Numbering Pages in a Report

This example shows how to edit a stylesheet cell to number the top right-hand side of all pages in the generated report. To do this, define a basic stylesheet cell in the Header Content cell group with a condition of `right`, and set the content to the current page number:

- 1 Open a PDF stylesheet in the Report Explorer.
- 2 Double-click **Header Content** in the Options pane in the middle.
- 3 Click **Position - Right** in the Properties pane on the right.

For `Position - Right`, the XML code for the condition and content appears as follows.

Edit Stylesheet: Cell

Value

If this condition is true, use this value for the current header/footer cell location.

Condition:

Value (XML):

Append template:

Help

This option specifies page header/footer content and placement. The "Value (XML)" field is XML code which specifies the text or graphics to appear in the header or footer. The "Condition" field controls where and on what type of pages the content is used.

The code for content is empty. Select a template from the **Append template** list and click **Append** to add the content that you need for your particular report.

Adding Graphics to Headers in PDF Reports

This example shows how to edit default header content for a PDF stylesheet to put an image in the center of each header in a PDF report and exclude the image only from the title page and the first page in each chapter. This example uses the report template `mfile-report.rpt`.

You can use any bitmap or jpeg file as image content. You must know the size of the image so that you can allow enough room for it in the header. The image used here is `sample_logo.bmp`, which is shown below.



Note PDF reports support only bitmap (.bmp) and jpeg (.jpg) images.

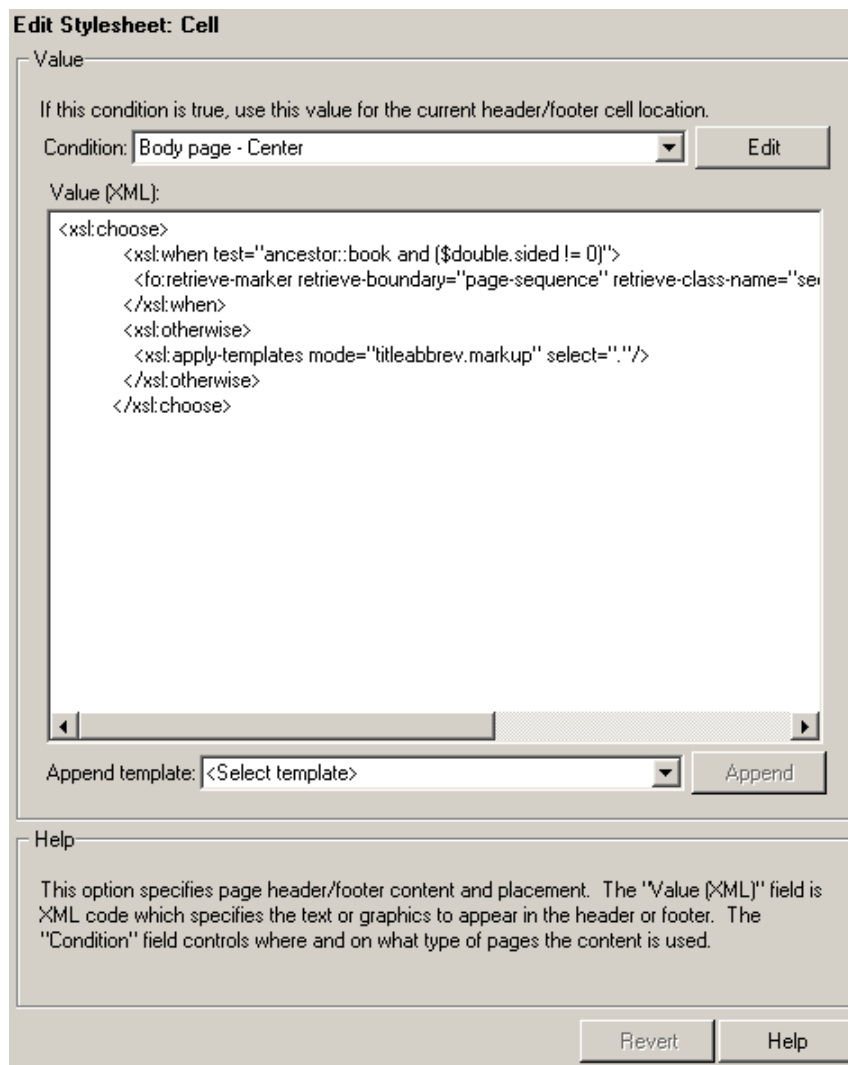
To include this image file in the center of each header in the body of a PDF report:

- 1 Open `mfile-report.rpt` by entering the following at the MATLAB command prompt:

```
setedit mfile-report
```

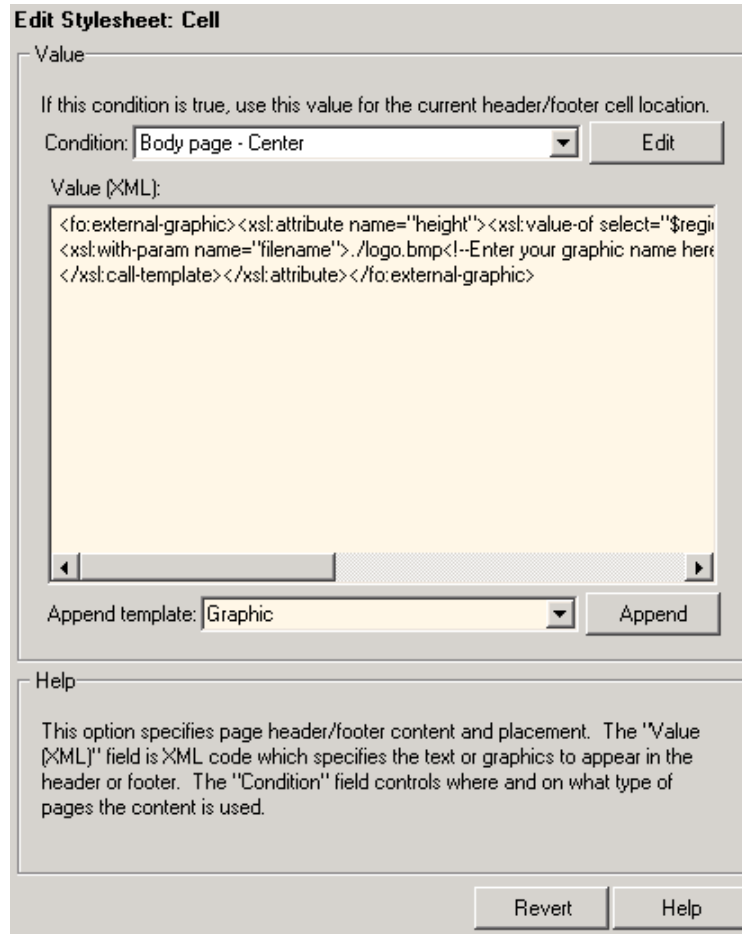
- 2 Create a new stylesheet.
 - a Select **Tools > Edit Stylesheet** in the menu bar of the Report Explorer.
 - b Click **New FO (PDF)** in the Properties pane on the right.
 - c As the **Display name**, enter `Logo stylesheet for PDF`.
 - d As **Description**, enter `Company logo in center of header`.
 - e Save the stylesheet as `logo_stylesheet.rgs` in a directory on your MATLAB path.
- 3 Open the cell group for editing.
 - a Scroll through the Options pane on the left to the **Pagination and General Styles** folder.
 - b Double-click **Header Content** in the Options pane.
 - c Click **Body – page Center** from the list of cells in the Properties pane on the right.

The Properties pane appears as shown.

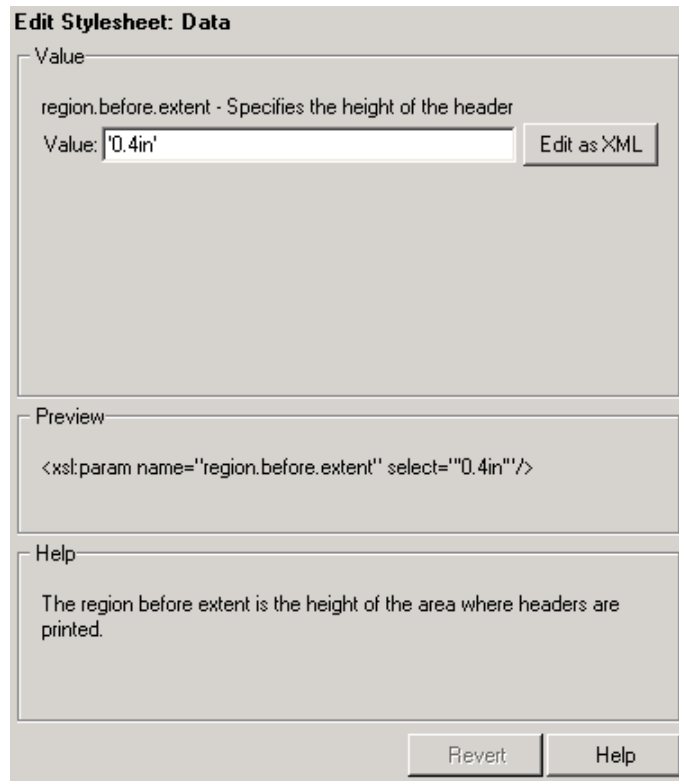


- d** Delete the text in the **Value (XML)** field.
- e** Select **Graphic** from the **Append template** selection list and click **Append**.

The Properties pane on the right shows the XML code that tells the File Converter to include the graphic.



- 4 By default, the name of the graphic is `logo.bmp`. Change all instances of this name to `sample_logo.bmp` in the **Value (XML)** field.
- 5 Save the stylesheet.
- 6 Make sure that the amount of room available in the header is large enough to accommodate the image file.
 - In the Options pane in the middle, double-click **Region Before Extent**, which is in the **Pagination and General Styles** folder.



- b** By default the value for the height of the header is 0.4 inches. Replace this value with 1.0in.
 - c** Save the stylesheet.
- 7** Generate the report with the new styles.
- a** Select **mfile-report.rpt** in the Outline pane on the left.
 - b** In the selection lists under the **Report Format and Stylesheet** area of the Properties pane on the right, specify Acrobat (PDF) for **File format** and Logo stylesheet for PDF.
 - c** Click **Report** on the toolbar to generate the report.

Changing Font Size, Page Orientation, and Paper Type of a Generated Report

This example shows how to:

- Generate an XML source file without converting it to a supported report format
- Make section headers in a report larger
- Change the report page orientation to landscape
- Change the report paper type to A4

Create a new stylesheet by editing an existing stylesheet to change the appearance of the `wsvar-report` report, which is provided with MATLAB Report Generator.

1 Generate a source file for the report.

- a** Open the report by entering the following command in the MATLAB Command Window:

```
setedit wsvar-report
```

- b** In the **Report Format and Stylesheet** area of the Properties pane, change the format to **DocBook (no transform)**.
- c** Check the **If report already exists, increment to prevent overwriting** check box.
- d** Select **File > Report** to generate the report.

MATLAB Report Generator creates an XML source file in the MATLAB editor.

2 Convert the report to PDF format.

- a** Select **Tools > Convert Source File** from the Report Explorer menu bar to open the File Converter.
- b** From the **Source file** selection list, enter **wsvar-report0.xml**.
- c** From the **File format** selection list, select **Acrobat (PDF)**.

d From the **Stylesheet** selection list, select **Unnumbered Chapters and Sections**.

e Click **Convert File**.

MATLAB Report Generator converts the XML source file for `wsvar-report` to PDF format, and then opens the PDF document.

3 Make the report headers more prominent.

a In the File Converter, click **Edit**.

The Report Explorer displays the **Unnumbered Chapters and Sections** stylesheet.

b In the Properties pane on the right, enter Custom Large Section Headers as the stylesheet name.

c Enter the description No chapter and section numbering, larger section titles.

d In the Outline pane on the left, select the **Custom Large Section Headers** stylesheet.

e In the Options pane in the middle, select **Section Title Level 1 Properties**.

f In the Properties pane on the right, click **Add to current stylesheet**.

The **Section Title Level 1 Properties** data item appears in the Outline pane on the left as a child of the Custom Large Section Headers stylesheet.

g In the Properties pane on the right, select the **Font Size** attribute.

The Properties pane on the right displays an XML expression specifying font size as a multiple of the Body Font Size attribute.

h Click **Edit as string**.

MATLAB Report Generator converts the XML expression to a simple string, which appears in a pane labeled **Value**.

i Enter the value 18pt.

The size of the font is now fixed at 18 points, rather than being a multiple of the body font size attribute.

- j** Select **File > Save** to save the stylesheet.
- k** Save the stylesheet as `customheader.rgs`, in a directory in your MATLAB path.

The `customheader.rgs` stylesheet appears as an available stylesheet in the Options pane in the middle of the Report Explorer. It also appears as an option in the File Converter.

- 4** Use the new stylesheet to convert the current XML source file.
 - a** In the **Stylesheet Editor: Main** Properties pane on the right, click **Send to File Converter**

The File Converter appears, with the `customheader.rgs` stylesheet selected.

- b** Click **Convert file**.

- 5** Change page orientation and paper type.
 - a** On the File Converter Properties pane, click **Edit**.
 - b** In the Options pane on the left, double-click the **Page Orientation** data item.
 - c** In the Properties pane on the right, use the selection list to change the value of the data item to Landscape.
 - d** In the Options pane in the middle, double-click **Paper Type** in the **Pagination and General Styles** folder.
 - e** In the Properties pane on the right, select **A4** from the selection list.
 - f** Save the stylesheet.

- 6** Generate the report `wsvar-report.xml` in PDF format using `customheader.rgs`.

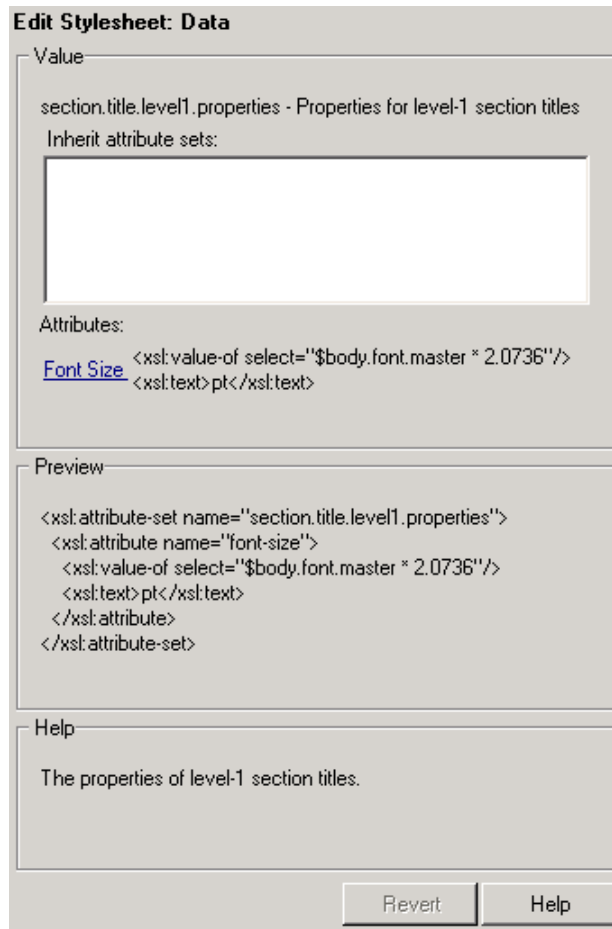
The PDF report appears with horizontally oriented pages of slightly different dimensions.

Editing Font Size as a Derived Value in XML

This example shows how to directly edit a PDF report's XML source to change the report's font size to a value derived from other values.

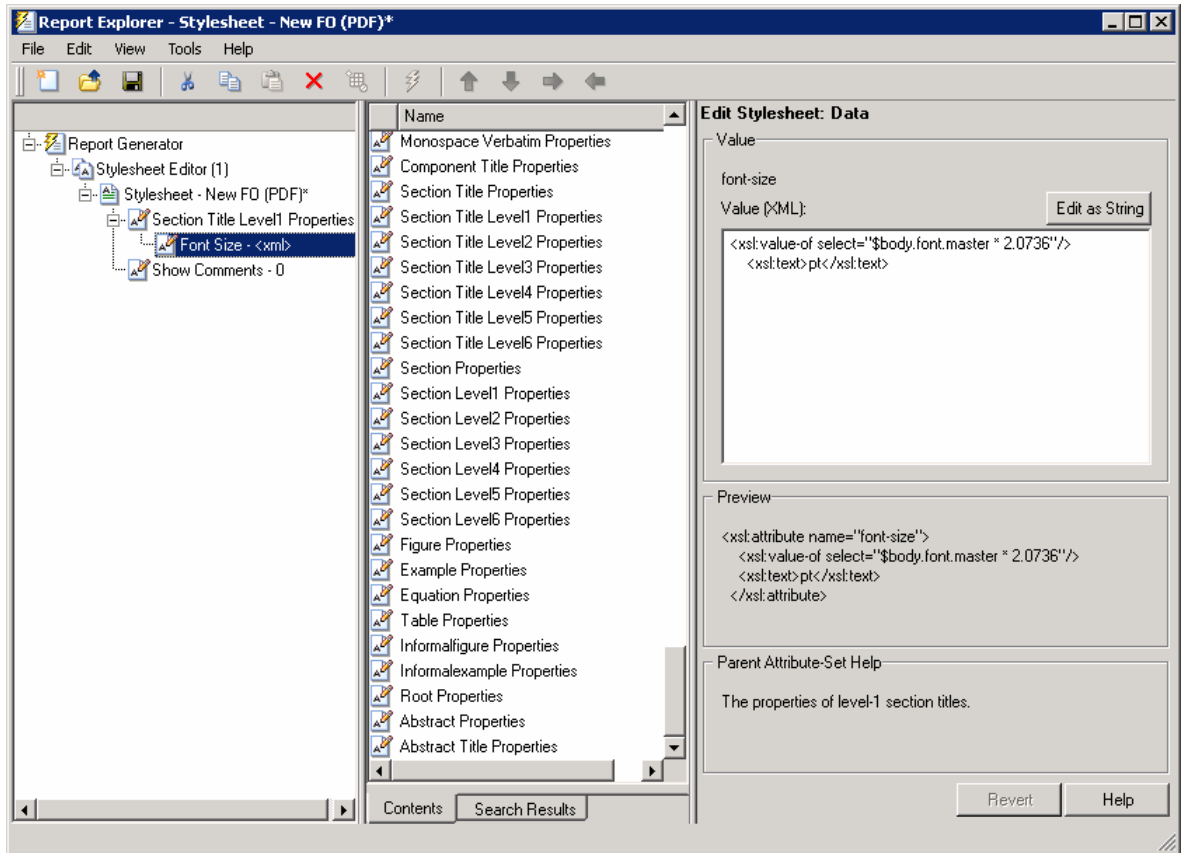
- 1** Open the default print stylesheet for PDF documents.
- 2** In the Options pane in the middle, select and expand the **Property Sets** folder.
- 3** In the Options pane, double-click the **Section Title Level1 Properties** data item.

The Properties pane on the right appear as follows.



- 4 In the **Attributes** area of the Properties pane on the right, click **Font Size - <xml>**.

The Report Explorer looks as follows.



The font size value is specified as a product of `$body.font.master` and 2.0736. To change the font size to a larger size, change the multiplication factor to 3.0736.

Tip The value for the `$body.font.master` data item is specified in the **Body Font Master** property, which is in the **Pagination and General Styles** category in the Options pane in the middle. The default value of this data item is 10. Changing this value causes the derived values to change accordingly.

MATLAB Components — By Category

Formatting (p. 7-2)

Handle Graphics (p. 7-4)

Logical and Flow (p. 7-6)

MATLAB (p. 7-7)

MATLAB Report Generator (p. 7-8)

Insert documentation elements

Report on Handle Graphics figures

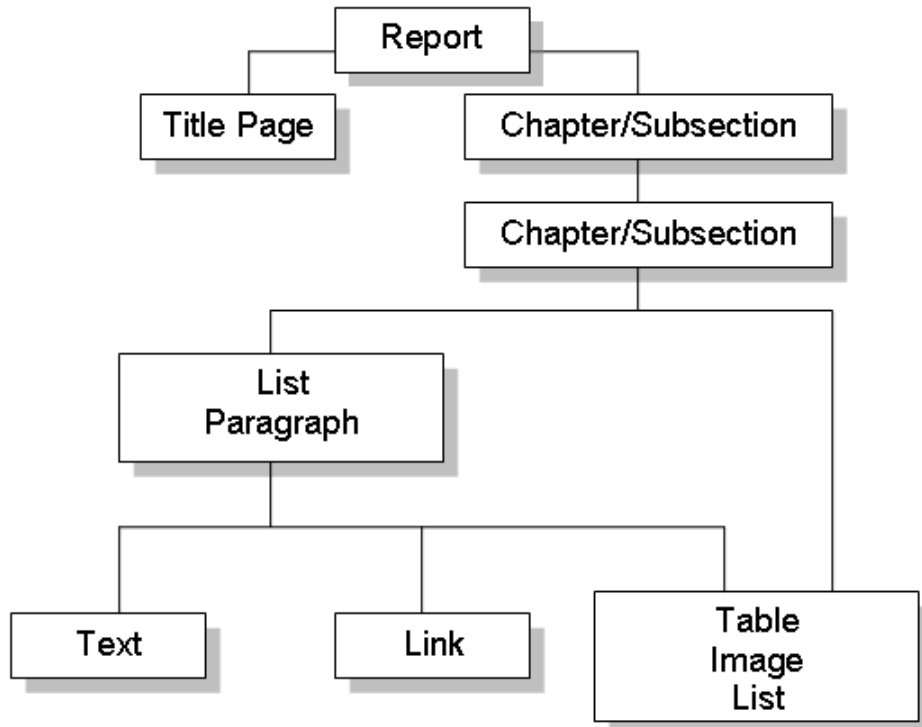
Specify conditions that must be met
for child components to execute

Interact with MATLAB workspace

General-purpose components

Formatting

Use Formatting components to insert documentation elements into your report. Formatting components must have the following parent/child relationships.



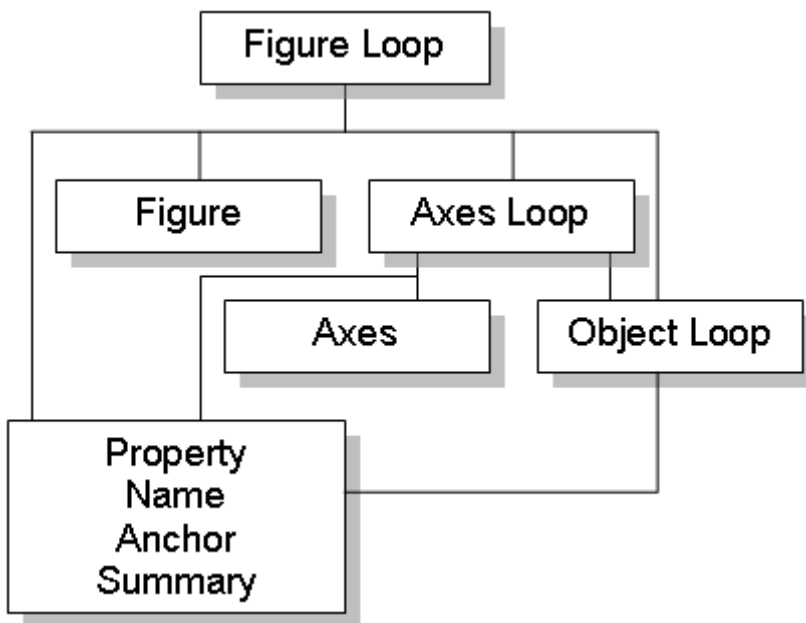
The following table describes the Formatting components.

Chapter/Subsection	Groups portions of the report into sections with titles
Empty Component	Groups components so that the components can be easily moved, activated, or deactivated, or creates a blank space in a list

Image	Inserts an image from an external file into the report
Link	Inserts linking anchors or pointers into the report
List	Creates a bulleted or numbered list from a cell array or from child components
Paragraph	Inserts paragraph text into the report
Table	Converts a rectangular cell array into a table and inserts the table into the report
Text	Formats and inserts text into the report
Title Page	Inserts a title page at the beginning of the report

Handle Graphics

Handle Graphics components report on Handle Graphics figures. The following figure shows the Handle Graphics component hierarchy.



The following table describes the Handle Graphics components.

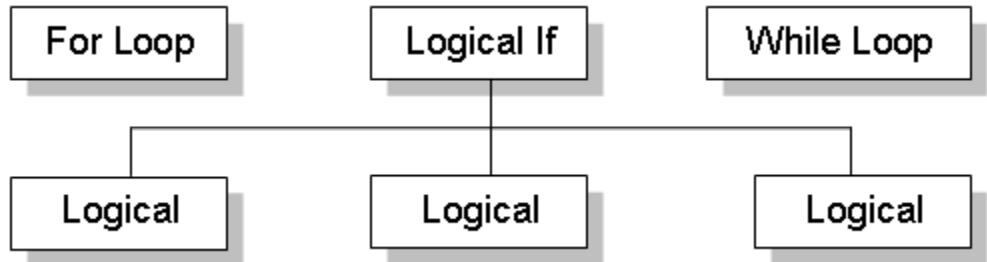
Axes Loop	Runs its child components for all axes objects in the MATLAB workspace
Axes Snapshot	Inserts an image of the selected axes objects in MATLAB into the generated report
Figure Loop	Applies each child component to specified graphics figures

Figure Snapshot	Inserts a snapshot of a Handle Graphics figure into the report
Graphics Object Loop	Runs its child components for each Handle Graphics object that is currently open in the MATLAB workspace
Handle Graphics Linking Anchor	Designates a location to which links point
Handle Graphics Name	Inserts the name of a Handle Graphics object into the report
Handle Graphics Parameter	Inserts a property name/property value pair from a Handle Graphics figure, axes, or other object
Handle Graphics Property Table	Inserts a table that reports on property name/property value pairs
Handle Graphics Summary Table	Inserts a table that summarizes Handle Graphics object properties

Logical and Flow

Logical and Flow Control components execute conditionally, enabling you to decide when a child component executes or how many times a child component executes.

These components have the following parent/child relationships.



For Loop

Iteratively executes its child components

Logical Else

Acts as an else when it is the child of the Logical If component

Logical Elseif

Acts as an elseif when it is the child of the Logical If component

Logical If

Acts as a logical if

Logical Then

Acts as a then when it is the child of the Logical If component

While Loop

Iteratively executes its child components while a specified condition is true

MATLAB

Use MATLAB components to interact with the MATLAB workspace. The following table describes these components.

Evaluate MATLAB Expression	Evaluates a specified MATLAB expression in the workspace
Insert Variable	Inserts variable values into the report
MATLAB Property Table	Inserts a table that reports on MATLAB object property name/property value pairs
MATLAB/Toolbox Version Number	Inserts a table that shows the version number, release number, and/or release date of MathWorks software
Variable Table	Inserts a table that displays all variables in the MATLAB workspace

MATLAB Report Generator

Report Generator components are “general utility” components. The following table describes these components.

Comment	Inserts a comment into the XML source file created by the report generation process
Import File	Imports an ASCII text file into the report
Nest Setup File	Allows one report template (.rpt file) to run inside another
Stop Report Generation	Halts report generation
Time/Date Stamp	Inserts the time and date of the report generation into your report

MATLAB Components — Alphabetical List

Axes Loop

Purpose Runs its child components for all axes objects in the MATLAB workspace

Description The Axes Loop component runs its child components for all axes objects in the MATLAB workspace. For details about working with looping components, see “Working with Looping Components” on page 4-3.

Object Selection

Loop type

To loop on all axes objects, select `All axes`. To loop on the currently selected axes object, select `Current axes`.

Exclude objects which subclass axes

Exclude objects such as legends and color bars.

Loop Menu

To loop only on visible axes objects, select `Loop on axes with handle visibility "on"`. To loop on all axes objects, select `Loop on all axes`.

Search terms

Enter search terms for the loop. For example, to search for `Tag` and `My Data`, enter `"Tag"`, `"My Data"`.

Section Options

Create section for each object in loop

Insert a section in the generated report for each object found in the loop.

Display the object type in the section title

Automatically insert the object type into the section title in the generated report.

Create link anchor for each object in loop

Create a hyperlink to the object in the generated report.

Insert Anything into Report?

Yes, inserts a section if the **Create section for each object in loop** option is selected.

File Name rptgen_hg.chg_ax_loop

Axes Snapshot

Purpose Inserts an image of the selected axes objects in MATLAB into the generated report

Description Inserts an image of the selected axes objects in MATLAB into the generated report

Format **Image file format**

Choose the image file format (e.g., JPEG, TIFF, etc.) from this list. Select Automatic HG Format to automatically choose 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 HG Format is the default option. Other options include:

- Automatic HG Format (uses the Handle Graphics file format selected in the Preferences dialog box)
- Adobe Illustrator
- 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

Capture figure from screen

Capture the figure for the generated report directly from the screen. Capture part of the figure by choosing the `Client area only` option, or capture the entire figure window by choosing `Entire figure window`.

Print Options

- **Paper orientation**

Choose `Landscape`, `Portrait`, or `Rotated`, or `Use figure orientation`. For information about paper orientation, see `orient` command in the MATLAB documentation.

- **Image size**

Set the image size in the report by selecting `Use figure PaperPositionMode` setting to set the `PaperPositionMode` property of the `Handle Graphics` figure. For information about paper position mode, see `orient` in the MATLAB documentation.

For the image in the report to be the same size as the image on the screen, select `Automatic (same size as onscreen)`.

To choose a custom image size, select `Custom`. Specify the image size in the **Size** field and **Units** list.

- **Size**

Axes Snapshot

Enter the size of the Handle Graphics figure snapshot in the form [w h] (width height). This field is active only if you choose Custom in the **Image size** list box.

- **Units**

Enter the 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**

Handle Graphics figures have an InvertHardcopy property. This parameter inverts colors for printing; i.e., it changes dark colors to light colors and vice versa.

Select one of the following options:

- Automatic. Select this option to automatically change dark axes colors to a light axes color. If the axes color is a light color, it is not inverted.
- Invert. Select this option to change dark axes colors to light axes colors and vice versa.
- Don't invert. Select this option if you do not want to change the colors in the image displayed on the screen for printing.
- Use figure's InvertHardcopy setting. Select this option to use the InvertHardcopy property set in the Handle Graphics image.
- Make figure background transparent. Select this option to make the image background transparent.

Display Options

- **Scaling**

To specify the number and type of units, select Fixed size

To specify the percentage, the maximum size, and the units of measure, select Zoom.

For the image in your report to be the same size as the image on the screen, select `Use image size`.

- **Size**

This field is active only if you choose `Fixed size` in the **Scaling** list. Enter the size of the snapshot in the form `w h` (width height) and press **Enter**.

- **Max size**

This field is active only if you choose `Zoom` from the **Scaling** list. Enter the maximum size of the snapshot in the form `w h` (width height) and press **Enter**.

- **Units**

This field is active only if you choose `Zoom` or `Fixed size` in the **Image size** list box. Enter the units for the size of the snapshot.

- **Alignment**

Options are `Auto`, `Right`, `Left`, and `Center`.

Note This option is supported only in PDF and RTF reports.

- **Title**

Enter the text to appear above the snapshot.

- **Caption**

Enter the text to appear below the snapshot.

**Insert
Anything
into
Report?**

Yes. Image.

File Name

rptgen_hg.chg_ax_snap

Chapter/Subsection

Purpose

Groups portions of the report into sections with titles

Description

This component groups portions of the report into sections. Each section has a title and content.

The following rules apply to the Chapter/Subsection component:

- Children of the Chapter/Subsection component are placed inside the section created by this component.
- The Chapter/Subsection component accepts paragraph-level children unless you select the **Get title from first child component** check box. In this case, a Chapter/Subsection component's first child must be a Text component.
- The Chapter/Subsection component can be a child of another Chapter/Subsection component.

Sections can be nested, and titles automatically become smaller inside subsections. There are seven levels of nesting possible. Chapter/Subsection components beyond the sixth section are not titled, although the child components of these sections insert information into the report.

Chapter Numbering

By default, chapters are numbered and sections are not numbered. Chapter and section numbering are specified by the stylesheet you choose in the Report Options attribute page. For more information about chapter and section numbering options in Web and print stylesheets, see "Setting the Report Format" on page 3-4.

Section Title

- **Title**
Enter a title for the display in the generated report. To generate a title automatically, select `Automatic`. To enter a title, select `Custom`.
- **Numbering**
Set a numbering style for the report. To number by context, select `Automatic`. To create your own numbering style, select `Custom`.

- **Section Type**

The item in the **Section Type** pane shows you in which level the selected section resides.

**Insert
Anything
into
Report?**

Yes. Chapter or section.

File Name

rptgen.cfr_section

Comment

Purpose

Inserts a comment into the XML source file created by the report generation process

Description

This component inserts a comment into the XML source file created by the report generation process. The comment is not visible in the generated report.

This component can have children. Child components insert their output into the XML source file, but this output appears within comment tags and does not appear in the generated report.

For comment text to appear in the report:

- 1 Edit the XML source file (which has the same name as your report file, but has a .xml extension).
- 2 Find the comment area in the XML source file by locating the comment tags `<--` and `-->`. Remove these tags.
- 3 Convert the XML source file using the `rptconvert` command.

For more information on the `rptconvert` command, see “Converting XML Documents” on page 3-11).

Properties

- **Comment text**

Enter comments you want to include in the report.

- **Show comment in Generation Status window**

Display comments in the **Generation Status** tab when the report is generating.

- **Status message priority level**

This option is only available if you select the **Show comment in Generation Status window** option. Choose the status message priority level that appears during report generation. The priority options range from 1) Error messages only to 6) All messages. The default is 4) Standard messages.

**Insert
Anything
into
Report?**

No. This component inserts comments, which can appear in the report, into the report's XML source file.

File Name

rptgen.crg_comment

Empty Component

Purpose	Groups components so that the components can be easily moved, activated, or deactivated, or creates a blank space in a list
Description	<p>This component does not insert anything into the report. It can have any component as a child. Use it to group components together so that the components can be easily moved, activated, or deactivated, or creates a blank space in a list.</p> <p>If MATLAB Report Generator does not recognize a component when loading a report template, it replaces the unrecognized component with the Empty Component.</p>
Insert Anything into Report?	No.
File Name	rptgen.crg_empty

Purpose	Evaluates a specified MATLAB expression in the workspace
Description	This component evaluates a specified MATLAB expression in the workspace. The code and/or command-line output can be inserted into the report.
Properties	<ul style="list-style-type: none">• Insert MATLAB expression in report<p>Select this option if you want the MATLAB expression that is evaluated by this component to appear in the report.</p>• Display command window output in report<p>Include the command window output that results when the MATLAB expression is evaluated.</p>• Expression to evaluate in the base workspace<p>Enter the expression that you want to evaluate in the MATLAB workspace.</p><p>Include this attribute in a loop component, such as a Block Loop, a Model Loop, a Signal Loop, or a System Loop to obtain a handle to the object in whose context the loop is executing. For example:</p><ul style="list-style-type: none">▪ Enter the following if you want to obtain a handle to the block in whose context a Block Loop is executing:<pre>%Access to block blockname = RptgenSL.getReportedBlock; blockHandle = get_param(blockname, 'handle');</pre>▪ Enter the following if you want to obtain a handle to the model in whose context a Model Loop is executing:<pre>%Access to model modelName = RptgenSL.getReportedModel; modelHandle = get_param(Modelname, 'handle');</pre>

Evaluate MATLAB Expression

- Enter the following if you want to obtain a handle to the signal in whose context a Signal Loop is executing:

```
%Access to signal
signalname = RptgenSL.getReportedSignal;
signalHandle = get_param(signalname, 'handle');
```

- Enter the following if you want to obtain a handle to the system in whose context a System Loop is executing:

```
%Access to system
systemname = RptgenSL.getReportedSystem;
systemHandle = get_param(systemname, 'handle');
```

Similarly, use set calls in place of the get calls in the previous examples to set your current block, model, signal, or system within the context of a loop component.

- **Evaluate this expression if there is an error**

Evaluate another MATLAB expression if the evaluated expression produces an error. Enter the expression that should be evaluated in case of an error in this field.

Insert Anything into Report?

Inserts text only if you select one of the following options:

- Insert MATLAB expression string in report
- Display command window output in report

File Name

rptgen.cml_eval

Purpose

Applies each child component to specified graphics figures

Description

Applies each child component to specified figures in the report. For more information about working with this component, see “Working with Looping Components” on page 4-3.

Figure Selection

Choose to loop on the following:

- The current figure
- All visible figures
- Figures with certain tags

- **Include figures**

To include only the current figure in the report, select `Current figure only`.

To loop on all visible figures, select `Visible figures`. This option executes child components for all figures that are currently open and visible. The `Data figures only` option is checked by default and excludes figures with `HandleVisibility = 'off'` from the loop.

To loop on figures with certain tags, select `All figures with tags`. When you select a tag, all figures with that tag are included, regardless of whether the figure is visible or whether `HandleVisibility` is 'on' or 'off'. The tag list box (located under `All figures with tags`) shows the selected tags. To add a tag to the tag list box, select the tag from the list on the right and click **Add**. The tag list displays a list of tags for figures that are open. If you want to add a tag that does not appear in the tag list, type in the name of the tag.

- **Loop Figure List**

Shows you all figures in the current set of figures that will be included in the loop.

Figure Loop

Note If the report template generates new figures or changes existing figures, figures in the **Loop Figure List** are not the figures that are reported on.

Section Options

- **Create section for each object in loop**
Select this option to insert a section in the generated report for each object found in the loop.
- **Display the object type in the section title**
Select this option to insert the object type automatically into the section title in the generated report.
- **Create link anchor for each object in loop**
Select this option to create a hyperlink to the object in the generated report.

Insert Anything into Report?

Yes, inserts a section if the **Create section for each object in loop** option is selected.

File Name

rptgen_hg.chg_fig_loop

Purpose	Inserts a snapshot of a Handle Graphics figure into the report
Description	This component inserts a snapshot of a Handle Graphics figure into the report.
Format	<ul style="list-style-type: none">• Image file format<p>Choose the image file format from this list. Select Automatic HG Format to automatically choose 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 HG Format is the default option. Other options include:</p><ul style="list-style-type: none">▪ Automatic HG Format (uses the Handle Graphics file format selected in the Preferences dialog box)▪ Adobe Illustrator▪ 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

Figure Snapshot

- JPEG high quality image
- JPEG medium quality image
- JPEG low quality image
- PNG 24-bit image
- TIFF - compressed
- TIFF - uncompressed
- Windows metafile

- **Capture picture from screen**

Capture a figure from the screen. Choose `Client` area only to capture a portion of the figure window, or `Entire` figure window to capture the whole window.

Print Options

Use the attributes in this section to set printing parameters.

- **Paper orientation**

Choose `Landscape`, `Portrait`, `Rotated`, or `Use figure` orientation. For more information about paper orientation, see the `orient` command in the MATLAB documentation.

- **Image size**

Select `Use figure` `PaperPositionMode` setting to use 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.

For the image in the report to appear as the same size as on the screen, select `Automatic` (same size as on screen).

Select `Custom` to choose a custom image size. Specify the image size in the **Size** field and **Units** list.

- **Size**

This field is active only if you choose Custom in the **Image size** list box. Enter the size of the Handle Graphics figure snapshot in the form wxh (width times height).

- **Units**

This field is active only if you choose Custom in the **Image size** list box. Enter the units for the size of the Handle Graphics figure snapshot.

- **Invert hardcopy**

Handle Graphics figures have an `InvertHardcopy` property. This parameter inverts colors for printing; i.e., it changes dark colors to light colors and vice versa.

Select one of the following options:

- Automatic. Automatically change dark axes colors to a light axes color. If the axes color is a light color, it will not be inverted.
- Invert. Change dark axes colors to light axes colors and vice versa.
- Don't invert. Do not change the colors in the image displayed on the screen for printing.
- Use figure's `InvertHardcopy` setting. Use the `InvertHardcopy` property set in the Handle Graphics image.
- Make figure background transparent. Make the image background transparent.

Display Options

- **Scaling**

To specify the number and type of units, select `Fixed` size.

To specify the percentage, the maximum size, and the units of measure, select `Zoom`.

For the image to appear in the report as the same size it does on the screen, select `Use image size`.

- **Size**

Figure Snapshot

This field is active only if you choose **Fixed** size in the **Scaling** list. Enter the size of the snapshot in the form **w h** (width height) and press **Enter**.

- **Max size**

This field is active only if you choose **Zoom** from the **Scaling** list. Enter the maximum size of the snapshot in the form **w h** (width height) and press **Enter**.

- **Units**

This field is active only if you choose **Zoom** or **Fixed** size in the **Image size** list box. Enter the units for the size of the snapshot.

- **Alignment**

Options are **Auto**, **Right**, **Left**, and **Center**.

Note This option is supported only in PDF and RTF reports.

- **Title**

Enter a title for the snapshot figure. Select **Custom** to enter a custom title. Select **Name** to specify the figure name as the title.

- **Caption**

Enter the text you want to appear below the snapshot.

Insert Anything into Report?

Yes. Image.

File Name

rptgen_hg.chg_fig_snap

Purpose

Iteratively executes its child components

Description

This component iteratively executes its child components. It must have at least one child component to execute properly. This component functions like the `for` loop in MATLAB, except that instead of executing a statement, it executes its child components while one of the following loops is active:

```
for varname=x:y:z
for varname=[a b c ...]
```

In the previous example, `varname` must be a string that is valid as a variable name. `x`, `y`, `z` can be scalars or workspace expressions that are evaluated in the workspace as scalars. If any one of these is not a scalar, the For Loop does not execute.

For example, if `red` is a workspace variable, the following is a valid `for` statement:

```
for i=1:1:length(red)
```

`length(red)` is 3, which is a scalar.

Loop Type

Choose the loop type to be incremented indices or a vector of indices. For information details on `for` loops and indices, see `for` in the MATLAB documentation.

- **Incremented indices**

Execute a `for` loop of the following form:

```
for varname=x:y:z
```

Enter appropriate values in the following fields:

- **Start** corresponds to `x`.
- **Increment** corresponds to `y`.
- **End** corresponds to `z`.

For Loop

- **Vector of Indices**

Execute a for loop of the following form:

```
for varname=[a b c ...]
```

Enter appropriate values in the **Vector** field in the form a b c

Workspace Variable

- **Show index value in base workspace**

Place the loop index (the value of the for loop variable) in the workspace while other components are executing.

- **Variable name**

Enter the variable name in this field. The default is RPTGEN_LOOP.

- **Remove variable from workspace when done**

This option is only available if you selected the **Show index value in base workspace** option. Remove the loop index from the workspace.

Insert Anything into Report?

No.

File Name

rptgen_lo.clo_for

Purpose	Runs its child components for each Handle Graphics object that is currently open in the MATLAB workspace
Description	The Graphics Object Loop component runs its child components for each Handle Graphics object that is currently open in the MATLAB workspace. The component inserts a table into the generated report.
Select Objects	<p>Exclude GUI objects (uicontrol, uimenu, ...) Select this option to exclude GUI objects, such as <code>uicontrol</code> and <code>uimenu</code>, from the loop.</p> <p>Loop list Use this list to select the loop level for Handle Graphics objects:</p> <ul style="list-style-type: none">• Loop on objects with handle visibility "on"• Loop on all objects <p>Search for Enter search terms in this field. Separate search terms with a space.</p>
Section Options	<p>Create section for each object in loop Select this option to insert a section in the generated report for each object found in the loop.</p> <p>Display the object type in the section title Select this option to insert the object type automatically into the section title in the generated report.</p> <p>Create link anchor for each object in loop Select this option to create a hyperlink to the object in the generated report.</p>
Insert Anything into Report?	Yes, inserts a section if the Create section for each object in loop option is selected.

Graphics Object Loop

File Name rptgen_hg.chg_obj_loop

Handle Graphics Linking Anchor

Purpose	Designates a location to which links point
Description	The Handle Graphics Linking Anchor component designates a location to which links point. This component should have a looping component as its parent.
Properties	<ul style="list-style-type: none">• Insert text Enter text to appear after the linking anchor in this field.• Link from current Set the current model, system, block, or signal as the linking anchor.<ul style="list-style-type: none">▪ Select Automatic to automatically select the appropriate figure, axes, or object as a linking anchor. If the Figure Loop component is the parent component, the linking anchor is set on the current figure. Similarly, if the Graphics Object Loop is the parent component, the linking anchor is inserted for the current object.▪ Select Figure to set the linking anchor to the current figure.▪ Select Axes to set the linking anchor to the current axes.▪ Select Object to set the linking anchor to the current object.
Insert Anything into Report?	Yes. Anchor.
File Name	rptgen_hg.chg_obj_anchor

Handle Graphics Name

- Purpose** Inserts the name of a Handle Graphics object into the report
- Description** This component inserts as text the name of a Handle Graphics object into the report.
- Use this component to create a section title based on the current figure by making it the first child component of a Chapter/Subsection component, and then select the **Get title from first child component** option in the Chapter/Subsection component.
- Properties**
- **Display name as**

Select Figure to show the name of the current Handle Graphics figure. The name of the current figure comes from the first nonempty figure parameter in the following order:

 - a** Name
 - b** Tag
 - c** file name
 - d** Handle
 - **Show name of current**

Select Axes to show the name of the current axes. The name of the current axes comes from the first nonempty axes parameter in the following order:

 - a** Title
 - b** Xlabel
 - c** Ylabel
 - d** Zlabel
 - e** Tag
 - f** Handle

Select the Other Object option to take the name of the current object from the figure's CurrentObject parameter and its first nonempty figure parameter in the following order:

- uicontrol
 - a** String
 - b** Style
 - c** Handle
- uimenu
 - a** Label
 - b** Accelerator
 - c** Tag
 - d** Handle
- uicontextmenu, image, light, line, patch, or surface
 - a** Tag
 - b** Handle
- text
 - a** String
 - b** Tag
 - c** Handle

Show name of current

Select one of the three options listed below **Name**:

- Figure
- Axes
- Other Object

Handle Graphics Name

**Insert
Anything
into
Report?**

Yes. Text.

File Name

rptgen_hg.chg_obj_name

- Purpose** Inserts a property name/property value pair from a Handle Graphics figure, axes, or other object
- Description** The Handle Graphics Parameter component inserts a property name/property value pair from a Handle Graphics figure, axes, or other object.
- Property Selection**
- **Get property from current**
Select a Handle Graphics object to report on:
 - Figure inserts a figure's property name/property value pair.
 - Axes inserts an axes' property name/property value pair.
 - Object inserts an object's property name/property value pair.
 - **Figure property**
Choose the type of property you want to include from the list in this section. All shows every parameter for the current object.
- Display Options**
- **Title**
Choose a title for the display in the generated report. The default is None. To generate a title automatically from the parameter, select Automatic. To enter a title, select Custom.
 - **Size limit**
Use this size to limit the width of the display in the generated report. Units are in pixels. When creating a table, the size limit is the hypotenuse of the table width and height [$\sqrt{w^2+h^2}$]. The size limit of text is equal to the number of characters squared. If the size limit is exceeded, the variable is rendered in condensed form, such as [64x64 double]. Setting a size limit of zero will ignore the size limit and always display the variable, no matter how large.
 - **Display as**
Choose a display style from the menu:

Handle Graphics Parameter

- Auto table/paragraph. Displays as a table or paragraph based on the information.
- Inline text. Displays as inline, which fits in line with the surrounding text.
- Paragraph. Displays as a text paragraph.
- Table. Displays as a table.
- **Ignore if value is empty**
Exclude empty parameters in the generated report.

Insert Anything into Report?

Yes. Text.

File Name

rptgen_hg.chg_property

Purpose

Inserts a table that reports on property name/property value pairs

Description

This component inserts a table that reports on property name/property value pairs.

For more information on using the Handle Graphics Property Table component, see “Working with Property Table Components” on page 4-9.

Select Graphics Object

- **Object type**

Choose the object type for the generated report:

- Figure
- Axes
- Object

- **Filter by class**

Specify a class or classes for the table. For example, when selecting type Object, filter by class line. If the object is type surface, it does not generate a property table.

Table

A preset table is a table that is already formatted and set up. You can select a preset table in the preset table list in the upper-left corner of the attributes page. To apply a preset table, select the table and click **Apply**.

- **Preset table**

Choose a type of table to display the object property table:

- Defaults
- Callbacks
- Graphics
- Printing
- Blank 4x4

Handle Graphics Property Table

- **Split property/value cells**

Split property name/property value pairs into separate cells. To have the property name and property value 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 and there can be only one property name/property value pair in a cell. If you have more than one name/property pair in a cell, only the first pair appears in the report. All subsequent pairs are ignored.

To have the property name and property value appear together in one cell, clear the **Split property/value cells** check box. This is nonsplit mode. Nonsplit mode supports more than one property name/property value pair. It also supports text.

Before switching from nonsplit mode to split mode, make sure that you have only one property name/property value pair per table cell. If you have more than one property name/property value pair or any text, only the first property name/property 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 **Title Properties** pane.

Cell Properties

The visible options in the **Title Properties** pane depend on the object selected in the **Table Cells** pane. If **Figure Properties** is selected, only the **Contents** and **Show** options are displayed. If any other object is selected in the **Table Cells** pane, the **Lower border** and **Right border** options are displayed.

- **Contents**

Modify the contents of the table cell selected in the **Table Cells** pane. Options include:

- Left

Handle Graphics Property Table

- Center
- Right
- Double justified
- **Show as**

Choose the format for the contents of the table cell. Options include:

 - Value
 - Property Value
 - PROPERTY Value
 - Property: Value
 - PROPERTY: Value
 - Property - Value
 - PROPERTY - Value
- **Alignment**

Align text in your cells. Options are:

 - Left
 - Center
 - Right
 - Double-justified
- **Lower border**

Display the lower border of the table in the generated report.
- **Right border**

Display the right border of the table in the generated report.

Handle Graphics Property Table

Creating Custom Tables

To create a custom table, select a preset table and then edit it. You might want to start with the Blank 4x4 preset 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 “Working with Property Table Components” on page 4-9.

Insert Anything into Report?

Yes. Table.

File Name

rptgen_hg.chg_prop_table

Handle Graphics Summary Table

Purpose	Inserts a table that summarizes Handle Graphics object properties
Description	This component inserts a table that summarizes Handle Graphics object properties. Each row in the table represents an object, and each column in the table represents a property. Specify object properties to be included in the report.
Properties	<ul style="list-style-type: none">• Object type Choose the object type to display in the generated report. Options include:<ul style="list-style-type: none">▪ figure▪ axes▪ objectThe available options in the Select Objects pane depend on your selection in the Object type menu.• Table title Choose a title for the summary table in the generated report. Options include:<ul style="list-style-type: none">▪ Automatic. Generate a title automatically from the parameter.▪ Custom. Enter a title manually.
Property Columns	<ul style="list-style-type: none">• Property columns This figure displays the object properties to be included in the summary table in the generated report. To add a property, select the appropriate property level in the menu. Next, in the contextual list below the menu, select the property that you want to add and click Add. To delete a property, select the property name and click Delete.

Handle Graphics Summary Table

Note Some entries in the list of available properties (such as Depth) are “virtual” properties which cannot be accessed 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**

Remove empty columns from the summary table in the generated report.

Object Rows

Insert anchor for each row

Insert an anchor for each row in the summary table.

Figure Selection

The options displayed in the **Figure Selection** pane depend on the object type selected in the **Object type** list.

- If figure is selected in **Object type**, the following options appear:

- **Include figures**

To include only the current figure in the report, select `Current figure only`.

To loop on all visible figures, select `Visible figures`. This option executes child components for figures that are currently open and visible. The `Data figures only` option is checked by default and excludes figures with `HandleVisibility = 'off'` from the loop.

To loop on figures with certain tags, select `All figures with tags`. All figures with the specified tag are included regardless of whether the figure is visible or whether `HandleVisibility` is 'on' or 'off'. The tag list box, located under the `All figures with tags` option, shows the selected tags. Enter a tag name to add it to the tag list box.

Handle Graphics Summary Table

- **Data figure only (Exclude applications)**
Check this box to show only data figures.
- **Loop Figure List**
Shows the figures within the current set of figures to be displayed.
- If axes is selected in **Object type**, the following options are displayed:
 - **Loop type**
To loop on all axes objects, select All axes. To loop on the selected axes object, select Current axes.
 - **Exclude objects which subclass axes**
Exclude objects such as legends and color bars.
 - **Loop Menu**
To loop only on visible axes objects, select Loop on axes with handle visibility "on". To loop on all axes objects, select Loop on all axes.
 - **Search terms**
Enter search terms for the loop. For example, to search for Tag and My Data, enter "Tag", "My Data".
- If object is selected in **Object type**, the following options appear:
 - **Exclude GUI objects (uicontrol, uimenu, ...)**
Exclude GUI objects, such as uicontrol and uimenu, from the loop.
 - **Loop menu**
Select the loop level for Handle Graphics objects:
 - Loop on objects with handle visibility "on"
 - Loop on all objects
 - **Search for**

Handle Graphics Summary Table

Enter space-delimited search terms in this field.

**Insert
Anything
into
Report?**

Yes. Table.

File Name

rptgen_hg.chg_summ_table

Purpose	Inserts an image from an external file into the report
Description	This component inserts an image from an external file into the report. It can have the Chapter/Subsection or Paragraph component as its parent. If the Paragraph component is its parent, the Insert as inline image check box must be selected.
File Name	<ul style="list-style-type: none">• File name Enter the image file name in this field or use the Browse button (...) to find the file that contains the image. Make sure that the image is in a format that your viewer can read. Your viewer depends on the report format you selected in the Report Options page in the Report Explorer. For more information on report format options, see “Setting Report Output Options” on page 3-2. An error like the following is produced if you specify the name of an image file that does not exist. No file name. Could not create graphic. The %<VariableName> notation is supported. For more details about this notation, see %<VariableName> Notation on the Text component reference page.• Copy to local report files directory Save a copy of the image to a local report files directory.
Display Options	<ul style="list-style-type: none">• Scaling To specify the number and type of units, select <code>Fixed size</code>. To specify the percentage, the maximum size, and the units of measure, select <code>Zoom</code>. For the image to appear in your report as the same size as on the screen, select <code>Use image size</code>.• Size

Image

This field is active only if you choose `Fixed` size from the **Scaling** list. Enter the size of the snapshot in the form `w h` (width height) and press **Enter**.

- **Max size**

This field is active only if you choose `Zoom` from the **Scaling** list. Enter the maximum size of the snapshot in the form `w h` (width height) and press **Enter**.

- **Units**

This field is active only if you choose `Zoom` or `Fixed` size in the **Image size** list box. Enter the units for the size of the snapshot.

- **Alignment**

Options are:

- Auto
- Right
- Left
- Center

Note This option is supported only in PDF and RTF reports.

- **Title**

Enter text to appear above the snapshot.

- **Caption**

Enter text to appear below the snapshot.

Preview

The **Preview** pane displays the image that you specify in the **Image file name** field. You cannot preview PostScript images or images with formats that are not supported by the `imread` function, such as `.gif`.

Clicking an image in the **Preview** pane causes it to appear in full size on the screen.

**Insert
Anything
into
Report?**

Yes. Image.

File Name

rptgen.cfr_image

Import File

Purpose Imports an ASCII text file into the report

Description This component imports an ASCII text file into the report.

Properties

- **File name**

Enter the name of the file to import into the text field in this section or use the **Browse** button (...) to find the file. This component executes only if you enter a valid file name in this field.

- **Import file as**

Indicates what kind of format to use for the imported file. Options include:

- **Plain text (ignore line breaks):** Import the file as plain text without any line breaks (no paragraphs). If you select this option, the Import File component acts like the Text component and should have the Paragraph component as its parent.

In this example and subsequent examples, the following text is used as the input file.

```
This is the first row of text from the imported file.  
The second row follows a line break in the first row.
```

```
There is a blank line above the third row.
```

Using Plain text (ignore line breaks), the following is produced:

```
This is the first row of text from the imported file.  
The second row follows a line break in the first row.
```

```
There is a blank line above the third row.
```

- **Paragraphs defined by line breaks:** Import the file as text contained in paragraphs defined by line breaks (hard returns or carriage returns).

Using Paragraphs defined by line breaks, the following is produced:

```
This is the first row of text from the imported file.  
The second row follows a line break in the first row.
```

```
There is a blank line above the third row.
```

- Paragraphs defined by empty rows: Import the file as text contained in paragraphs defined by empty rows (rows that do not contain text).

Using Paragraphs defined by empty rows, the following is produced:

```
This is the first row of text from the imported file.  
The second row follows a line break in the first row.
```

```
There is a blank line above the third row.
```

- Text (retain line breaks): This is the default option. Select this option to import the file as plain text, including line breaks.

Using Text (retain line breaks), the following is produced:

```
This is the first row of text from the imported file.  
The second row follows a line break in the first row.
```

```
There is a blank line above the third row.
```

- Fixed-width text (retain line breaks): Import the file as fixed-width text (all letters have the same width or size) including line breaks. This option is best suited for importing M-files.

Using the Fixed-width text (retain line breaks), the following is produced:

```
This is the first row of text from the imported file.  
The second row follows a line break in the first row.
```

Import File

There is a blank line above the third row.

- DocBook XML: Select this option if the inserted text file is in an XML source file. The format of the inserted XML file is not changed.
- Formatted Text (RTF/HTML): Select this option if the inserted text file is in an RTF or HTML source file. The format of the inserted RTF or HTML source file is not changed.
- Syntax highlighted M-code: Select this option if the inserted text file is an M-file containing highlighted M-code.

The **File Contents** field displays the first few lines of the file to be imported.

Insert Anything into Report?

Yes. Inserts text if you select one of the following options:

- Plain text (ignore line breaks)
- Text (retain line breaks)
- Fixed-width text (retain line breaks)

Inserts paragraphs if you select one of the following options:

- Paragraphs defined by line breaks
- Paragraphs defined by empty rows

Inserts the contents of the XML file if you select the DocBook XML option. This can include text, paragraphs, tables, etc.

Inserts the contents of the RTF or HTML file if you select the Formatted text (RTF/HTML) option. This can include text, paragraphs, tables, etc.

Inserts a link to a file if you import the file into an HTML report.

File Name

rptgen.crg_import_file

Purpose	Inserts variable values into the report
Description	<p>This component inserts the value (and optionally, the name) of each the following variables into the report:</p> <ul style="list-style-type: none">• A variable from the MATLAB workspace• A variable from a MAT-file• A global variable
Source	<ul style="list-style-type: none">• Variable name<p>Enter the name of the variable in this field.</p><p>Use %<VariableName> to insert the value of a variable from the MATLAB workspace. For example, if you enter the following expression in this field:</p><pre>%<my_variable></pre><p>And you have the following variables in your MATLAB workspace:</p><pre>my_variable='velocity' velocity=50</pre><p>Then your report will contain the following:</p><pre>50</pre><p>If you enter %<velocity> in this field, and your MATLAB workspace contains the same information, your report will also contain 50.</p>• Variable location<p>Select one of the following options:</p><ul style="list-style-type: none">▪ Base Workspace. Gets a variable from the MATLAB workspace.▪ MAT File. Gets a variable from a binary file with a .mat extension.▪ Global variable. Gets a global variable.

Insert Variable

Display Options

- **Title**

Choose a title for the display in the generated report. To generate a title automatically from the parameter, select Automatic. To enter a title, select Custom. To specify no title, select None.

- **Size limit**

Limits 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 table width and height $[\sqrt{w^2+h^2}]$. The size limit of a given block of text is the number of characters squared. If the size limit is exceeded, the variable renders in condensed form, such as `[64x64 double]`. Setting a size limit of zero causes the size limit to be ignored and always displays the variable regardless of size.

- **Display as**

Choose a display style from the menu:

- Table. Displays as a table.
- Paragraph. Displays as a text paragraph.
- Inline text. Displays as inline, which fits in line with the surrounding text.
- Auto table/paragraph. Displays as a table or paragraph based on the information.

- **Ignore if value is empty**

Exclude empty parameters in the generated report.

Insert Anything into Report?

Yes. Text.

File Name

rptgen.cml_variable

Purpose	Inserts linking anchors or pointers into the report
Description	This component inserts linking anchors or pointers into the report.
Properties	<ul style="list-style-type: none">• Link type<p>Select the type of link to insert into the report. options include:</p><ul style="list-style-type: none">▪ Linking anchor: An anchor is the object to which something links; i.e., links “link” to an anchor.▪ Internal document link: This link takes you to the location in your report (specified by an anchor).▪ URL (external) link: A link to a Web site. It takes you to the location on the Web to which your link points.• Link identifier<p>This field marks the location to which the link points. It can contain only ASCII characters, and it is not visible in the generated report.</p><p>The link identifier options are context-sensitive, and their formats differ depending on the link type you select. For example, say you want to link to an external file <code>foo.txt</code>. On UNIX , the link would look something like this:</p><pre>file:///home/janedoe/foo.txt</pre><p>While on Windows, it would look something like this:</p><pre>H:\foo.txt</pre>• Link text<p>Enter text to use in the link.</p>• Emphasize link text<p>Italicize the link text.</p>

Link

**Insert
Anything
into
Report?**

Yes. Text or anchor.

File Name

rptgen.cfr_link

Purpose	Creates a bulleted or numbered list from a cell array or from child components
Description	This component creates a bulleted or numbered list from a cell array or from child components.
List Content	<ul style="list-style-type: none">• Create list from workspace cell array Enter the name of the 1-by-n or n-by-1 cell array from which the list will be created in this field. This option is not available when this component has child components, because in this case the list is automatically generated from the child components.• List title Enter the title of the list.
List Formatting	<ul style="list-style-type: none">• List style Select either <code>Bulleted list</code> or <code>Numbered list</code>.• Numbering style Select a numbering style for numbered lists. Options include:<ul style="list-style-type: none">▪ 1,2,3,4,...▪ a,b,c,d,...▪ A,B,C,D,...▪ i,ii,iii,iv,...▪ I,II,III,IV,...• Show parent number in nested list (1.1.a) Show all level numbers in a nested list. Create a nested list by putting one cell array inside another or by nesting one List component inside another. Following is an example of how a list displays with this option selected.<ol style="list-style-type: none">1. Example

List

- 2. Example
 - 2.1. Example
 - 2.2. Example
 - 2.2.a. Example
 - 2.2.b. Example
- 3. Example

Note This option is not available if you select Show only current list value (a).

- **Show only current list value (a)**

Select this option to show only the current list value. Following is an example of how a list displays with this option selected.

- 1. Example
- 2. Example
 - 1. Example
 - 2. Example
 - 1. Example
 - 2. Example
- 3. Example

Note This option is not available if you select Show parent number in nested list (1.1.a).

Example 1: Creating a Nested List

Create a nested list by putting one cell array inside another or by nesting one List component inside another.

Consider the following report template, with one List component nested inside another List component:

```
[ - ] Report - Unnamed.rpt
```



```
[ - ] Bulleted list from child components
  [ ] Text - sky
  [ ] Table - varname
  [ ] Image - test.jpg
  [ ] Text - grass
[ - ] Bulleted list from child components
  [ ] Text - clouds
  [ ] Text - sun
[ - ] Paragraph - information
```

This report template generates a report that includes the following bulleted lists:

- sky
- varname, the table from the variable
- test.jpg, a snapshot of the image
- grass
 - clouds
 - sun
- information

Example 2: Creating the Same List with Child Components and a Cell Array

To generate a report that includes the following bulleted list:

- red
- green
- blue

Follow one of these steps:

- 1** Create the following report template:

List

```
[ - ] Report - Unnamed.rpt
      [ - ] Bulleted list from child components
            [ ] Text - red
            [ ] Text - green
            [ ] Text - blue
```

2 Configure a report template to call a cell array, colors:

```
[ - ] Report - Unnamed.rpt
      [ - ] Bulleted list from cell array called colors
```

Wherecolors is defined as:

```
colors={'red','green','blue'}
```

Insert Anything into Report?

Yes. List.

File Name

rptgen.cfr_list

Purpose Acts as an else when it is the child of the Logical If component

Description This component acts as an else when it is the child of the Logical If component. There are two ways to use this component:

```
if
  then
  else

or

if
  then
  elseif
  elseif
  .
  .
  .
  else
```

Properties **If component has no children, insert text**

When the Logical Else component has no child components, the string in this field is inserted into your report. In this case, this component acts like the Text component.

Insert Anything into Report? Depends on whether if or elseif statement is true.

File Name rptgen_lo.clo_else

Logical Elseif

Purpose	Acts as an elseif when it is the child of the Logical If component
Description	<p>This component acts as an elseif when it is the child of the Logical If component. You must use this component in this format:</p> <pre>if then elseif elseif . . . else</pre>
Properties	<ul style="list-style-type: none">• Test expression Enter a MATLAB expression to evaluate.• If component has no children, insert text When the Logical Elseif component has no child components, the string in this field is inserted into the report. In this case, this component acts like the Text component.
Insert Anything into Report?	Yes, only when parent if statement is false.
File Name	rptgen_lo.clo_else_if

Purpose

Acts as a logical if

Description

This component acts as a logical if; it can have the Logical Then, Logical Elseif, and Logical Else components as its child components. This component executes its child components when the given workspace expression is true. It displays a specified string when it has no child components. There are three ways to use this component:

```
if
  then
```

or

```
if
  then
  else
```

or

```
if
  then
  elseif
  elseif
  .
  .
  .
  else
```

Properties

- **Test expression**

Enter a MATLAB expression to evaluate.

- **If component has no children, insert text**

When the Logical If component has no child components, the string in this field is inserted into your report.

Logical If

**Insert
Anything
into
Report?**

Depends on attributes.

File Name

rptgen_lo.clo_if

Purpose Acts as a then when it is the child of the Logical If component

Description The Logical Then component acts as a then when it is the child of the Logical If component. There are three ways to use this component:

```
if
  then

or

if
  then
  else

or

if
  then
  elseif
  elseif
  .
  .
  .
  else
```

Attributes **If component has no children, insert text**

When the Logical Then component has no child components, the string in this field is inserted into your report. In this case, this component acts like the Text component.

Insert Anything into Report? Yes, when parent if statement is true.

File Name rptgen_lo.clo_then

MATLAB Property Table

Purpose Inserts a table that reports on MATLAB object property name/property value pairs

Description This component inserts a table that reports on MATLAB object property name/property value pairs.

Table A preset table is a table that is already formatted and set up. Select a preset table in the preset table list in the upper-left corner of the attributes page. To apply a preset table, select the table and click **Apply**.

- **Preset table**

Choose a type of table to display the object property table:

- Default
- Blank 4x4

- **Split property/value cells**

Split property name/property value pairs into separate cells. Select the **Split property/value cells** check box for the property name and property value to appear in adjacent cells. In this case, the table is in split mode, so there can be only one property name/property value pair in a cell. If more than one name/property pair exists in a cell, only the first pair appears in the report; subsequent pairs are ignored.

For a given property name and property value to appear together in one cell, clear the **Split property/value cells** check box. This is nonsplit mode. Nonsplit mode supports more than one property name/property value pair. It also supports text.

Before switching from nonsplit mode to split mode, make sure that you have only one property name/property value pair per table cell.

- **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

The visible options in the **Cell Properties** pane depend on the object selected in the **Table Cells** pane. If **Workspace Properties** is selected, only the **Contents** and **Show** options are displayed. If any other object is selected in the **Table Cells** pane, the **Lower border** and **Right border** options are displayed.

- **Contents**

Modify the contents of the table cell selected in the **Table Cells** pane.

- **Show as**

Choose the format for the contents of the table cell. Options include:

- Value
- Property Value
- PROPERTY Value
- Property: Value
- PROPERTY: Value
- Property - Value
- PROPERTY - Value

- **Alignment**

Choose the alignment of the contents of the selected table cell in the **Table Cells** field. Options include:

- Left
- Center
- Right
- Double justified

- **Lower border**

MATLAB Property Table

Display the lower border of the table in the generated report.

- **Right border**

Display the right border of the table in the generated report.

Creating Custom Tables

To create a custom table, select a preset table and edit it. You might want to start with the Blank 4x4 preset table. Add and delete rows and add properties. To open the Edit Table dialog box, click **Edit**.

For details about using this dialog box to create custom property tables, see “Working with Property Table Components” on page 4-9.

Insert Anything into Report?

Yes. Table.

File Name

rptgen.cml_prop_table

Purpose	Inserts a table that shows the version number, release number, and/or release date of MathWorks software
Description	This component inserts a table that shows the version number, release number, and/or release date of MathWorks software that you are currently using. Get a list of all your toolboxes by typing <code>ver</code> at the MATLAB command line.
Table Title	Table title Enter the title of the table in this field. The default is <code>version number</code> .
Table Columns	<ul style="list-style-type: none">• Version number Include the version number of any MathWorks software you are currently using.• Release number Include the release number of any MathWorks software you are currently using.• Release date Include the release date of any MathWorks software you are currently using.
Insert Anything into Report?	Yes. Table.
File Name	<code>rptgen.cml_ver</code>

Nest Setup File

Purpose

Allows one report template (.rpt file) to run inside another

Description

This component runs another report template at the point where the Nest Setup File component is located in the current report template.

The components of the inserted report template enter the current report template at the level of the Nest Setup File component. Thus, inserted components have the same parent component as the Nest Setup File component. Make sure that the inserted components can have this parent component.

Properties

- **Setup file to run**

Enter the name of the report template to import and run in this field or use the **browse** button (...) to find the file. This component executes only if you enter a valid file name in this field.

- **Inline nested report in this report**

Select this option to insert the nested report at the place in the original report where this component is located.

- **Insert link to external report**

Select this option to create two separate reports, one with the original report template and one with the nested report template.

- **Recursion limit**

If you want to nest a report template inside itself, set a recursion limit in this field. The recursion limit sets a limit on the number of times the report template can run itself.

- **Nest all reports with specified file name**

Select this option to nest all reports with the same name as specified in the **Setup file to run** option.

Example

In the example shown below, the report template R2.rpt is nested in R1.rpt:

```
[ - ] Report - R1.rpt
  [ ] Chapter
    [ - ] B
      [ ] Nest Setfile - R2.rpt
        [ ] C
          [ ] D
    [ ] D
```

```
[ - ] Report - R2.rpt
  [ ] 1
  [ ] 2
  [ - ] Chapter
    [ ] 4
    [ ] 5
```

The generated report is identical to the one generated by the following report template:

```
[ - ] Report - R1.rpt
  [ ] Chapter
    [ - ] B
      [ ] 1
      [ ] 2
      [ - ] Section 1
        [ ] 4
        [ ] 5
      [ ] C
    [ ] D
```

Components that determine their behavior from their parents, such as Chapter/Subsection, are affected by components in the parent report template (as shown above). See the Chapter/Subsection reference page for more details on the Chapter/Subsection component.

Insert Anything into Report?

Yes, if the nested report template produces visible output.

File Name rptgen.crg_nest_set

Paragraph

Purpose

Inserts paragraph text into the report

Description

This component inserts a paragraph into the report. The paragraph text is taken from a child text component or from text that you enter in the **Paragraph Text** field.

Title Options

- **No paragraph title**

Select this option (the default) if you do not want the paragraph to have a title.

- **Get title from first child**

The title of the paragraph is taken from the first child component. The first child component should be a Text component.

- **Custom title**

Enter the title of the paragraph in this field if you do not want MATLAB Report Generator to get the title from the first child component.

Paragraph Text

Enter paragraph text in this field. If the Paragraph component has child components, the paragraph content is taken from the child components; otherwise, the Paragraph component inserts text from this field.

If the Paragraph component does not have any child components and you do not enter any text in this field, the Paragraph component does not insert anything into the report.

Use the %<VariableName> notation in this field if you want to insert the value of a variable from the MATLAB workspace. For more details about this notation, see %<VariableName> Notation on the Text component reference page.

Style

- **Bold**

Make the text bold.

- **Italic**

Make the text italic.

- **Underline**

Underline the text.

- **Strikethrough**

Strike through the text.

- **Retain spaces and carriage returns**

Cause the formatting of the text to appear in the generated report as it was entered.

- **Show text as syntax-highlighted M-code**

Show the text as syntax-highlighted M-code.

- **Color**

Choose the color of the text from the menu.

Insert Anything into Report?

Yes. Can be text, an image, etc.; depends on child components.

File Name

rptgen.cfr_paragraph

Stop Report Generation

Purpose Halts report generation

Description This component acts like **Stop** during report generation.

Use this component inside an if/then statement by using Logical and Flow Control components to halt the report generation process when specified condition is true. When report generation is halted, an XML source file is produced, but not converted.

Confirmation Properties

- **Confirm before stopping generation**

Generate a confirmation dialog box before stopping report generation.

- **Confirmation question**

Enter a confirmation question for the prompt. The default is “Stop generating the report?”

- **Halt button name**

Enter a name for the button that stops report generation. The default is “Halt Generation”.

- **Continue button name**

Enter a name for the button that continues report generation. The default is “Continue Generation”.

Example

The report template in this example creates a simple report that takes a snapshot of the current figure. If there is no current figure, the report generation automatically halts.

```
[ - ] Report - figure-report.rpt
[ - ] if (isempty(get(0,'CurrentFigure')))
[   ] Stop Generation
[ - ] Figure Loop - current
[ - ] Chapter - <Title from SubComponent1>
[   ] Figure Name
[   ] Graphics Figure Snapshot
[   ] Figure Prop Table - Figure Properties
```


Stop Report Generation

**Insert
Anything
into
Report?**

No.

File Name

rptgen.crg_halt_gen

Table

Purpose	Converts a rectangular cell array into a table and inserts the table into the report
Description	This component converts a rectangular cell array into a table and inserts the table into the report.
Table Content	<ul style="list-style-type: none">• Workspace variable name Enter the workspace variable name with which to construct the table.• Collapse large cells to a single description Consolidate large cells into one description.
Formatting Options	<ul style="list-style-type: none">• Table title Enter the title of your table. If you have a title for a table, it is included in the list of tables if the stylesheet you chose in the Report Options attribute page supports it. To find stylesheets that have a list of tables, see “Setting the Report Format” on page 3-4.• Cell alignment Select a cell alignment option (left, center, right, or double justified).• Column widths Input a vector with m elements, where m equals the number of columns in the table. Column sizing is relative and normalized to page width. For example, say you have a 2-by-3 cell array and input the following into the Column widths field: [1 2 3] The report output format for the cell array is such that the second column is twice the width of the first column and the third column is three times the width of the first column. If the vector is greater than the number of columns in the table, the vector is truncated so that the number of elements equals the number of columns. If m is less

than the number of columns in the table, the vector is padded with 1s so that the number of elements equals the number of columns.

- **Table grid lines**

Display grid lines in your table. Grid lines create borders between the fields.

- **Table spans page width (HTML only)**

Cause the table to have the same width as the page on which it appears.

Header/Footer Options

Designating a row as a header or footer row causes the contents of the row to be displayed in boldface.

- **Number of header rows**

Enter the number of header rows (from the top): 0 = none, 1 = 1 row from the top, 2 = two rows from the top, etc.

- **Footer list**

For no footers to appear in the report, select No footer. To select a footer that is different from your header, select Last N rows are footer and enter the number of footer rows (from the bottom) in the corresponding field: 0 = none, 1 = one row from the bottom, 2 = two rows from the bottom, etc.

Example

Consider the following cell array in the MATLAB workspace:

```
{'foo', 'bar'; [3], [5]}
```

The resulting cell table has the following appearance.

foo	bar
3	5

Table

Note that the table has no headers or footers and no title.

**Insert
Anything
into
Report?**

Yes. Table.

File Name

rptgen.cfr_table

Purpose	Formats and inserts text into the report
Description	This component formats and inserts text into the report. It must have the Paragraph component as its parent.
Properties	<p>Text to include in report</p> <p>Enter the text you want to include in your report in this field.</p> <p>%<VariableName> Notation</p> <p>Enter %<VariableName> in this field (and in any field where the text appears blue) to include the value of a variable from the base MATLAB workspace. You cannot enter more than one variable in %<>. If you enter an invalid variable name, the report will contain the text %<VariableName> instead of the value of the variable.</p> <p>Example</p> <ol style="list-style-type: none">1 Enter the following text: <pre>I have a %<ObjName> and it has %<NumLeaves> leaves. The word '%<ObjName>' has %<size(ObjName)> letters.</pre>2 Set ObjName = "plant" and NumLeaves = 3 in the MATLAB workspace.3 Generate the report. The report includes the following text: <pre>I have a plant and it has 3 leaves. The word 'plant' has 5 letters.</pre>
Style	<ul style="list-style-type: none">• Bold Make the text bold.• Italic Make the text italic.• Underline

Text

Underline the text.

- **Strikethrough**

Strike through the text.

- **Retain spaces and carriage returns**

Cause the formatting of the text to appear in the generated report as it was entered.

- **Show text as syntax-highlighted M-code**

Show the text as syntax-highlighted M-code.

- **Color**

Choose the color of the text from the menu.

Insert Anything into Report?

Yes. Text.

File Name

rptgen.cfr_text

Purpose	Inserts the time and date of the report generation into your report
Description	This component inserts the time and date of the report generation into your report as text. It must have the Paragraph or Chapter/Subsection component as its parent.
Prefix	Include text before stamp Select this option to include text before the time/date stamp, and enter the text in the corresponding field.
Time Stamp Properties	<ul style="list-style-type: none">• Include current time in stamp Insert the current time in the time/date stamp.• Time display Select the appearance of the time display. Options include:<ul style="list-style-type: none">▪ 12-hour▪ 24-hour• Time Separator Select a separation marker between hours, minutes, and seconds. The following options are available for the separator:<ul style="list-style-type: none">▪ Blank space () - For example, Hour Minute Second▪ Colon (:) - For example, Hour:Minute:Second▪ Period (.) - For example, Hour.Minute.Second▪ None () - For example, HourMinuteSecond• Include seconds in time stamp Display seconds in the time/date stamp.

Time/Date Stamp

Date Stamp Properties

- **Include current date in stamp**

Insert the current date in the time/date stamp.
- **Date order**

Select the order in which the day, month, and year should appear. Options include:

 - Day Month Year
 - Month Day Year
 - Year Month Day
- **Date separator**

Select a separation marker between day, month, and year. Options include:

 - Blank space (). For example, Day Month Year
 - Colon (:). For example, Day:Month:Year
 - Slash (/). For example, Day/Month/Year
 - Period (.). For example, Day.Month.Year
 - None (). For example, DayMonthYear
- **Month display**

Select the appearance of the month display. Options include:

 - Long (December)
 - Short (Dec)
 - Numeric (12)
- **Year display**

Select the appearance of the year display. Options include:

 - Long (2007)
 - Short (07)

Preview This pane displays the time/date stamp as it will appear in the report.

**Insert
Anything
into
Report?** Yes. Text.

File Name rptgen.crg_tds

Title Page

Purpose Inserts a title page at the beginning of the report

Description This component inserts a title page at the beginning of the report. Use it as a child component of a Chapter/Subsection component or alone in a report template.

Properties The text fields on this property pane can contain the %<VariableName> notation. For more details about this notation, see %<VariableName> Notation on the Text component reference page.

Main Tab **Title**

- **Title**

Enter the title of the report in this field. The title will appear in a large font.

- **Subtitle**

Enter the subtitle in this field. The subtitle will appear under the title in a smaller font than the title.

Options

- **Author**

Options include:

- Custom. This is the default. Enter a name for your report's author.
- No author. No author's name appears on the title page.
- Automatic author. Automatically include your user name as the author name.

The author name appears under the subtitle, in a smaller font than the subtitle.

- **Include report creation date**

Include the report creation date, and choose the date format in the corresponding list.

- **Include copyright holder and year**

Include copyright holder and year information.

Image Tab

File

- **File name**

Enter the file name for an image you want to appear on the title page. The image appears under the subtitle.

- **Copy to local report files directory**

MATLAB Report Generator copies the image file into the directory in which the report file is located.

Display Options

- **Scaling**

To specify the number and type of units, select `Fixed size`.

To specify the percentage, the maximum size, and the units of measure, select `Zoom`.

For the image to appear in your report as the same size it does on the screen, select `Use image size`.

- **Size**

This field is active only if you choose `Fixed size` in the **Scaling** list. Enter the size of the snapshot in the form `w h` (width height) and press **Enter**.

- **Alignment**

Options are:

- Auto
- Right
- Left
- Center

Title Page

Note This option is supported only in PDF and RTF reports.

Abstract Tab

Abstract Text

Enter an optional abstract for the report.

Style

- **Bold**
Make the text bold.
- **Italic**
Make the text italic.
- **Underline**
Underline the text.
- **Strikethrough**
Strike through the text.
- **Retain spaces and carriage returns**
Cause the formatting of the text to appear in the generated report as it was entered.
- **Show text as syntax-highlighted M-code**
Show the text as syntax-highlighted M-code.
- **Color**
Choose the color of the text from the menu.

Legal Notice Tab

Legal Notice Text

Enter an optional legal notice for the report.

Style

- **Bold**

Make the text bold.

- **Italic**

Make the text italic.

- **Underline**

Underline the text.

- **Strikethrough**

Strike through the text.

- **Retain spaces and carriage returns**

Cause the formatting of the text to appear in the generated report as it was entered.

- **Show text as syntax-highlighted M-code**

Show the text as syntax-highlighted M-code.

- **Color**

Choose the color of the text from the menu.

Insert Anything into Report?

Yes. Title page.

File Name

rptgen.cfr_titlepage

Variable Table

Purpose Inserts a table that displays all variables in the MATLAB workspace

Description This component inserts a table that displays all variables in the MATLAB workspace.

Tip Find all the variables in the MATLAB workspace by typing `whos` at the command line.

Source Workspace

Read variables from

Select one of the following options:

- Base workspace. Read variables from the MATLAB workspace.
- MAT-file. Read variables from a binary file with a `.mat` extension. Use the `%<VariableName>` notation. For more details about this notation, see `%<VariableName>` Notation on the Text component reference page.

Table Title

- **Table title**

Select one of the following options for the table title:

- Automatic (Variables from MATLAB workspace). The table title is the name of a variable from the MATLAB workspace.
- Custom. Enter the title in this field if you want a custom title for the variable table.

- **Table Columns**

Select information to be contained in table columns. Options include:

- Variable dimensions (M×N). Include the size of the variable.
- Variable memory bytes. Include the number of bytes of memory occupied by the variable.
- Variable class. Include the variable class.

- Variable value. Include the value of the variable.

Note Large variable arrays are collapsed to [MxN CLASS]. For example, if you have a 300-by-200 double array, it appears in the report as [300x200 DOUBLE].

Example

The following is an example of a variable table that includes size, memory bytes, and value information in the table columns.

Name	Size	Bytes	Value
aCell	1x2	238	{ [1 2 3 4] Speed (kph) }
aNumber	1x1	8	1
aString	1x11	22	Speed (kph)
aStructure	1x1	302	[struct w/ fields. Inputs, Outputs]
aVector	1x4	32	[1 2 3 4]

Insert Anything into Report?

Yes. Table.

File Name

rptgen.cm1_whos

While Loop

Purpose Iteratively executes its child components while a specified condition is true

Description This component iteratively executes its child components while a specified condition is true. The While Loop component must have at least one child component; the purpose of this component is to run its children several times. If it does not have any children, this component does not add anything to the report.

Tip Limit the number of repetitions to prevent infinite loops.

Logic Properties

- **Continue looping if this expression is true**

Enter a string to be evaluated in this field. This string should be a valid MATLAB expression that evaluates to 1 or 0 (true or false) in the workspace. For example, if $a = 1$, $b = 2$, and $c = 3$ in the MATLAB workspace, and you enter the following at the command line:

```
d=(a>b/c)
```

MATLAB returns the following:

```
d = 1
```

Because 1 is greater than b/c ($2/3$), this expression is true, and evaluates to 1.

- **Limit number of loops to**

Use this option to prevent infinite loops. Use the left and right arrows to increase or decrease the number of loops.

- **Initialize with this expression**

Enter a valid MATLAB expression to initialize the loop.

**Insert
Anything
into
Report?**

Yes, as long as it has a child component.

File Name

rptgen_lo.clo_while

While Loop

Examples

Use this list to find examples in the documentation.

Working with Components

“Working with Looping Components” on page 4-3

“Working with Property Table Components” on page 4-9

“Working with Summary Table Components” on page 4-21

Customizing Components

“Fetching Securities Data and Displaying It in a Table” on page 5-22

“Displaying Securities Data in Two Tables” on page 5-27

Customizing Stylesheets

“Numbering Pages in a Report” on page 6-27

“Adding Graphics to Headers in PDF Reports” on page 6-28

“Changing Font Size, Page Orientation, and Paper Type of a Generated Report” on page 6-33

“Editing Font Size as a Derived Value in XML” on page 6-36

A

- attribute sets
 - editing 6-18
- attributes
 - editing 6-18
- Axes Loop 8-2
- Axes Snapshot 8-4

C

- cells
 - stylesheet 6-22
 - stylesheet cell groups 6-21
- Chapter/Subsection 8-8
- child components
 - option, creating custom components 5-9
- Comment 8-10
- components
 - about 4-2
 - creating
 - examples 5-22
 - customizing
 - examples 5-22
 - Handle Graphics components 7-4
 - Logic and Flow components 7-6
 - looping components
 - about 4-3
 - MATLAB components 7-7
- converting English strings to other languages 3-9
- converting XML documents
 - using Report Explorer 3-11
 - using the command line 3-13
- converting XML documents to different format 3-11
- custom components
 - about 5-2
 - building 5-13
 - category name 5-8
 - child components 5-9

- creating 5-3
- creating properties 5-9
- defining 5-6
 - required data 5-6
- description 5-8
- display name 5-7
- editing M-files
 - modifying properties display 5-19
 - specifying additional component properties 5-19
- modifying existing components 5-13
- rebuilding 5-13
- removing 5-13
- setting display options 5-7
- specifying behavior 5-15
- specifying component string 5-18
- specifying file location 5-6
- specifying report output format and content 5-16

D

- data items
 - about 6-3
 - categories 6-10
 - deleting 6-20
 - editing 6-10
 - boolean and enumerated values 6-16
 - simple vs. advanced edit mode 6-14
 - strings 6-16
 - title placement properties 6-18
 - TOC properties 6-17
 - varpair values 6-19
 - XML expressions 6-17
 - working with 6-15
- default
 - content for headers and footers 6-22
- display name of stylesheet 6-3
- DocBook SGML report 3-5

E

Empty Component 8-12
Evaluate MATLAB Expression 8-13

F

Figure Loop 8-15
Figure Snapshot 8-17
flow object tree (XML) 3-5
font size
 changing
 example 6-33
 modifying by editing XML 6-36
footers
 adding graphics to 6-25
 content 6-21
 content properties 6-26
 specifying content of 6-22
 templates 6-24
For Loop 8-21

G

graphics
 inserting in header 6-28
Graphics Object Loop 8-23

H

Handle Graphics
 components 7-4
 creating figures 4-3
 figure property table 4-9
 HandleVisibility 4-4
 invisible figures 4-3
 tags 4-4
Handle Graphics Name 8-26
Handle Graphics Parameter 8-29
Handle Graphics Property Table 8-31
Handle Graphics Summary Table 8-35

headers

 adding graphics to 6-25
 example 6-28
 content 6-21
 content properties 6-26
 specifying content of 6-22
 templates 6-24
HTML report
 stylesheets 3-5
HTML reports
 list of editable HTML styles 6-13

I

Image 8-39
images
 inserting in header 6-28
Import File 8-42
Insert Variable 8-45

L

legacy report templates
 working with 3-17
Link 8-47
List 8-49
localization 3-9
log files
 creating 3-15
Logic and Flow components 7-6
Logical Else 8-53
Logical Elseif 8-54
Logical If 8-55
Logical Then 8-57
looping components
 about 4-3

M

M-code
 generating from report templates 3-16

- MATLAB components 7-7
- MATLAB Property Table 8-58
- MATLAB Report Explorer
 - Options pane 2-5
 - Outline pane 2-4
 - Properties pane 2-6
- MATLAB Report Generator
 - creating reports in 2-1
 - generating reports in 2-41
 - interaction with MATLAB 1-4
 - overview 1-2
 - report templates
 - adding components to 2-8
 - setting report options in 2-3
 - workflow 1-3
- MATLAB/Toolbox Version Number 8-61
- Microsoft Word report 3-5

N

- Nest Setup File 8-62
- numbering pages
 - example 6-27

P

- pages
 - numbering
 - example 6-27
- pagination styles
 - example 6-28
- paper size
 - customizing
 - example 6-33
- paper type
 - customizing
 - example 6-33
- Paragraph 8-64
- property
 - data type menu 5-22

- property name field 5-22
- property tables
 - adding and deleting columns and rows 4-19
 - creating custom tables 4-20
 - displaying or hiding table and cell
 - borders 4-18
 - editing table titles 4-16
 - entering text into table cells 4-16
 - modifying properties in a table 4-17
 - property name/property value pairs 4-12

R

- report formats 1-6
- report templates
 - autosaving on report generation 3-9
- reports
 - descriptions 3-10
 - generating 3-1
 - preferences 3-2
 - printing 3-8
 - setting output options 3-2
- rich text format 3-5
- RTF reports
 - list of editable RTF styles 6-12

S

- section head
 - changing format
 - example 6-33
- SGML report 3-5
- Simulink Report Generator
 - generating reports 3-1
- Stop Report Generation 8-66
- stylesheet
 - cells 6-22
- stylesheet cells
 - about 6-21
 - cell groups 6-21

- stylesheets 3-5
 - about 6-2
 - built-in vs. custom 6-2
 - creating 6-8
 - customizing 6-1
 - data items. *See* data items
 - deleting 6-9
 - editing 6-3 6-5
 - HTML 3-7
 - list of editable HTML styles 6-11
 - list of editable RTF styles 6-12
 - PDF 3-6
 - properties 6-2
 - RTF (DSSSL Print) and Word 3-8
 - saving 6-8
 - specifying name for customized
 - stylesheet 6-3
 - transform type 6-3
 - Web 3-7
- Summary Table components
 - working with 4-21

T

- Table 8-68
- templates
 - for stylesheets 6-22
- Text 8-71

- Time/Date Stamp 8-73
- Title Page 8-76
- transform type 6-3
- troubleshooting 3-19

V

- variable notation 8-71
- Variable Table 8-80

W

- Web stylesheet 3-5
- While Loop 8-82

X

- XML documents
 - converting to different format 3-11
 - converting to different formats
 - using Report Explorer 3-11
 - using the command line 3-13
- XML report 3-5
- XML source files
 - editing 3-14
- XSLT
 - definition 6-24