Simulink Design Documenter Tool

Text Formatting Guide

June 2017

McMaster Centre for Software Certification (McSCert)

1 Introduction

The purpose of the Simulink Design Documenter is to facilitate the production of useful Software Design Description documents for software systems developed with Simulink. Text DocBlocks[1] as well as external text files may be used to provide the content of the reports, however, it is possible to format and markup the text in order to add items such as newlines and links. This document explains some of the formatting behaviour as well as the available markup language made available by the Simulink Design Documenter for plain text content.

2 Formatting and Markup

In principle, the input text for the report from Text type DocBlocks or from external text files can be written without concern for the formatting which will be outlined here, however, it may be useful to be aware of the features which can be used as they may be useful in preparing the report. The behaviour and tags are explained in the following sections. The markup elements/tags are highlighted red in the provided code snippets, while elements for the user to change with text of their choice are highlighted blue.

2.1 Whitespaces

Whitespace such as tabs and consecutive spaces appear in the output report as a single space. Note, this appears to be a limitation of Simulink Report Generator and may be changed by MathWorks in the future.

2.2 Newlines

Newlines in Text DocBlocks and external text files appear differently in the generated report than one might expect. A single newline is ignored, and consecutive newlines beyond the first will act as newlines in the generated document. The rationale for this is that single newlines are needed to break up long lines into paragraphs, in order to make the text file itself readable. If a single newline corresponded to a newline in the report, then the text file would likely be unreadable as it would have too much on a single line (i.e. it would have a whole paragraph on a single line). Therefore, to include one newline in the report, the text must have two newlines in a row. Similarly, to include an empty line in the report, the text must have three newlines in a row.

2.3 Comments

Lines of text may be commented out if desired. This is achieved by beginning the line with the percent character (%), which will result in that line being ignored during report generation. Unlike MATLAB code, the % cannot be added

[1] https://www.mathworks.com/help/simulink/slref/docblock.html

at any point in the line, just at the beginning. Otherwise, the percent character will be interpreted as regular plain text.

```
% Comment text that is not to be included in the report
Regular text of the report that uses % as a regular character.
```

2.4 Images

Images can be included in the report by including the `image` tags, where `imagePath` is the system path of the file to be included, and `imageTitle` will appear above the image as its title.

```
[image=imagePath]imageTitle[/image]
```

2.5 Links

The markup language supports linking to external sources, linking to other parts of the report, and anchors for those to be linked to.

2.5.1 External Links

To create a link to a source external to the document, such as a file or website, use the `url` tags shown below.

```
[url=myLinkID]linkText[/url]
```

In this example, only `linkText` will appear in the document and it can be clicked to open `myLinkID`, where `myLinkID` can be the path to a file (e.g. `C:\Users\File_Name`), or a website (e.g. `www.website.com`).

2.5.2 Internal Links

It is possible to create links to different parts of a report. This section explains the tags to do so.

Anchors
One can create a link anchor within the text, in order to be able to link to that location from other points in the document. To do so, use the `anchor` tags below.

```
[anchor=myLinkID]linkText[/anchor]
```

In the generated document, only the `linkText` will appear. The unique link identifier of the anchor is `myLinkID` and it is used to identify the point in the document where the anchor is placed. Note that there should be only one anchor with a given link ID in order to avoid ambiguity.

Links to Anchors
Then, a link can be created to link to an anchor point in the document. To do so, use the following `goto` tags.

```
[goto=myLinkID]linkText[/goto]
```

Again, only `linkText` will appear in the generated document, and clicking `linkText` will jump to the anchor with link ID of `myLinkID`. Note the use of the `goto` and `anchor` tags allows for linking across all parts of the report. Goto linking is not limited to the DocBlock or text file in which a link is created.

Links to Automatically Generated Anchors
The Simulink Design Documenter creates a number of link anchors automatically, and it is possible to link to them. For example, to link to an item in a table in the appendix, use `app:ItemName` in place of `myLinkID`, as shown below.

```
[goto=app:ItemName]linkText[/goto]
```

In this case `ItemName` should be replaced with the contents of the leftmost column of the row that this should link to. For example, if there is a table in the appendix in which the leftmost column is "Name", the top-left cell in the table will also be "Name" (as it is the header). Thus, the below tags will link to the first row in that table.

```
[goto=app:Name]linkText[/goto]
```

Multiple rows may have the same entry in the first column, in which case this method of linking will choose one of the options arbitrarily. This ambiguity can be handled by using a longform syntax in which `app:ItemName` should be replaced as shown below:

```
[goto=%<linkMap('Table_TableTitle_ItemName')>]linkText[/goto]
```

In this case `TableTitle` is the title of the table, and `ItemName` is the same as with the regular `app:ItemName` syntax.

2.6 Code

MATLAB code may be used in the blue fields of each *tag* presented in this document (i.e. image, url, goto, and anchor tags). If the appropriate code was `my code`, then it cana be run as MATLAB code by entering it as follows:

```
%<my code>
```

An example of using this MATLAB code is shown in the last code snippet given in Section 2.5.2, where `%<linkMap('Table_TableTitle_ItemName')>` may be written in place of `app:ItemName`.