

MadCap Software

What's New Guide

Flare 12



Copyright 2016 MadCap Software. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of MadCap Software.

MadCap Software
7777 Fay Avenue
La Jolla, California 92037
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1 Introduction	7
Summary of Major New Features—March 2016	8
Summary of Additional New Features—March 2016	9
Summary of New Features—May 2016	12
CHAPTER 2 Multilingual Output	13
Multiple Languages in Targets	14
Language Skin Improvements	31
CHAPTER 3 Responsive Layout	37
Responsive Layout Window Versus Foundation Framework	38
The 12-Column Grid System	39
Layout and Media Query Integration	43
Responsive Layout Styles	46
Creating a Responsive Layout	50
CHAPTER 4 Snippet Enhancements	69
Snippet Groups	70
Pinning Snippets	72
Snippet Conditions on Individual Snippets	76
Snippet Variables	87
CHAPTER 5 Stylesheet Enhancements	93
Allow Local and Master Stylesheets	94
Redesigned Display of Properties	109



Multiple Medium View	115
Add Selector Button Results in More Options	122
Media Queries	160
Multi-Select and Delete	173
Conditions	177
Tooltips Show Inheritance	178
Units of Measurement—New Options	180
CHAPTER 6 Tracked Changes	183
Change Bars	184
Tracked Changes in PDF Output	190
Tracked Changes in Word Output	194
CHAPTER 7 Additional New Features—March 2016	199
64-bit Support	201
Build Log Save	202
Byte Order Mark	203
Condition Enhancements	204
Drop-Down and Expanding Text Enhancements	213
Equation Editor	219
Glossary Enhancements	224
Help—Live or Local	238
Indexes—Sorting	242
Link to Styles	245
Macros—Assign Shortcuts in Dialog	247
Master Page Enhancements	251
Output—Choose Specific Browser or Viewer	256
Page Break—Independent of HTML Element	257
Report—Word Count	263
Skins—Runtime	265
Smart Quotes Per Language	278

Source Control Enhancements	279
Tables—Advanced Sorting	289
Variable Enhancements	295
Word Enhancements	302
XML Editor Enhancements	312
CHAPTER 8 New Features—May 2016	315
Linking to Lingo Projects	316
Sync to Lingo Project Updates	323
APPENDIX PDFs	325

CHAPTER 1

Introduction

This version of Flare contains several new features and enhancements.

For more information about each feature discussed in this manual, open the online Help and refer to the "What's New in this Version" topic. Links are provided in each new feature description, taking you to topics that contain additional information and steps.

This chapter discusses the following:

Summary of Major New Features—March 2016	8
Summary of Additional New Features—March 2016	9
Summary of New Features—May 2016	12



Summary of Major New Features—March 2016

Following are the *major* new features introduced in this version of Flare.

MAJOR NEW FEATURES		
Feature	What's New?	For More Information
Multilingual Output	<ul style="list-style-type: none">» Generate multilingual output from one target» Language skin improvements	See "Multilingual Output" on page 13.
Responsive Layout	<ul style="list-style-type: none">» Responsive Layout window pane» Create responsive content quickly» Different settings for web, tablet, and mobile	See "Responsive Layout" on page 37.
Snippet Enhancements	<ul style="list-style-type: none">» Snippet groups» Pinning snippets» Snippet conditions on individual snippets» Snippet variables	See "Snippet Enhancements" on page 69.
Stylesheet Enhancements	<ul style="list-style-type: none">» Allow local and master stylesheets» Redesigned Stylesheet Editor (Advanced view)» Multiple medium view» More options to create selectors» Create media queries» Conditions on styles	See "Stylesheet Enhancements" on page 93.
Tracked Changes	<ul style="list-style-type: none">» Change bars» Preserve tracked changes in PDF and Word outputs	See "Tracked Changes" on page 183.

Summary of Additional New Features—March 2016

Following are several *additional* new features and enhancements introduced in this version of Flare.

ADDITIONAL NEW FEATURES		
Feature	What's New?	For More Information
64-Bit Support	<ul style="list-style-type: none">» Flare is now 64-bit» Better access to RAM	See "64-bit Support" on page 201.
Build Log Save	<ul style="list-style-type: none">» Choose location where build logs are saved	See "Build Log Save" on page 202.
Byte Order Mark	<ul style="list-style-type: none">» Can enable/disable from Flare, instead of system registry	See "Byte Order Mark" on page 203.
Conditions	<ul style="list-style-type: none">» Conditions on glossary terms» Conditions on styles» Tracked changes to conditions	See "Condition Enhancements" on page 204.
Drop-Down and Expanding Text Enhancements	<ul style="list-style-type: none">» Simplified process	See "Drop-Down and Expanding Text Enhancements" on page 213.
Equation Editor	<ul style="list-style-type: none">» Add MathML and LaTeX equations from third-party editors	See "Equation Editor" on page 219.
Glossaries	<ul style="list-style-type: none">» Case-sensitivity and variants» Conditions on glossary terms» Select term for glossary term links» Sorting» Variables in terms and definitions» Other enhancements	See "Glossary Enhancements" on page 224.
Help—Live or Local	<ul style="list-style-type: none">» View Flare's online Help either externally (live) or locally	See "Help—Live or Local" on page 238.

ADDITIONAL NEW FEATURES		
Feature	What's New?	For More Information
Indexes	» Sorting enhanced (applies to both first- and second-level terms)	See "Indexes—Sorting" on page 242.
Link to Styles	» Point to content based on a style when creating a link (e.g., cross-reference, hyperlink)	See "Link to Styles" on page 245.
Macros	» Assign shortcuts in the Manage Macros dialog	See "Macros—Assign Shortcuts in Dialog" on page 247.
Master Pages	» Associating master pages with stylesheets using Properties dialog » Associating master pages with topics now easier	See "Master Page Enhancements" on page 251.
Output—Viewing	» Choose specific browser or viewer from Project ribbon or Target Editor when viewing output	See "Output—Choose Specific Browser or Viewer" on page 256.
Page Breaks	» Insert page breaks independent of HTML elements	See "Page Break—Independent of HTML Element" on page 257.
Reports	» Word count	See "Report—Word Count" on page 263.
Skins	» Select different skins at runtime	See "Skins—Runtime" on page 265.
Smart Quotes	» Different smart quotes per language	See "Smart Quotes Per Language" on page 278.

ADDITIONAL NEW FEATURES		
Feature	What's New?	For More Information
Source Control	<ul style="list-style-type: none"> » Publish output to source control » Offline mode » Bind detection » Other enhancements 	See "Source Control Enhancements" on page 279.
Tables	<ul style="list-style-type: none"> » Advanced sorting 	See "Tables—Advanced Sorting" on page 289.
Variables	<ul style="list-style-type: none"> » Insert variables in more places throughout the interface » Snippet variables » System variables use language in Flare » Other enhancements 	See "Variable Enhancements" on page 295.
Word Output	<ul style="list-style-type: none"> » Tracked changes » Variables in indexes » Table rows avoid page breaks » Other enhancements 	See "Word Enhancements" on page 302.
XML Editor	<ul style="list-style-type: none"> » Layout views for tablet and mobile media queries » Apply a style ID to content 	See "XML Editor Enhancements" on page 312.

Summary of New Features—May 2016

Following are new features and enhancements introduced in this version of Flare.

Feature	What's New?	For More Information
Link to Lingo Projects	» Creation of multilingual projects by linking to Lingo projects directly	See "Linking to Lingo Projects" on page 316.
Sync to Lingo Project Updates	» Sync to linked Lingo projects to retrieve necessary updates	See "Sync to Lingo Project Updates" on page 323.

CHAPTER 2

Multilingual Output

Flare's multilingual functionality has been improved to make it easier to create and build multilingual output. You can now seamlessly create multilingual outputs for each of Flare's output types. Additionally, language skins are now stored and maintained within the Flare interface, which makes it easier to manage their translation.


This chapter discusses the following:

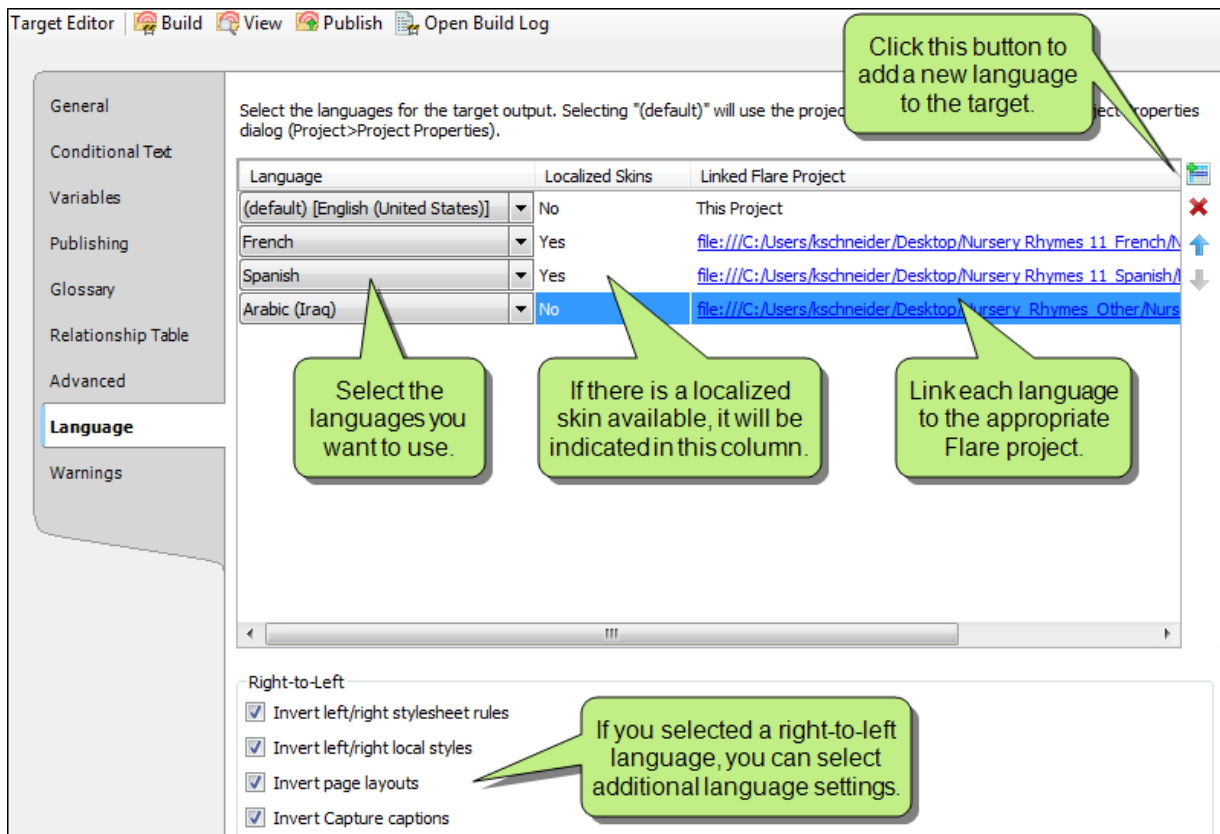
Multiple Languages in Targets	14
Language Skin Improvements	31



Multiple Languages in Targets

You can now select the languages that you want to include in the output and rearrange them to set the order in which they will appear in the final output. You can link each language to its corresponding translated Flare project (e.g., a French project, a Spanish project, and an English project).

To do this, open the target you want to build and select the **Language** tab. From the **Language** drop-down, select the language you want to use for the current, or master, project. Then click  to add another row to the grid. Click the link (...) in the **Linked Flare Project** column to select the project you want to link to the master project. When you do this, the Language drop-down automatically updates to reflect the default language set in the linked project (you can change this setting if you want). If default translations are available for that language's skin, it will be noted in the **Localized Skin** column. Continue adding rows for each language that you want to add.



The screenshot shows the Target Editor interface with the **Language** tab selected. The left sidebar contains a list of tabs: General, Conditional Text, Variables, Publishing, Glossary, Relationship Table, Advanced, **Language**, and Warnings. The main area displays a table for selecting languages for the target output. Above the table, a note states: "Select the languages for the target output. Selecting '(default)' will use the project's default language (Project > Project Properties)." The table has three columns: Language, Localized Skins, and Linked Flare Project. It contains four rows: (default) [English (United States)], French, Spanish, and Arabic (Iraq). A callout points to the '+' button in the top right corner of the table, stating: "Click this button to add a new language to the target." Another callout points to the 'Language' column, stating: "Select the languages you want to use." A third callout points to the 'Localized Skins' column, stating: "If there is a localized skin available, it will be indicated in this column." A fourth callout points to the 'Linked Flare Project' column, stating: "Link each language to the appropriate Flare project." Below the table, there is a section titled "Right-to-Left" with four checked options: "Invert left/right stylesheet rules", "Invert left/right local styles", "Invert page layouts", and "Invert Capture captions". A callout points to this section, stating: "If you selected a right-to-left language, you can select additional language settings."

Language	Localized Skins	Linked Flare Project
(default) [English (United States)]	No	This Project
French	Yes	file:///C:/Users/kschneider/Desktop/Nursery Rhymes 11 French/
Spanish	Yes	file:///C:/Users/kschneider/Desktop/Nursery Rhymes 11 Spanish/
Arabic (Iraq)	No	file:///C:/Users/kschneider/Desktop/Nursery Rhymes Other/Nursery Rhymes 11 Arabic (Iraq)/

Right-to-Left

- ☒ Invert left/right stylesheet rules
- ☒ Invert left/right local styles
- ☒ Invert page layouts
- ☒ Invert Capture captions

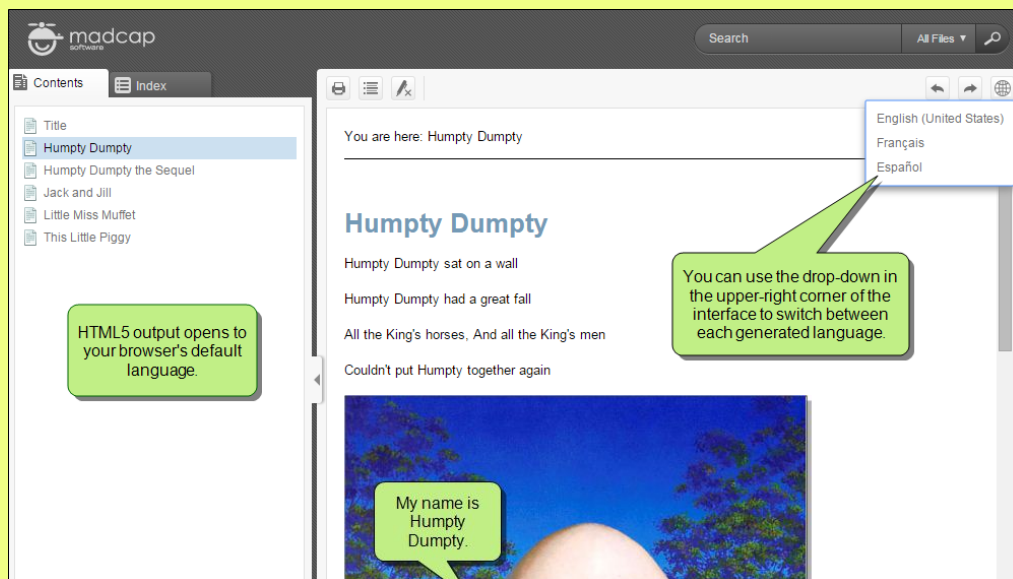
When you are ready to build your output, open the target you want to generate and click **Build**. Flare will generate the linked target from each project in the order you specified. The final files will be generated differently depending on the output you are generating.

HTML5 AND WEBHELP

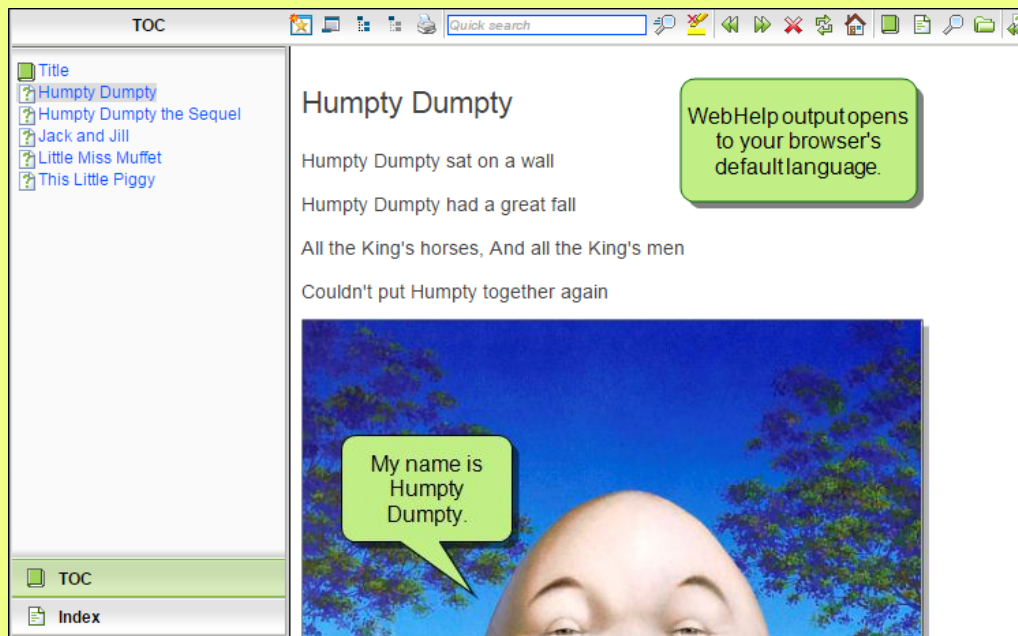
Flare will create a subfolder in your Output folder for each language. When you view the output, Flare will display the output for your browser's default language. In HTML5 outputs, you can use a drop-down menu to switch between each generated language.

EXAMPLE

HTML5 output will look like this.



WebHelp output will look like this.

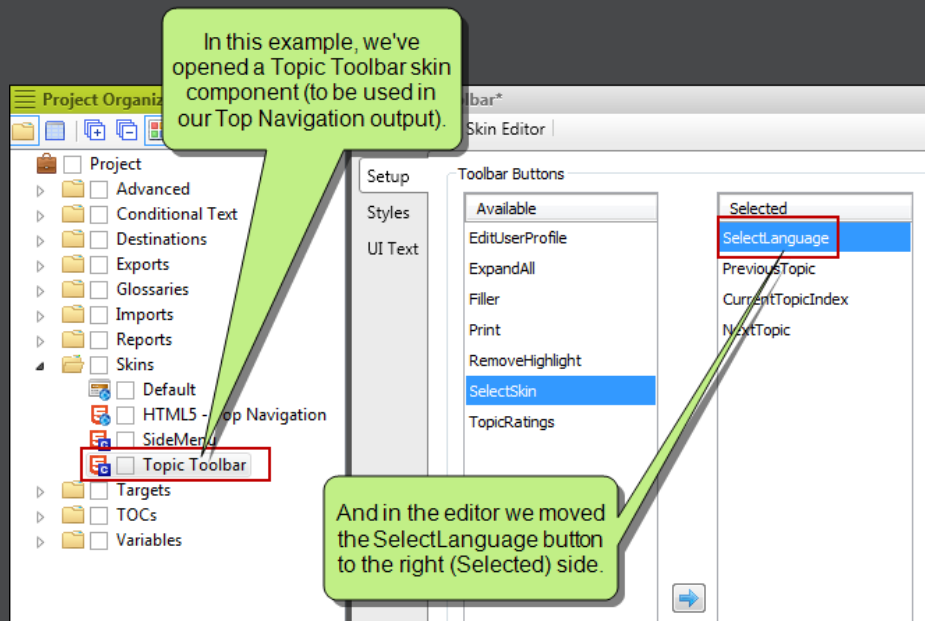


Note: Be sure that the file names are the same in your master project and each of your linked projects. This is especially important for HTML5 targets, so you can switch between languages using the Select Language button in the output.



Note: If you are using an HTML5 output, be sure to enable the SelectLanguage button in the skin for your master project and each linked project. This will allow you to change languages using a drop-down.

1. If you are creating Top Navigation output, you should have a Topic Toolbar skin component in your project. Open the skin component, and select the **Setup** tab.
If instead you are creating Tripane output, open the regular skin and select the **Toolbar** tab.
2. If necessary, move the **SelectLanguage** button from the **Available** section to the **Selected** section.



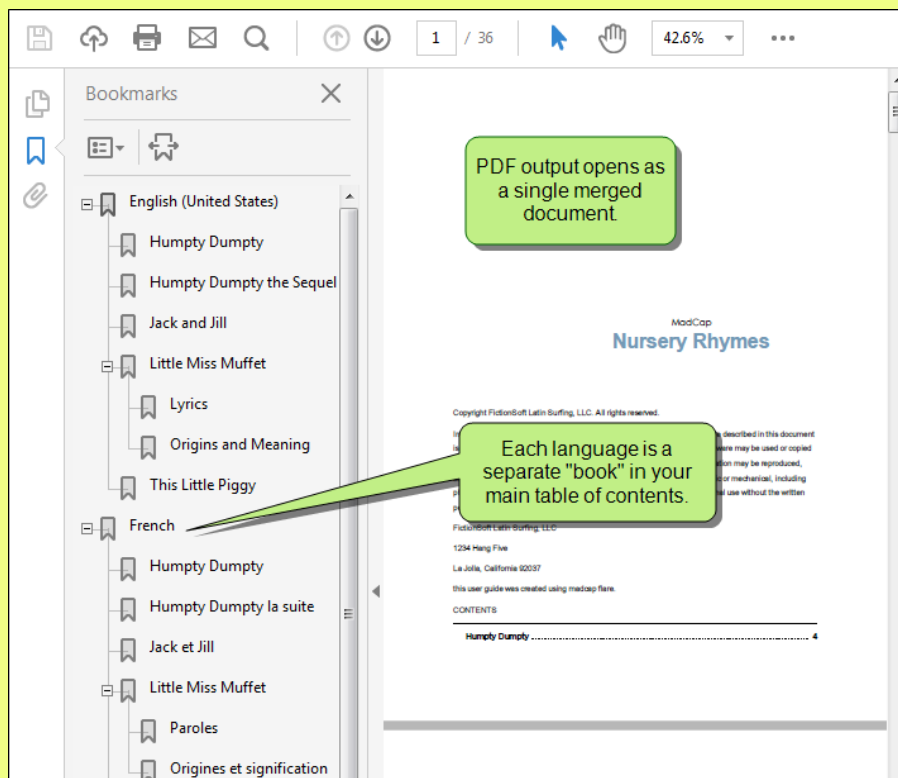
After you add the SelectLanguage button to your Topic Toolbar, you also need to be sure the toolbar appears in your master page. If the topic toolbar does not appear in each of your master pages, you will not be able to switch languages from those page types.

PDF

The output is merged (or “stitched”) into a single document, with each language appearing in the order you specified. You will see a single file in your Output folder.

EXAMPLE

PDF output will look like this.





Note: Generating a PDF output using a multilingual target uses the same process as PDF stitching. However, creating a multilingual target automates the process and allows you to work from multiple Flare projects at once. Additionally, because you do not need to generate each PDF separately every time the document is updated, generating output from a multilingual PDF target saves considerable time.



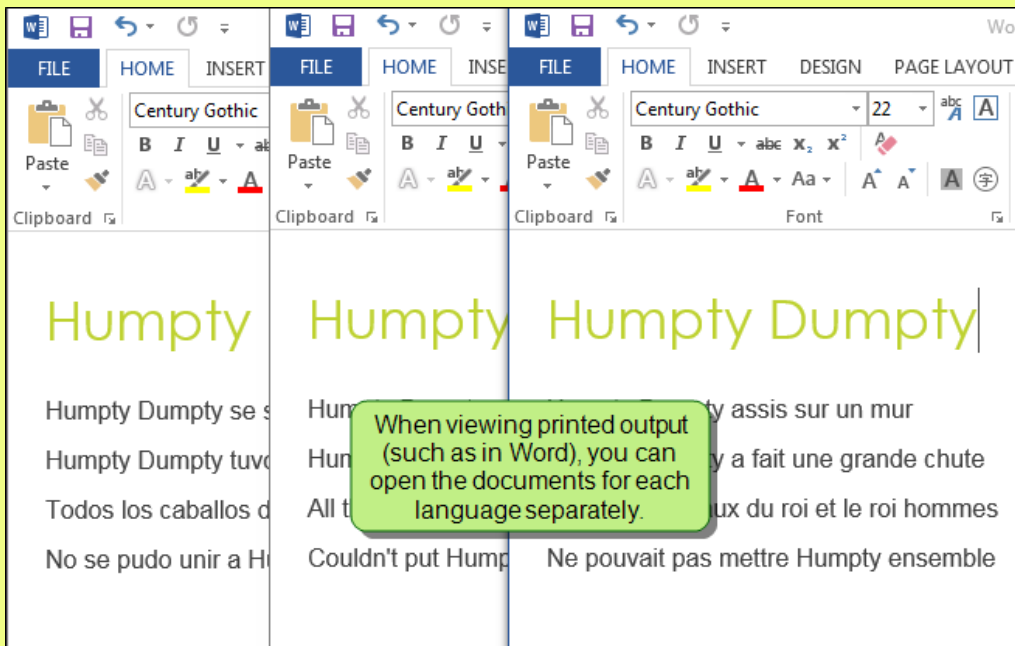
Note: You can access individual PDFs for each language in the Temporary folder in your Output folder.

OTHER OUTPUT TYPES

Flare will create subfolders for each language in your Output folder. You can merge printed output types manually (e.g., in Word) if you need to create a single file.

EXAMPLE

Word output will look like this.

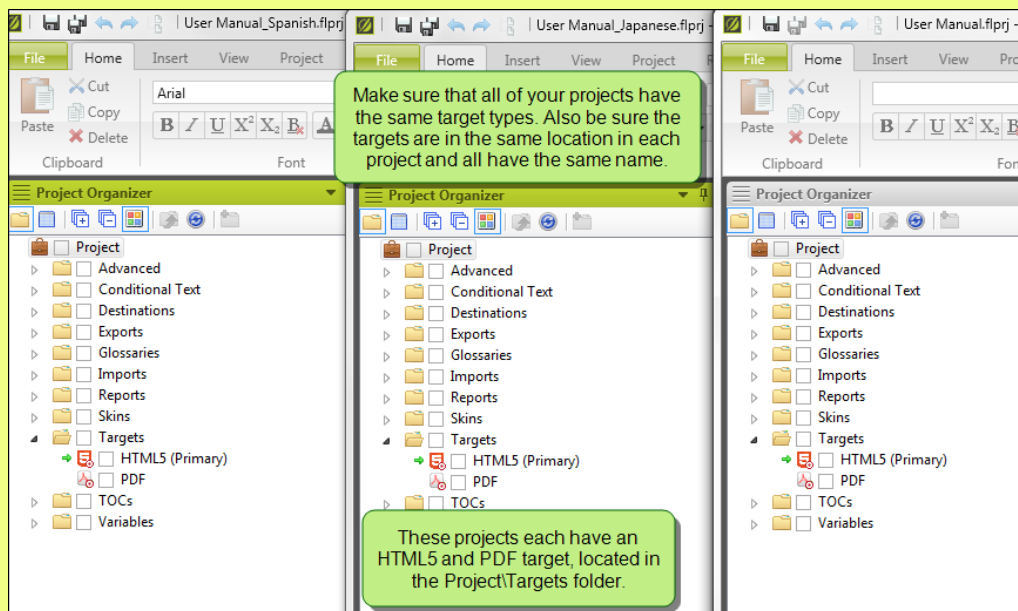


Note: If you are building Eclipse Help, you will need to open your Output folder to open your desired language output.


EXAMPLE—LINKING TO FLARE PROJECTS

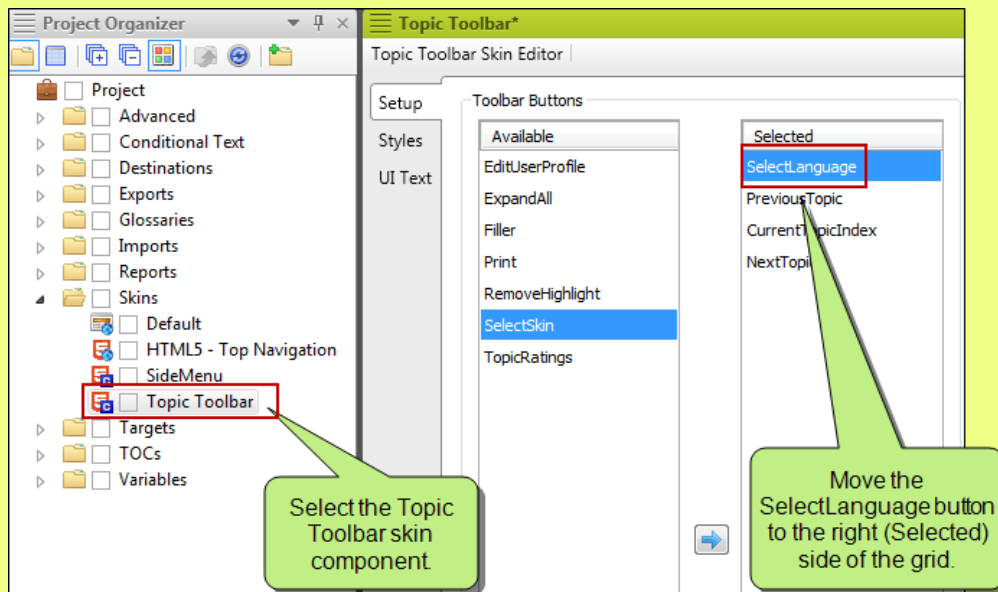
Let's say you want to create HTML5 and PDF outputs in US English, Spanish, and Japanese. You have already worked with a translator to translate your documents, so you have three different Flare projects: one for each language.


Before you can create your multilingual output in Flare, make sure that each Flare project has both an HTML5 and a PDF target available. This will allow your master project to build from the targets in the other two linked projects. Open the Project Organizer in each project, and then open the Targets folder. If you already have the targets you need, you do not need to create any new targets. If you do not have both targets in all three projects, you can create the missing targets from the Add File dialog.

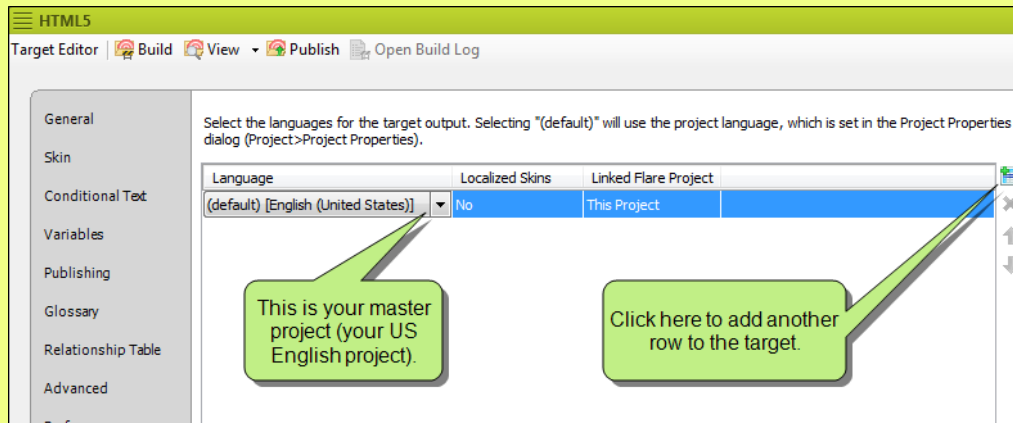


Because you want to build multilingual HTML5 output, you need to make sure that each of your projects has the SelectLanguage button in its HTML5 toolbar. If the button isn't there, users won't be able to switch from one language to another in your output.

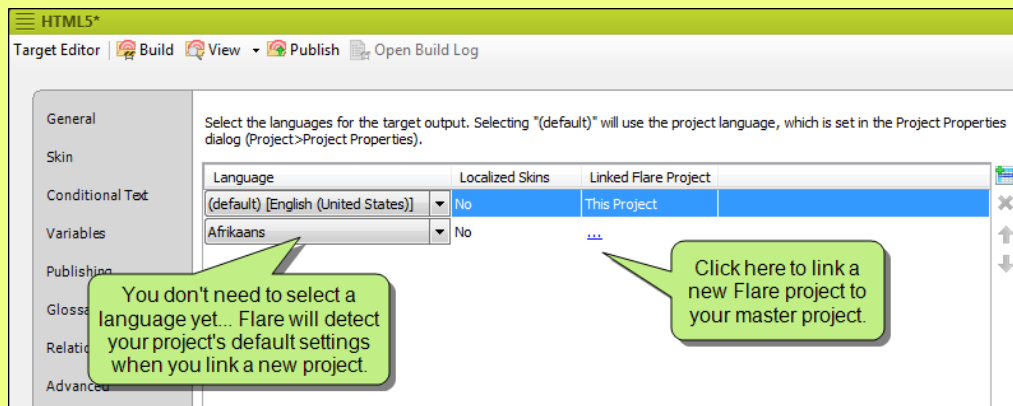
Open the Project Organizer. In the **Skins** folder, open the Topic Toolbar skin component. On the **Setup** tab, select the **SelectLanguage** button, click  to add it to your skin, and then save your work.



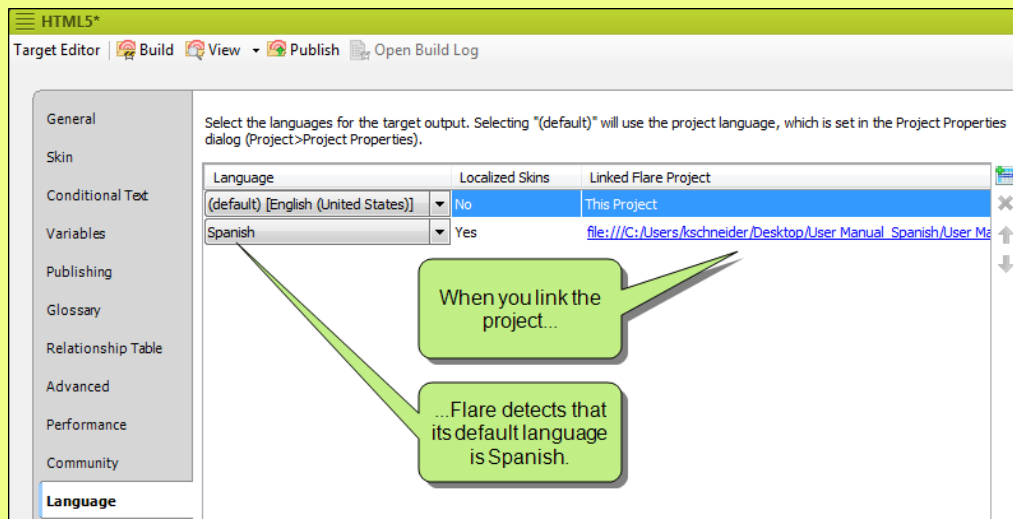
Now you can prepare your multilingual target. Open your master project (in this case, the US English project), and open the **Targets** folder. Open the HTML5 target. In the Target Editor, select the **Language** tab. This is where you will link your other two projects. Click  to add a new row.



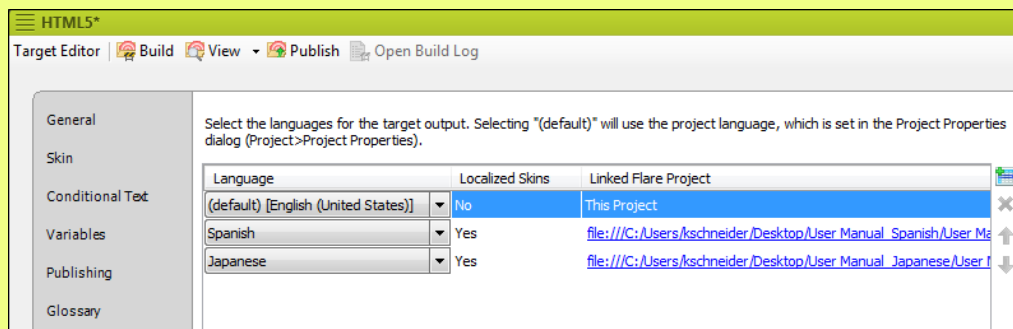
Click the link in the **Linked Flare Project** column.




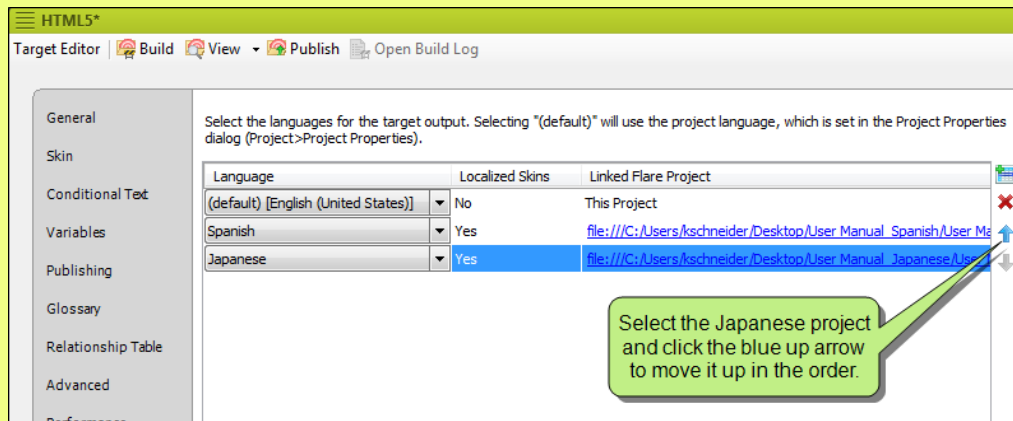
In the dialog that opens, navigate to your Spanish project and click **Open**. This links the Spanish project to the US English project. The Language drop-down automatically updates when Flare detects the default Spanish language settings in your linked project.



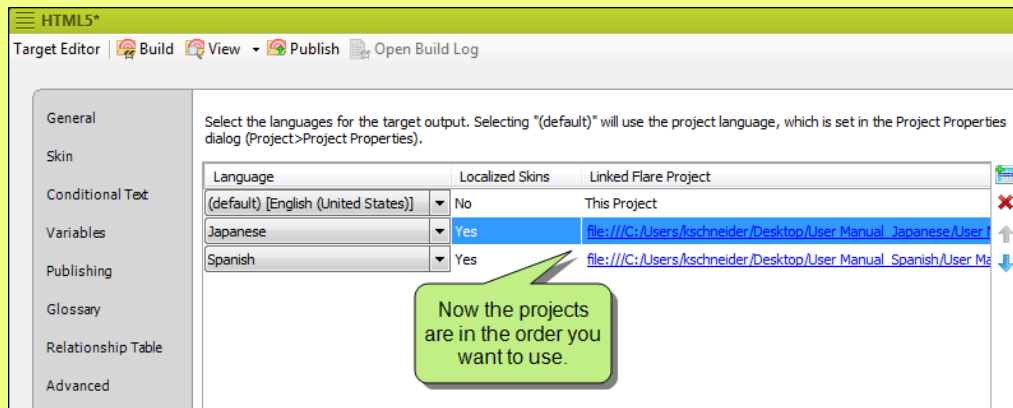
Add your Japanese project in the same way.



You want the Japanese project to appear second in the Help system, so you select the Japanese project and click  to rearrange the projects.

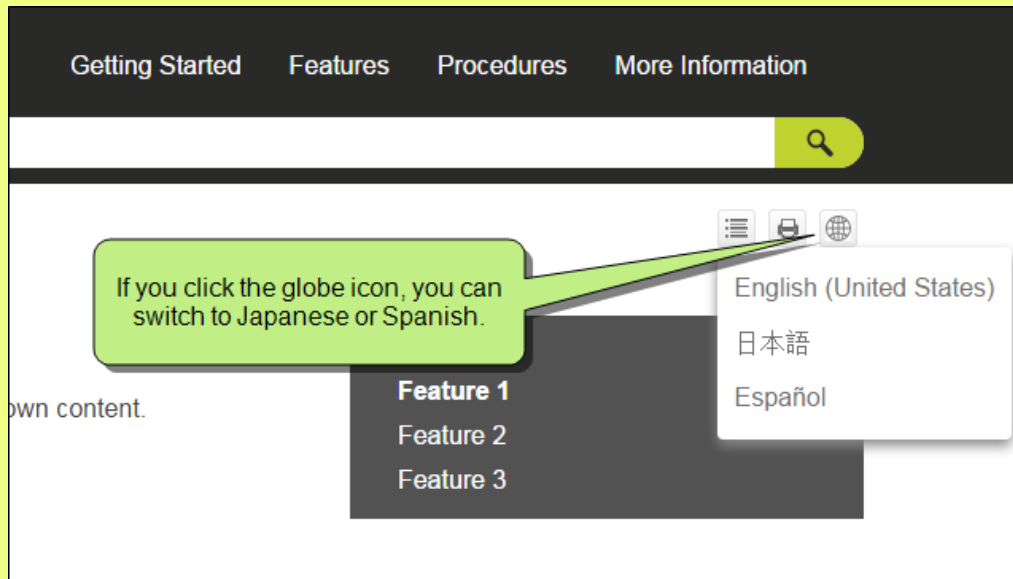


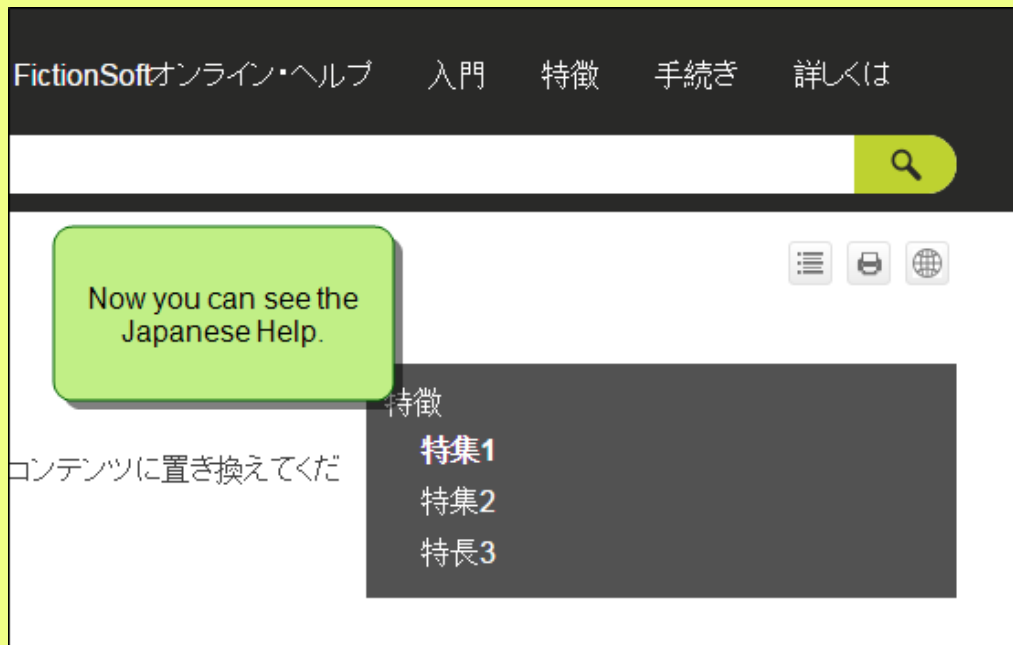
Now the Japanese project is listed before the Spanish project.



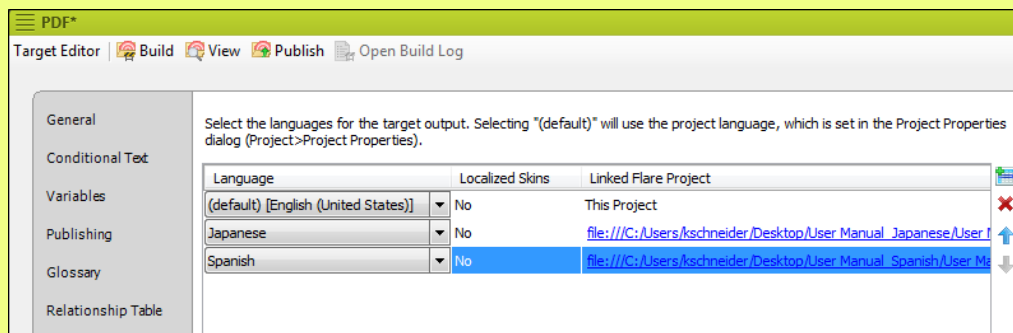
You are ready to build, so save your work and build the target. When you open the HTML5 output, it defaults to the US English Help because that is your browser's default setting.

But remember the SelectLanguage button you added to your Topic Toolbar skin component? You can use that button to change the language setting using a drop-down menu.





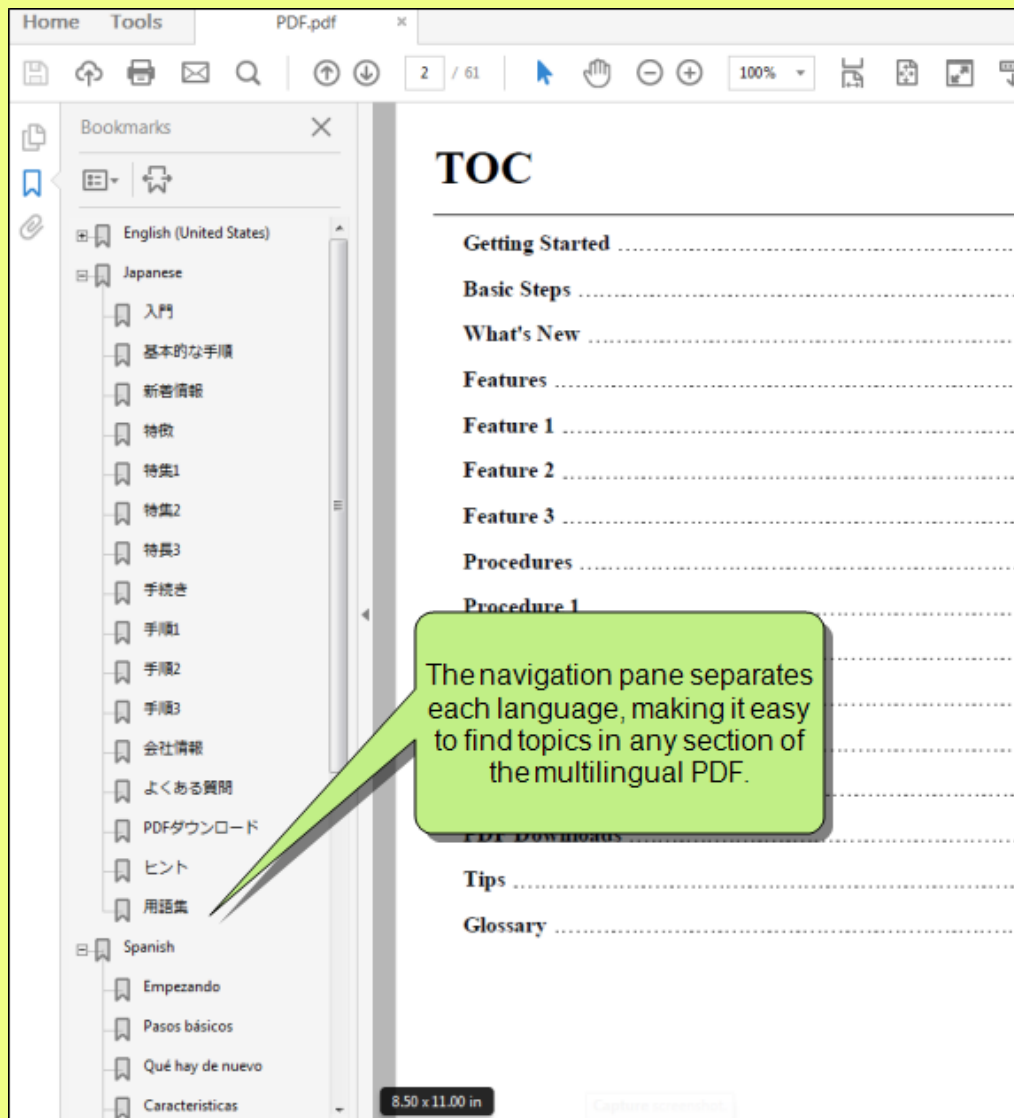
Next you create the PDF target. Open the PDF target in the Target Editor and create it in the same way you did the HTML5 output. Since you want the Japanese output to appear first, you add that project first, then you add the Spanish project.



Save the PDF target and build it. When you look in the Output folder, you only see one document. This is because Flare stitched all three documents together into a single PDF.



When you open the PDF, you see all three languages in the document's table of contents. Each language has its own title page so you can find it quickly.





Note: You can use a tool like MadCap Lingo (or another computer-aided translation tool) to translate your Flare content. Although you can set the language for your project in Flare, this does not mean that Flare automatically outputs translated content.



Note: You must link each language to a Flare or Lingo project before you can close the Target Editor. Similarly, if you make changes to your linked projects and their links are no longer valid when you build the project, you will see a warning message before the build starts and you will be unable to build.



Note: Because Flare generates output from the linked targets in each of your project files, each linked project must have its own target file for the output you want to build (e.g., if you are creating PDF output, each linked project needs its own PDF target). If you do not have a needed target file in one of your linked projects, you will see a warning message before the build starts and you will be unable to build. Target files must be in the same relative location in each project.



Note: If you have created a new language skin for a language, Flare will use it when you build the project. The language skin must reside in the project that uses that language.

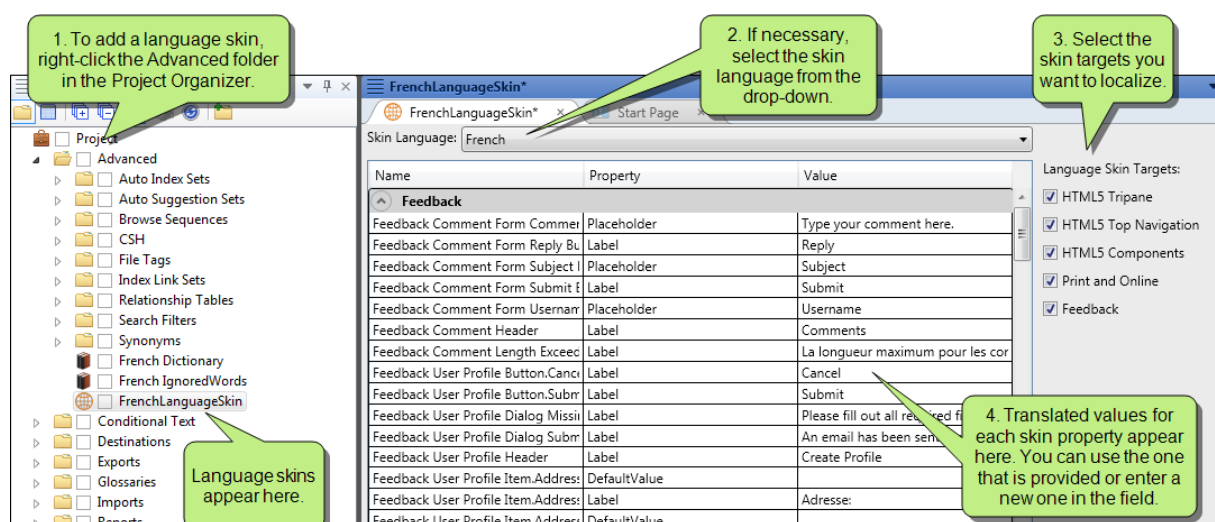


Note: If you are using right-to-left language settings in a linked project, you must enable these settings in your master project target. Language settings in the master project target control those for each of the linked projects, regardless of the settings in your linked projects.

Language Skin Improvements

If you write your documentation in a language other than US English, it is now easier to translate interface text using a language skin. Previously, language skins were created outside of the project and resided in your AppData\Roaming folder (specifically, C:\Users\[username]\AppData\Roaming\MadCap Software\Flare\Language Skins). In Flare 12, language skins are created in the Add File dialog and reside in the Advanced folder in the Project Organizer. This makes them easier to access and edit.

You can edit language skins using the redesigned Language Skin Editor. In this editor, you can select the skin type(s) you want to translate (e.g., Tripane, Top Navigation) and translate the value associated with each UI text string. Values are grouped by the skin type(s) in which they appear, so you do not need to translate the value more than once. Many values are prepopulated, and you can also enter custom values.



HOW TO IMPORT A LEGACY LANGUAGE SKIN

If you created language skins in a previous version of Flare, you can continue to use them in Flare 12. Flare will read your legacy language skins from the AppData\Roaming folder, or you can move them to the Advanced folder so you can edit them in Flare.

You can also use your legacy language skins as templates when creating a new language skin from the Add File dialog.



Note: Moving your legacy language skins to the Advanced folder also allows you to edit newer skin types, such as the HTML5 Top Navigation skin.

HOW TO CREATE A NEW LANGUAGE SKIN

If you want to create a new language skin, you can do this from the Add File dialog.



1. Do one of the following, depending on the part of the user interface you are using:
 - » **Ribbon** Select the **Project** ribbon. In the **Content** section select **New>Advanced>Add Language Skin**.
 - » **Right-Click** In the Project Organizer, right-click on the **Advanced** folder and from the context menu select **Add Language Skin**.

The Add File dialog opens.

2. In the **File Type** field at the top, make sure **Language Skin** is selected.
3. In the **Source** area select one of the following:
 - » **New from template** This lets you choose either the factory template file or one of your own customized template files as a starting point. The new file will take on all of the settings contained in the template. If you want to use the factory template provided by Flare, expand the **Factory Templates** folder and click on a template file. If you want to use your own customized template file, expand the appropriate folder and click on a file. For more information about templates, see the online Help.



Note: If you have existing language skins in your AppData folder, you can use them as templates for new language skins. Select **New as template**, then select an existing language skin from the **Language Skins** folder.

- » **New from existing** This lets you choose an existing file of the same type as a starting point for your new file. As with template files, your new file will take on all of the settings contained in the file you select. To use this option, click the browse button , use the Open File dialog to find a file, and double-click it.
4. (Optional) If you want to place the file into a subfolder that you previously created in the Project Organizer, in the **Folder** field click  and select the subfolder. Otherwise, keep the default location.

5. From the **Language** drop-down, select the language you want to use for the language skin.



Note: The **Language** drop-down defaults to the current project language.

6. In the **File Name** field, type a new name for the language skin.
7. Click **Add**. The language skin is added to the Advanced folder in the Project Organizer. The Language Skin Editor opens to the right, with the new language skin shown. Depending on the language you selected when creating the language skin, the skin may include default translations for some UI text strings.



Important: Creating multiple language skins for a single language is not recommended. Flare applies the first language skin it finds for a language—alphabetically—in the Advanced folder of the Project Organizer. As a best practice, you should maintain all of your translations for a language in a single language skin to avoid confusion and incorrect translations.



Important: Once you enable a language skin target, you should not disable it unless you do not plan to use it in your project. Disabling the language skin target removes any translated strings from your skin file. If you do not need to see a target in the editor, you should instead collapse the target.



Note: You can also translate interface values in your stylesheet and—for HTML5 skins—on the UI Text tab of the Skin Editor (see). When you set these values in multiple places, they are prioritized as follows:

1. Stylesheet
2. HTML5 skin
3. Language skin
4. Legacy language skin, still in your AppData\Roaming folder

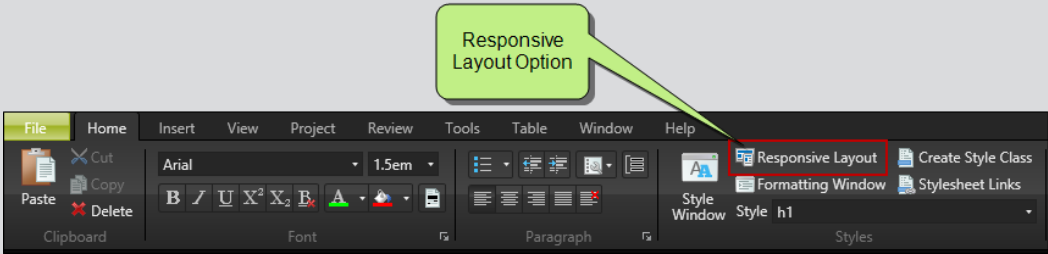
As a best practice, you should try to translate each interface value in a single location to prevent conflicts.



Note: If you use the Language Skin Editor to edit an HTML5 skin value, your changes will be reflected in the HTML5 skin file (on the UI Text tab of the Skin Editor). However, changes made in the Skin Editor are not reflected in the Language Skin Editor.

Responsive Layout

A new window pane called "Responsive Layout" has been added to Flare to assist you with creating responsive content. This window pane can be opened from the Styles section of the Home ribbon in Flare.



It displays by default on the right side of the interface.

This lets you create a "one-row grid" to hold content. The benefit of putting your content into such a grid is that it allows it to shift and be responsive to screens of different sizes.

This chapter discusses the following:

- Responsive Layout Window Versus Foundation Framework38
- The 12-Column Grid System39
- Layout and Media Query Integration43
- Responsive Layout Styles46
- Creating a Responsive Layout50



Responsive Layout Window Versus Foundation Framework

Flare's responsive layout framework is similar to the system created by Zurb Foundation (for more information, see <http://foundation.zurb.com/>). The Foundation framework was introduced in Flare 11, making an appearance in the Top Navigation project templates.

So what is the difference between the Foundation framework and the new system used in this version of Flare?

- » **Simplified** First of all, you will notice that there are only a few options in Flare's Responsive Layout window pane. Basically you are dealing with div style classes that comprise rows and cells, with width, offset (left margin), and gutter settings for your cells. The Foundation framework is much more extensive than just these few settings. Flare has scaled these back to the few tasks that most authors are likely to use.
- » **User Interface** The Foundation framework can be very useful and powerful, but in order to use it, you need to work in a topic's code (i.e., the Internal Text Editor). The Responsive Layout window pane lets you accomplish some of the same tasks as Foundation, but you do not need to use the code. You can use the Responsive Layout window pane in conjunction with the XML Editor. In other words, it's easier to use for those who are not familiar with HTML and the Foundation system.
- » **Supported in XML Editor** If you look at a topic using the Foundation framework (e.g., the Home page in Top Navigation project templates in Flare 11), you will notice that the sections using this system do not appear in the XML Editor the same way they appear in the output. That is because the Foundation framework is not fully supported in the XML Editor. On the other hand, the simplified system using Flare's Responsive Layout window pane is supported in the XML Editor. Therefore, what you see in the editor is much closer to what you will see when you generate the output.

This does not mean that you cannot continue to use Foundation in your Flare projects. It is still supported. But you will notice that Flare's Top Navigation project templates now use Flare's new system instead of Foundation.

The 12-Column Grid System

Both the Flare and Foundation systems are based on a 12-column grid system (i.e., a row has the potential of using 12 "columns" of content). But keep this very important point in mind:

Responsive content columns are not the same as the columns you are used to with tables. With responsive content, columns are purely used for measuring width (in percentages).

Flare considers the entire width of the screen displaying your output, and it divides this into 12 imaginary columns. The maximum width is 100% of the space, so 12 columns equals 100%. Half the width of the available screen space is 50%, or 6 imaginary columns. One-third of the width is 33.333%, or 4 columns. It is no more complicated than that. Assigning a number of columns in this type of layout is simply a shortcut; it is a quick way to tell Flare how wide a particular area should be.

When you create a new row using the Responsive Layout window pane, it contains a certain number of cells, and you can add more cells to the row as necessary. Then you assign a certain number of imaginary columns to each cell to set its width.

The screenshot shows the 'Responsive Layout' window in Flare. The main area displays a row with three cells: 'Cell 1' (orange), 'Cell 2' (light blue), and 'Cell 3' (yellow). A green callout bubble points to the row, stating: 'Here is a responsive layout row with three cells. We've added a background color to each cell to make it easier to see each one.' Another green callout bubble points to 'Cell 1', stating: 'Currently, the cursor is placed in Cell 1. This cell has a width of 50% (or 6 columns). The other two cells have a width of 25% (or 3 columns) each.' A third green callout bubble points to the 'Cell Width' field in the configuration panel, stating: 'The number of columns is set here. Because the cursor is in Cell 1, we see the number of columns assigned to it.' A fourth green callout bubble points to the 'width: 50%' property in the 'Cell Style' panel, stating: 'The percentage is shown here.'

The configuration panel on the right shows the following settings:

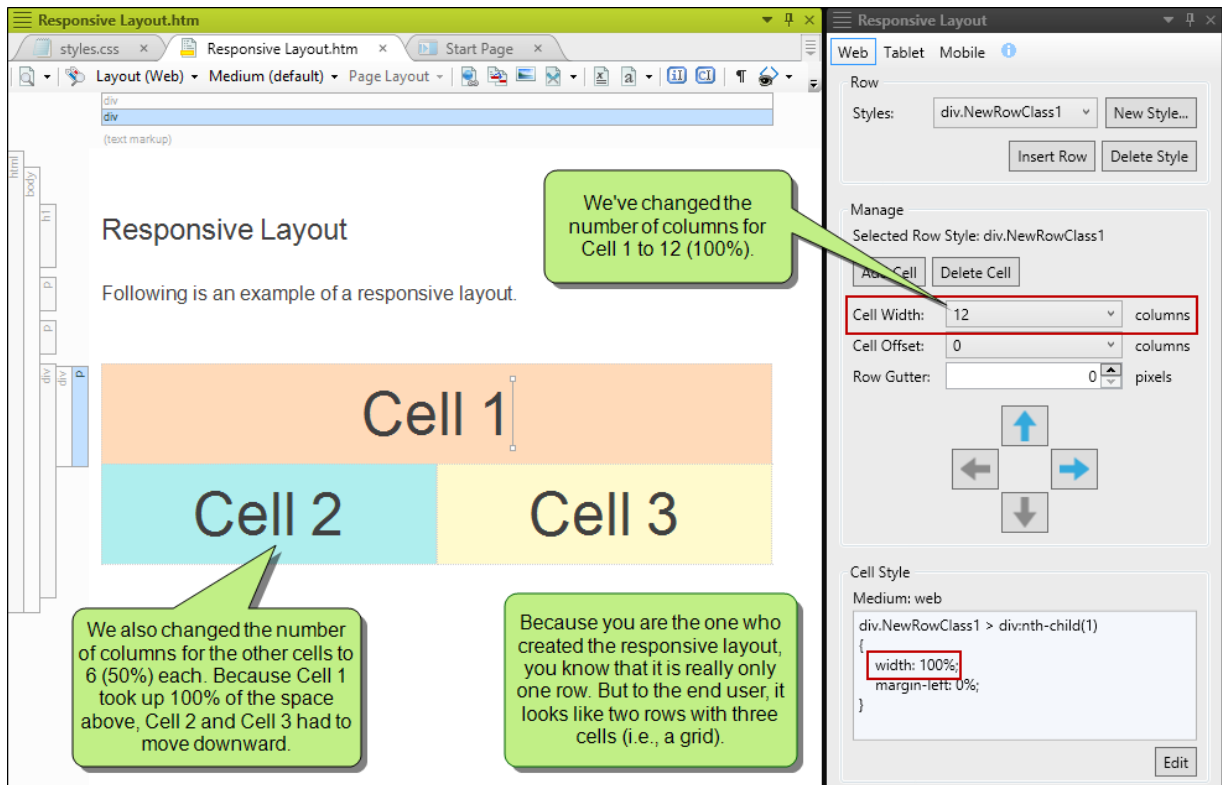
- Row: div.NewRowClass1
- Cell Width: 6 columns
- Cell Offset: 0 columns
- Row Gutter: 0 pixels
- Cell Style: Medium: web, div.NewRowClass1 > div:nth-child(1) { width: 50%; margin-left: 0%; }

Why is this called a "grid" if it has only one row? Well, it has to do with what the end user perceives when looking at the output. Although each responsive layout consists of only one row (i.e., all of the content is contained within one `<div>` tag), the widths that you assign to the various cells can cause the layout to look like multiple rows in the output. Therefore, it appears to the end user to be a grid.

The key to this is the fact that for any row, you can exceed 100%. This usually happens when you assign column widths to cells for tablet and mobile screens.

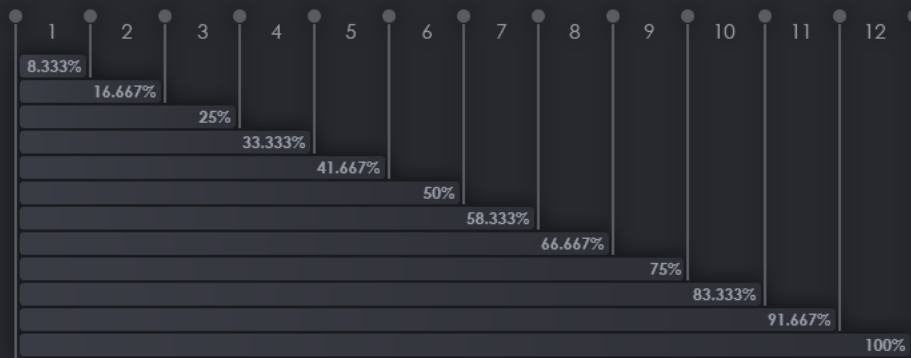
For your large web layouts, you usually want the total number of columns in a row to add up to 100%. However, you can use fewer columns if you want (e.g., you might end up using only 10 columns, or roughly 83% of the space). In fact, there might be times when you use more than 100%, even for the large web view.

But for smaller devices to show the same content, you are likely to exceed 100% for the entire row. When the total width exceeds 12 columns (100%), the next cell moves downward. That way the content can be reconfigured and will be easier to see on a small screen. To the end user, it might look like you're using extra rows to display the same content. But it's actually still just one row, with the extra content shifted downward.



You can also use the Offset field in the Responsive Layout window pane to give any cell a left margin percentage (by assigning a number of columns to it). As far as right margins are concerned, you can accomplish that by editing your stylesheet (see "Responsive Layout Styles" on page 46).

The following graphic illustrates how the 12-column grid system works. Three different layouts are shown with various column (cell width) settings. One layout also uses offset (left margin) settings.



Layout 1

Cell 1 (4 = 33.333%)

Cell 2 (4 = 33.333%)

Cell 3 (4 = 33.333%)

Layout 2

Cell 1 (6 = 50%)

Cell 2 (6 = 50%)

Cell 3 (12 = 100%)

Layout 3

Left Margin
(2 = 16.667%)

Cell 1 (8 = 66.667%)

Right Margin
(2 = 16.667%)

Left Margin
(2 = 16.667%)

Cell 2 (5 = 41.667%)

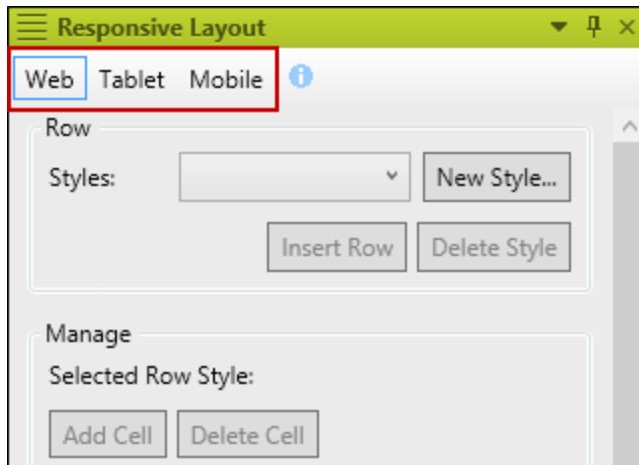
Cell 3 (3 = 25%)

Right Margin
(2 = 16.667%)

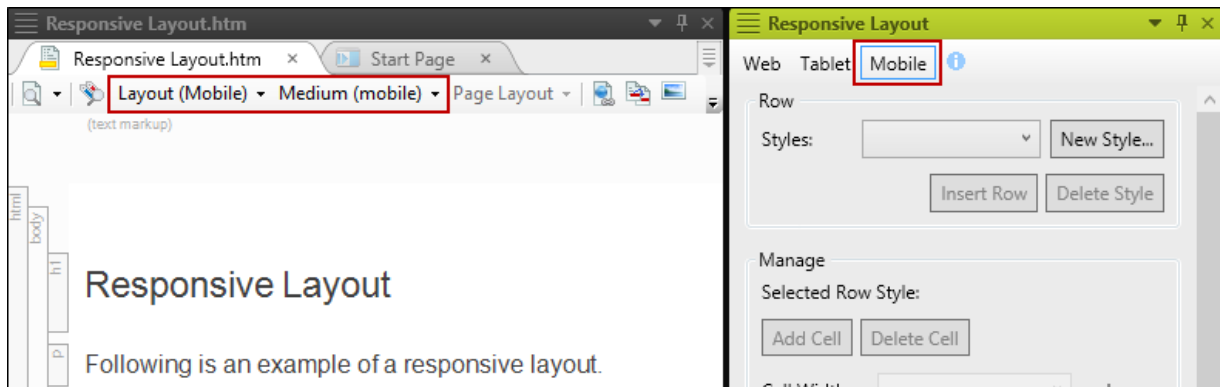
Left margin can be set in the Responsive Layout pane (Offset field), as well as in the stylesheet.
Right margin can be set in the stylesheet.

Layout and Media Query Integration

Just as you can specify skin settings for web, tablet, and mobile mediums, you can do the same with responsive layouts. You will see buttons for each medium at the top of the Responsive Layout window pane. Actually, these are more accurately referred to as "media queries," but they are sometimes called by both names.



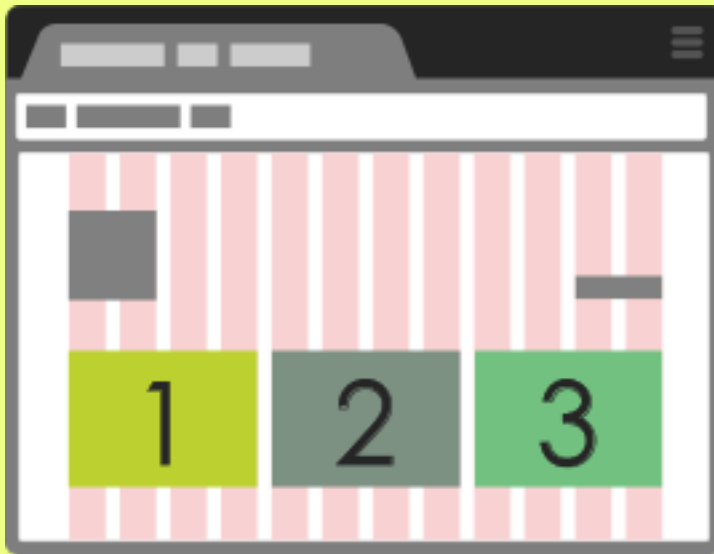
These are synchronized with the layout and medium settings in the XML Editor. So when you select a particular layout in the XML Editor, the appropriate medium is also selected in both the XML Editor and the Responsive Layout window pane. And if you choose a medium in the Responsive Layout window pane, the corresponding layout and medium are selected in the XML Editor.



So when a particular layout/medium is selected, the settings you apply will be used for that size of screen in the output.

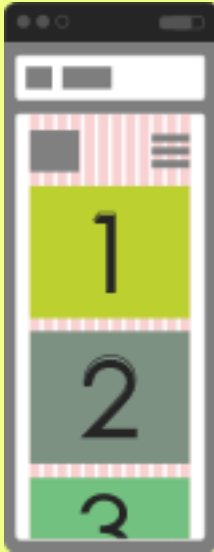
EXAMPLE

For a large screen, you might have a row consisting of three cells, with each cell set to use four columns (33.333% each).



The three cells each use an equal portion of the possible space when shown on a large screen that can handle that much content.

But this is too much for an iPhone. So for your Mobile layout, you tell Flare to use 12 columns in each of those three cells.

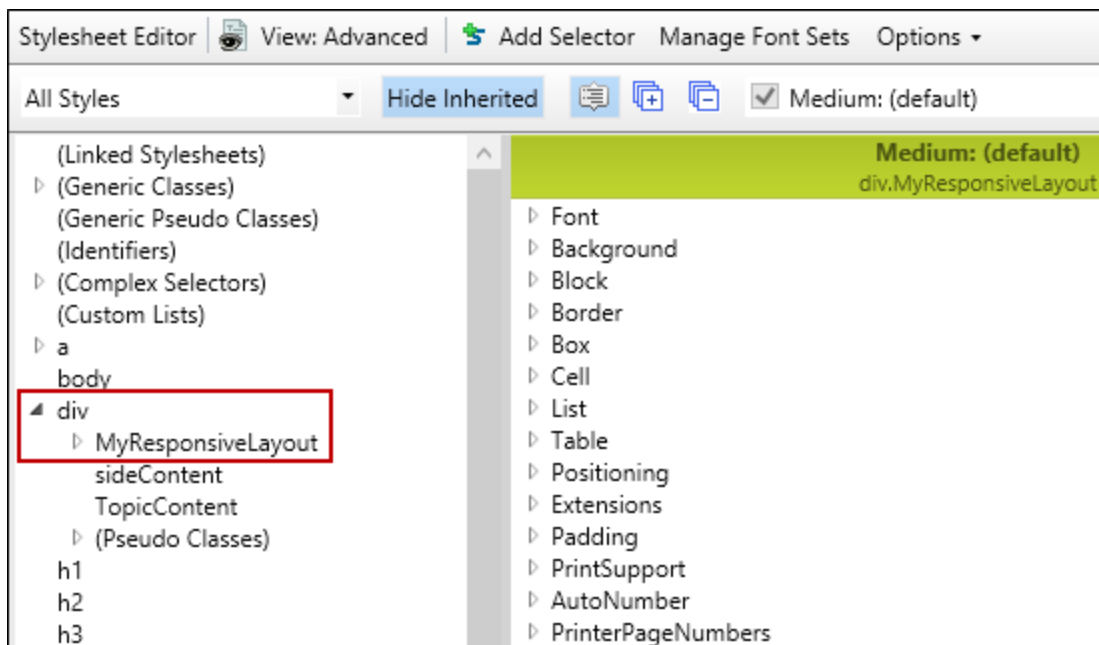


As the image above shows, the first cell took up the largest amount of space (12 columns, or 100%), so Flare had to shift the next cell downward. The second cell also used 100% of the space, so Flare had to move the third cell down even farther.

Responsive Layout Styles

If you've used the Foundation framework, you know that there are several styles involved. In the simplified Flare system, there are only a few styles to know about.

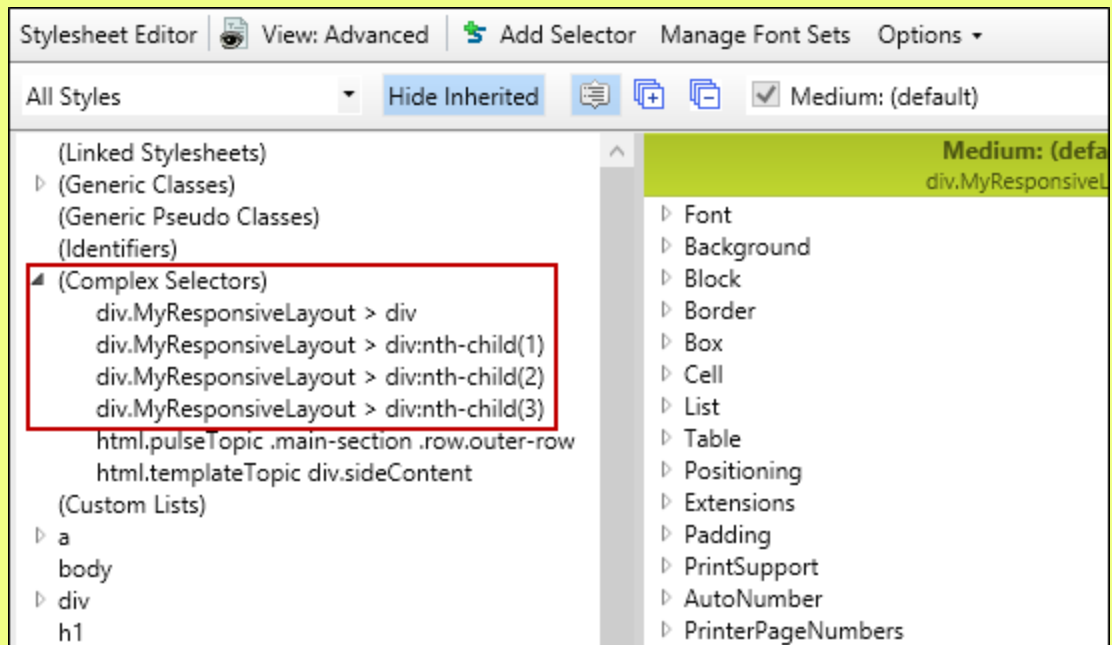
When you create a responsive layout in Flare, you are asked to create a new "row" style, or use an existing one. It's called a row style because it controls everything that is going on inside that one-row grid. However, in CSS it is actually a div style.



In fact, when you are all done adding cells to the row and assigning columns, you will also end up with multiple complex selectors in your stylesheet. These selectors consist of div styles, using the nth-child pseudo class for each cell.

EXAMPLE

In the following image, we have a responsive layout with three cells, so we have three complex selectors; the number at the end in parentheses represents the order of the cells.



Maybe you want to make formatting adjustments to the second cell in this row style. In that case, you would select **div.MyResponsiveLayout > div:nth-child(2)** and modify the properties for it on the right side of the editor.

When you click in a cell in the XML Editor, the Responsive Layout window pane will show the relevant complex selector and the definitions for it. You cannot edit this part of the Responsive Layout window pane, but you can copy and paste it elsewhere if necessary. If you click the **Edit** button below this section, the stylesheet opens and automatically chooses the appropriate selector so that you can edit its properties.

Responsive Layout

Web

Tablet

Mobile

Row

Styles:

div.MyResponsiveLayout

New Style...

Insert Row

Delete Style

Manage

Selected Row Style: div.MyResponsiveLayout

Add Cell

Delete Cell

Cell Width:

4

columns

Cell Offset:

0

columns

Row Gutter:

0

pixels

Cell Style

Medium: web

div.MyResponsiveLayout > div:nth-child(1)

{
width: 33.33333%;
margin-left: 0%;
}

Edit

Settings

☒ Auto-save Stylesheet



Tip: Depending on the type of content you add to your responsive layouts, certain additional styles and settings might be especially useful for you. If you create a grid where the content in each cell is exactly the same size, you're likely to have a much easier time configuring your layout. But if you have different types of content in your cells, or content that requires more space in some cells than others, you might need to use additional styles and properties in your stylesheet. For example, the "height" property can be quite useful to keep all of your cells looking the same. In the Stylesheet Editor, you would expand **(Complex Selectors)**, choose your main selector (e.g., if your layout is named "Grid," you would select **div.Grid > div**) and then complete the height or other settings to the right.

Also, embedded videos can be tricky within responsive layouts. That's why we created the following additional video-related styles in the StylesForHomePage stylesheets in our Top Navigation project templates: `..video`, `video-section`, and `.video-wrapper`. If you open the Home page topic in one of these Top Navigation templates, you will see that we used a combination of these styles and responsive layouts to display the videos. If you want to include videos in your responsive grids, you might consider creating additional styles like these, or you can just import these same styles from the template into your own stylesheet.

Creating a Responsive Layout

Following are steps showing how to create a layout in a topic that is responsive in web, tablet, and mobile views.

HOW TO CREATE A RESPONSIVE LAYOUT

1. Open the content file (e.g., topic, snippet).
2. Select the **Home** ribbon. In the **Styles** section, select **Responsive Layout**. The Responsive Layout window pane opens.
3. At the top of the Responsive Layout window pane, select the appropriate medium/media query: **Web**, **Tablet**, or **Mobile**.

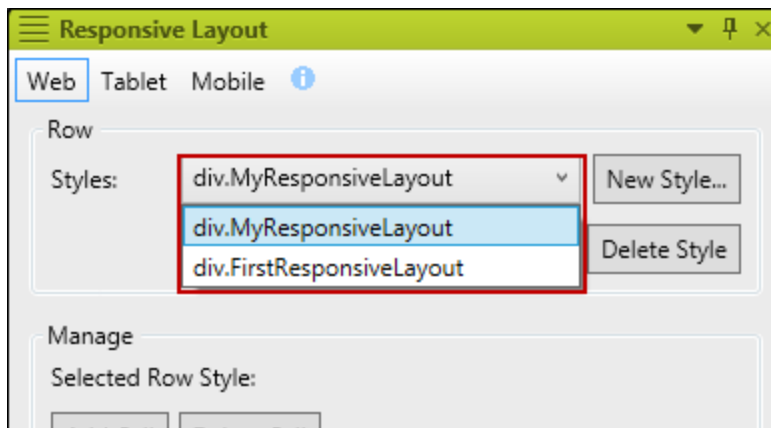
Alternatively, you can select the appropriate layout view in the top local toolbar of the XML Editor.

It is recommended that you start with Web to create a layout for large screens. After you finish those settings, you can move on to the Tablet and then finally Mobile.

4. In the content file, place your cursor where you want to create the layout. (The row will be added below the location of your cursor.)
5. In the Responsive Layout window pane, do one of the following.

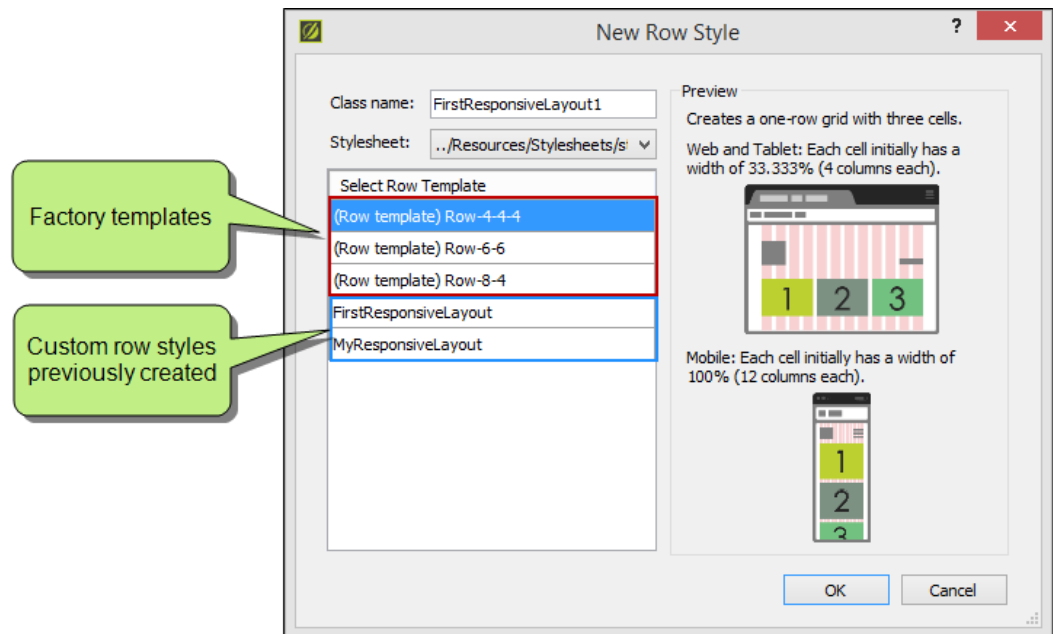
SELECT AN EXISTING STYLE

If you want to use the same row style for a responsive layout that you've created previously, click in the **Styles** drop-down and select it.



CREATE A NEW STYLE

- Click **New Style**. The New Row Style dialog opens.
- In the **Class name** field, enter a name for the new row style (or you can keep the default name).
- (Optional) In the **Stylesheet** drop-down, you can select a stylesheet where the new style will be added.
- On the left side of the dialog, select a template. This will list three factory templates provided by Flare. Also, any of your previously created row styles will also be listed in this dialog as available templates.



As you click on each factory template, a preview to the right describes how each one is configured. If you click on one of your previously created row styles, you will see some default text. This text was added as a comment for that new style. You can open your stylesheet and replace this text with whatever you want to describe your layout. This can be especially useful if you work with a team of authors who will be creating responsive layouts. For more information on adding comments to styles, see the online Help.

Selecting a template simply makes a copy of the style settings in that template and places them in your new style. You can then make changes to those styles when you choose different settings in the Responsive Layout window pane. So when you choose a template, pick one that is closest to the design that you plan on creating.

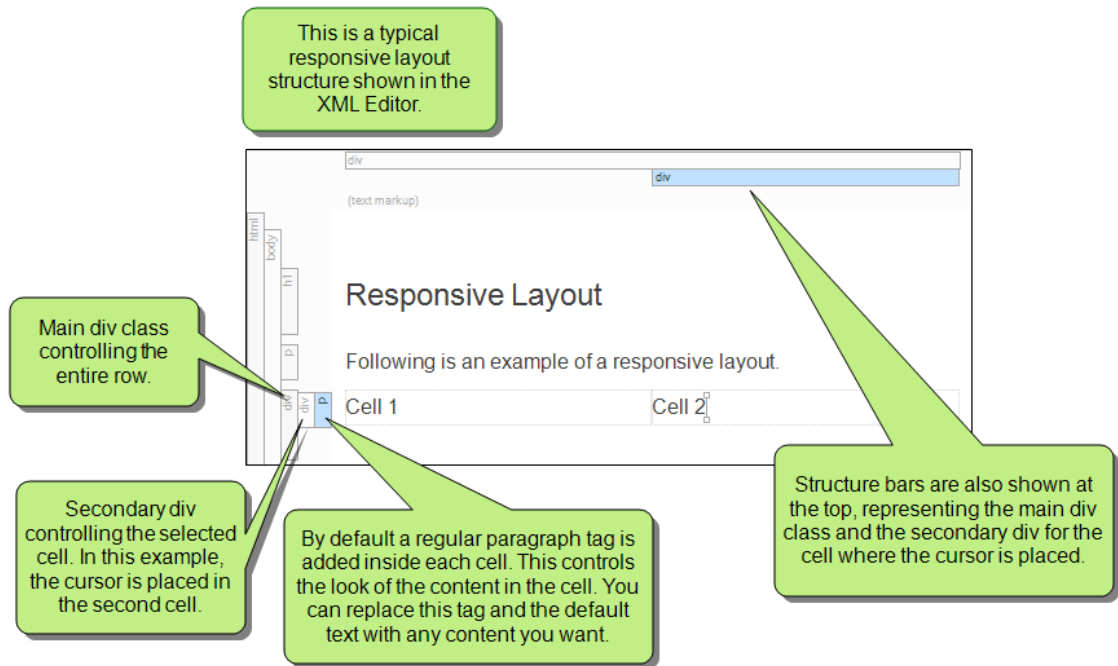
Following are the three factory templates:

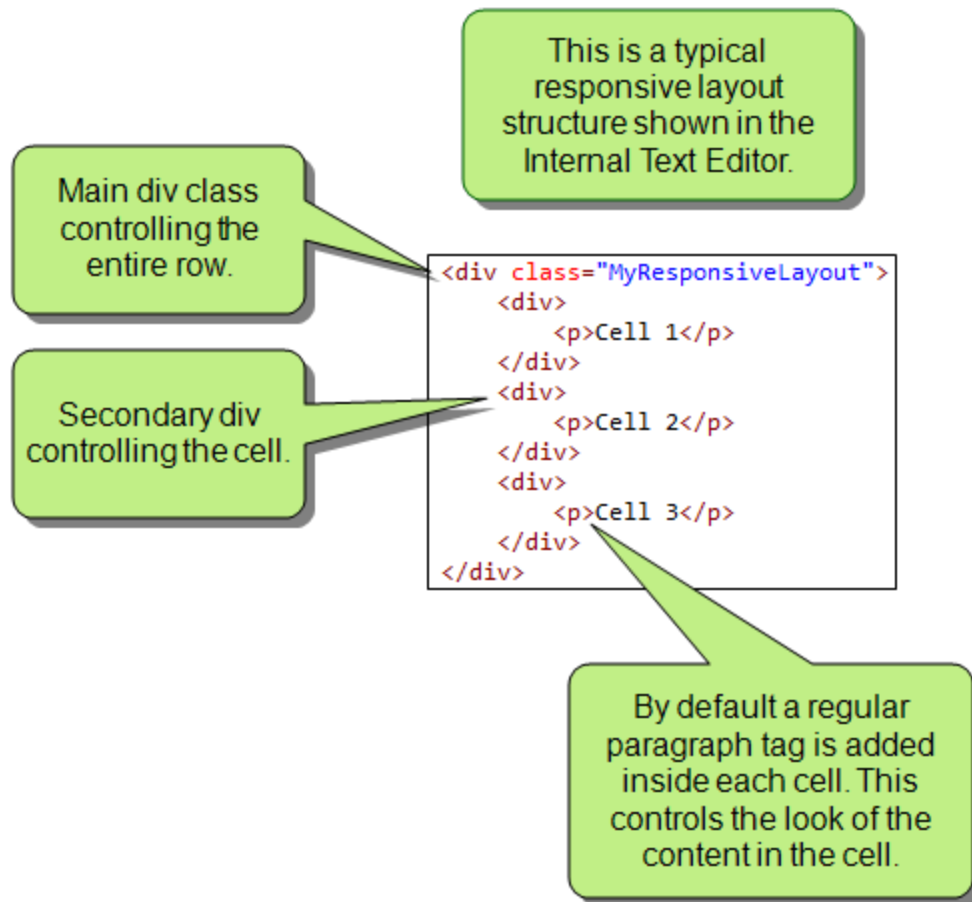
- » **Row-4-4-4** This template starts you out with three cells, each taking up the space of four columns (33.333%) for Web and Tablet layouts, and 12 columns (100%) for Mobile layouts.
 - » **Row-6-6** This template starts you out with two cells, each taking up the space of six columns (50%) for Web and Tablet layouts, and 12 columns (100%) for Mobile layouts.
 - » **Row-8-4** This template starts you out with two cells. For Web and Tablet layouts, the first cell takes up the space of eight columns (66.667%) and the second cell takes up the space of four columns (33.333%). For Mobile layouts, both cells take up the space of 12 columns (100%).
- e. Click **OK**. The Row Styles field in the Responsive Layout window pane is now populated with the new style. Also, your stylesheet will open behind the topic (if it isn't already open).



Note: If you no longer want to use an existing row style anywhere in your project, you can select it from the **Styles** drop-down and click **Delete Style**. This removes the appropriate styles from your stylesheet.

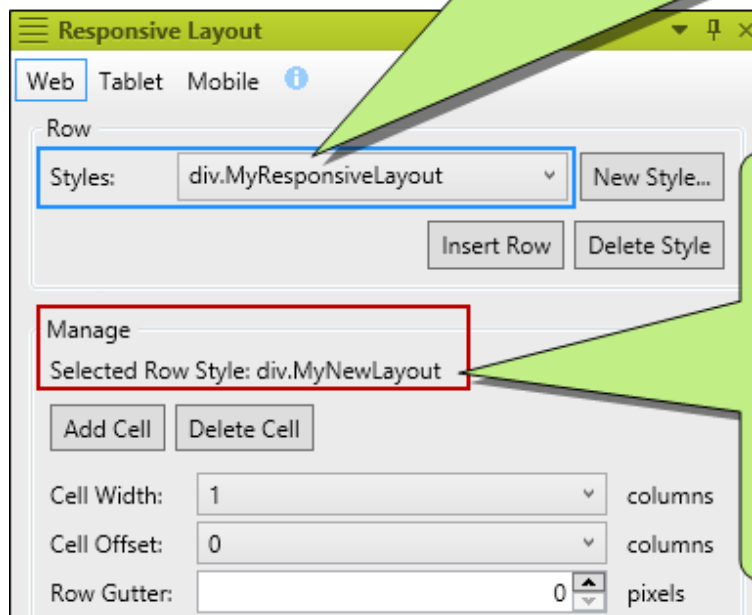
6. Click **Insert Row**. In the XML Editor, the row is inserted below your cursor location. A `<div>` tag—using the style name that you created or selected in the previous step—acts as the container for the row. Within that `<div>` tag there will be additional plain `<div>` tags, one for each cell.





7. In the XML Editor, click in the cells, remove the default text, and add your own content. If you want, you can save this task until after you configure your row using the options in the next step.
8. Click in a cell in the XML Editor. The row style for that layout is indicated at the top of the Manage section in the Responsive Layout window pane. So any changes you make will be applied to that style in your stylesheet.

In this example, a row style called "div.MyResponsiveLayout" is shown at the top of the window pane. This is the most recent style that was created or selected from this drop-down. But it doesn't necessarily mean it is the style that is being changed by the fields in the bottom section of the window pane.



A style called "div.MyNewLayout" is currently selected in the Manage section (i.e., in the XML Editor you've clicked in a responsive layout that is using this style). Therefore, any changes you make in the fields below will affect this style.




Use any of the following options in the Responsive Layout window pane to enhance the cell and your layout.





Warning: Keep in mind that your changes in this window pane will be auto-saved to your stylesheet by default. This means that anyone making changes to a particular row style using the Responsive Layout window pane will be affecting any other responsive layouts that are using that same style wherever they have been inserted throughout the project.



You can remove the check mark from the **Auto-save Stylesheet** option at the bottom of the window pane; however, if you do that, you will need to save your changes to both your content file *and* your stylesheet as you work.

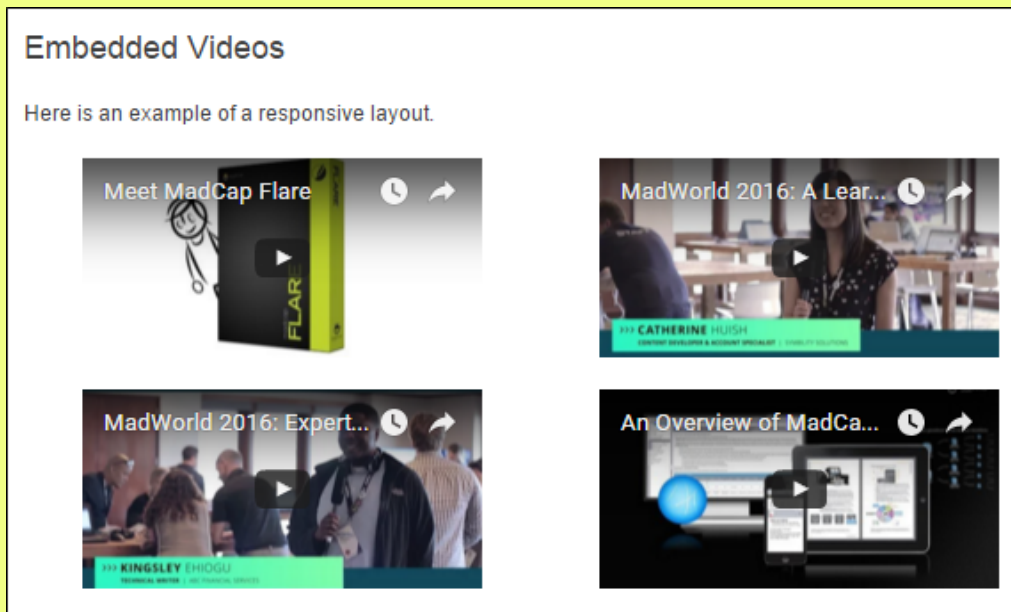
Option	Description
Add Cell	Adds a new cell to the right edge of the row. By default, a new cell will start out with a width of one column (8.333%).
Delete Cell	Deletes the selected cell from the layout. A message asks if you want to remove the corresponding style from the stylesheet as well.
Cell Width	Select a number of columns to set the width of the current cell (in percentage). For example, six columns equals 50%. As you select each number the new percentage changes in the Cell Style preview area below.
Cell Offset	Select a number of columns to provide the offset (left margin) for the current cell (in percentage). For example, four columns equals a left margin of 33.333%. When you select a number, the left margin percentage changes in the Cell Style preview area below.
Row Gutter	Enter the number of pixels for a gutter in the entire row. This adds left and right padding for each of the cells. For example, if you enter 20 pixels for a gutter, each cell will have 10 pixels of left padding and 10 pixels of right padding.
	Moves the entire row up. This simply moves the layout above whatever block element is next to it (at the same level) in the XML Editor. Alternatively, you can click and drag the outermost div structure bar.
	Moves the active cell to the left in the row.
	Moves the active cell to the right in the row.

Option	Description
	Moves the entire row down. This simply moves the layout below whatever block element is next to it (at the same level) in the XML Editor. Alternatively, you can click and drag the outermost div structure bar.
Edit	<p>Opens the stylesheet so that the style (i.e., complex selector) for the current cell is selected. You can then make any additional style changes (e.g., add a border, add a background color, set the right margin).</p> <p>You can also make style changes for the entire row. For more information, see the online Help.</p>
Auto-save	If this is selected any changes made to the selected style in this window pane will be applied in the stylesheet. If you disable this option, you must save changes in both the XML Editor and Stylesheet Editor as you work.

9. When you are finished configuring the row for your current layout, select a different media query (e.g., Tablet, Mobile) at the top of the Responsive Layout window pane. (Alternatively, you can change the layout in the local toolbar of the XML Editor.)
10. Repeat steps 8 and 9 until you are finished designing the row for all of your mediums/media queries.
11. Click  to save your work.

EXAMPLE

Let's say you want to embed four YouTube videos in a topic. When viewed on a large screen, you want the videos to display like this, two videos above the other two:

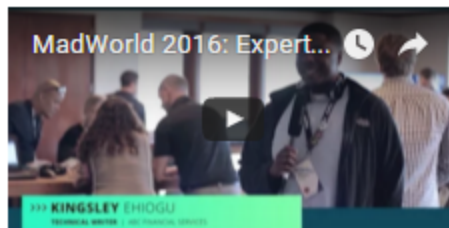


You want this same configuration to be used when the topic is viewed on a smaller tablet device.

But when viewed on a smart phone, you want each video to be displayed by itself, like this:

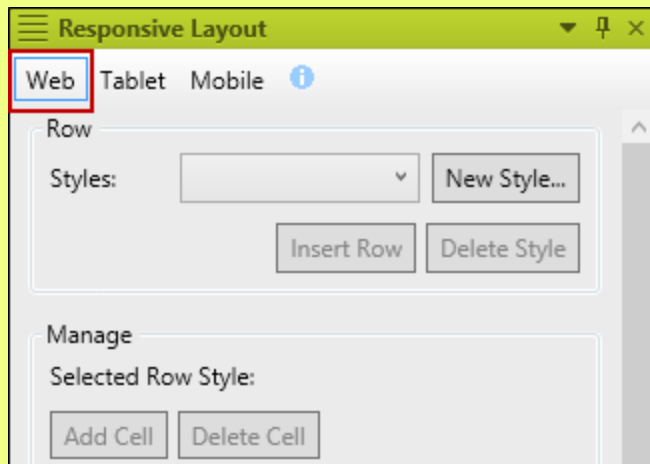
Embedded Videos

Here is an example of a responsive layout.

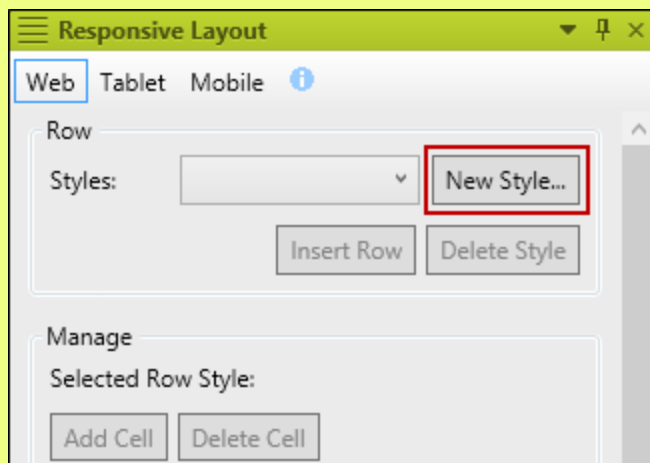


To make this happen, you first open the topic where you want to insert the videos, and you also open the Responsive Layout window pane.

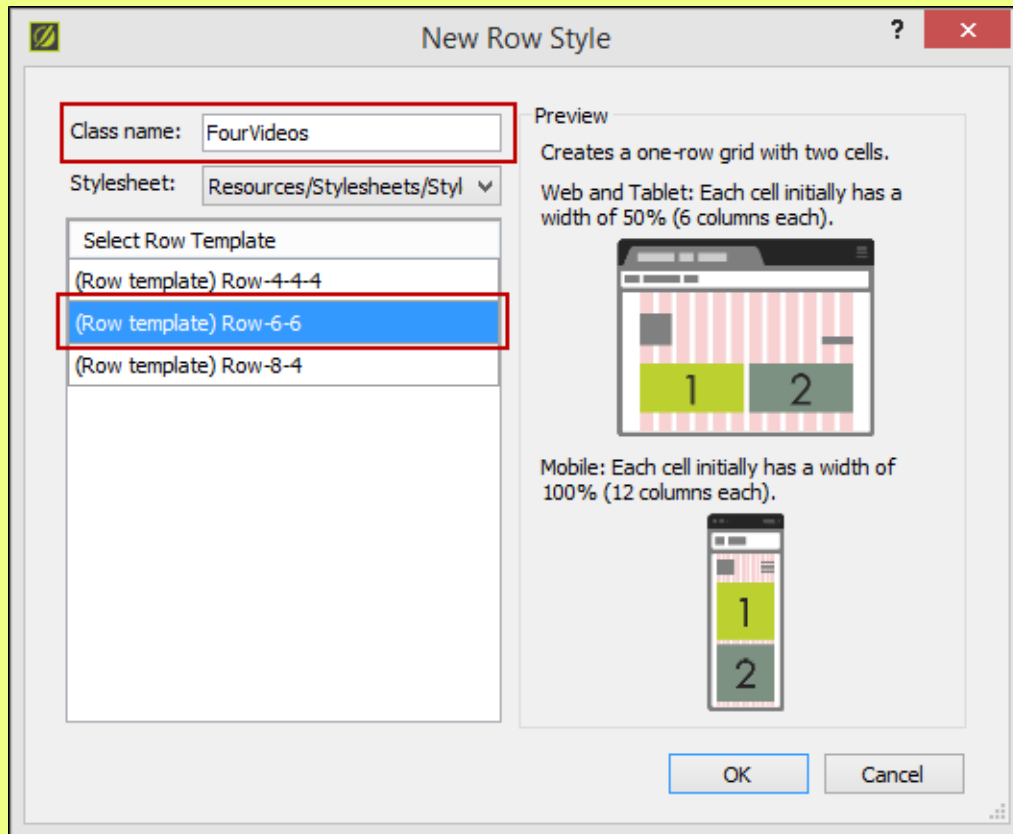
Make sure the **Web** medium is selected in the Responsive Layout window pane.



Then place the cursor where you want to insert the layout. We're going to assume you haven't created any row styles yet in your project, so in the Responsive Layout window pane, click **New Style**.

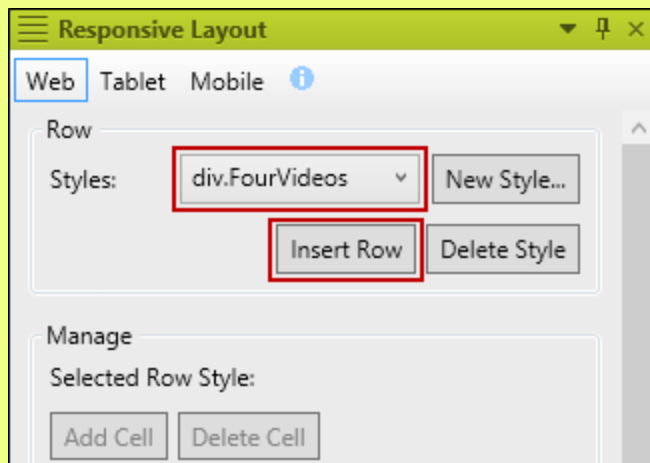


In the New Row Style dialog, give the new style a name. Let's say you call it **FourVideos**. Your stylesheet is already selected in the dialog, but you need to choose a template. Click **Row-6-6**, which will start you out with two cells in the row, each taking up half the total space.

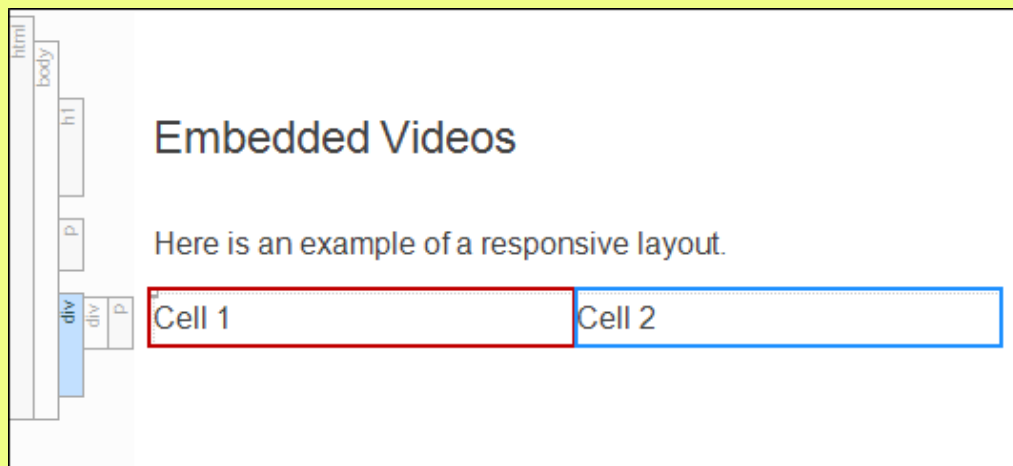


After you click **OK**, the row style is added to your stylesheet and is selected in the Responsive Layout window pane. Also, your stylesheet will open behind the topic.

Now click **Insert Row**.



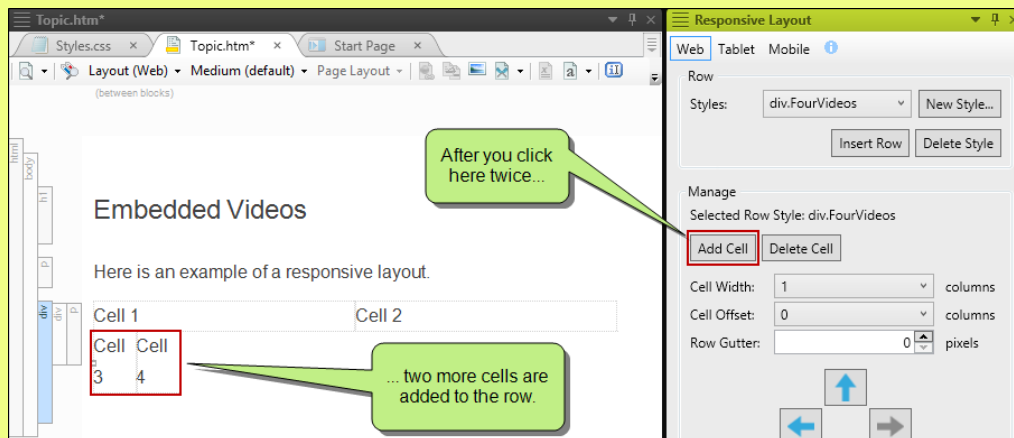
In your topic, you will see two cells, indicated by the dummy text "Cell 1" and "Cell 2."



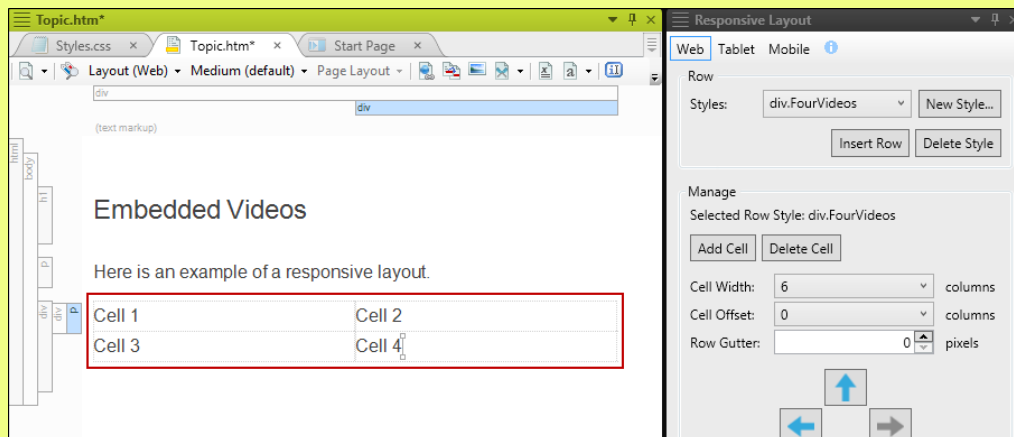
Click in either of the cells. The Responsive Layout window pane will change, showing the buttons that you can select, as well as the current style settings for that cell.

The screenshot shows a web editor interface with a 'Responsive Layout' pane on the right. The main editor displays a page titled 'Topic.htm' with a section 'Embedded Videos' and a sub-section 'Here is an example of a responsive layout.' Below this, there are two columns labeled 'Cell 1' and 'Cell 2'. 'Cell 2' is highlighted with a red border. The 'Responsive Layout' pane on the right has tabs for 'Web', 'Tablet', and 'Mobile'. The 'Web' tab is active. It shows a 'Row' section with 'Styles: div.FourVideos' and buttons 'New Style...', 'Insert Row', and 'Delete Style'. Below this is a 'Manage' section with 'Selected Row Style: div.FourVideos' and buttons 'Add Cell' and 'Delete Cell'. The 'Cell Width' is set to '6' columns, 'Cell Offset' is '0' columns, and 'Row Gutter' is '0' pixels. There are also directional arrows. The 'Cell Style' section shows 'Medium: web' and a complex selector 'div.FourVideos > div:nth-child(2)' with a red box around it. Below the selector, the 'width' is set to '50%' and 'margin-left' is '0%', both with red boxes around them. An 'Edit' button is at the bottom right of the 'Cell Style' section. Several green callout boxes with arrows point to specific elements: one points to 'Cell 2' with the text 'In this example, we clicked in Cell 2.'; another points to the 'div.FourVideos' style with the text 'This indicates the style being used for the row where your cursor is placed.'; a third points to the 'Add Cell' and 'Delete Cell' buttons with the text 'These buttons are now enabled.'; a fourth points to the 'div:nth-child(2)' selector with the text 'The cell uses this complex selector in the stylesheet. It's the second cell, and that's why you see the nth-child(2) pseudo class at the end.'; and a fifth points to the 'width: 50%' and 'margin-left: 0%' settings with the text 'The width of the cell is currently 50%. That's because 6 columns have been set in the Cell Width field above.'

You've got two cells, but you need two more. So click **Add Cell**. This inserts a third cell in your topic. Click **Add Cell** again. This adds a fourth cell.



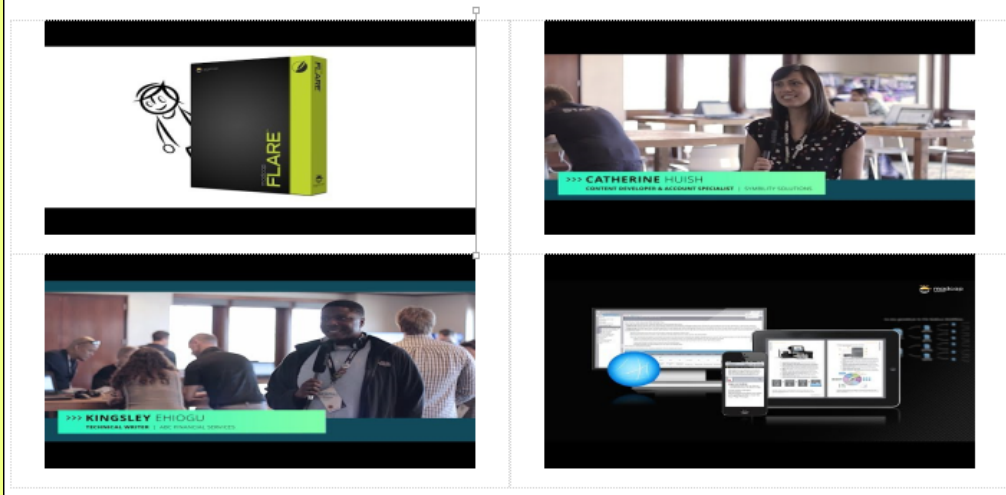
So you've got four cells for the four videos, but you need to adjust their widths. Click in the third cell and in the Responsive Layout window pane, change the **Cell Width** to 6. Do the same for the fourth cell. Now each of the four cells has a width of 50%.



Next, click in each cell and replace the default text with your embedded YouTube videos. In the XML Editor, it might look something like this:

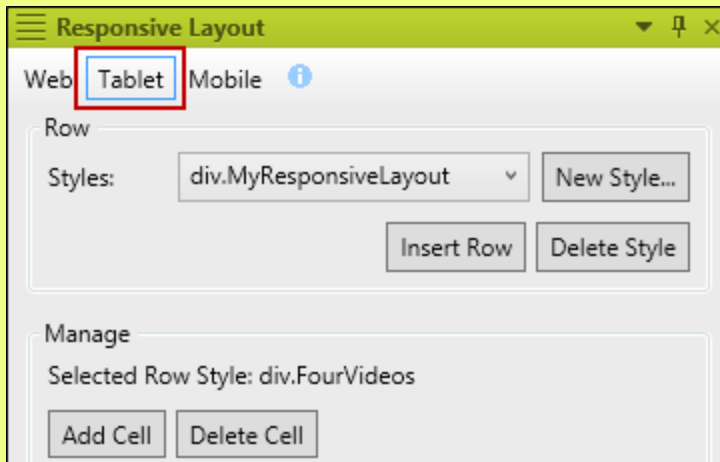
Embedded Videos

Here is an example of a responsive layout.



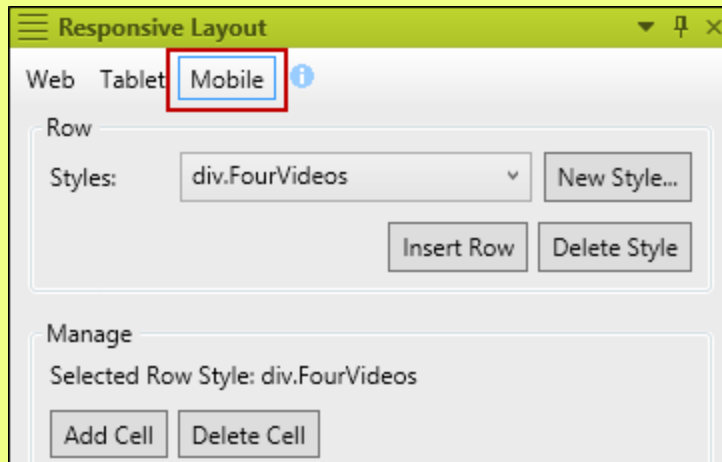
You can also apply other styles to your content as needed. For example, you might want to center your content.

At the top of the Responsive Layout window pane, click **Tablet**. The layout view and medium in the XML Editor change automatically.



Set all four cells to the same width (6 columns, 50%).

At the top of the Responsive Layout window pane, click **Mobile**. The layout view and medium in the XML Editor change automatically.



Click in each cell and in the Responsive Layout window pane, make sure the **Cell Width** is set to **12**. This results in a width of 100% for each cell.

When you generate your HTML5 output, you should see the configuration shown at the beginning of this example for each of the three layouts (Web, Tablet, Mobile).

CHAPTER 4

Snippet Enhancements

Several enhancements have been made to snippets in Flare 12. You can now pin your favorite snippets from a group of recently used snippets. Snippet conditions have been enhanced so you can apply them to multiple instances of a snippet within a single topic. Additionally, you can now set variable definitions for individual snippets. Each of these changes gives you more flexibility when working with snippets in Flare.

This chapter discusses the following:

Snippet Groups	70
Pinning Snippets	72
Snippet Conditions on Individual Snippets	76
Snippet Variables	87

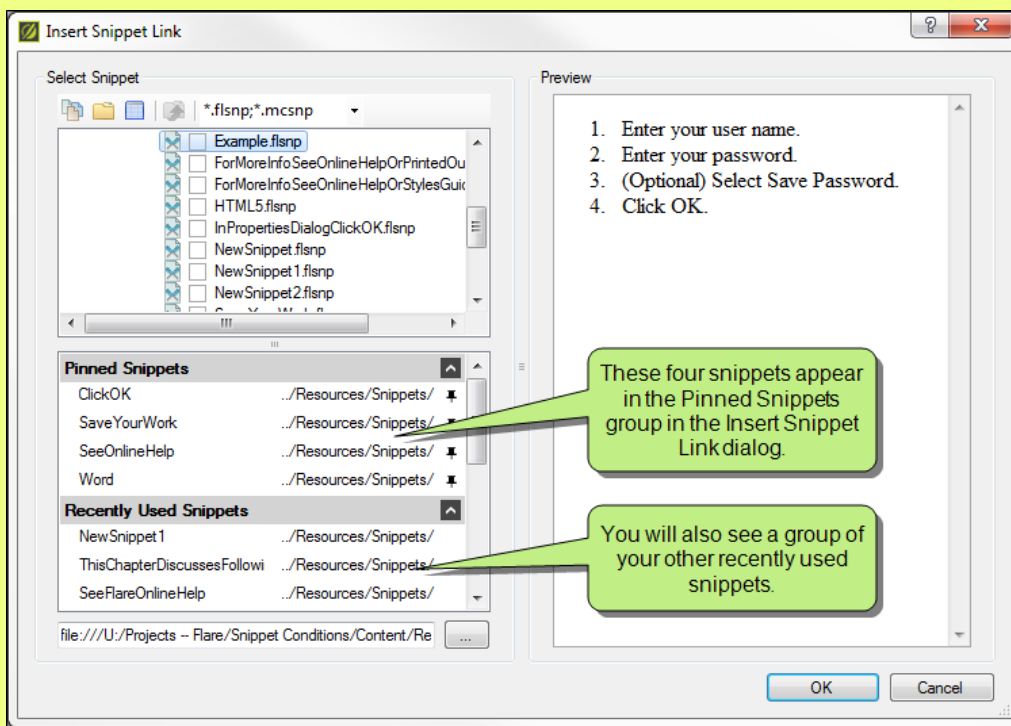


Snippet Groups

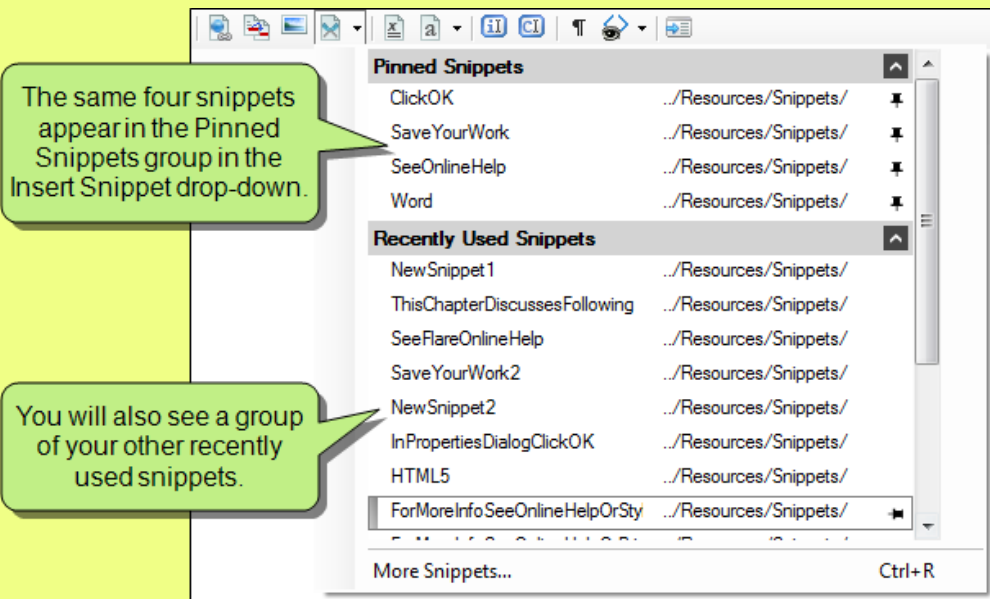
You can now use groups for your pinned and most recently used snippets. Snippet groups are available for pinned snippets and recently used snippets. You can access snippet groups from both the Insert Snippet Link dialog and the Insert Snippet drop-down in the local toolbar of the XML Editor.

EXAMPLE

Let's say that you have pinned four snippets. You have also recently used 15 snippets. In the Insert Snippet Link dialog, you will see the Pinned Snippets and Recently Used Snippets groups.



In the Insert Snippet drop-down in the local toolbar of the XML Editor, you will see the same groups with the same snippets in each group.



Note: The Recently Used Snippets and Pinned Snippets groups show the relative path location of each snippet. Hover over any snippet to see its full file path.



Note: You can expand or collapse the Recently Used Snippets and Pinned Snippets groups by clicking the arrows next to each group name.

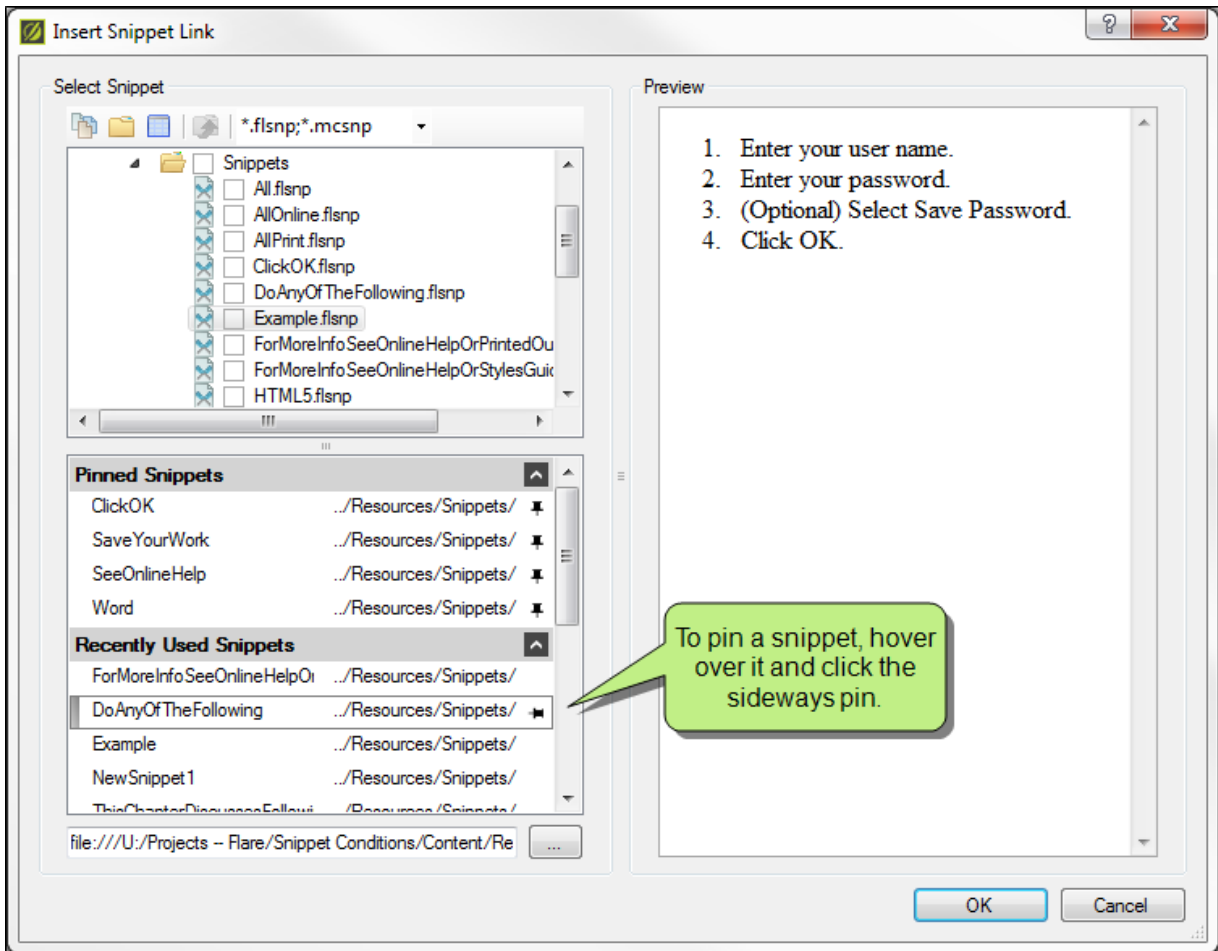


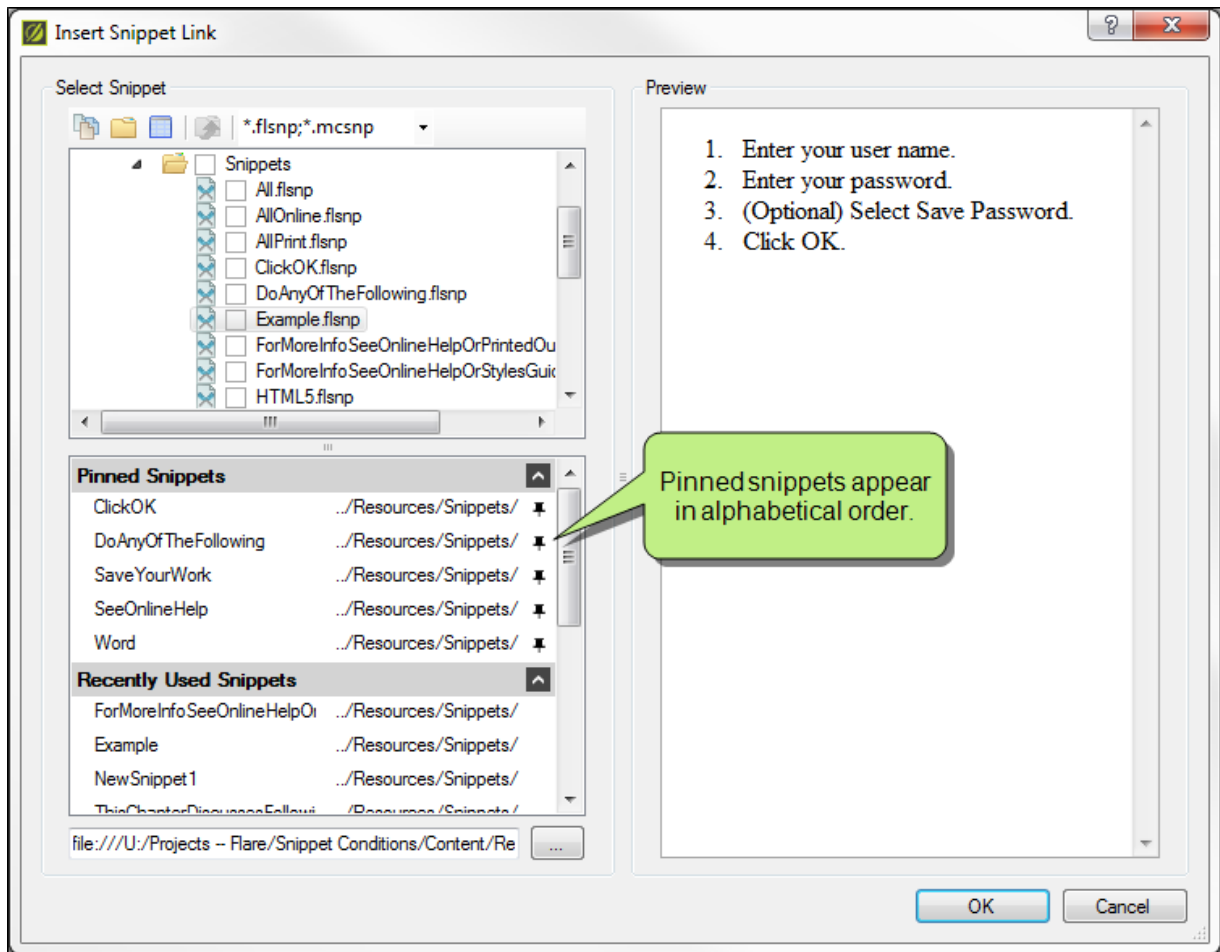
Note: If you need to insert a snippet that does not appear in the Pinned Snippets or Recently Used Snippets groups, you can select **More Snippets** to open the Insert Snippet Link dialog.

Pinning Snippets

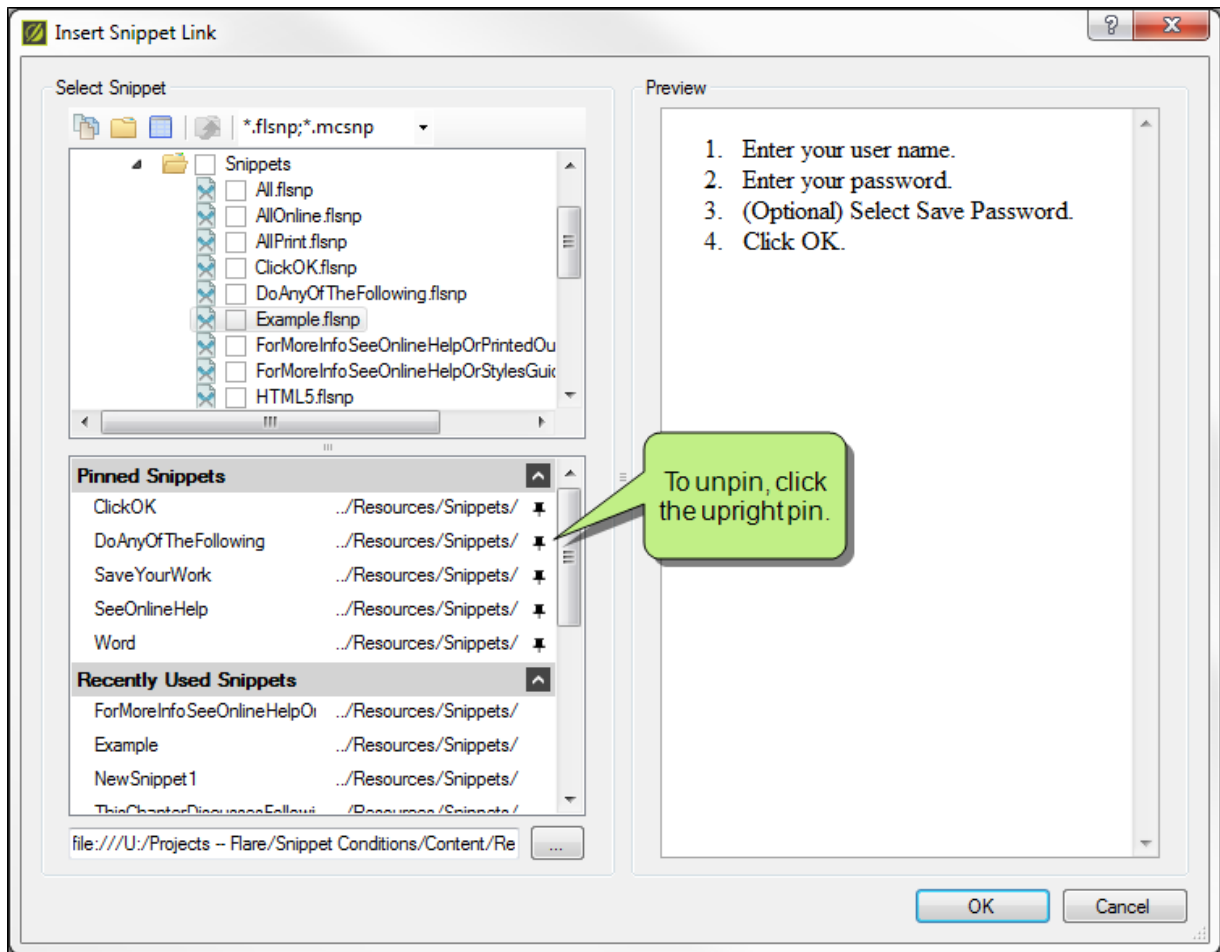
Flare 11 introduced pinning for styles and fonts. In Flare 12, you can also pin your favorite snippets. When you use a snippet file, it is added to a group of recently used snippets. You can pin any snippet found in the Recently Used Snippets group, which lists your 20 most recently used snippets. You can access this group from a drop-down in the local toolbar of the XML Editor, and also from the Insert Snippet Link dialog (**Insert>Snippet**).

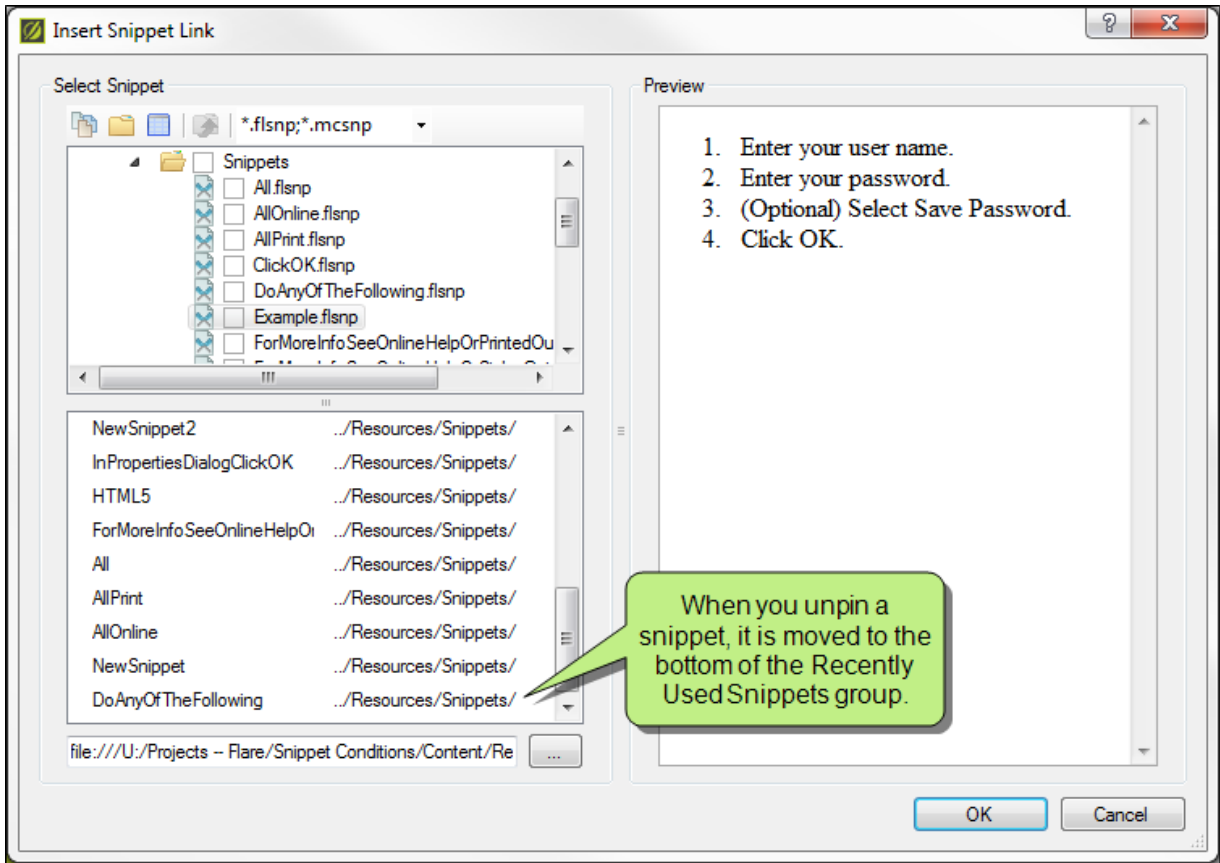
To pin a snippet, open the Insert Snippet Link dialog or expand the Insert Snippet drop-down in the XML Editor local toolbar. Hover over the snippet you want to pin and click the little pin. Pinned snippets are added to the Pinned Snippets group in alphabetical order.





To unpin a snippet, just click the pin again and it will be moved back to the bottom of the Recently Used Snippets group.





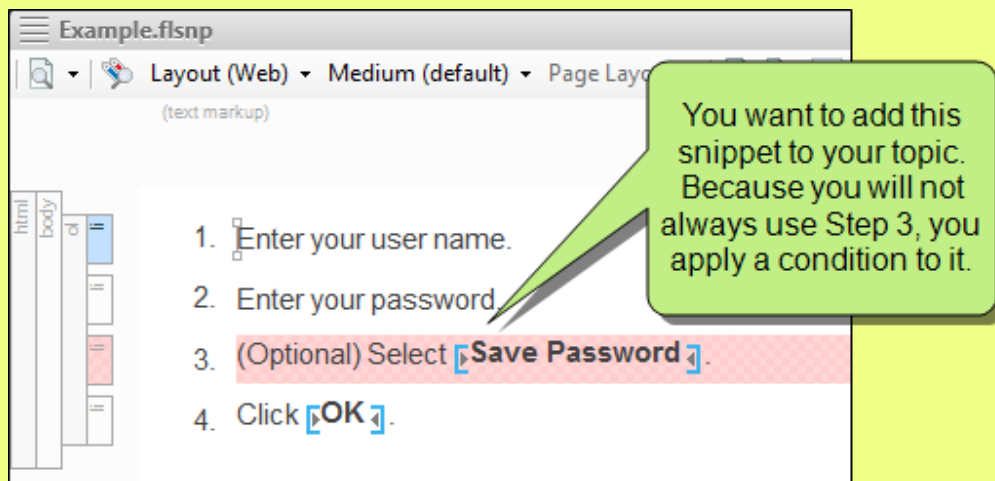
Note: Snippets must appear in the Recently Used Snippets group before they can be pinned. If the snippet you want to pin does not appear in the Recently Used Snippets group, you can add it to your project from the Content Explorer or the Insert Snippet Link dialog.

Snippet Conditions on Individual Snippets

You can now apply snippet conditions to your content and enable the conditions at the snippet level, rather than at the topic level. Doing this allows you to use different versions of the same snippet in a single topic. This makes single-sourcing easier by preventing duplicated content and multiple snippets. Snippet conditions applied to individual snippets override snippet conditions applied at the topic level.

EXAMPLE

Let's say you are writing a user guide. There is a three-step process that users need to repeat very frequently, so you create a snippet for those steps. However, there occasionally is a fourth step in the process, where users can optionally select an extra check box before completing the process. You sometimes need to include this process multiple times in the same topic, so you create a snippet for the four-step process, and apply a condition to the optional step.



You add your snippet to the topic...

The screenshot shows a topic editor window titled "Steps.htm*". The main content area displays a topic titled "Accessing the Flex Plan" with a list of steps. A snippet titled "HOW TO ENTER YOUR LOGIN INFORMATION" is being added to the topic. The snippet contains a list of steps, with the third step, "(Optional) Select **Save Password**", highlighted in red. A green callout bubble points to the snippet title, stating "You want to keep this snippet as is." Another green callout bubble points to the third step of the snippet, stating "But you want to exclude Step 3 in this instance of the snippet." The snippet is being added to the topic, and the topic's structure is visible in the left sidebar.

Steps.htm*

Layout (Web) Medium (default) Page Layout

(text markup)

Accessing the Flex Plan

1. Open the HR website.
2. In the Login dialog, enter your login information.
 - HOW TO ENTER YOUR LOGIN INFORMATION
 - 1. Enter your user name.
 - 2. Enter your password.
 - 3. (Optional) Select **Save Password**.
 - 4. Click **OK**.
3. On the HR website homepage, click **Flex Plan**.
4. In the Flex Plan Login dialog, enter your flex plan login information.
 - HOW TO ENTER YOUR FLEX PLAN LOGIN INFORMATION
 - 1. Enter your user name.
 - 2. Enter your password.
 - 3. (Optional) Select **Save Password**.
 - 4. Click **OK**.
5. On the Flex Plan homepage, click **View Details**. You will be able to see your current balance, outstanding claims, and submit a new claim.

...and then exclude the snippet condition on the second instance of the snippet. By doing this, you won't see the optional step in that instance of the snippet.

4. In the Flex Plan Login dialog, enter your flex plan login information.

☐ HOW TO ENTER YOUR FLEX PLAN LOGIN INFORMATION

1. Enter your user name.
2. Enter your password.
3. (Optional) Select \$
4. Click OK.

5. On the Flex Plan homepage claim.

Right-click on the snippet and choose **Snippet Conditions**.

<MadCap:snippetBlock>

- ✓ Select Node
- Cut
- Copy
- Paste Before
- Paste After
- Delete
- text
- Snippet...
- Open Snippet File With ▶
- Conditions... Ctrl+Shift+C
- Snippet Conditions...
- Snippet Variables...

The screenshot shows a web editor interface with a sidebar on the left containing a tree view of document elements. The main content area displays two snippets under the heading "Accessing the Flex Plan".

Snippet 1: TO ENTER YOUR LOGIN INFORMATION

- 1. Enter your user name.
- 2. Enter your password.
- 3. (Optional) Select **Save Password**.
- 4. Click **OK**.

Snippet 2: TO ENTER YOUR FLEX PLAN LOGIN INFORMATION

- 1. Enter your user name.
- 2. Enter your password.
- 3. Click **OK**.

Callout Box 1: "Since you want to see all four steps here, you leave this snippet as is." (Points to step 3 of Snippet 1)

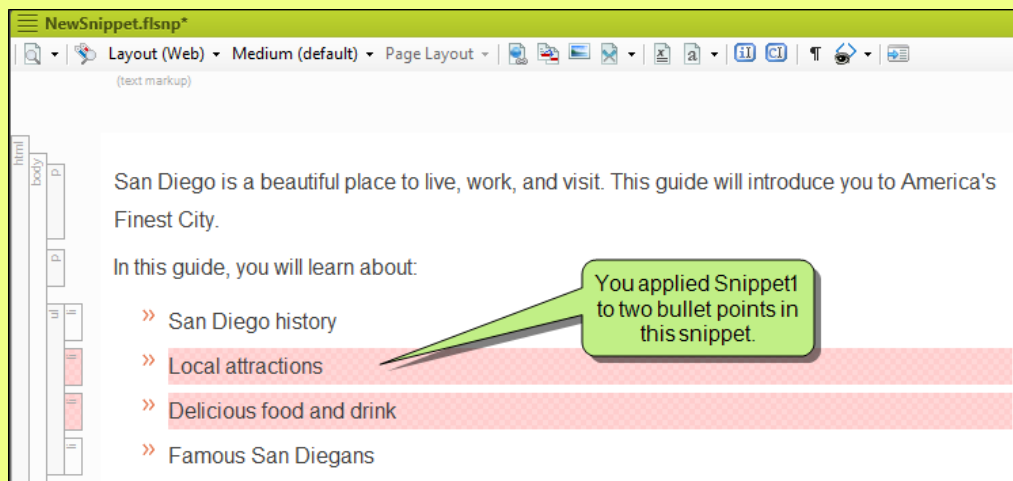
Callout Box 2: "But you apply an Exclude condition to this snippet so you do not see the optional step." (Points to step 3 of Snippet 2)

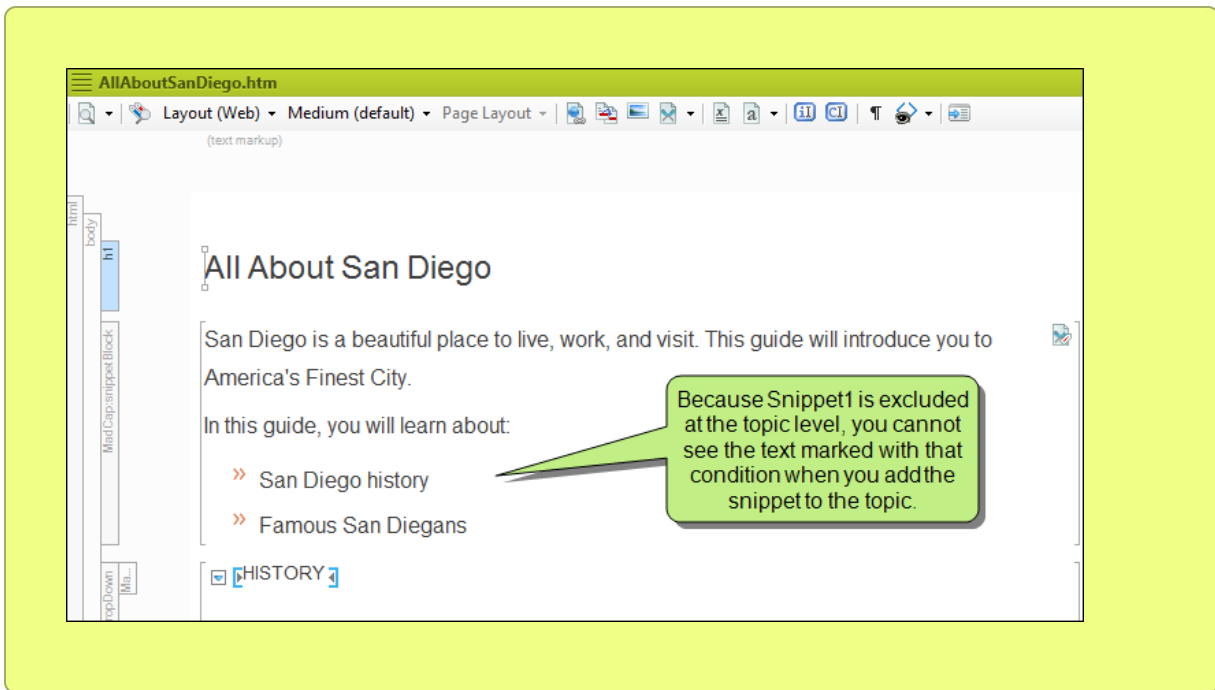
Step 5: On the Flex Plan homepage, click **View Details**. You will be able to see your current balance, outstanding claims, and submit a new claim.

Later, if you need to edit the content, you only have one snippet to edit.

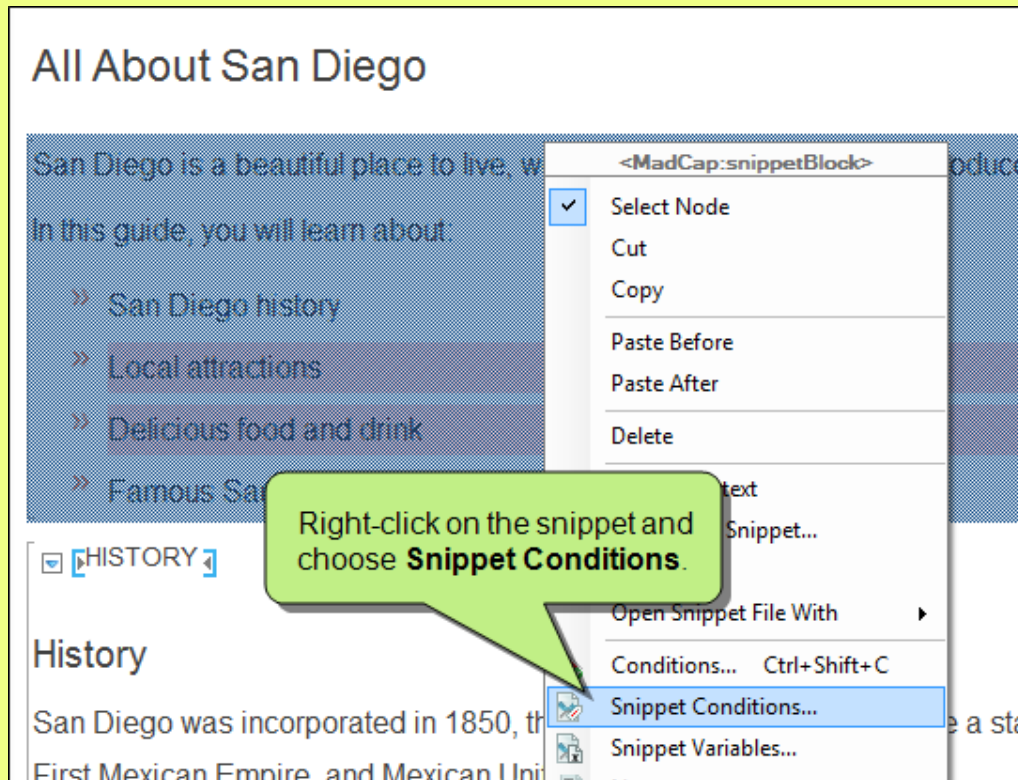
EXAMPLE

Let's say you are working in a topic that has several snippet conditions set at the topic level. When you insert an existing snippet into your topic, you notice that some of the text in the snippet is excluded using the condition tag Snippet1. However, the Snippet1 condition tag is also excluded at the topic level in the topic you are working on, so that text does not appear in the XML Editor or your output.

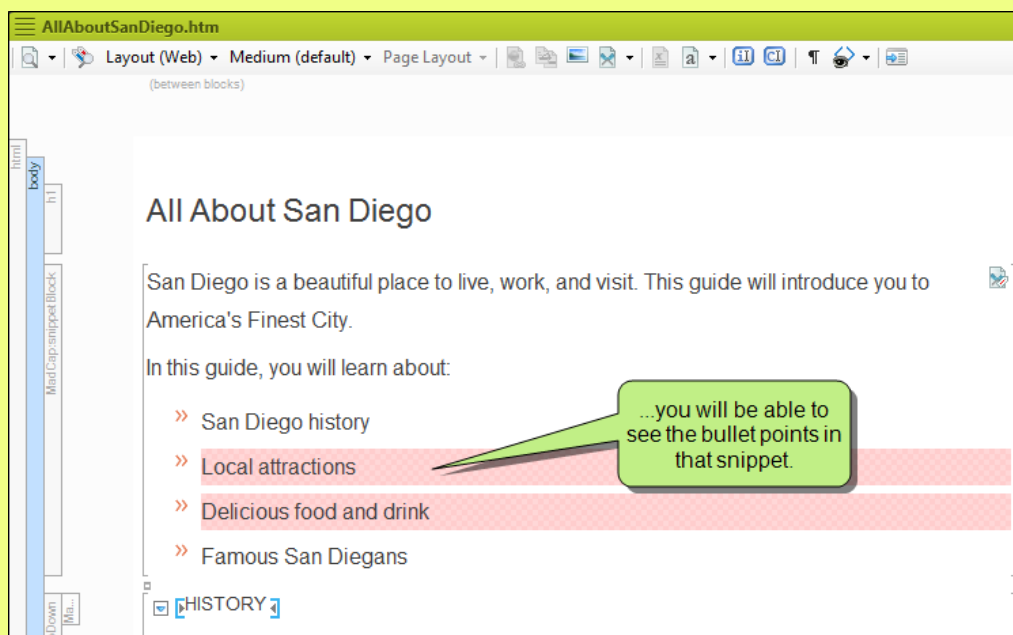
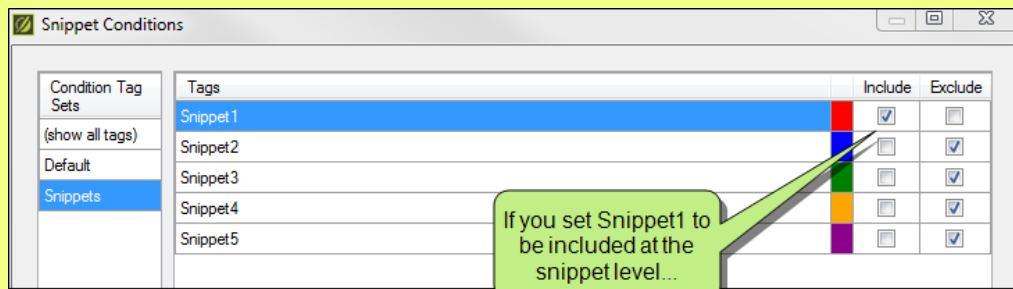




Snippet conditions on individual snippets override those at the topic level. So in the topic you right-click the snippet and choose **Snippet Conditions**.



In the Snippet Conditions dialog, you set the conditions on this snippet to *include* Snippet1. While Snippet1 will still be excluded at the topic level, in this particular snippet, you will be able to see the text that is marked with that tag.



HOW TO USE SNIPPET CONDITIONS IN A TOPIC

1. Create all of the condition tags that you plan to use for separating your snippet content.




Note: The more variations that you will have in your snippet content, the more condition tags you will need to create. An average number is four to eight condition tags, but you may find that you require fewer or more tags than this for snippet content. It is always preferable to create more condition tags than you may need, rather than to realize later that you have too few. You can even create a separate condition tag set to be used specifically for holding your snippet condition tags (while keeping your regular condition tags in other tag sets).



Tip: It is also a good idea to include the word "Snippet" somewhere in the name of a new snippet condition tag. This way, you can easily identify it, rather than confusing it with the regular condition tags that you use for other purposes (e.g., online output or printed output condition tags).

2. Create and open the snippet.
3. Apply the necessary condition tags to the text or other content in the snippet.



Tip: It is a good idea to click the **Hide/Show Conditional Indicators** button  at the bottom of the editor so that you can clearly see where the different tags have been applied in the snippet.


4. Insert the snippet into the necessary topic, or determine which topic already contains the snippet.
5. In the topic, right-click the first instance of the snippet and select **Snippet Conditions** from the context menu. The Snippet Conditions dialog opens.
6. If you want to exclude a condition tag from the snippet, click the **Exclude** check box next to it. If you want to make sure a condition tag is included in the snippet, click the **Include** check box next to it. The primary reason for having Include check boxes is to account for possible conflicts. Snippet conditions set on a snippet override snippet conditions set at the topic level.





Note: If you want to create more advanced condition tag expressions, you can switch to Advanced mode.

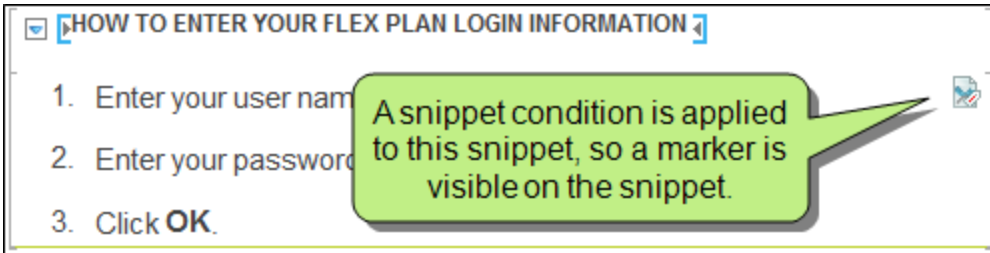



Warning: If you do not select any check boxes at all, all of the tags will automatically be *included*. If you include only some tags and exclude others, any remaining tags without check marks for either Include or Exclude will automatically be *included*. However, if you select Include for any of the tags (even just one) and do not select Exclude for any tags, all of the other tags that do not have Include check marks will automatically be *excluded*. If you want to be safe and always know that the correct tags are included or excluded, you can make sure that all of the tags have check marks with either Include or Exclude.


7. Click **OK**.
8. (Optional) Repeat steps 5-7 for the other instances of the snippet in your topic.
9. Click  to save your work.

SNIPPET CONDITIONAL INDICATOR COMMAND

A Snippet Conditional Indicator command has been added to the **Show tags** drop-down arrow  in the XML Editor. This command enables or disables a marker  that overlays snippets that have snippet conditions applied to them.



 **Note:** If you enable this indicator, you will see it on snippets if you have snippet conditions enabled at either the topic level or the snippet level.

 **Note:** Remember that snippet conditions only apply to a particular instance of a snippet. If you add new snippet conditions to an existing snippet, be sure to use the Link Viewer to check that the content in other instances of the snippet is still visible (or hidden) the way you want it to be.

Snippet Variables

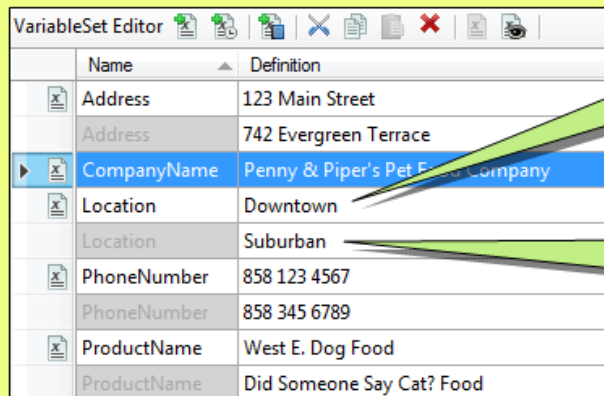
You have always been able to override variable definitions at the target level. Now you can also override definitions for variables in snippets. You can do this at the topic level or the snippet level. This gives you more ways to customize your snippets. This also is beneficial for single-sourcing because you do not need to create multiple versions of the same content or convert content to text.

EXAMPLES—DIFFERENT TYPES OF VARIABLE DEFINITIONS

With the introduction of snippet variables, Flare now offers several different types of variable definitions. Use the examples below to help you determine which type of variable definition you should use.

- » **Original Variable Definitions** Used in most cases. These are the variable definitions you enter in the Variable Set Editor. You can enter primary and alternate variable definitions.

These are the original variable definitions.

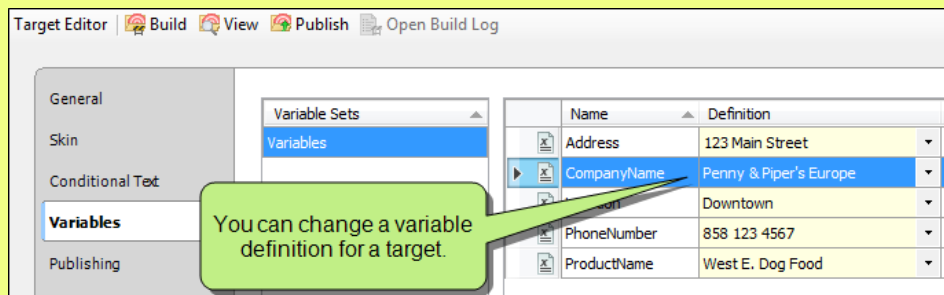


	Name	Definition
	Address	123 Main Street
	Address	742 Evergreen Terrace
	CompanyName	Penny & Piper's Pet Food Company
	Location	Downtown
	Location	Suburban
	PhoneNumber	858 123 4567
	PhoneNumber	858 345 6789
	ProductName	West E. Dog Food
	ProductName	Did Someone Say Cat? Food

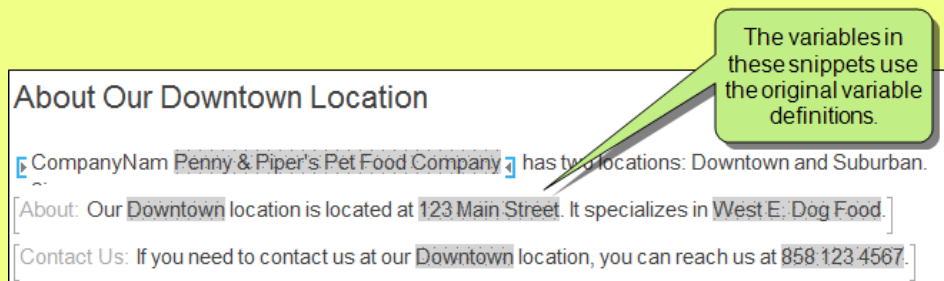
The first entry for a variable is the primary definition.

The second entry is an alternate definition.

- » **Target-Level Variable Definitions** Used when generating output for specific targets. Overrides the project-level variable definition, but only for the defined target. For example, if you use a company name variable in your documents, you may want to change it to say "ABC Corporation" in one document and "XYZ Company" in another.



- » **Topic-Level Variable Definitions** Used to change a variable definition in a single topic. Applies only to variables in snippets, and overrides project-level and target-level variable definitions. For example, if you use a phone number variable throughout your project, but a handful of topics are about a different location (and therefore use a different phone number), you can set a different variable definition that applies only to those topics.



About Our Suburban Location

[CompanyName Penny & Piper's Pet Food Company] has two locations: Downtown
[About: Our Suburban location is located at 742 Evergreen Terrace. It specializes in Did Someone Say Cat? Food.]
[Contact Us: If you need to contact us at our Suburban location, you can reach us at 858.345.6789.]

You can edit the variable definitions at the topic level if you want to use the same snippets in multiple topics.

- » **Snippet-Level Variable Definitions** Used to change a variable definition in a single snippet. Applies only to variables in snippets, and overrides project- and target-level variable definitions, as well as topic-level snippet variable definitions. For example, if you use a product name variable throughout your project but a few snippets refer to a different product, you can set a different variable definition that applies only to those snippets.

Our Products


[CompanyName Penny & Piper's Pet Food Company] makes delicious [Product] cat food. We specialize in West E. Dog Food and Did Someone Say Cat? Food.
[Product: Our Downtown location makes West E. Dog Food.]
[Product: Our Suburban location makes Did Someone Say Cat? Food.]

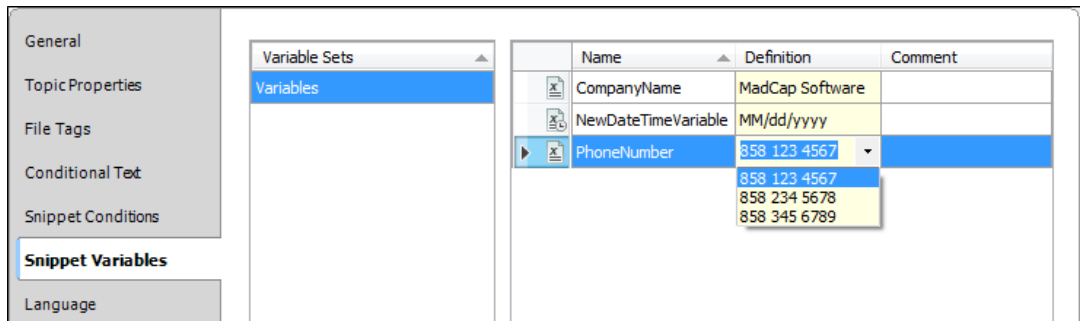
This snippet uses the original variable definitions.


This snippet uses variable definitions edited at the snippet level.

HOW TO OVERRIDE A SNIPPET VARIABLE DEFINITION FOR A TOPIC

Use this option if you want to change the definition of all of the variables that appear in snippets in a single topic. Topic-level snippet variable definitions take priority over your original variable definitions or target-level variable definitions.

1. Open the Content Explorer.
2. Locate the topic in which you want to override a variable definition.
3. In the local toolbar, click . The Properties dialog for the topic opens.
4. Click the **Snippet Variables** tab. The Snippet Variables tab shows all of the variable sets in your project and their associated variables.
5. Select the appropriate variable set.
6. Do one of the following, depending on whether the variable has only one or multiple definitions in the Variable Set Editor.
 - » **Single Definition** Click in the **Definition** cell for the variable that you want to override and press **F2** on your keyboard. Then type a different definition and press **Enter** on your keyboard.
 - » **Multiple Definitions** In the **Definition** cell, click the down arrow and select an alternative definition.



7. Click **OK**.
8. Click  to save your work. The alternative variable definition will be used for all instances of the variable that appear in snippets in this topic. However, if the variable appears elsewhere in the topic (i.e., not in a snippet), or in other topics in the project, the original variable definition (or the target-level variable definition, if applicable) will be used.

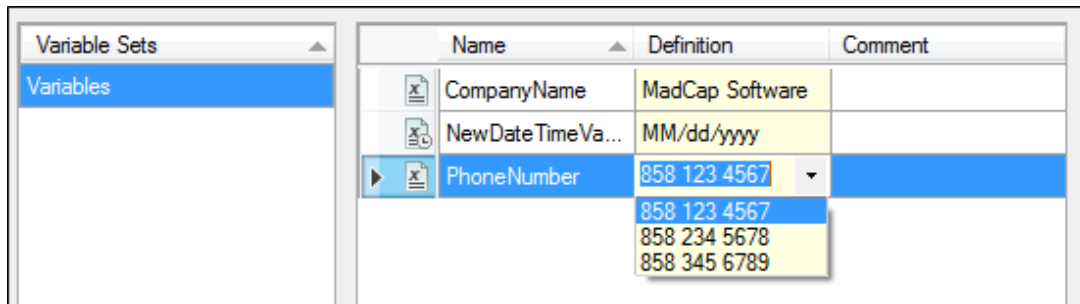
HOW TO OVERRIDE A SNIPPET VARIABLE DEFINITION FOR A SNIPPET


Use this option if you want to change the definition of a variable within a specific snippet. Snippet-level snippet variable definitions take priority over topic-level snippet variable definitions.

1. Open the Content Explorer.
2. Locate the topic in which you want to override a variable definition.
3. In the topic, right-click the instance of the snippet in which you want to override the variable definition and select **Snippet Variables** from the context menu.

The Snippet Variables dialog opens. The Snippet Variables dialog shows all of the variable sets in your project and their associated variables.

4. Select the appropriate variable set.
5. Do one of the following, depending on whether the variable has only one or multiple definitions in the Variable Set Editor.
 - » **Single Definition** Click in the **Definition** cell for the variable that you want to override and press **F2** on your keyboard. Then type a different definition and press **Enter** on your keyboard.
 - » **Multiple Definitions** In the **Definition** cell, click the down arrow and select an alternative definition.



6. Click **OK**.
7. Click  to save your work. The alternative variable definition will be used for all instances of the variable that appear in the selected snippet. However, if the variable appears elsewhere in the topic (i.e., in other snippets or outside of snippets) or in other topics, the topic-level snippet variable definition, the target-level variable definition (if applicable), or the original variable definition will be used.

CHAPTER 5

Stylesheet Enhancements

There have been several improvements to the way you can use styles and stylesheets. This includes a major redesign of Flare's Stylesheet Editor. Many of the changes to the Stylesheet Editor affect only the Advanced view, but some can be used when you are in the Simplified view as well.

This chapter discusses the following:

Allow Local and Master Stylesheets	94
Redesigned Display of Properties	109
Multiple Medium View	115
Add Selector Button Results in More Options	122
Media Queries	160
Multi-Select and Delete	173
Conditions	177
Tooltips Show Inheritance	178
Units of Measurement—New Options	180

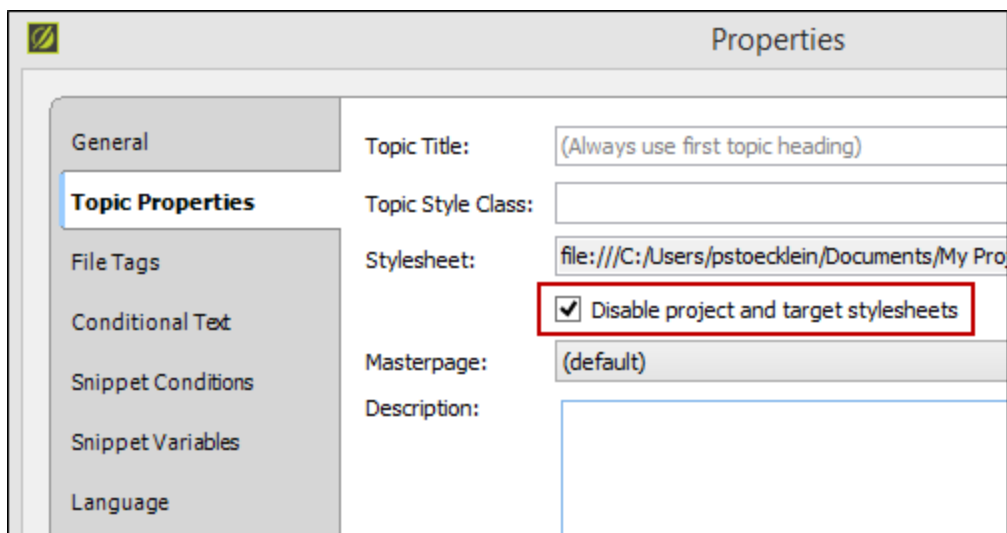


Allow Local and Master Stylesheets

In previous versions you could set a master stylesheet at the project or target level, but if you did that, you then were not allowed to associate stylesheets with individual topics. Starting with this version, you can have both local and master stylesheets in a project. As a general rule, stylesheets assigned at the topic level have precedence over any master stylesheets, and a master stylesheet at the target level has precedence over a master stylesheet at the project level. However, there are options at each level that give you even more control over multiple stylesheets set in a project.

LOCAL STYLE SHEET (TOPICS)

In the Properties dialog for a topic, you can open the **Topic Properties** tab to choose a stylesheet. The **Stylesheet** field has a new check box under it labeled **Disable project and target stylesheets**.



The screenshot shows the 'Properties' dialog box with the 'Topic Properties' tab selected. The left sidebar contains a list of tabs: General, Topic Properties (selected), File Tags, Conditional Text, Snippet Conditions, Snippet Variables, and Language. The main area of the dialog contains the following fields:

- Topic Title: (Always use first topic heading)
- Topic Style Class: (empty text box)
- Stylesheet: file:///C:/Users/pstoecklein/Documents/My Pro
- ☒ Disable project and target stylesheets (This checkbox is highlighted with a red rectangle)
- Masterpage: (default)
- Description: (empty text box)

If you select this check box, any target or project master stylesheet will be ignored when it comes to the formatting of the selected topic. However, the master stylesheets may still be used for other topics.

LOCAL STYLESHEET (MASTER PAGES)

You can associate a master page with a stylesheet using the Properties dialog. In previous versions, the only way to do this was to use the Stylesheet Links dialog.

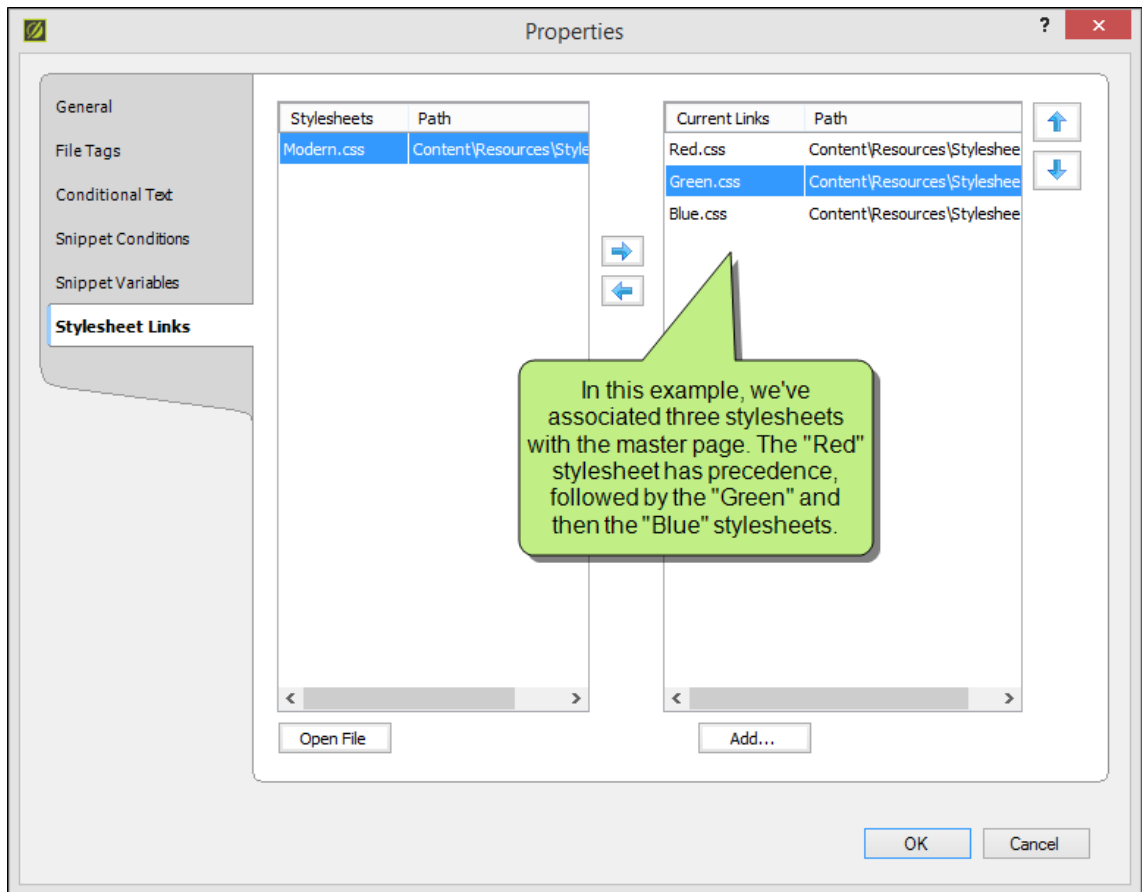
HOW TO ASSOCIATE A MASTER PAGE WITH A STYLESHEET

1. If you have master stylesheets set on a target or at the project level, you must open the Target Editor or Project Properties dialog and select the option to **Allow local stylesheets**. If you do not have a master stylesheet set, you can skip this step.
2. In the Content Explorer, right-click your master page and select **Properties**. The Properties dialog opens.



Note: If you want to associate a stylesheet with multiple master pages, you can open the File List window pane (**View>File List**), hold down your **SHIFT** or **CTRL** key to select the master pages, then right-click and choose **Properties**.

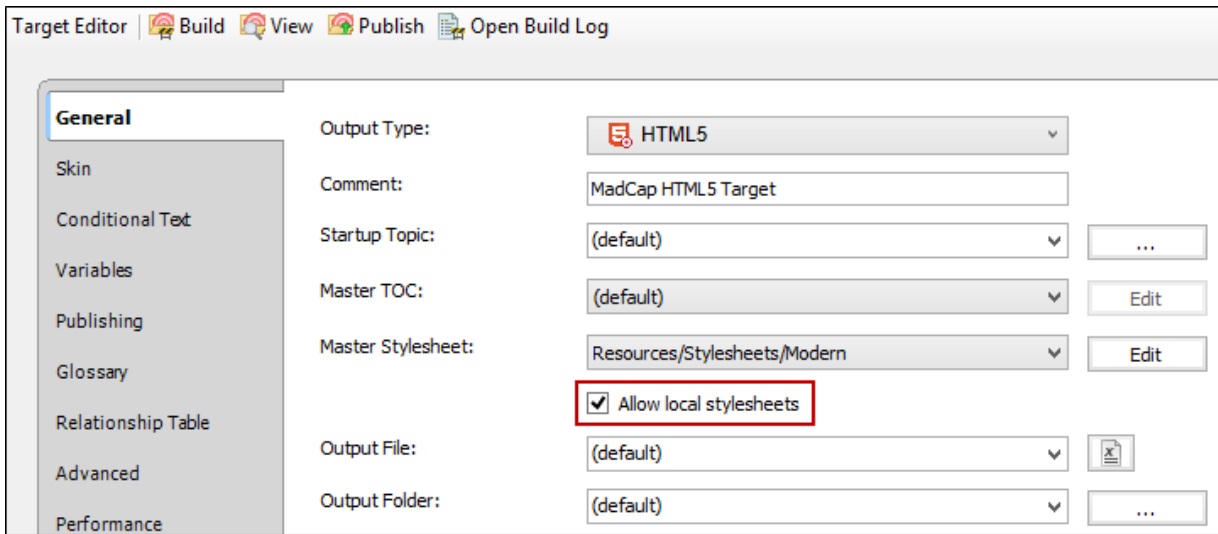
3. Click the **Stylesheet Links** tab.
4. On the left side, double-click the stylesheet you want to associate with the master page. That stylesheet is moved to the right side of the dialog. If you add multiple stylesheets, you can use the up and down arrows to change the order. This affects the preferred formatting in case a conflict exists between the stylesheets.



5. Click **OK**.

MASTER STYLESHEET (TARGETS)

In the Target Editor, you can open the **General** tab to choose a master stylesheet. The **Master Stylesheet** field has a check box under it labeled **Allow local stylesheets**.



The screenshot shows the Target Editor window with the General tab selected. The left sidebar contains a list of tabs: General, Skin, Conditional Text, Variables, Publishing, Glossary, Relationship Table, Advanced, and Performance. The main area displays the following settings:

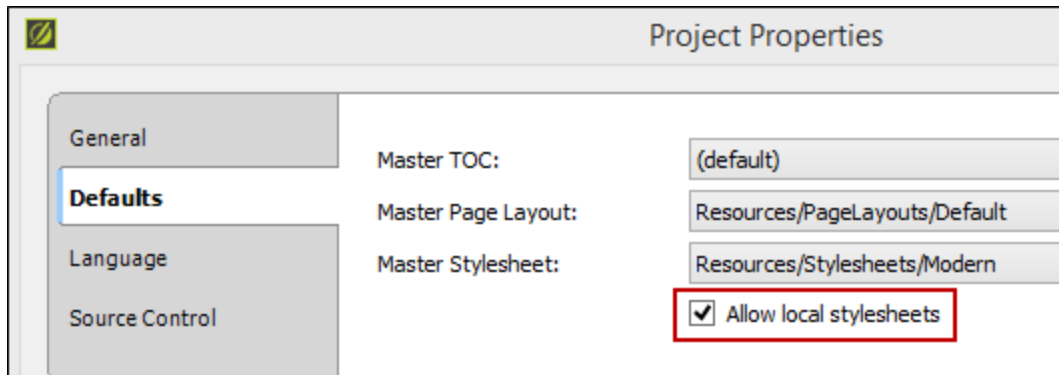
Field	Value	Action
Output Type:	HTML5	
Comment:	MadCap HTML5 Target	
Startup Topic:	(default)	...
Master TOC:	(default)	Edit
Master Stylesheet:	Resources/Stylesheets/Modern	Edit
<input checked="" type="checkbox"/> Allow local stylesheets		
Output File:	(default)	...
Output Folder:	(default)	...

If you select this check box, you will be allowed to set stylesheets locally on individual content files. Your master stylesheet will still be used as well (as long as you don't disable it from the content file's Properties dialog).

If you do not select this check box, only the master stylesheet will be used. And if you attempt to associate a stylesheet locally with a content file, you will see a message asking if you want to disable the master stylesheet.

MASTER STYLESHEET (PROJECT)

In the Project Properties dialog, you can open the **Defaults** tab to choose a master stylesheet. The **Master Stylesheet** field has a check box under it labeled **Allow local stylesheets**.



If you select this check box, you will be allowed to set stylesheets locally on individual content files. Your master stylesheet will still be used as well (as long as you don't disable it from the content file's Properties dialog).

If you do not select this check box, only the master stylesheet will be used. And if you attempt to associate a stylesheet locally with a content file, you will see a message asking if you want to disable the master stylesheet.

PRECEDENCE FOR STYLESHEETS

Because you are allowed to have multiple stylesheets set on different files and at different levels, you need to understand how precedence works, both in the XML Editor and the output.

PRECEDENCE FOR TOPICS IN XML EDITOR

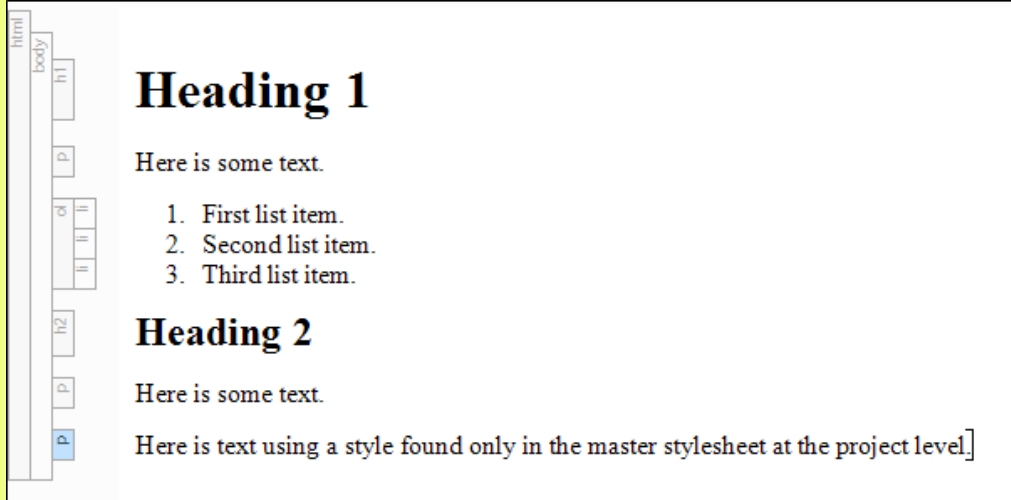
When you are working on a topic in the XML Editor, this is how precedence will work if you have multiple stylesheets.

1. Local Stylesheet Associated with Topic
2. Master Stylesheet (styles from only one master stylesheet can be used):
 - a. *Primary* Target
 - b. Project

EXAMPLE

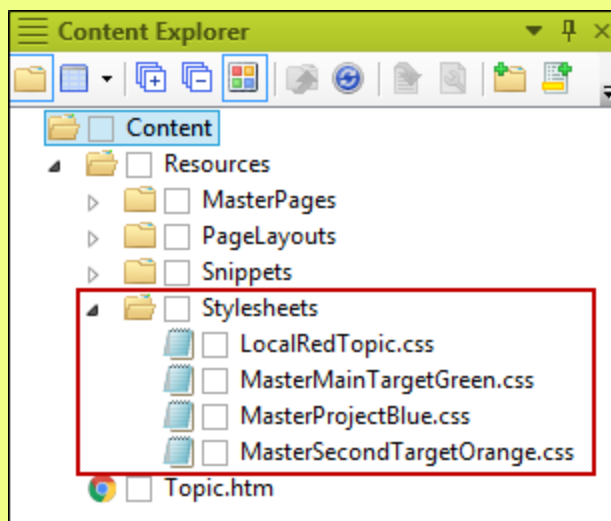
Let's say you have a project and you want to use multiple stylesheets in it.

When you first open a topic, it looks very plain, because no stylesheets are yet associated with in the project.



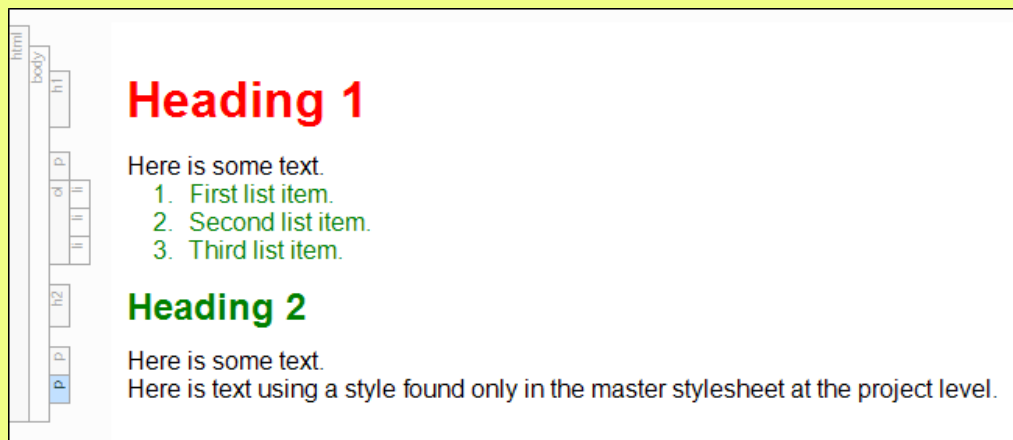
You add the following stylesheets and associate them at different levels:

- » **LocalRedTopic** You set this stylesheet locally on a topic. And you tell Flare to use a red font for the H1 style.
- » **MasterMainTargetGreen** You associate this stylesheet with your primary target (called "Main Target"). And you tell Flare to use a green font for the H1, H2, and ol styles.
- » **MasterSecondTargetOrange** You associate this stylesheet with another target (called "Second Target"). And you tell Flare to use an orange font for the H1, H2, and ol styles.
- » **MasterProjectBlue** You associate this stylesheet with the entire project. And you tell Flare to use a blue font for the H1, H2, and p styles.



For all of the master stylesheets, you tell Flare to allow local stylesheets. In other words, you can use any of the stylesheets in your project.

Now when you open the topic, it looks like this:



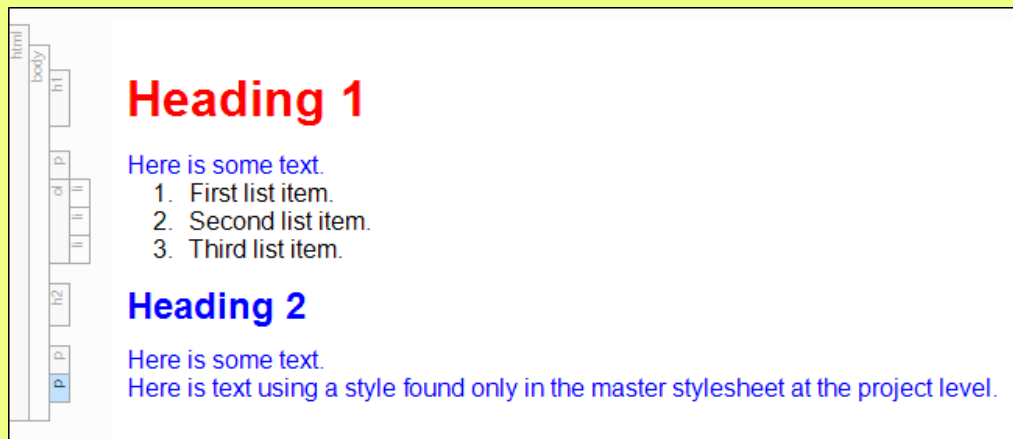
The first-level heading is red because that's what we set in the local stylesheet that is associated with the topic. The other stylesheets have different colors for this style, but the local stylesheet has precedence, so red wins.

The next stylesheet that has precedence is the one for the primary target that has green fonts. That's why you see the second-level heading and the numbered list (which uses the ol style) in green. We didn't set anything for the H2 and ol styles in the local stylesheet, so Flare goes to the next one (the primary target stylesheet) and uses its settings.

What about the regular paragraphs? They look better, but they're not displaying in any color. Well, the only stylesheet where we specified a setting for the p style is the master stylesheet at the project level. But because there's another master stylesheet on the primary target that has precedence, the blue from the project master stylesheet is completely ignored. Instead, Flare uses the default settings from a factory stylesheet where Flare is installed, and that's why the font family is different from the plain font that you first saw.

So what happens if you make a change? Suppose you remove the master stylesheet that is set on the primary target.

In that case, you would see this in the XML Editor:



The first-level heading is the same because it is coming from the local stylesheet. But now that the master stylesheet on the primary target is gone, Flare moves to the next master stylesheet at the project level, which shows the second-level heading and the regular paragraphs in blue. The numbered list no longer has a color because the ol style is not set in either the local stylesheet or the project master stylesheet. It is set in the secondary target, but Flare doesn't use those settings to display content in the XML Editor. It only uses the settings from the primary target. Of course, you can always use the drop-down field on the left side of the local toolbar in the XML Editor to preview the topic as it will be seen in the secondary target, and in that case you will see some orange text.

PRECEDENCE FOR OUTPUT

When you generate output, this is how precedence will work if you have multiple stylesheets.

1. Local Stylesheet Associated with Topic
2. Local Stylesheet Associated with Master Page

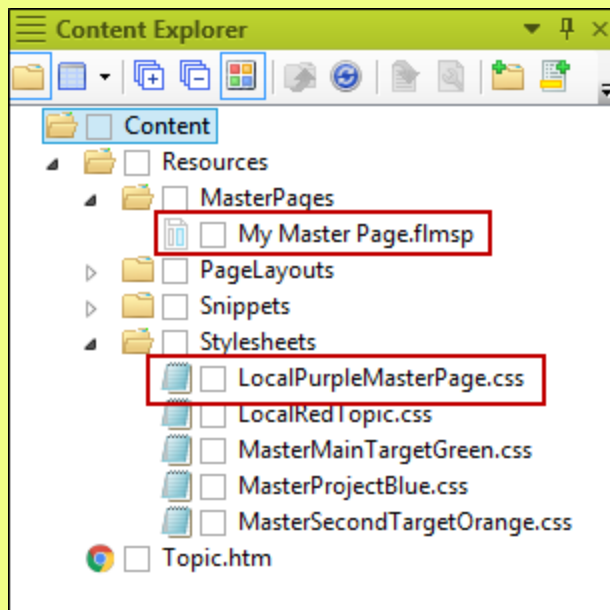
3. Master Stylesheet (styles from only one master stylesheet can be used):
 - a. *Any Target*
 - b. *Project*

EXAMPLE

Take a look at the previous example for precedence in the XML Editor before continuing with this example.

Let's say you add another stylesheet to your project to be used for a master page.

- » **LocalPurpleMasterPage** You set this stylesheet locally on a master page. And you tell Flare to use a purple font for the p style.



You've also reset the "MasterMainTargetGreen" stylesheet on your primary target. So you now have all of the stylesheets from the previous example set as they were, and you've added one more local stylesheet to the mix. The master page content does not come into play when you're working in the XML Editor, but what happens in the output? For the primary target, it initially it would look like this:

Heading 1

Here is some text.

1. First list item.
2. Second list item.
3. Third list item.

Heading 2

Here is some text.

Here is text using a style found only in the master stylesheet at the project level.

Master text

It's similar to what you saw in the previous example when looking at the XML Editor. But now you see that the regular paragraphs are purple. That's because the local stylesheet for the topic didn't specify a color, and the next stylesheet with precedence is the one used on the master page. You'll also see an extra paragraph at the bottom ("Master text"); this content is coming from the master page, not the topic.

If you generate the secondary target, it would look like this:

Heading 1

Here is some text.

1. First list item.
2. Second list item.
3. Third list item.

Heading 2

Here is some text.

Here is text using a style found only in the master stylesheet at the project level.

Master text

It's much like the other output, except the green text is now orange, because it's coming from the secondary target stylesheet, which has precedence over the master stylesheet at the project level.

But if you remove the master stylesheet link from either of the targets, the output will look like this:

Heading 1

Here is some text.

1. First list item.
2. Second list item.
3. Third list item.

Heading 2

Here is some text.

Here is text using a style found only in the master stylesheet at the project level.

Master text

Now that the target master stylesheets are out of the way, Flare looks to the master project stylesheet, which uses blue for the second-level heading. Most of the regular paragraphs remain purple because they are coming from the local master page stylesheet, which has a higher precedence. However, notice the long sentence near the end of the topic. That sentence is blue because it is actually using a class of the main paragraph style (p.SpecialClass). This class is found in the master project stylesheet, but not in the local stylesheet for the master page. That's why it's blue instead of purple.

If you remove the stylesheet link on the master page, the output will look like this:

Heading 1

Here is some text.

1. First list item.
2. Second list item.
3. Third list item.

Heading 2

Here is some text.

Here is text using a style found only in the master stylesheet at the project level.

Master text

Now we see more blue.

And finally, if you remove the local stylesheet link on the topic, the output will look like this:

Heading 1

Here is some text.

1. First list item.
2. Second list item.
3. Third list item.

Heading 2

Here is some text.

Here is text using a style found only in the master stylesheet at the project level.

Master text

Now it's all about the project-level master stylesheet, because all of the others are out of the way. But the numbered list is still getting its look from the default factory stylesheet because the project-level master stylesheet doesn't tell Flare to change it in any way.

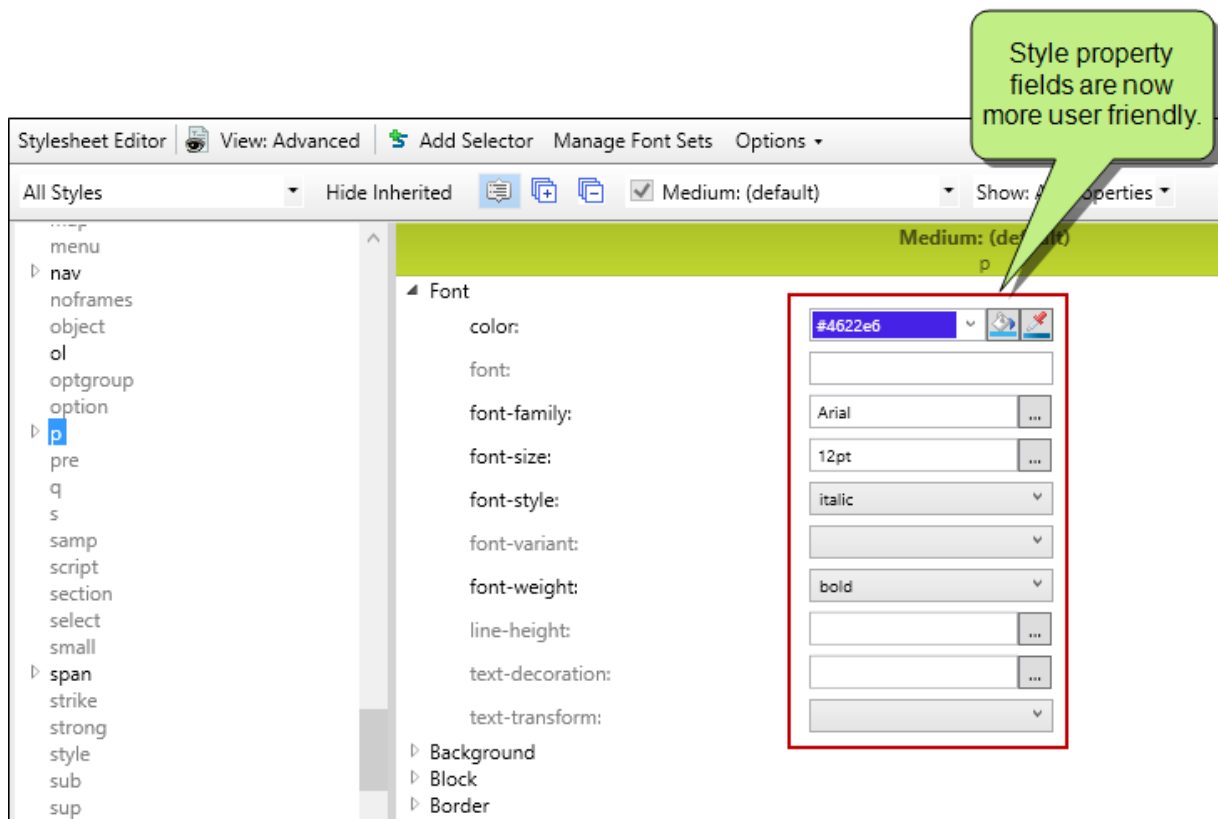


Note: Although it is possible to associate a stylesheet locally with a snippet, the only reason to do this is if you do not have any master stylesheets in your project. Without a master stylesheet, a snippet's content will look very plain when you open it. That's because Flare doesn't know which styles to use for it. In order to work in that snippet and apply styles to the content, you will need to associate the snippet with a stylesheet.

You can associate multiple snippets with a stylesheet (or remove a stylesheet from those snippets). The procedure is similar to the one used for associating a stylesheet with master pages. You can open the File List window pane (**View>File List**), hold down your **SHIFT** or **CTRL** key to select the snippets, then right-click and choose **Properties**. In the Properties dialog, select the **Stylesheet Links** tab and choose your stylesheet, moving it to the right side of the dialog.

Redesigned Display of Properties

One of the first changes you will notice in the Stylesheet Editor (Advanced view) is that the properties look different, making it easier to see what you are modifying.

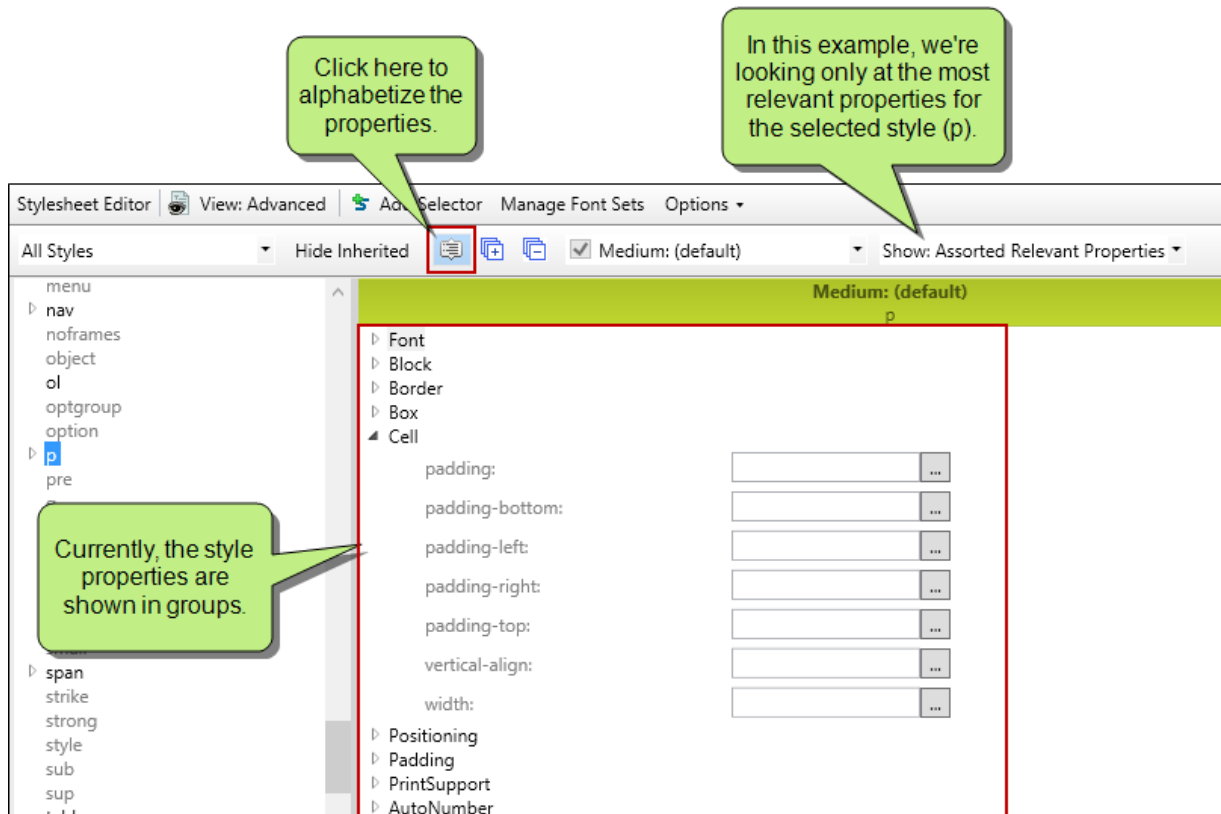


The following enhancements have also been made.

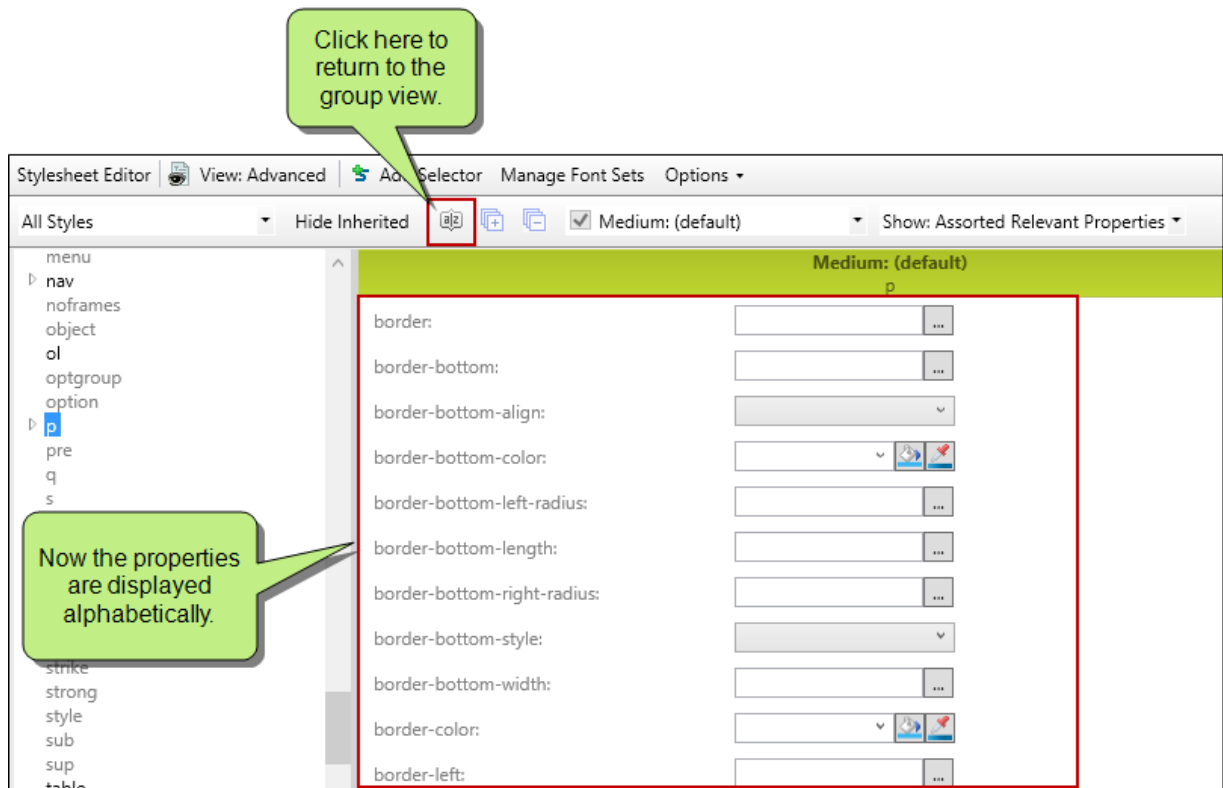
ALPHABETICAL VIEW ON ALL FILTERS

In previous versions, there was an alphabetical view option in the drop-down filter field. This has been moved to the local toolbar, allowing you to alphabetize any of the other filtered views: Set (Locally) Properties, Set Properties, Assorted Relevant Properties, and All Properties.

If you are viewing properties in groups, click . This displays the properties alphabetically.

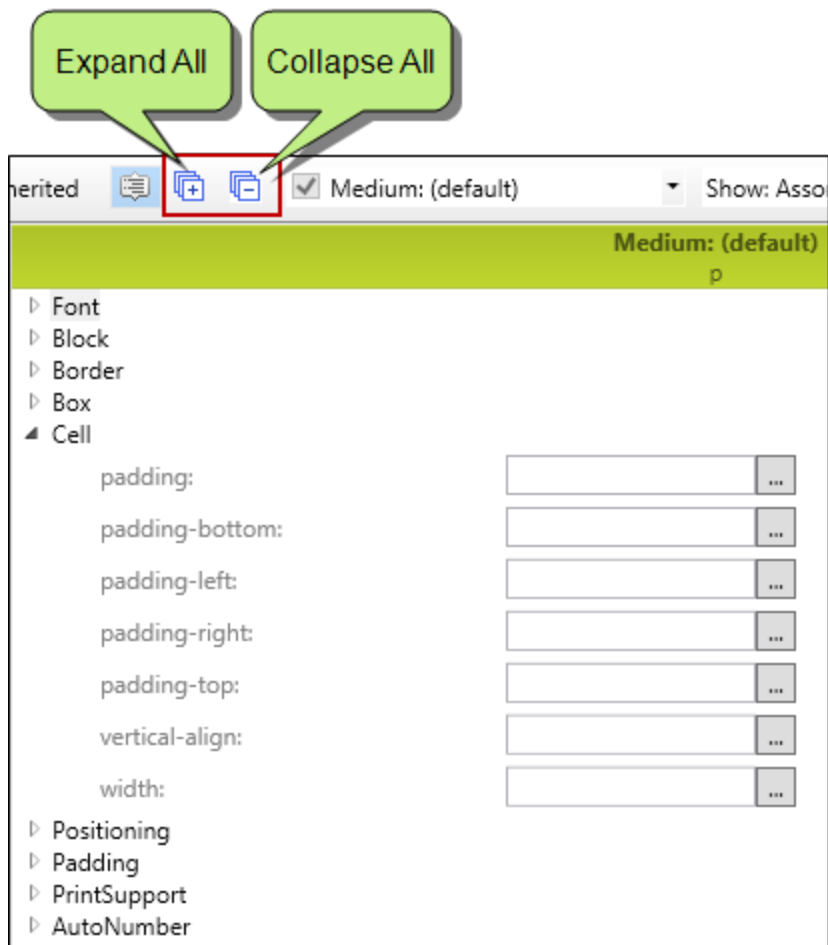


To return to the group view, click .



EXPAND/COLLAPSE ALL OPTIONS

If you are viewing properties in group view, you can use the "Expand All" and "Collapse All" buttons to expand or collapse all groups.

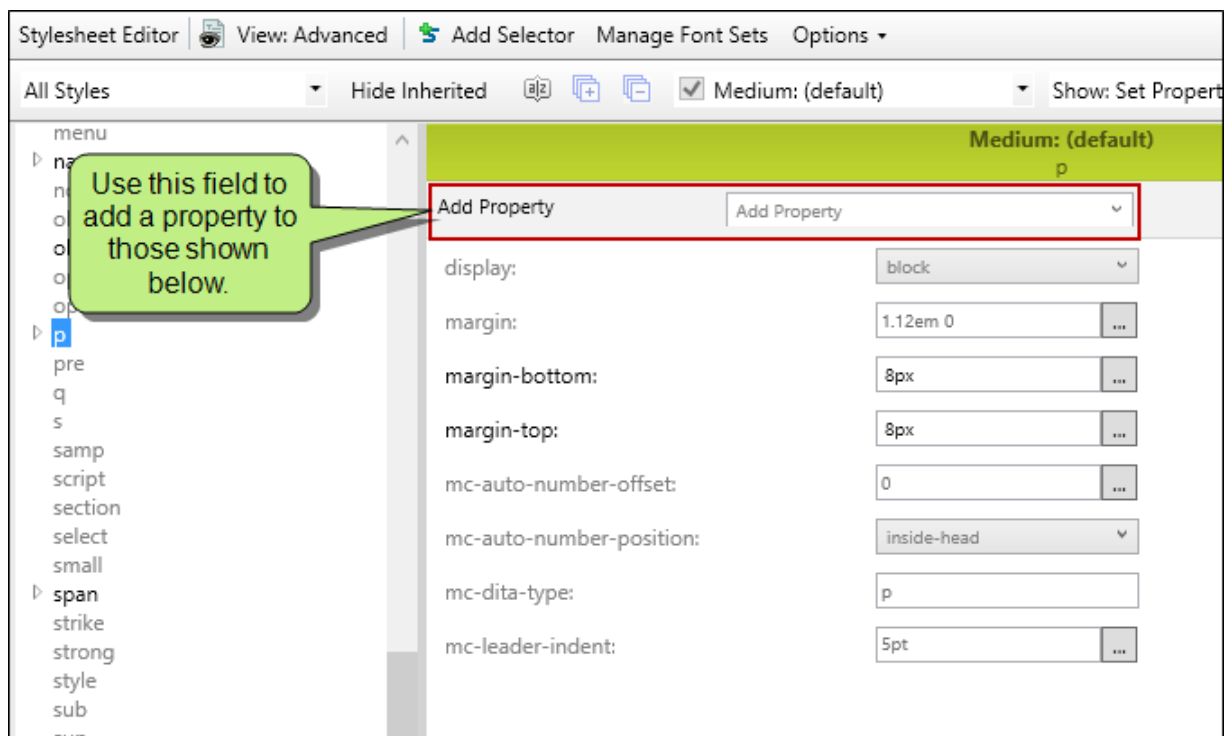


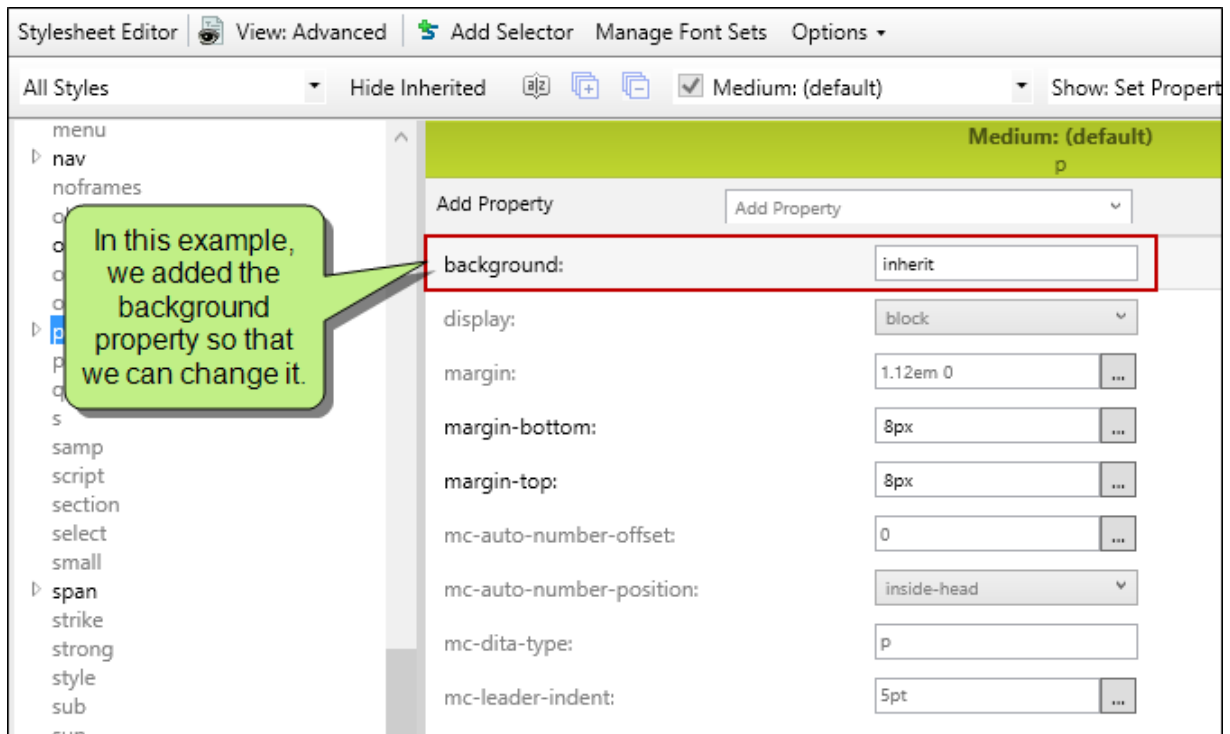
ADD PROPERTY CONTROL

If you are viewing style properties in one of the "Set" filtered views—"Set (Locally) Properties" or "Set Properties"—there is a field at the top that lets you add a property to the display.

- » **Set (Locally) Properties** Displays (in the Properties section below) properties that have been set explicitly in the stylesheet.
- » **Set Properties** Displays (in the Properties section below) all properties that have been set explicitly in the stylesheet. It will also show properties that have been set in an imported stylesheet or inherited properties that have been set in a factory stylesheet.

You can select a property from the drop-down or just type it in the field.





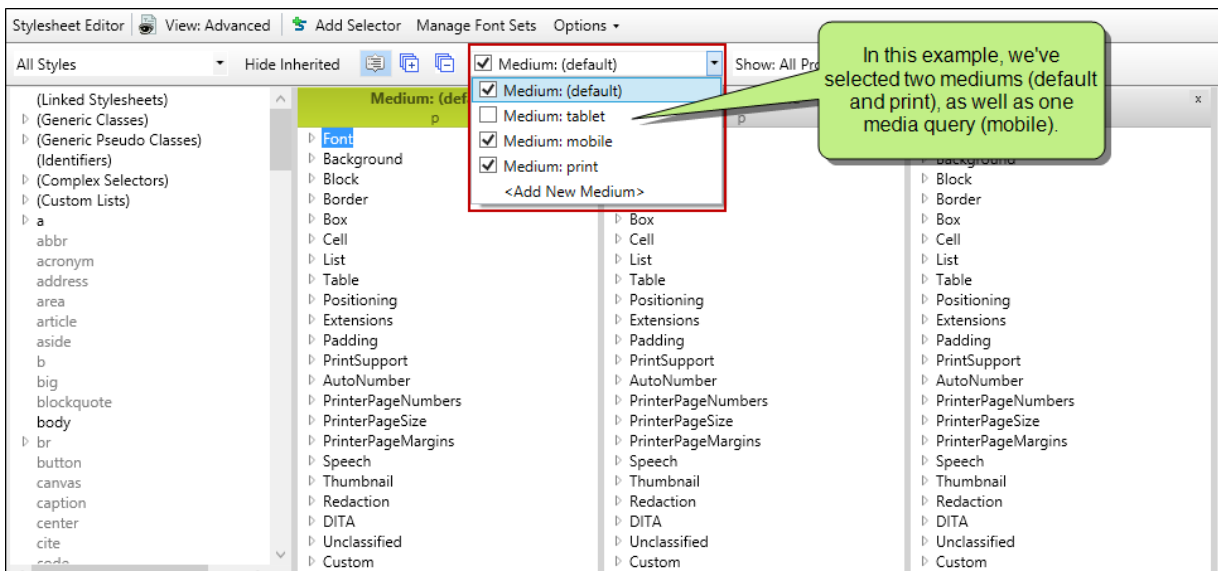
The name that you enter in this field must be a valid CSS property. You cannot enter just any text.

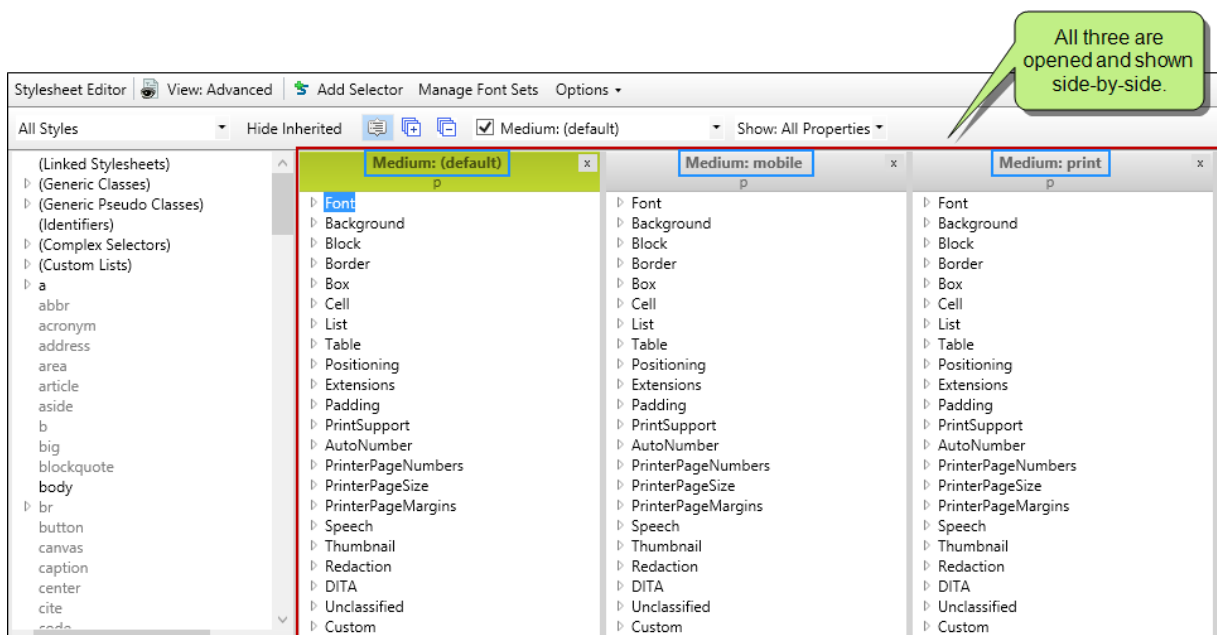
Multiple Medium View

If you are working in the Advanced View of the Stylesheet Editor, you can open and edit multiple mediums at same time. You can do this by clicking the **Medium** drop-down field in the local toolbar and choosing the mediums you want to see so that they have a check mark next to them. Media queries are also listed in the drop-down and can be opened at the same time as well.

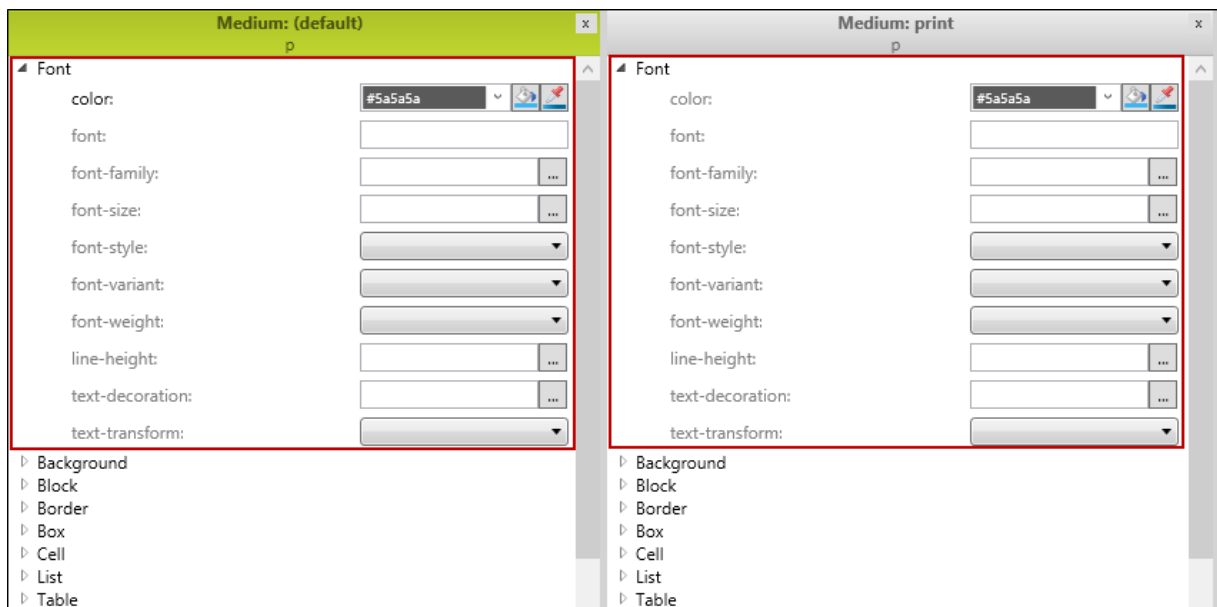
When choosing a medium/media query from the drop-down, you can click on the check mark or the name of the medium/media query. What happens next depends on how many mediums/media queries are currently open in the editor.

- » **If One Medium or Media Query is Open** If only one medium/media query is open and you click a *check box* next to another medium/media query, that second medium/media query will open next to the first one. However, if you select the *name* of the medium/media query in the drop-down, it will open and the first medium/media query will close.
- » **If Multiple Mediums or Media Queries are Open** If two or more mediums/media queries are open, the next medium/media query you select will open next to the others. This is true whether you select the check box or the name of the medium/media query.

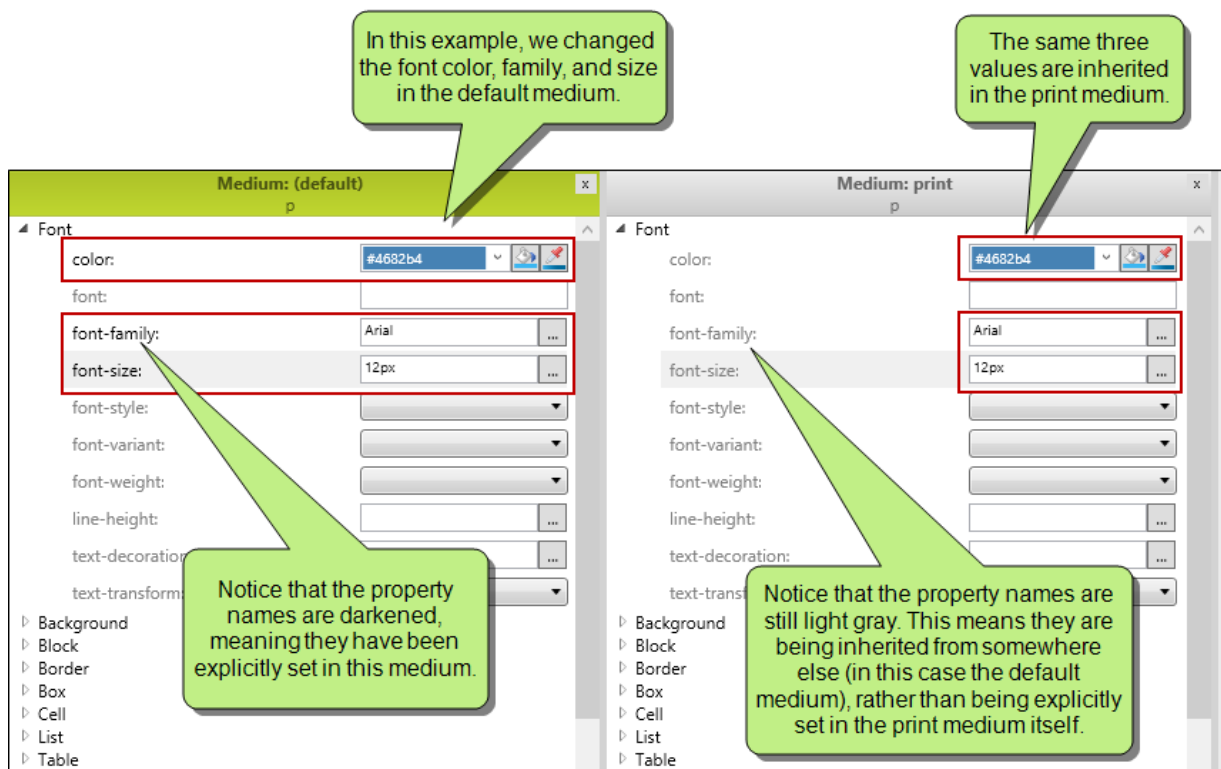




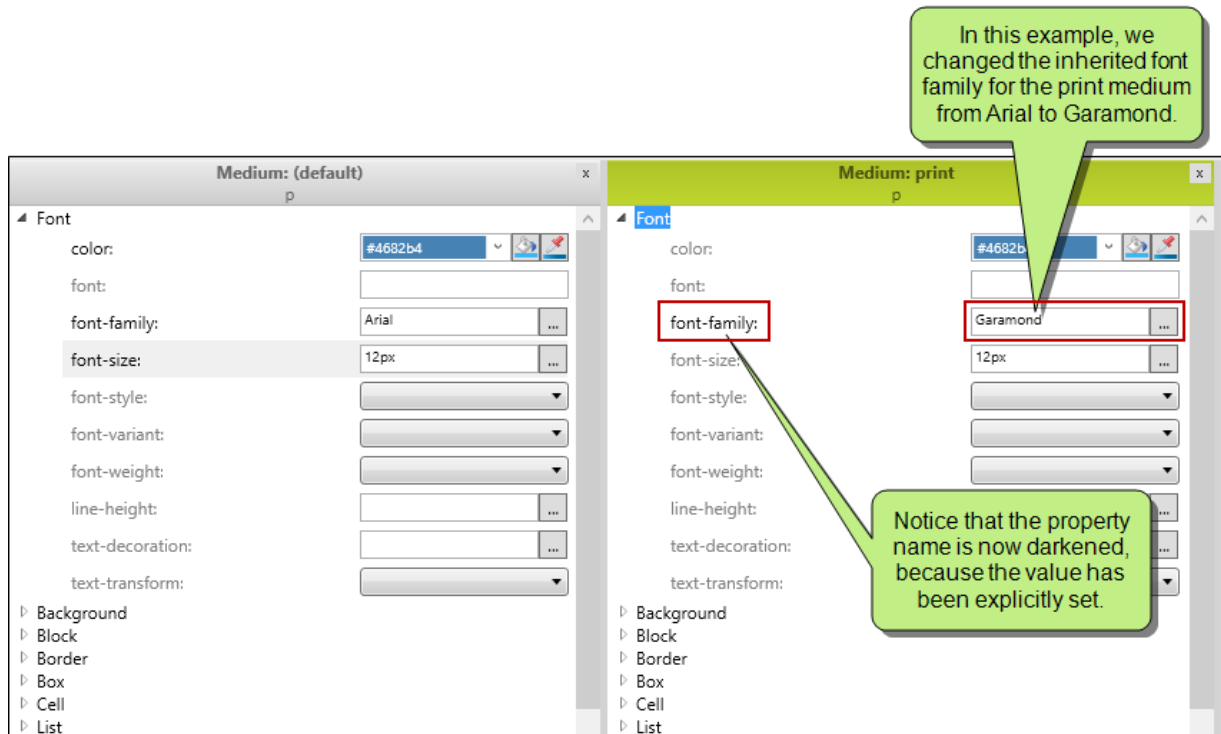
When you expand a property group in one medium or media query, the same property group also expands in any others that are open.



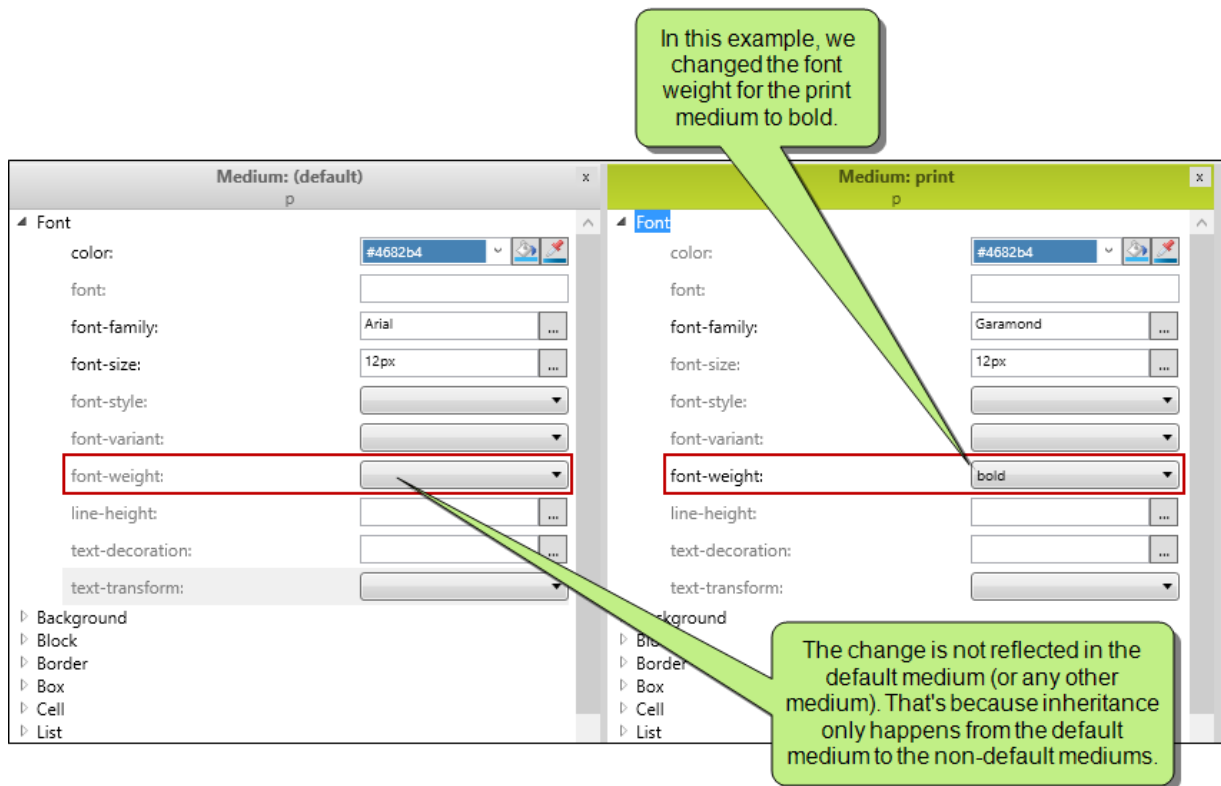
If you make a change in default medium, you see it applied also to the other mediums and media queries, because they inherit whatever is added in the default medium.



Of course, you can override any inherited property value in a specific medium or media query.

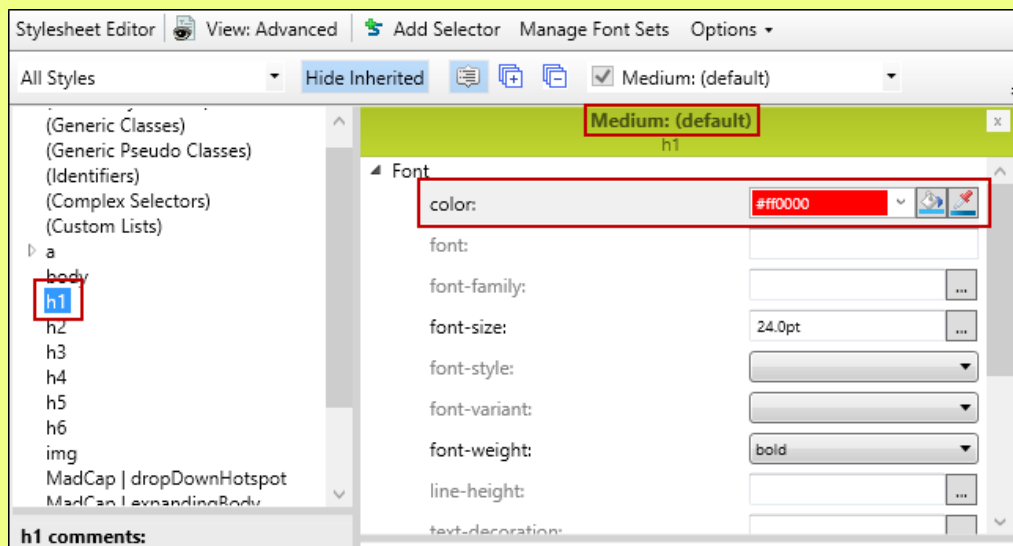


If you make a change in a non-default medium or in a media query, you see it only in that place.

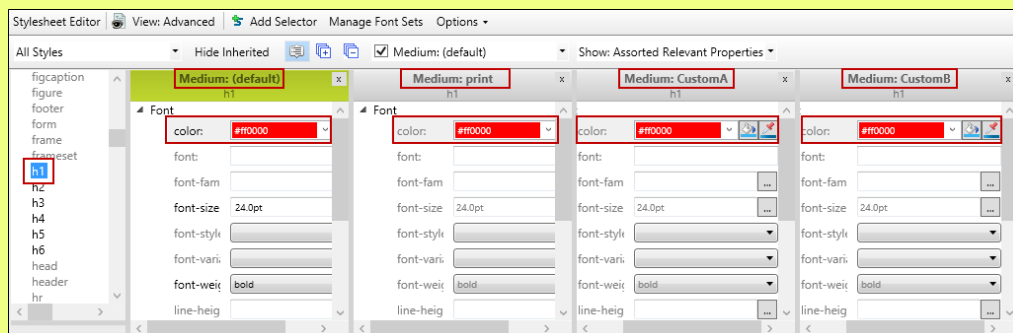


EXAMPLE

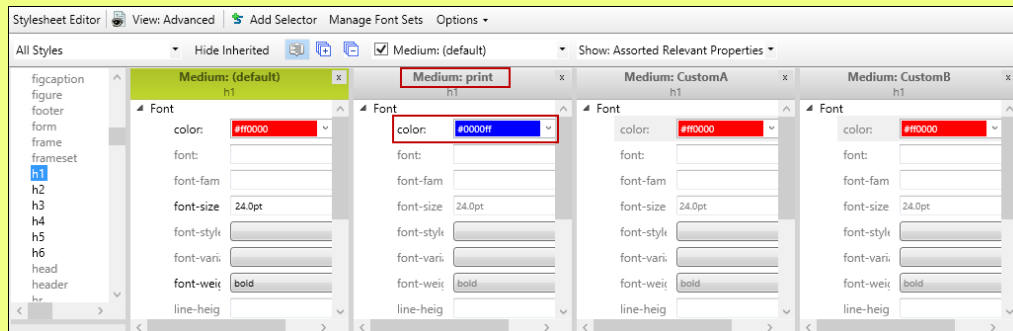
Suppose you are using four different mediums (default, print, CustomA, CustomB) in your project. Let's say you specify that the font color for the h1 style in the default medium should be red.



If you were to then open any of the other three mediums, you would see that the font color for the h1 style in each of those is also red.



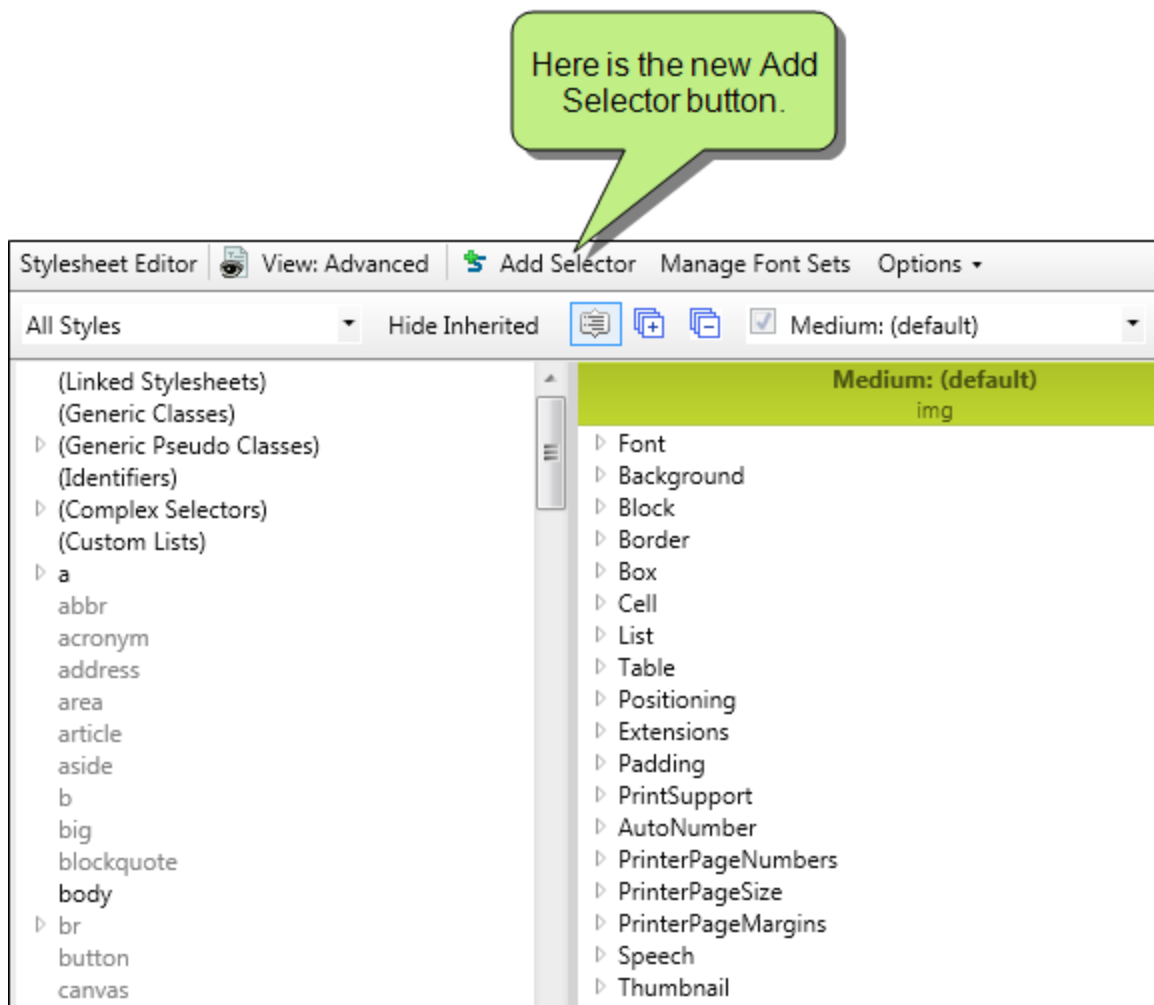
If you were to open the print medium and change the font color for the h1 style to blue, that color will be used for that medium. However, the h1 style will continue to be displayed in red for the default, CustomA, and CustomB mediums.



Note: To access the mediums and media queries provided by Flare (print, tablet, mobile), you might need to make sure the **Hide Inherited** option in the local toolbar of the Stylesheet Editor is *not* selected. However, this is not necessary once you make an explicit change in one of those mediums or media queries; after that, it will show up in the Medium drop-down whether you use the Hide Inherited option or not.

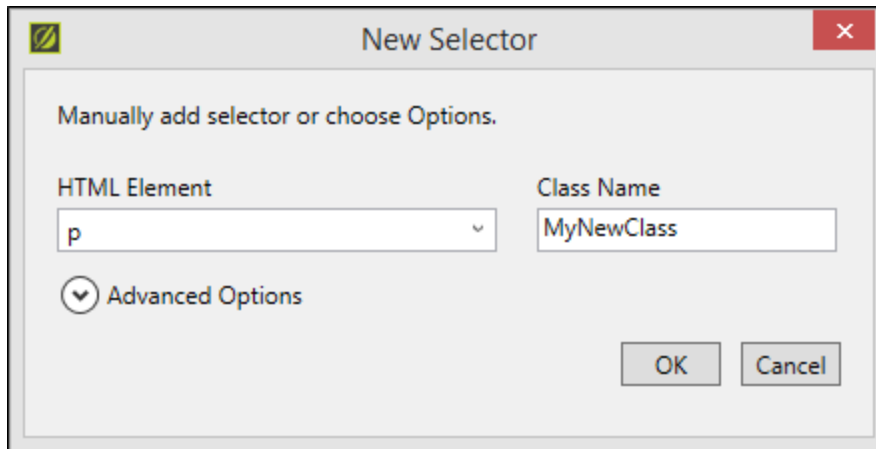
Add Selector Button Results in More Options

In previous versions, there were buttons in the local toolbar of the Stylesheet editor to add new classes or pseudo classes. These have been replaced by a button labeled "Add Selector." (You can also access this option by right-clicking a style.)



Not only does this new button combine the processes for adding new classes and pseudo classes, but it also allows you to use additional options.

When you click this button, the New Selector dialog opens. The selected element (or "parent style") is shown in the first field, but you can choose a different one from the drop-down field. In the second field, you can add a new class for the element.



You can click **Advanced Options** to enhance the style even more by creating an advanced selector, including pseudo classes and identifiers (IDs).

ADVANCED SELECTORS

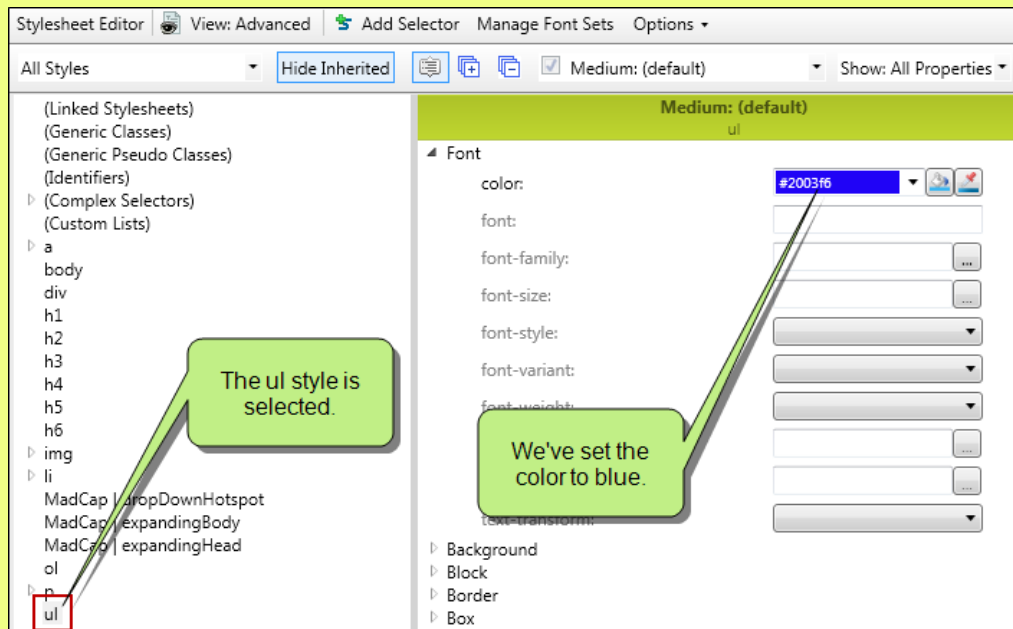
An advanced (or "complex") selector lets you format content based on very specific criteria. There are many ways to create advanced selectors in CSS. For details, see:

http://www.w3.org/community/webed/wiki/Advanced_CSS_selectors

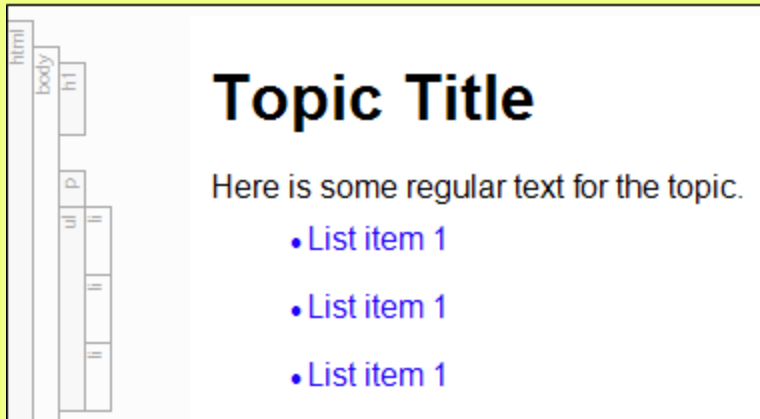
You can use the New Selector dialog in Flare to create advanced selectors by entering them directly in the Advanced Selector field. You would need to do this, for example, if you need to create a descendant selector. This is a selector that applies formatting when one selector is found within another.

EXAMPLE

Let's say you style your unordered (bulleted) lists so that the text is blue.



Here is how it looks
in the XML Editor.



But if an unordered list (tag) is found within a numbered list (tag) in the markup, you want the bulleted list text to be green. To make this happen, you would create a descendant selector by separating the two selectors by white space, with the ol selector first.

New Selector

Manually add selector or choose Options.

HTML Element: ol

Class Name:

Advanced Options

Advanced Selector: ol ul

Pseudo Class:

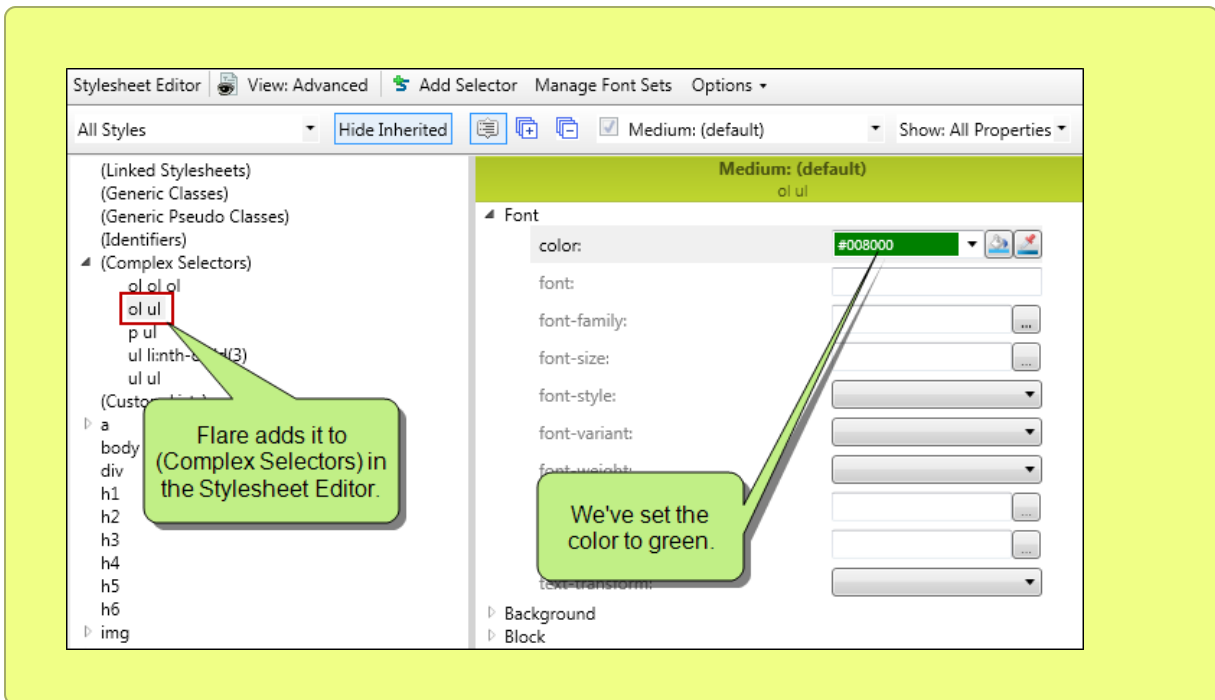
Pseudo Class Expression:

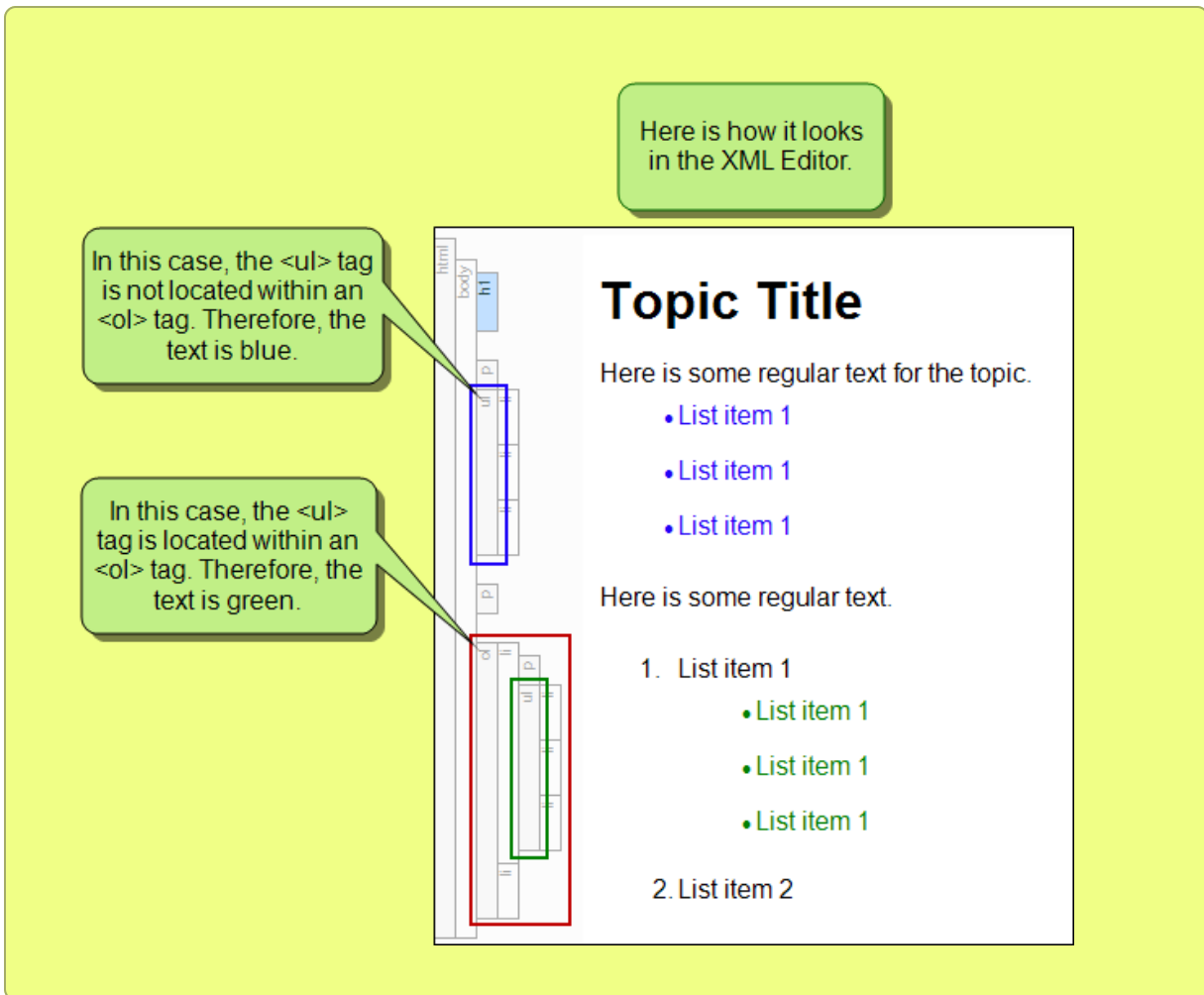
Pseudo Element:

Identifier (ID):

Comments:

OK Cancel





You can also create advanced selectors by completing the various fields in the New Selector dialog. As you complete the different fields, the Advanced Selector field is populated accordingly. The reverse is also true. As you enter text in the Advanced Selector field directly, the other fields in the dialog are automatically populated.

CLASS

In CSS there are primary styles that correspond to the different HTML elements (e.g., h1, h2, p, img). You can think of these as parent styles, because in a way, they can have children. A class is the most common type of child for a style. Some classes might already be included in your stylesheet when you first create a project.

You cannot create new parent styles, but you can create a class under any of those parent styles to give you more variety and flexibility.

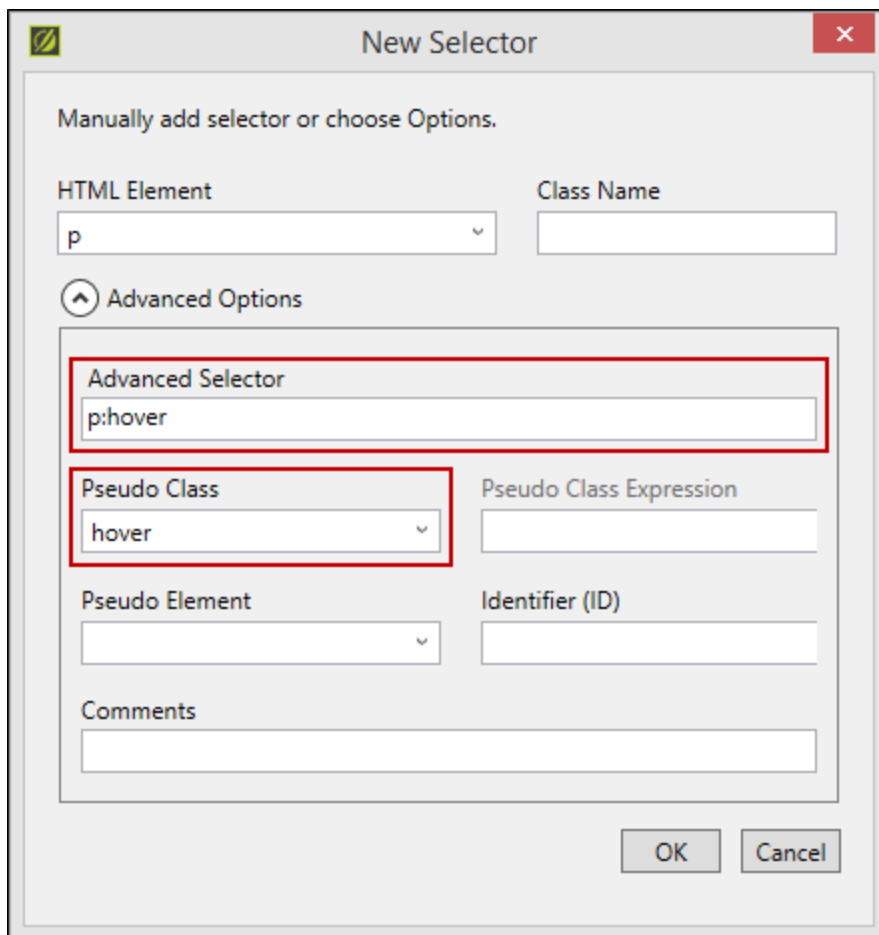
To create a class, you simply enter text in the **Class Name** field. Alternatively, you can type the main HTML element (e.g., p, h1) followed by a period and the name of your class in the **Advanced Selector** field.

The image shows a 'New Selector' dialog box. At the top, it says 'Manually add selector or choose Options.' Below this, there are two input fields: 'HTML Element' with a dropdown menu showing 'p', and 'Class Name' with a text field containing 'Tip'. Below these is an 'Advanced Options' section. The 'Advanced Selector' field is highlighted with a red box and contains 'p.Tip'. Below this are four more fields: 'Pseudo Class' (dropdown), 'Pseudo Class Expression' (text), 'Pseudo Element' (dropdown), and 'Identifier (ID)' (text). At the bottom is a 'Comments' text area.

PSEUDO CLASS

Pseudo classes are a special group of style classes in CSS that pertain to elements when they're in a certain state (e.g., the font turns orange when a user hovers over it). They are often (but not exclusively) used for styles associated with hyperlinks.

In the New Selector dialog you can use the **Pseudo Class** field to enter or select a pseudo class for the HTML element. In the **Advanced Selector** field, the main HTML element is followed by a colon and then the pseudo class.



The image shows a 'New Selector' dialog box with a title bar containing a maximize icon, the text 'New Selector', and a close button. The main content area has the instruction 'Manually add selector or choose Options.' Below this are two input fields: 'HTML Element' with a dropdown menu showing 'p' and 'Class Name' with an empty text box. A section titled 'Advanced Options' is expanded, showing several fields: 'Advanced Selector' with a text box containing 'p:hover', 'Pseudo Class' with a dropdown menu showing 'hover', 'Pseudo Class Expression' with an empty text box, 'Pseudo Element' with a dropdown menu, and 'Identifier (ID)' with an empty text box. At the bottom of the dialog is a 'Comments' text box and two buttons labeled 'OK' and 'Cancel'.

Manually add selector or choose Options.

HTML Element: p

Class Name:

Advanced Options

Advanced Selector: p:hover

Pseudo Class: hover

Pseudo Class Expression:

Pseudo Element:

Identifier (ID):

Comments:

OK Cancel

PSEUDO CLASS EXPRESSION

For a handful of pseudo classes, you can also add an expression. If you select one of the valid pseudo classes (e.g., `nth-child`, `not`), you can then enter something in the **Pseudo Class Expression** field (e.g., `3`, `5n+5`, `odd`, `even`).

EXAMPLE

Let's say you want to show the third item in every bulleted list in a blue font.

To accomplish this, you select the **ul** (unordered list) style in the Stylesheet Editor and click the **New Selector** button. Then you click the **Advanced Options** arrow to show the fields at the bottom of the dialog. The Advanced Selector field starts out showing only your main HTML element (`ul`).

The screenshot shows the 'New Selector' dialog box. It has a title bar with a close button. The main content area is titled 'Manually add selector or choose Options.' Below this, there are two main sections. The top section has a 'HTML Element' dropdown menu with 'ul' selected, and a 'Class Name' text input field. The bottom section is titled 'Advanced Options' and is expanded. It contains an 'Advanced Selector' text input field with 'ul' entered. Below this are four more fields: 'Pseudo Class' (dropdown), 'Pseudo Class Expression' (text input), 'Pseudo Element' (dropdown), and 'Identifier (ID)' (text input). Red boxes highlight the 'HTML Element' dropdown and the 'Advanced Selector' text input field.

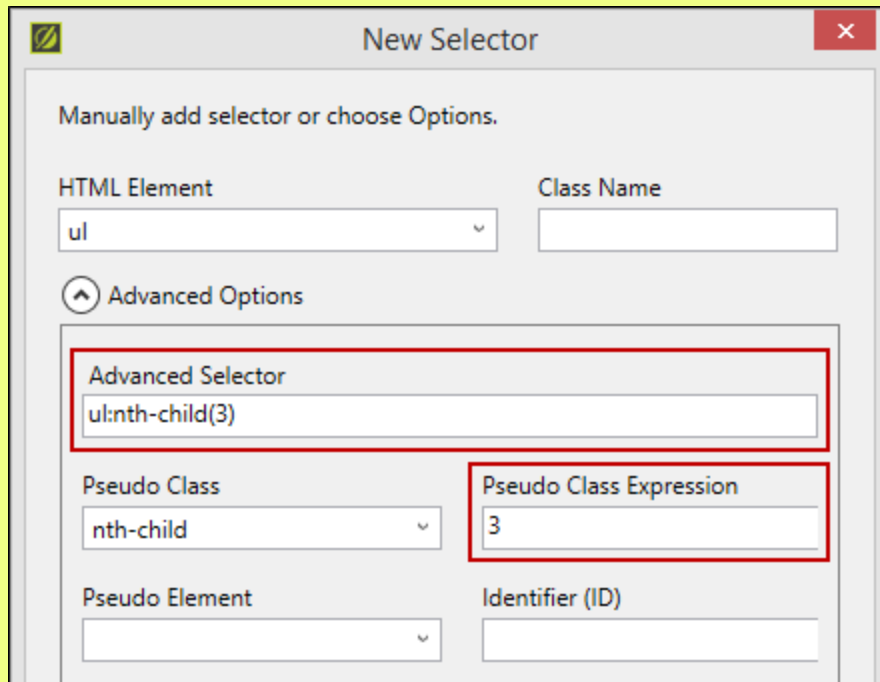
From the **Pseudo Class** field you select **nth-child**. This adds that pseudo class to the Advanced Selector field, after a colon.

The image shows a 'New Selector' dialog box with a close button (X) in the top right corner. The dialog contains the following fields and sections:

- Manually add selector or choose Options.**
- HTML Element:** A dropdown menu with 'ul' selected.
- Class Name:** An empty text input field.
- Advanced Options:** A section with an expand/collapse arrow icon.
- Advanced Selector:** A text input field containing 'ul:nth-child'.
- Pseudo Class:** A dropdown menu with 'nth-child' selected.
- Pseudo Class Expression:** An empty text input field.
- Pseudo Element:** A dropdown menu.
- Identifier (ID):** An empty text input field.

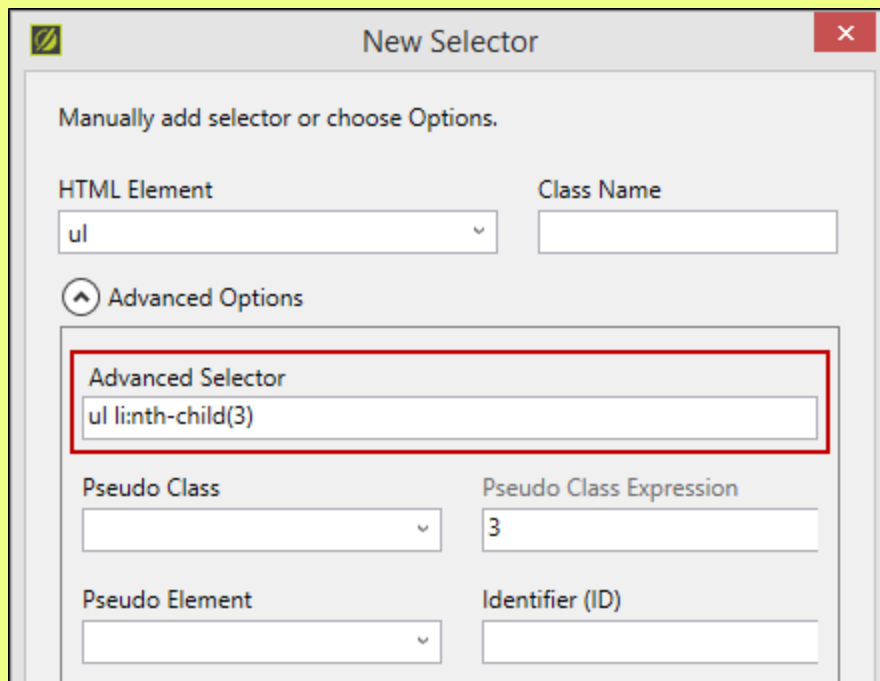
Red rectangular boxes highlight the 'Advanced Selector' field (containing 'ul:nth-child'), the 'Pseudo Class' dropdown (showing 'nth-child'), and the 'Pseudo Class Expression' field.

In the **Pseudo Class Expression**, you type 3 (because you only want the third list item to be affected). This adds the number between parentheses in the Advanced Selector field.



The screenshot shows a 'New Selector' dialog box with a close button in the top right corner. The main instruction is 'Manually add selector or choose Options.' Below this, there are two input fields: 'HTML Element' with a dropdown menu showing 'ul', and 'Class Name' with an empty text box. A section titled 'Advanced Options' is expanded, showing four fields: 'Advanced Selector' with the text 'ul:nth-child(3)', 'Pseudo Class' with a dropdown menu showing 'nth-child', 'Pseudo Class Expression' with the text '3', and 'Pseudo Element' with a dropdown menu. The 'Identifier (ID)' field is empty. Red rectangular boxes highlight the 'Advanced Selector' field and the 'Pseudo Class Expression' field.

You're almost done, but there is one more thing to do. In the Advanced Selector field, you need to place your cursor between the "ul" text and the colon, and you need to type a space followed by `li`. This tells Flare that it's not just the unordered list (ul) in general that this applies to, but rather to a specific item (li) within that list.



New Selector

Manually add selector or choose Options.

HTML Element:

Class Name:

Advanced Options

Advanced Selector:

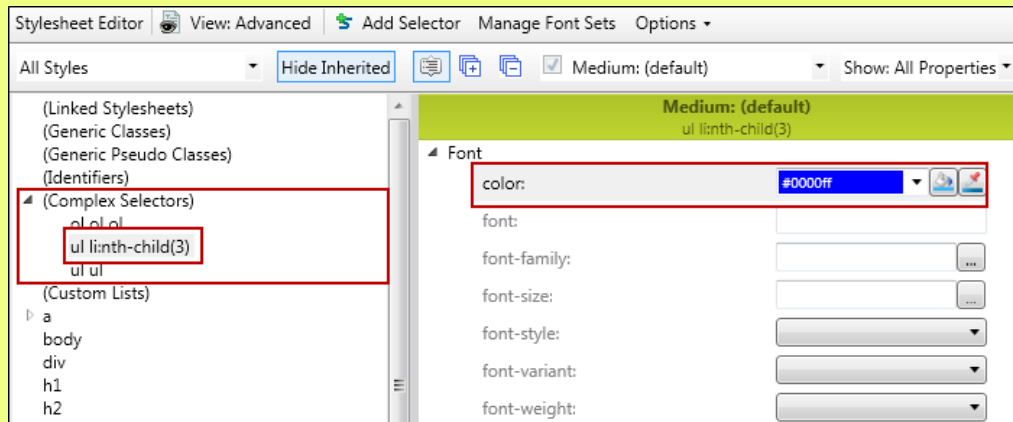
Pseudo Class:

Pseudo Class Expression:

Pseudo Element:

Identifier (ID):

After you click **OK**, the new advanced selector is added to your stylesheet. In the Advanced view of the Stylesheet Editor, it will be shown under (Complex Selectors). With this advanced selector highlighted, you change the font color to blue.

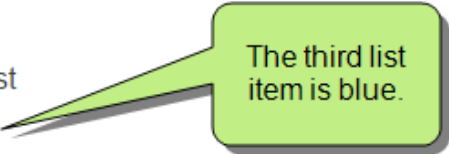


And this is what you get as a result in the output:

Simple Bulleted List

Here is some general text for a topic. Replace this with your own content.

- » First item in list
- » Second item in list
- » [Third item in list](#)
- » Fourth item in list
- » Fifth item in list
- » Sixth item in list
- » Seventh item in list



The third list
item is blue.

PSEUDO ELEMENT

In addition to pseudo classes, you can add pseudo elements to a style. Whereas a pseudo class focuses on the state of an element (e.g., change font color when hovered), a pseudo element focuses on a specific part of an element.

In the New Selector dialog you can use the **Pseudo Element** field to make a selection. In the **Advanced Selector** field, the two colons are added, followed by the pseudo element.

The screenshot shows the 'New Selector' dialog box. It has a title bar with a close button. The main area is titled 'Manually add selector or choose Options.' Below this, there are two input fields: 'HTML Element' (a dropdown menu showing 'p') and 'Class Name' (an empty text box). Below these is a section titled 'Advanced Options' with a collapse/expand icon. Inside this section, there is a text box labeled 'Advanced Selector' containing 'p::after'. Below this are two more input fields: 'Pseudo Class' (a dropdown menu) and 'Pseudo Class Expression' (a text box). Below these are two more input fields: 'Pseudo Element' (a dropdown menu showing 'after') and 'Identifier (ID)' (a text box). At the bottom of the 'Advanced Options' section is a 'Comments' text box. At the bottom right of the dialog are 'OK' and 'Cancel' buttons. Red boxes highlight the 'Advanced Selector' field, the 'Pseudo Element' dropdown, and the 'Pseudo Class' dropdown.

EXAMPLE

Let's say you want the first paragraph in some of your topics to start out with a letter that is larger and bolder than the rest of the characters.

To accomplish this, you select the **p** (paragraph) style in the Stylesheet Editor and click the **New Selector** button. In the **Class Name** field, you enter some text, such as `FirstParagraph`. Now you've got a style class that lets you make some paragraphs different from the rest.

Then you click the **Advanced Options** arrow to show the fields at the bottom of the dialog. The Advanced Selector field starts out showing only your main HTML element, followed by a period and the name of your class (`p.FirstParagraph`).

New Selector

Manually add selector or choose Options.

HTML Element: p

Class Name: FirstParagraph

Advanced Options

Advanced Selector: p.FirstParagraph

Pseudo Class: [dropdown]

Pseudo Class Expression: [text field]

Pseudo Element: [dropdown]

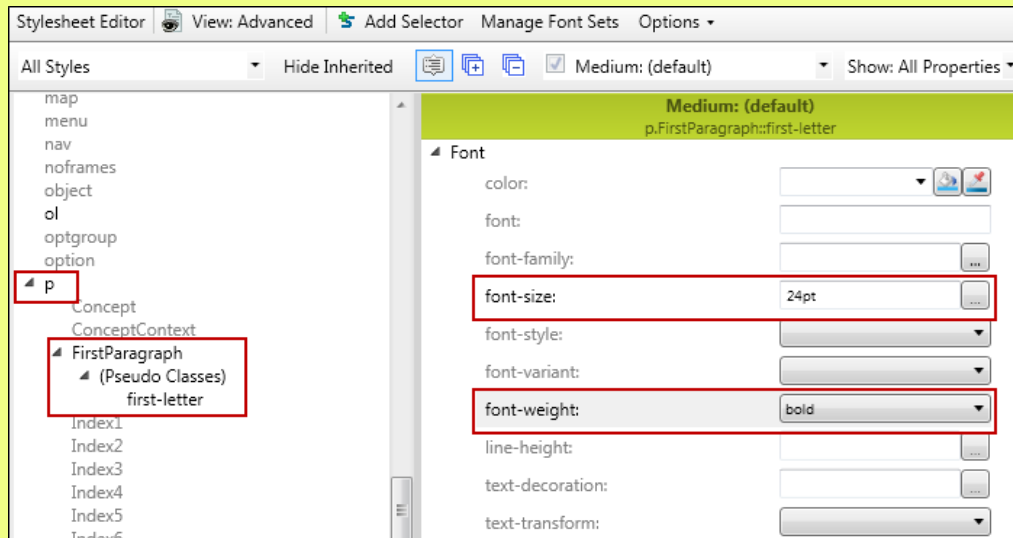
Identifier (ID): [text field]

From the **Pseudo Element** field you select **first-letter**. This adds that pseudo class to the Advanced Selector field, after two colons.

The screenshot shows a 'New Selector' dialog box with the following fields and values:

- HTML Element:** p
- Class Name:** FirstParagraph
- Advanced Options:**
 - Advanced Selector:** p.FirstParagraph::first-letter
 - Pseudo Class:** (empty)
 - Pseudo Class Expression:** (empty)
 - Pseudo Element:** first-letter
 - Identifier (ID):** (empty)

After you click **OK**, the new advanced selector is added to your stylesheet. With this the first-letter pseudo element highlighted, you change the font size to **24 pt** and the weight to **bold**.



Finally, for any paragraph that you want to use this advanced selector, you apply it in the XML Editor.

And this is what you get as a result in the output:

Pseudo Element Example

Here is the first paragraph of this topic. We've applied a style class to it so that the first letter is larger than the others.

This is the second paragraph of this topic. Here is some more text.

This is the third paragraph of this topic. Here is some more text.

IDENTIFIER

Identifiers (IDs) are similar to classes, except that IDs are unique. An element in your stylesheet can have only one ID on it, whereas it can have multiple classes. And each page of your output can have only one element with a particular ID. For many authors, using an ID may not be important, but for others—such as those making use of Javascript—IDs can be very useful.

In the New Selector dialog use the **Identifier (ID)** field to give the ID a name. In the **Advanced Selector** field, the ID name is added after a hash tag.

As with a class, an ID can be added after a specific HTML element, such as an h1 style...

The image shows a 'New Selector' dialog box with the following fields and values:

- HTML Element:** A dropdown menu with 'h1' selected.
- Class Name:** An empty text field.
- Advanced Options:** A section with a blue arrow icon and the text 'Advanced Options'.
- Advanced Selector:** A text field containing 'h1#chapter1'.
- Pseudo Class:** A dropdown menu.
- Pseudo Class Expression:** A text field.
- Pseudo Element:** A dropdown menu.
- Identifier (ID):** A text field containing 'chapter1'.
- Comments:** A text field.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

... or like a class, it can be generic, standing alone.

New Selector

Manually add selector or choose Options.

HTML Element

Class Name

Advanced Options

Advanced Selector

#header-section

Pseudo Class

Pseudo Class Expression

Pseudo Element

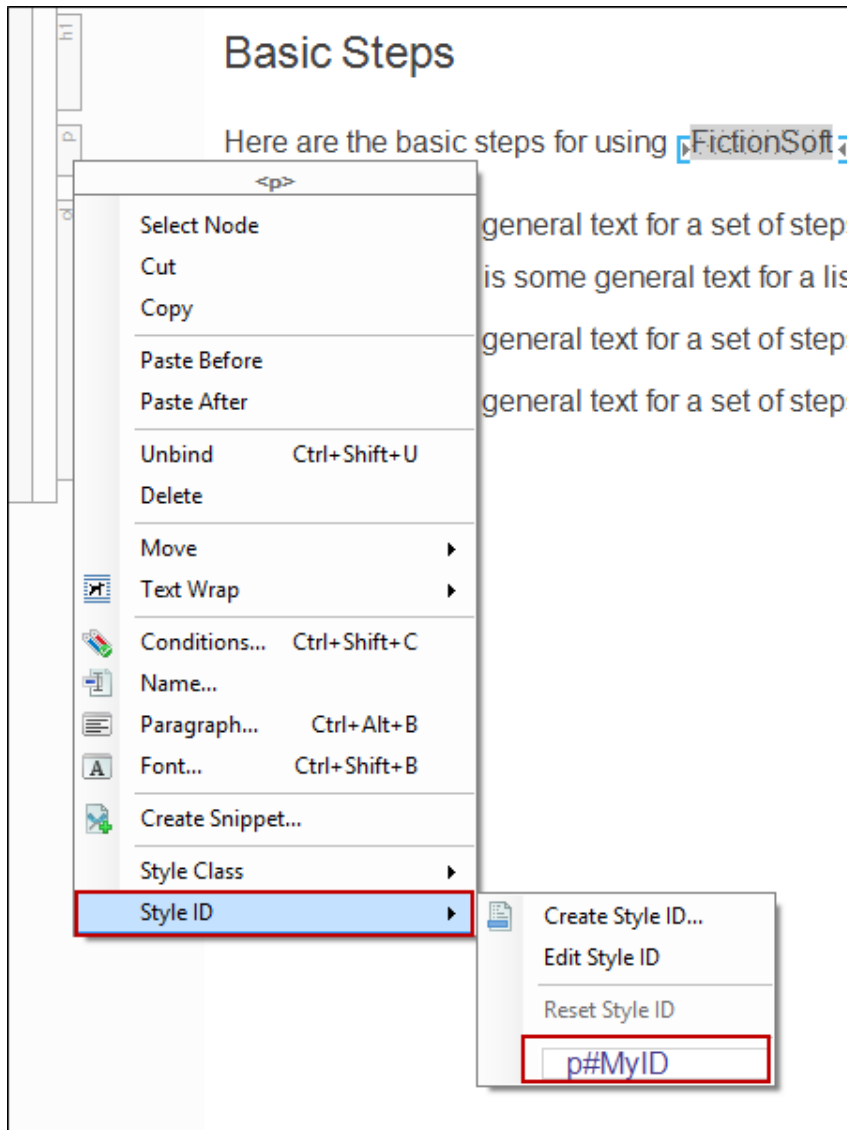
Identifier (ID)

header-section

Comments

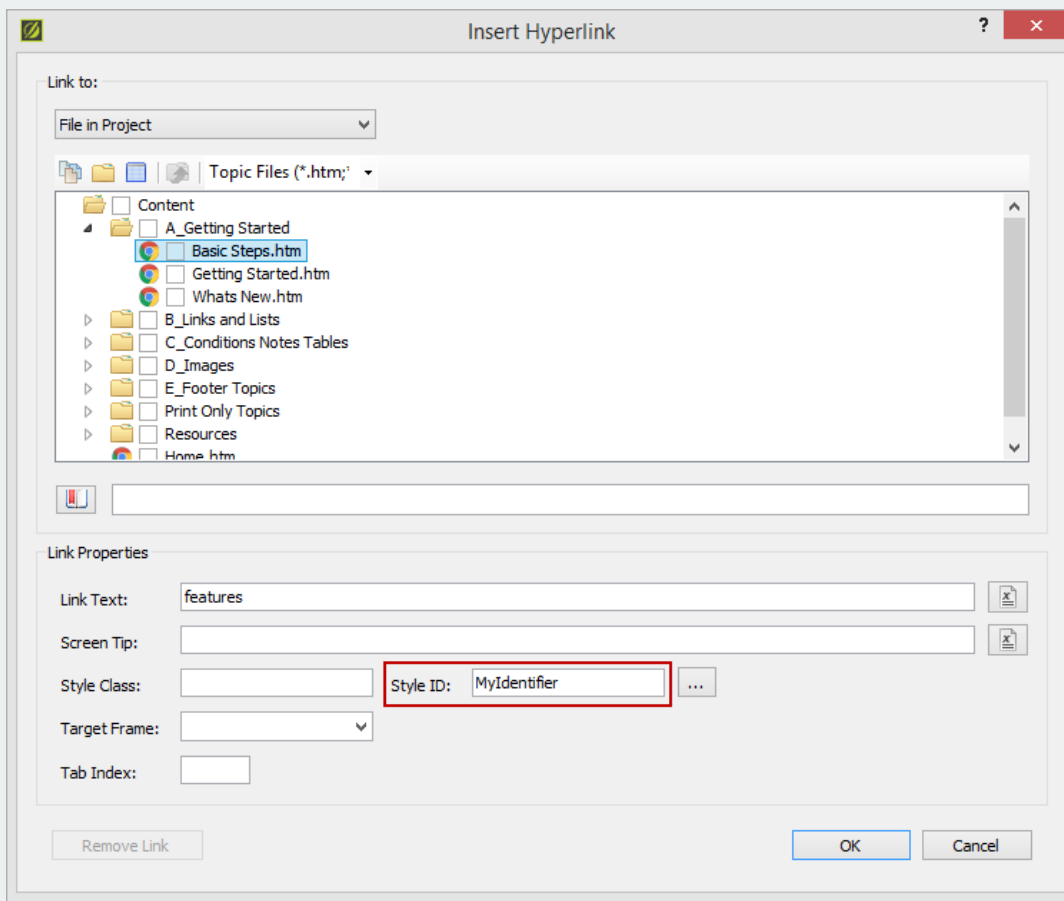
OK Cancel

After creating an ID, you can apply it to content in much the same way you would apply a class. In the interface (e.g., Styles window pane, Styles field in Home ribbon), you can identify IDs by looking for the hash tag before the name. You can right-click a structure bar in the XML Editor, and after selecting **Style ID**, you can choose any available ID.

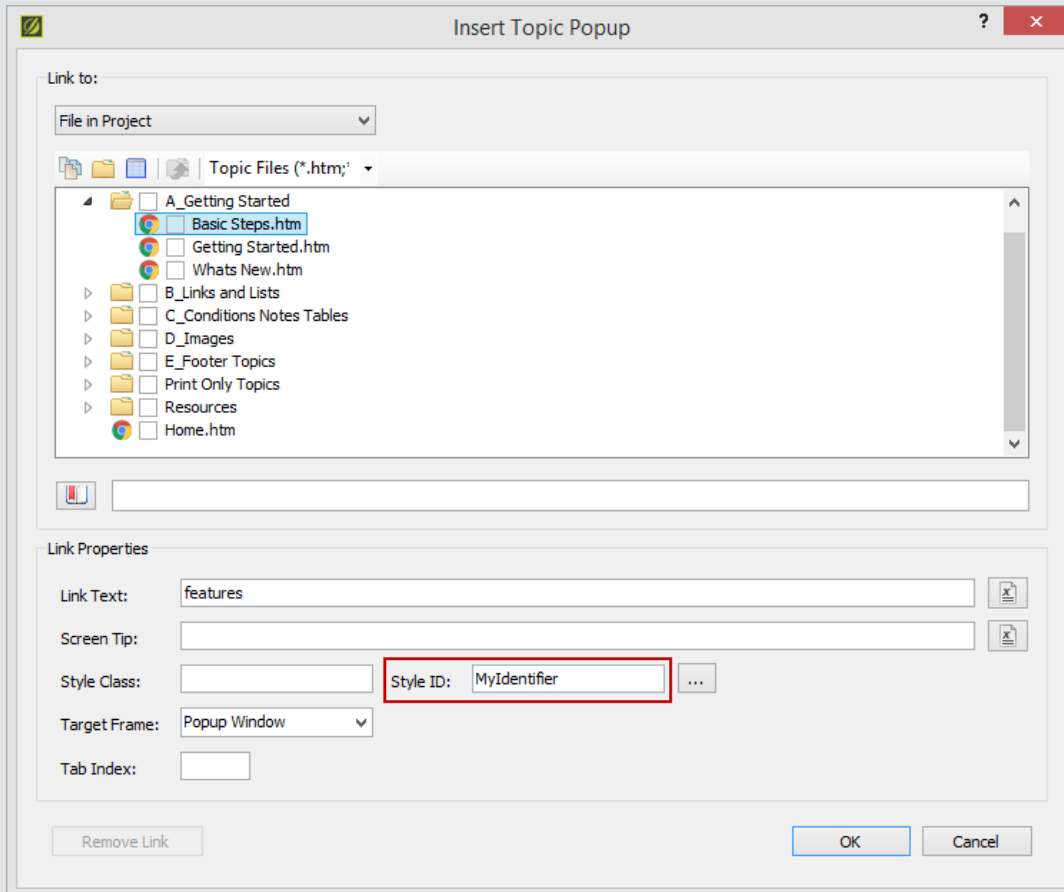


ID support has been added to the following dialogs:

Insert Hyperlink Dialog



Insert Topic Popup Dialog



COMMENTS

You can add any internal comments you want to a selector. This can be especially useful if you are working on a team and need to educate all of the authors about the selectors you use in your stylesheet.

New Selector

Manually add selector or choose Options.

HTML Element:

Class Name:

Advanced Options

Advanced Selector:

Pseudo Class:

Pseudo Class Expression:

Pseudo Element:

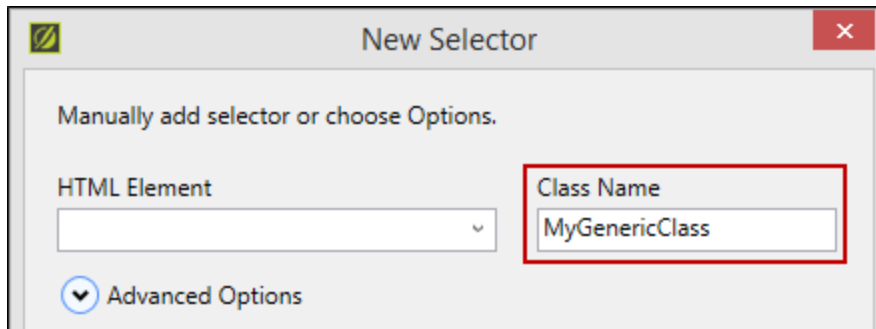
Identifier (ID):

Comments:

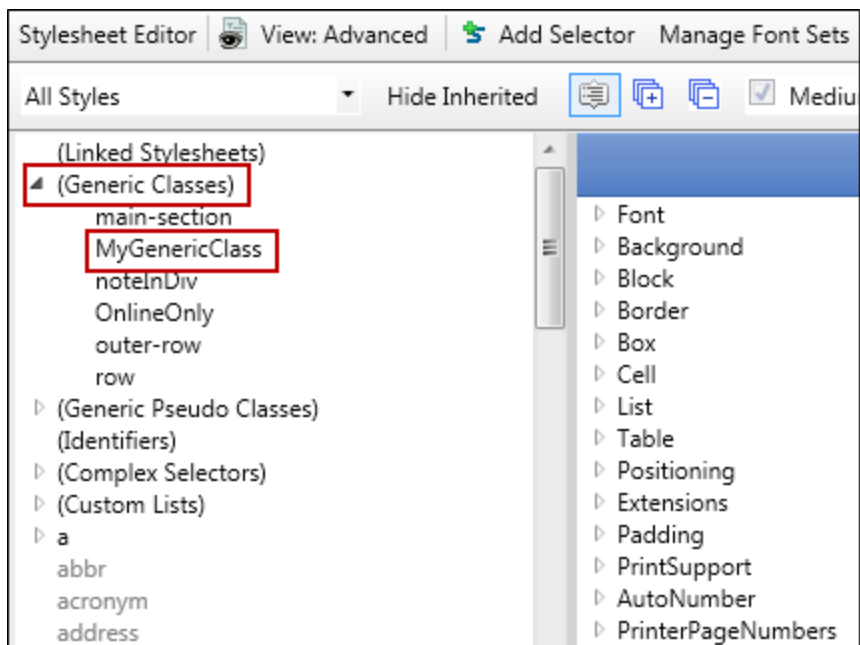
OK Cancel

GENERIC STYLE CLASSES

You can create a generic class by clearing the **HTML Element** field and entering text in the **Class Name** field.



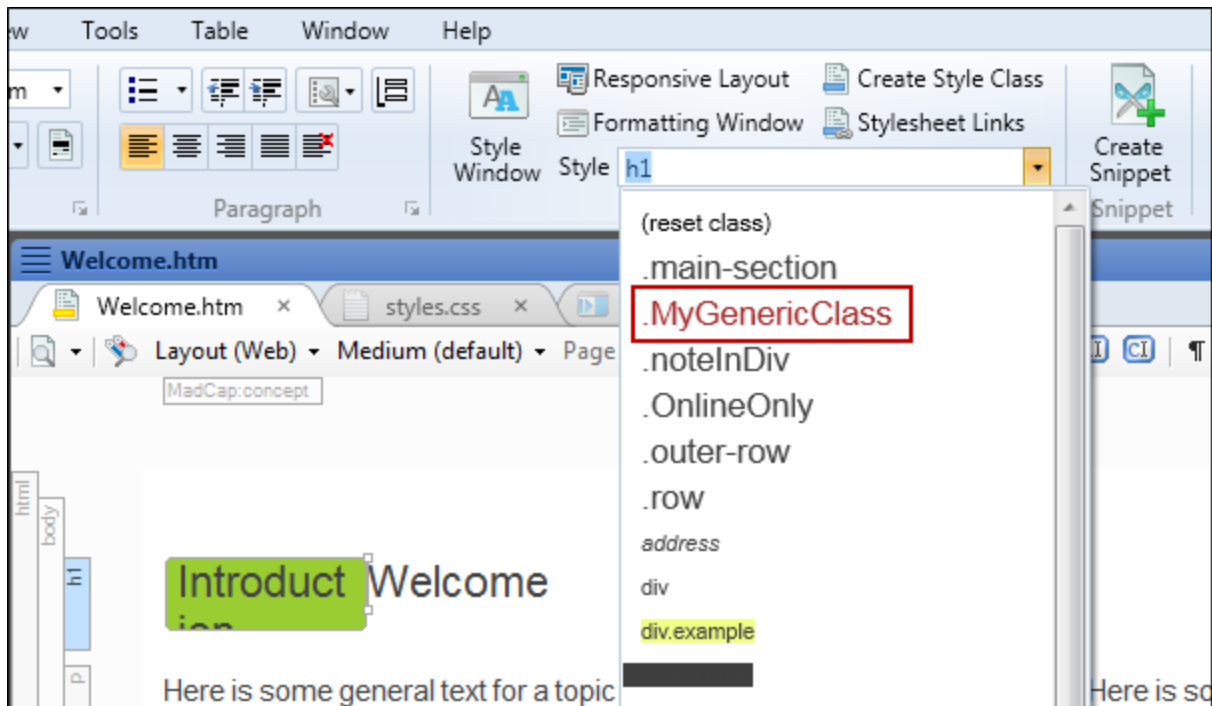
In the Advanced view of the Stylesheet Editor, it will be shown under **(Generic Classes)**.



In the Simplified view of the Stylesheet Editor, the Internal Text Editor, and the XML Editor, it will be shown with a period at the beginning.

Stylesheet Editor View: Simplified Add Selector Manage Font Sets Options ▾						
All Styles ▾		<input type="checkbox"/> Hide Inherited	<input type="checkbox"/> Hide Properties	Medium: (default) ▾		
Name ▲	Tag	Class	ID	Pseudo Class	Preview	
.main-section		main-section			MadCap Softv	
.MyGenericClass		MyGenericClass			MadCap Softv	
.noteInDiv		noteInDiv			MadCap Softv	
.outer-row		outer-row			MadCap Softv	
.row		row			MadCap Softv	
::after				after	MadCap Softv	
::before				before	MadCap Softv	
::marker				marker	MadCap Softv	
a	a				MadCap Softv	
a.GlossaryPage...	a	GlossaryPageTe...			MadCap Softv	

```
.MyGenericClass
{
    color: #a52a2a;
}
```



A generic class can be very useful if you need to apply the same formatting to several pieces of content, even if they have different HTML elements (parent styles).

EXAMPLE

Let's say you want some content to be centered every now and then. Sometimes it's a paragraph, sometimes it's a div tag (container), sometimes it's a list, and so on.

So rather than using the local formatting button on the Home ribbon to center each piece of content (which you should really never do), you create a generic style class. First, you open your stylesheet, and in the local toolbar you click **Selector**.

In the New Selector dialog, you clear the **HTML Element** field (if anything is in there); alternatively, you can select **(generic)**. And in the **Class Name** field, you give your new generic class a name, maybe something like `Centered`.

Either clear this field or select **(generic)**.

New Selector

Manually add selector or choose Options.

HTML Element

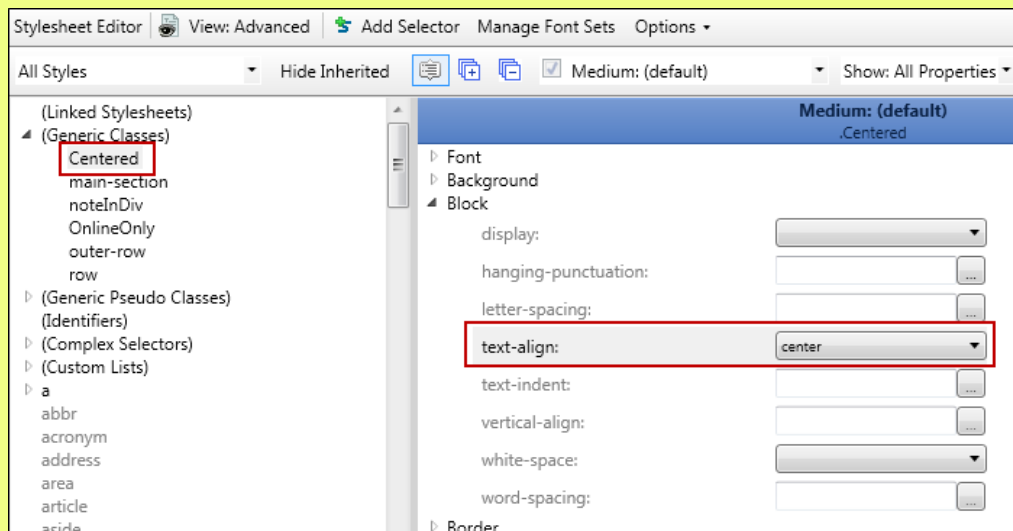
Class Name

Centered

☒ Advanced Options

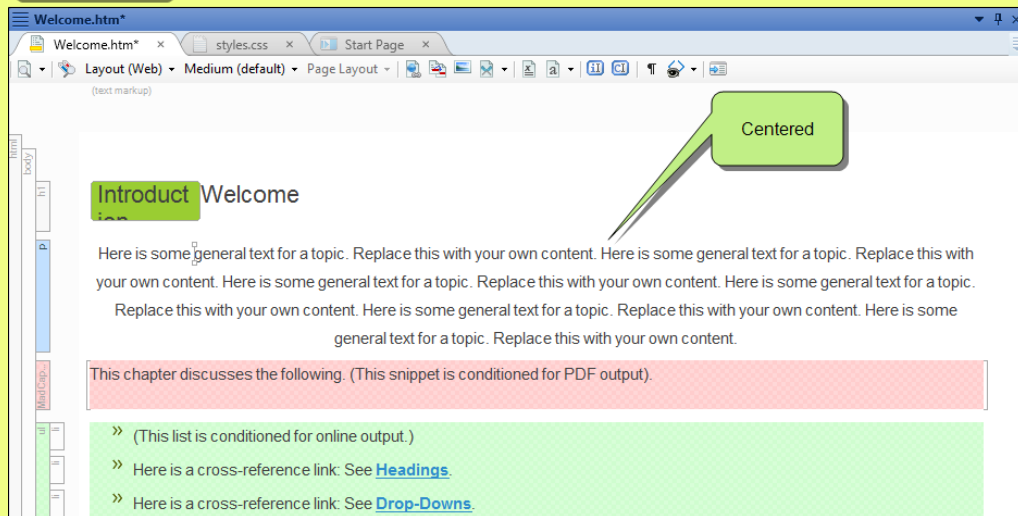
OK Cancel

Then you find the **text-align** property and set it to **center**.

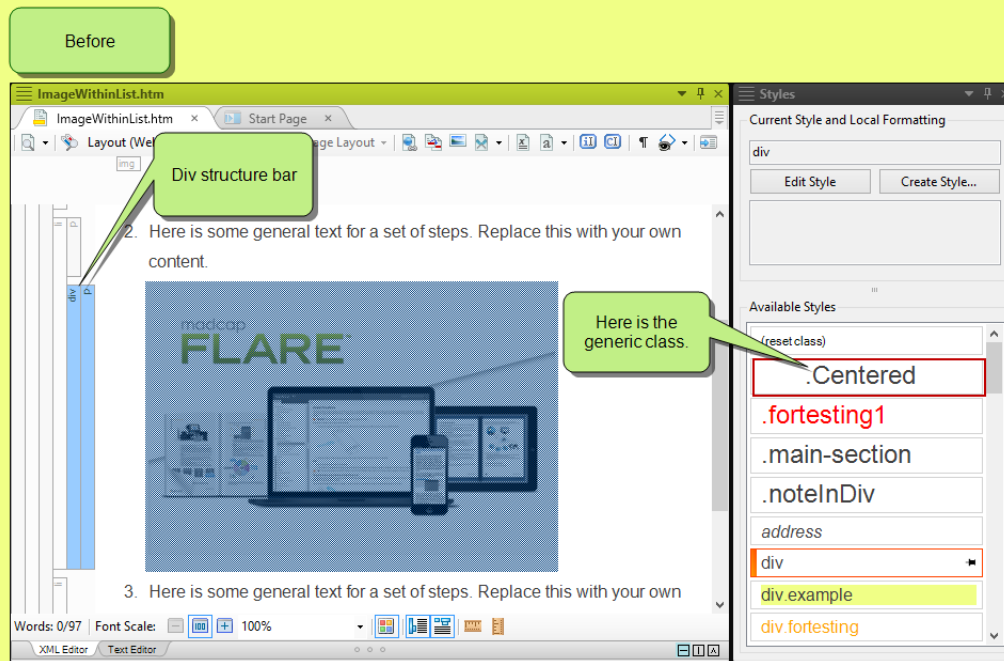


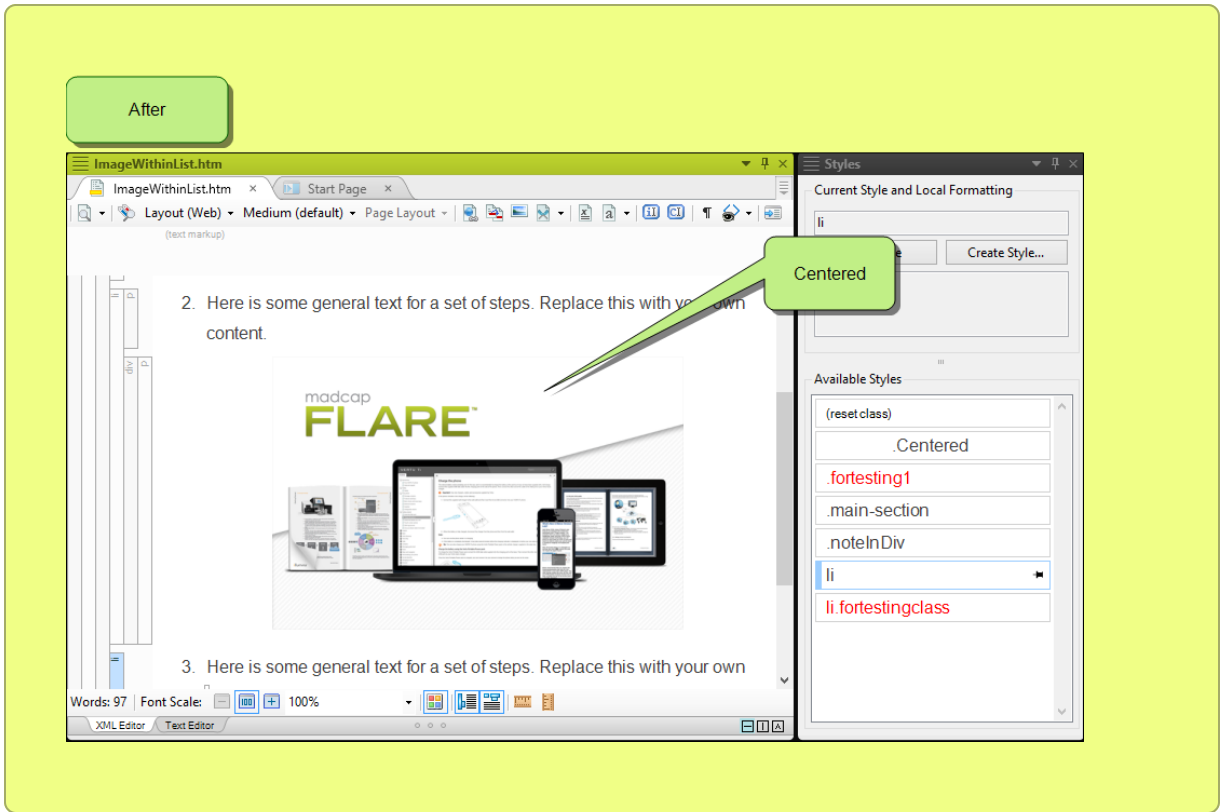
In the XML Editor, when you come across a paragraph you want to center, you just click in that paragraph, and from the **Home** ribbon's **Style** field you choose the generic **.Centered** class.

After

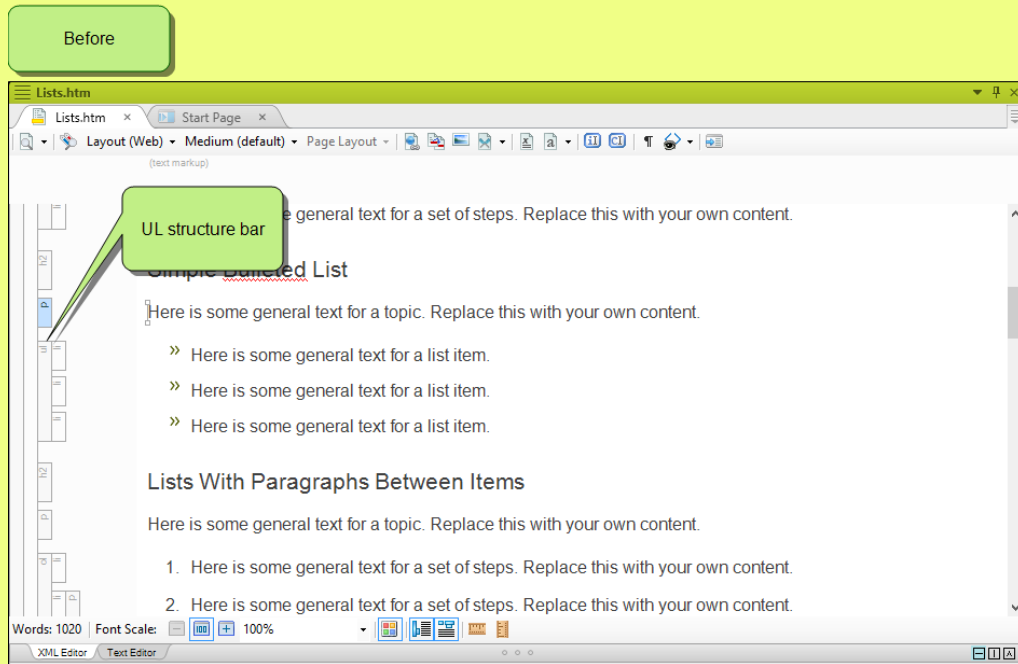


Suppose you prefer to use the Styles window pane instead of the Home ribbon. That's fine. Say you come across a div you want to center, such as the following div that has an image inside it. You can click on the **div** structure bar to select it and then choose the generic **.Centered** class from the Styles window pane.

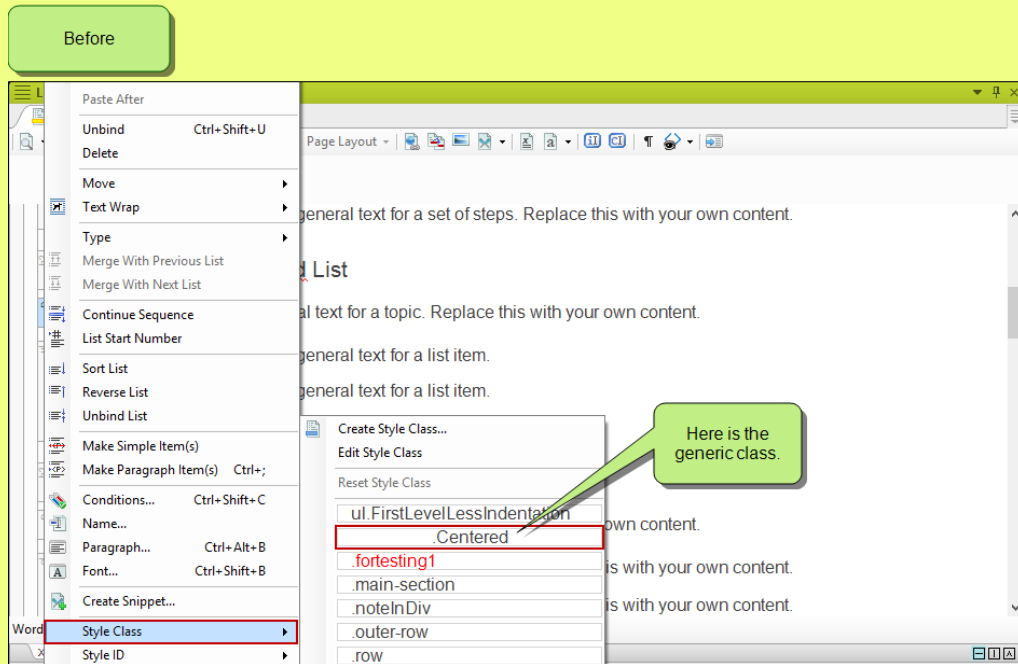


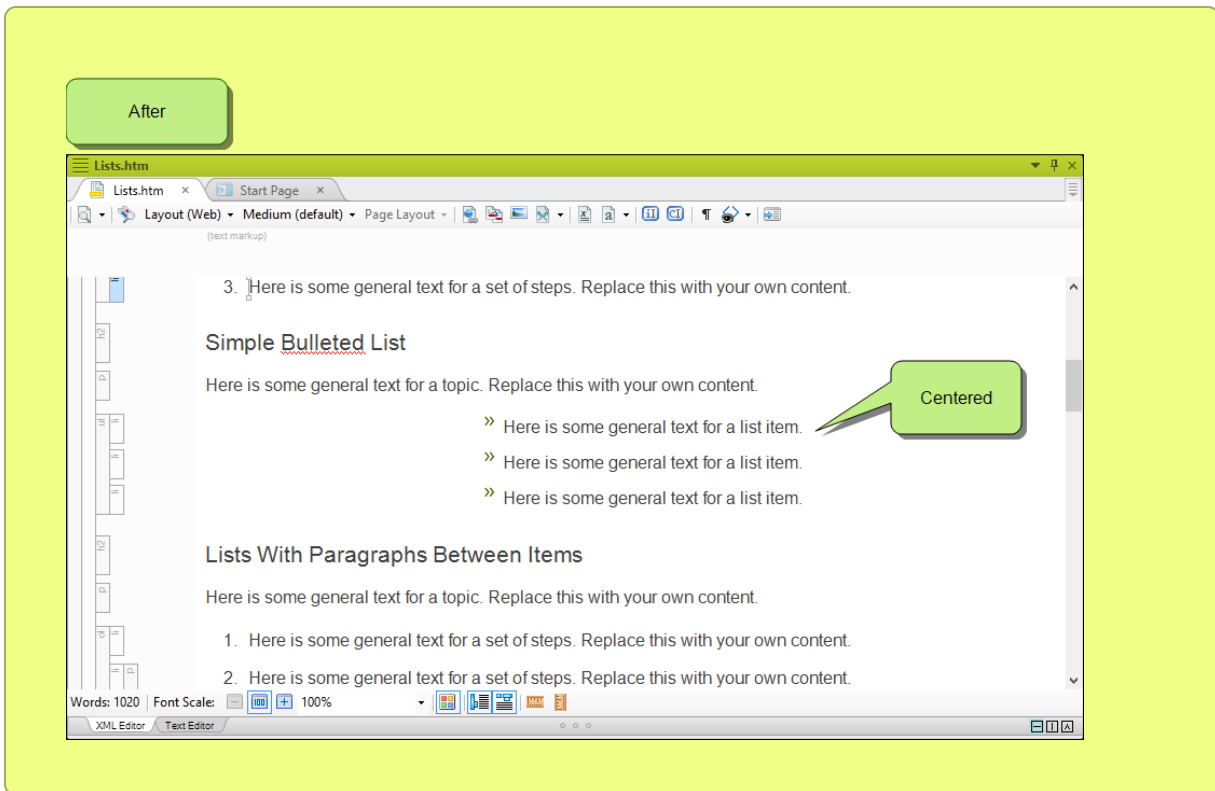


Or maybe you like to use context menus. Suppose you come across a bulleted list you want to center. So you right-click the **ul** structure bar.



Then from the context menu you choose **Style Class**, then **.Centered**.





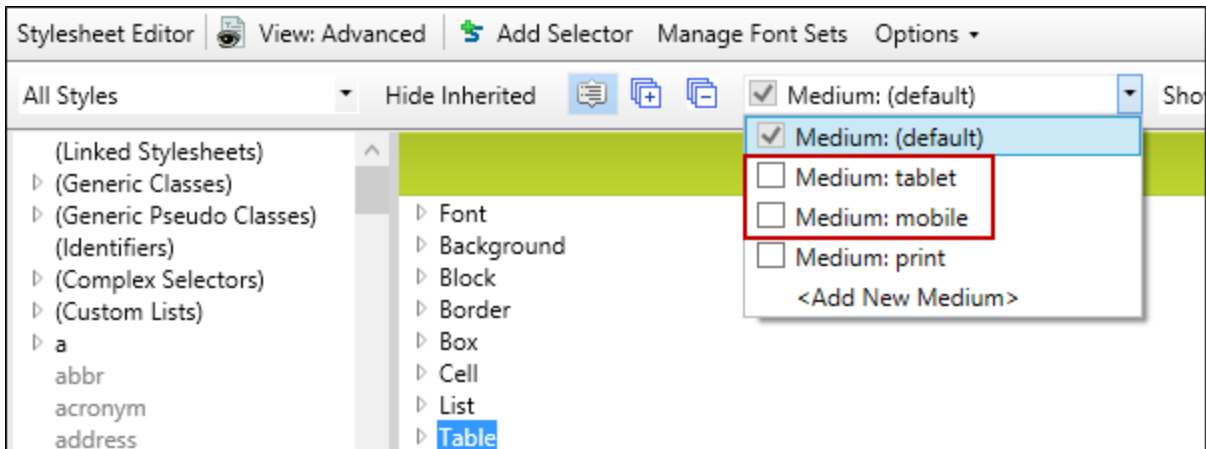
Media Queries

You can create and use both mediums and media queries in Flare. The two concepts are very similar; in fact, you will see them side by side in different places in Flare's user interface. However, mediums and media queries are not the same. So what is the difference between a medium and a media query?

- » **Medium** A medium is an alternative group of settings in a stylesheet and can be very useful when you are generating multiple kinds of outputs. Unless you tell Flare otherwise, default style settings will be used for the different outputs you generate. But there may be times when you want to override a default style setting for a particular output; that's why you would use a medium. *You need to explicitly tell Flare which medium you want a particular target to use.* This is done from the Advanced tab of the Stylesheet Editor.
- » **Media Query** A media query is an alternative group of settings in a stylesheet. These settings are automatically used under certain conditions, such as when a screen of a certain size is displaying the output. Media queries are able to do this because they are configured with specific criteria (e.g., maximum width of the screen, orientation, resolution). When the criteria are met, the style settings in the media query are used to display the output. *You do not tell a Flare target to use a media query; it just happens automatically.*

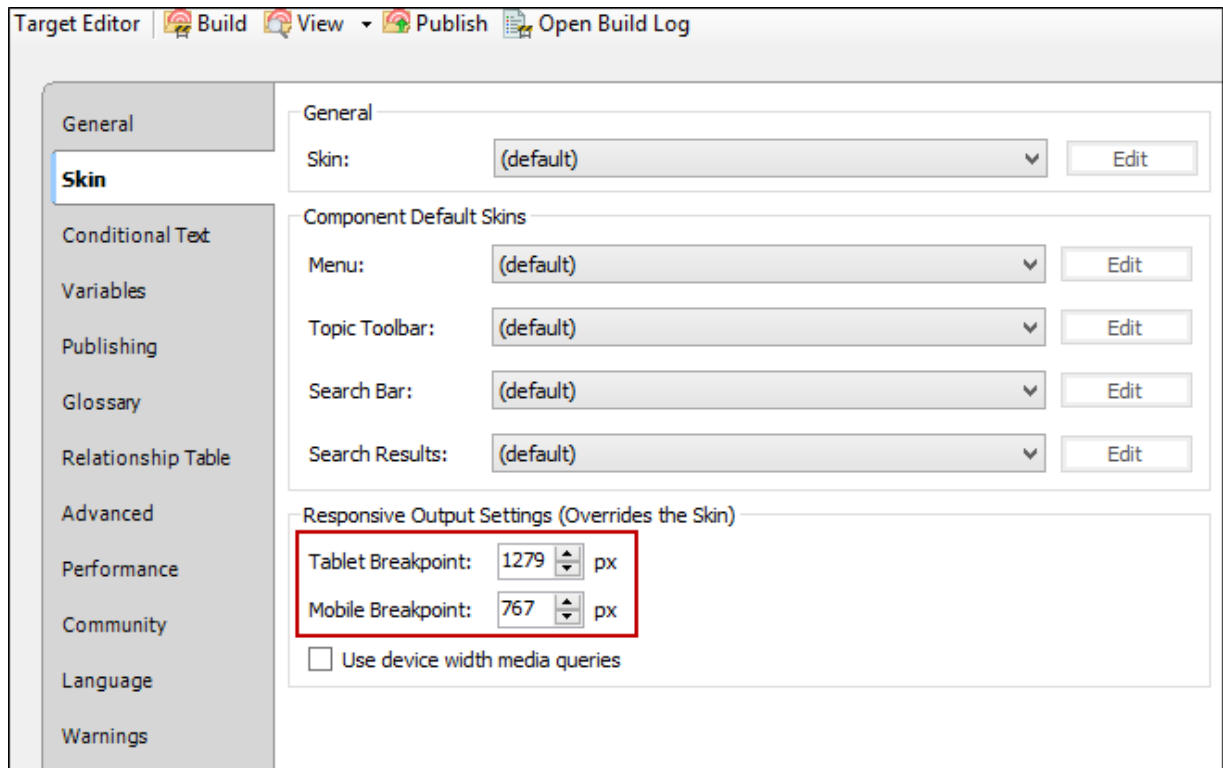
TABLET AND MOBILE MEDIA QUERIES

A couple of media queries (tablet and mobile) are already provided in Flare.



You can create additional media queries from the Stylesheet Editor if you want, but most authors will be able to do everything they need with just the tablet and mobile media queries. The tablet media query is designed to be used on medium-sized screens, such as iPads. The mobile media query is designed to be used on smaller screens, such as smart phones.

The tablet and mobile media queries are tied to the responsive output settings on the Skin tab of the Target Editor for HTML5 targets. The breakpoints provided in the target determine the point at which your media queries will become active in displaying its style settings.



USES FOR MEDIA QUERIES

You can place any valid CSS style settings in a media query, just as you can in a medium. For example, if you want paragraph text to suddenly turn blue when a topic is viewed on an iPhone, you can edit the mobile media query, telling it to use a blue font for all paragraph styles.

One of the most common reasons to use a media query is to account for how the structure of content needs to shift or change when viewed on screens of different sizes. You can use Flare's Responsive Layout window pane to do exactly that. See "Responsive Layout" on page 37.

MEDIA QUERIES AND MEDIUMS TOGETHER

You do not need to choose between using a medium or a media query for a target. They can be used alongside one another when you generate output.

E X A M P L E

Let's say you have custom medium called "OnlineBlue," which you associate with one of your targets to show content with a blue theme on some styles. But in addition to that medium, you also make some edits to the tablet and mobile media queries in your stylesheet so that content is adjusted for smaller devices.

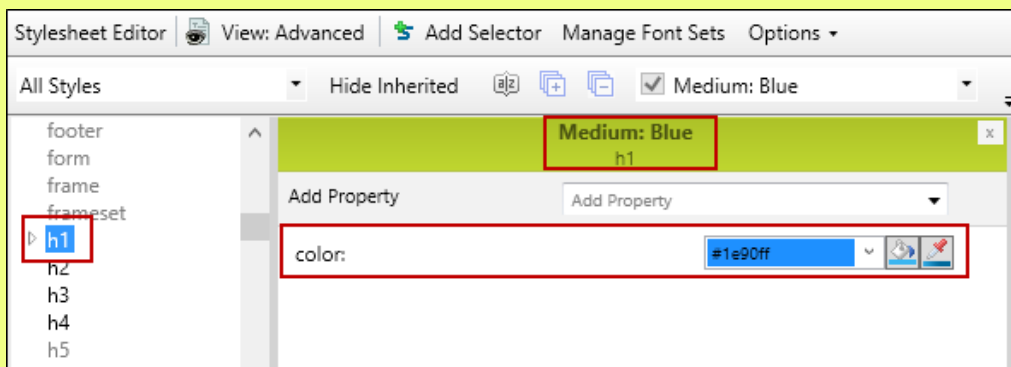
When you view the output on a large monitor, the content will look a certain way that makes sense on a big screen. When you view it on a tablet, the content might shift a bit so that it looks better on that smaller screen. And when you view it on an iPhone, the content might shift again to account for that device. But in all three cases, the content is still adhering to the blue theme from the OnlineBlue medium.

CONFLICTS AND THE ORDER OF MEDIA QUERIES

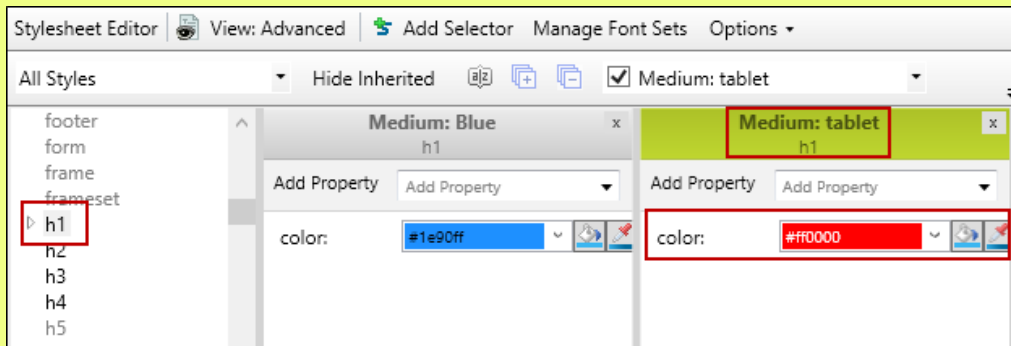
What if there is a conflict between your selected medium and a media query, or a conflict between media queries themselves? Media queries will always have precedence over your selected medium. As far as multiple media queries are concerned, the end result depends on the order of the media queries in your stylesheet (when viewing it in the Internal Text Editor). The general rule is that priority is given to whichever media query is listed last (i.e., the one that was added most recently) in the stylesheet. Then the next one above has the next highest priority, and so on. You can always open your stylesheet in the Internal Text Editor and change the order of the media queries.

EXAMPLE

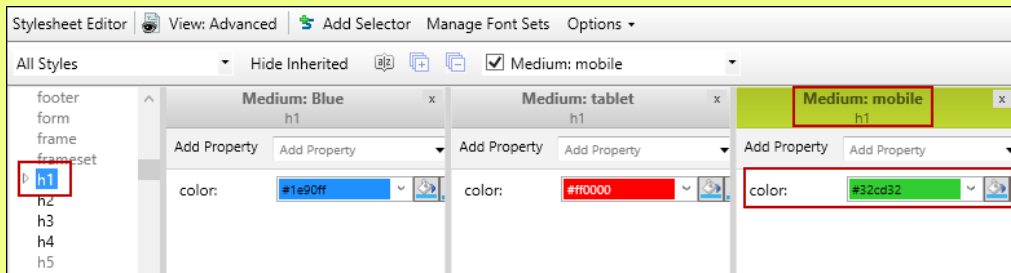
Let's say you create a custom medium, naming it "Blue." You associate this medium with your HTML5 target. And when you open your stylesheet, you select this medium and tell Flare that the h1 style should be blue.



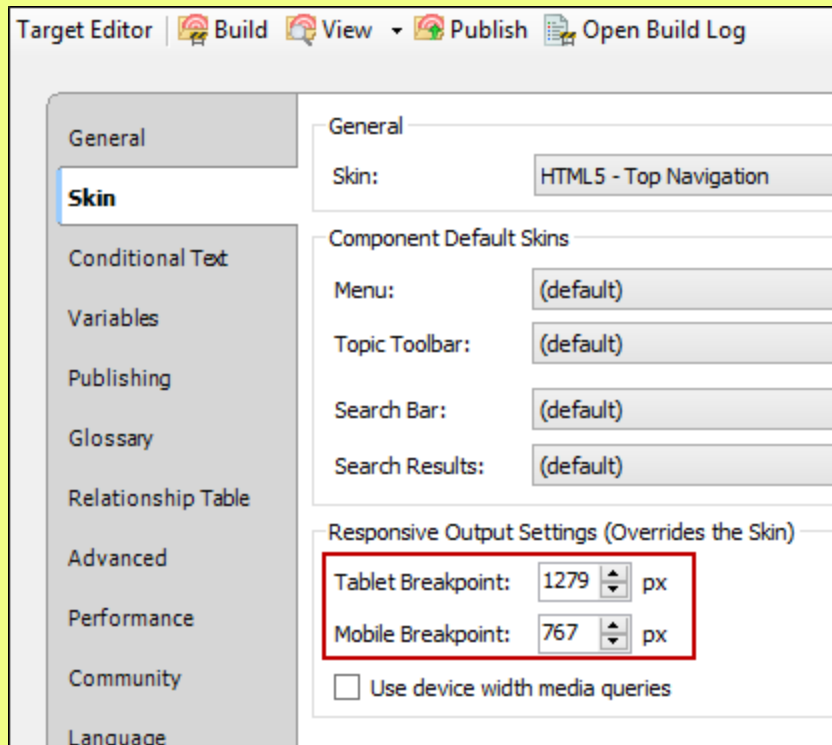
Next, you edit the factory tablet media query and you specify that the h1 style should be red.



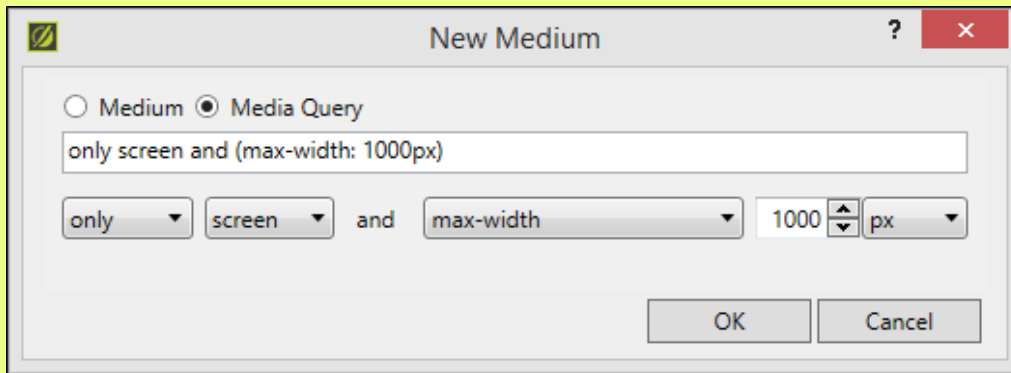
After this, you edit the factory mobile media query and tell Flare to make the h1 style green.



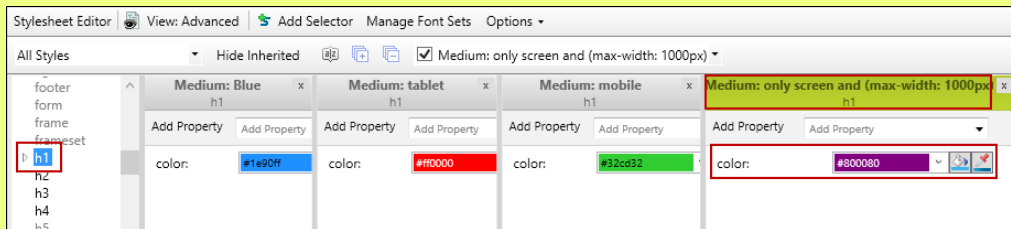
If you open your HTML5 target and look at the **Skin** tab, you will see that the tablet maximum width is set to 1279 pixels, and the mobile maximum width is set to 767 pixels.



You decide that you want a different look when the screen is between 1279 and 767 pixels. Therefore, you create a new media query, setting it to a maximum width of 1000 pixels, like this:



In the stylesheet, you set the h1 style for this media query to purple.



If you open the Content Explorer, right-click on your stylesheet, and select **Open with>Internal Text Editor**, the stylesheet will open in that editor. When you scroll all the way to the bottom, you will see the custom medium and media queries in order, like this:



```
1004: @media Blue
1005: {
1006:     h1
1007:     {
1008:         color: #1e90ff;
1009:     }
1010: }
1011:
1012: @media tablet
1013: {
1014:     h1
1015:     {
1016:         color: #ff0000;
1017:     }
1018: }
1019:
1020: @media mobile
1021: {
1022:     h1
1023:     {
1024:         color: #32cd32;
1025:     }
1026: }
1027:
1028: @media only screen and (max-width: 1000px)
1029: {
1030:     h1
1031:     {
1032:         color: #800080;
1033:     }
1034: }
```

Now you generate the HTML5 target and view the output. When the browser is maximized, you will notice that the h1 headings are blue, because that's what you set on the "Blue" medium.

Then you reduce the width of the browser. Once it gets to 1279 pixels, the h1 headings turn from blue to red, because that's what you set in the tablet media query.

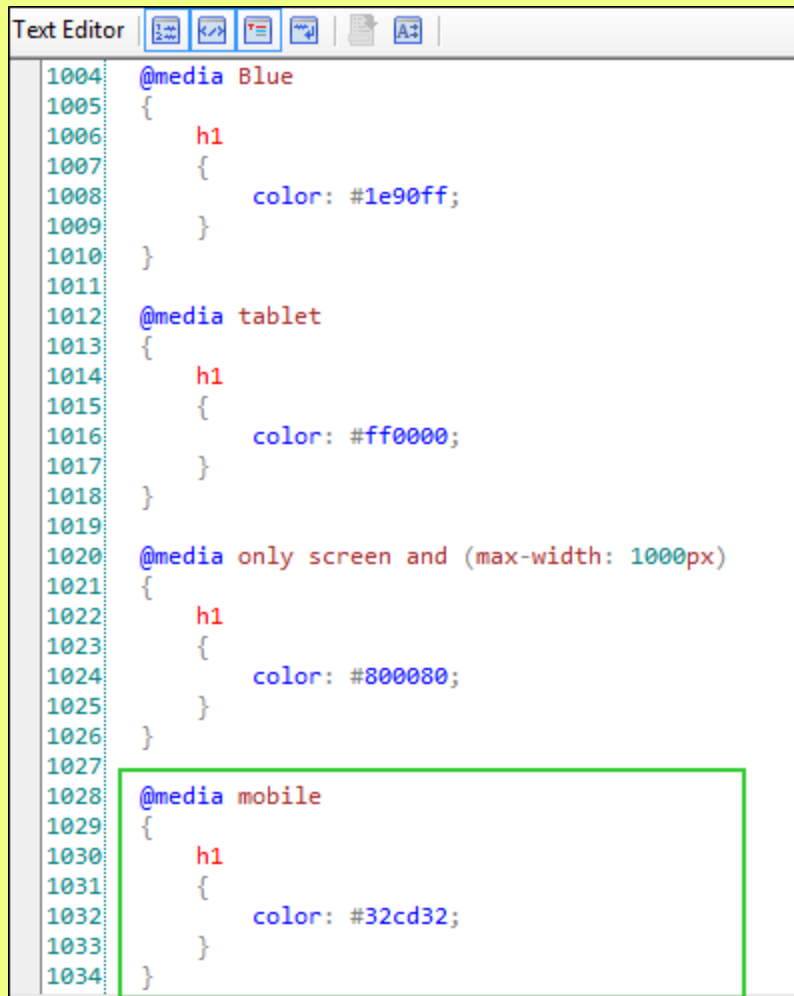
You reduce the width of the browser even more. When you get to 1000 pixels, the h1 headings turn from red to purple, because that's what you set in your custom media query.

You continue to reduce the width of the browser. You're expecting the h1 headings to display in green (the color in the mobile media query), but they never do. What happened?

Remember that you added your custom media query last, so it appears at the bottom of the stylesheet. The tablet color (red) showed up because its width is higher than the custom media query width. As soon as it hit 1279 pixels, your browser saw that there were instructions to change the color to red. Even though the custom media query appears at the bottom of the stylesheet (therefore with the highest priority), its lower width setting meant that it wasn't in conflict with the tablet setting at that point. It was only at the point when the browser got to 1000 pixels that there was a conflict, so at that moment, the browser chose the media query that is lower in the stylesheet and displayed the text in purple.

The problem with the mobile media query is that by the time the browser was reduced to 767 pixels, it was already using a color for a media query with a higher priority. So the mobile color for h1 never got an opportunity to display.

To fix this, you can cut the mobile media query in the Internal Text Editor and paste it last in the stylesheet, like this:

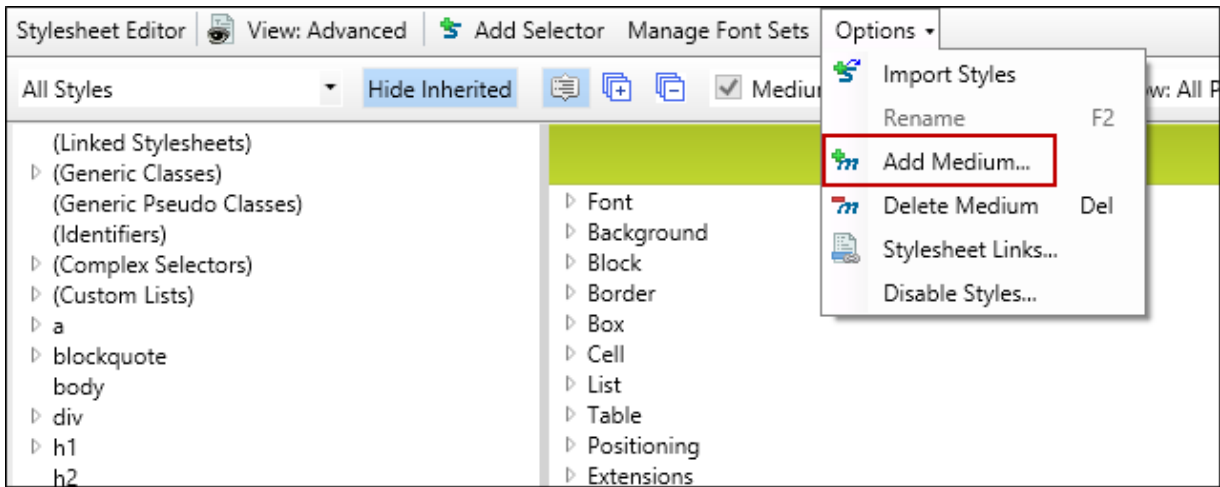


```
1004 @media Blue
1005 {
1006     h1
1007     {
1008         color: #1e90ff;
1009     }
1010 }
1011
1012 @media tablet
1013 {
1014     h1
1015     {
1016         color: #ff0000;
1017     }
1018 }
1019
1020 @media only screen and (max-width: 1000px)
1021 {
1022     h1
1023     {
1024         color: #800080;
1025     }
1026 }
1027
1028 @media mobile
1029 {
1030     h1
1031     {
1032         color: #32cd32;
1033     }
1034 }
```

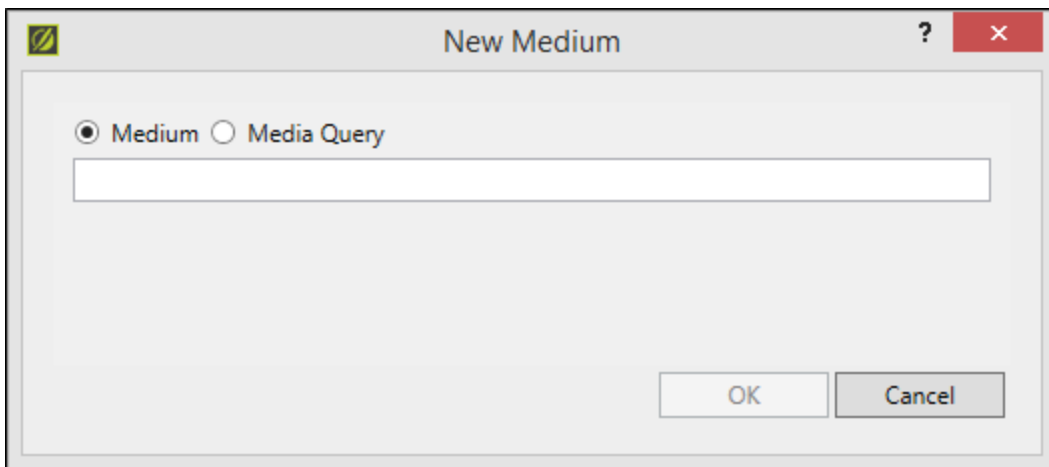
Now when you generate and view the output, it will work as you expect when you reduce the browser width: from blue to red to purple to green.

CREATING MEDIA QUERIES

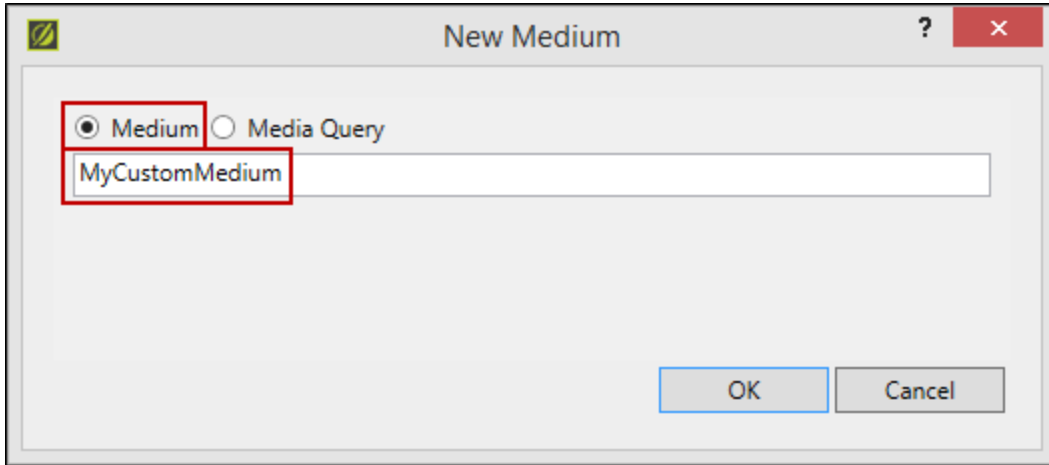
You have been able to create media queries in Flare stylesheets previously, but you had to do so using the Internal Text Editor. Starting with this version, you can create media queries in the Stylesheet Editor. The steps are similar to those followed when creating a medium. In the local toolbar, you can click **Options** and then choose **Add Medium**.



This opens the New Medium dialog.

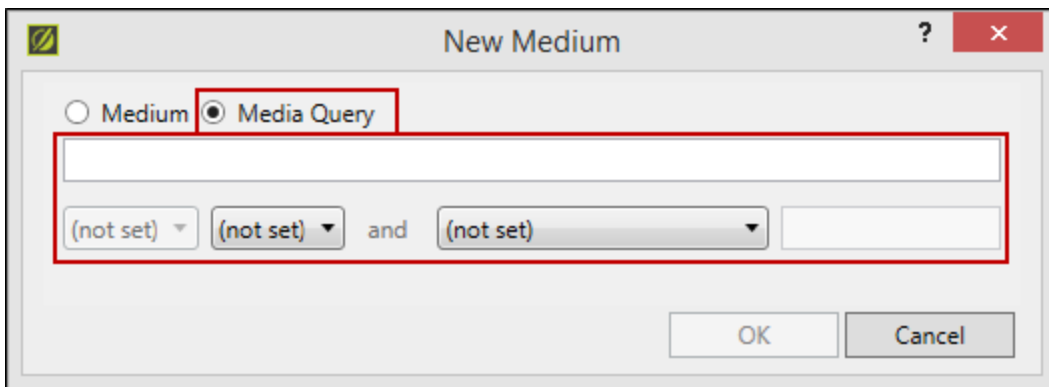


To create a basic medium, make sure **Medium** is selected. Then enter a name for it in the field.



The 'New Medium' dialog box is shown with the 'Medium' radio button selected. A text field below it contains the text 'MyCustomMedium'. The 'OK' and 'Cancel' buttons are at the bottom right.

To create a media query, select **Media Query**. Extra fields are shown in the dialog.



The 'New Medium' dialog box is shown with the 'Media Query' radio button selected. Below the radio buttons is a large empty text field. Below that are three dropdown menus, each showing '(not set)', followed by the word 'and' and another dropdown menu showing '(not set)'. The 'OK' and 'Cancel' buttons are at the bottom right.

You can enter your media query directly in the text field (as long as you use valid media query syntax), or you can use the fields below to choose criteria for the media query. The text field is populated based on your selections in the drop-down fields.

Please note that some fields in this dialog will change depending on the options you choose in other fields.

For more information about creating custom media queries and the options available in the New Medium dialog, see <http://www.w3.org/TR/css3-mediaqueries/>.

Multi-Select and Delete

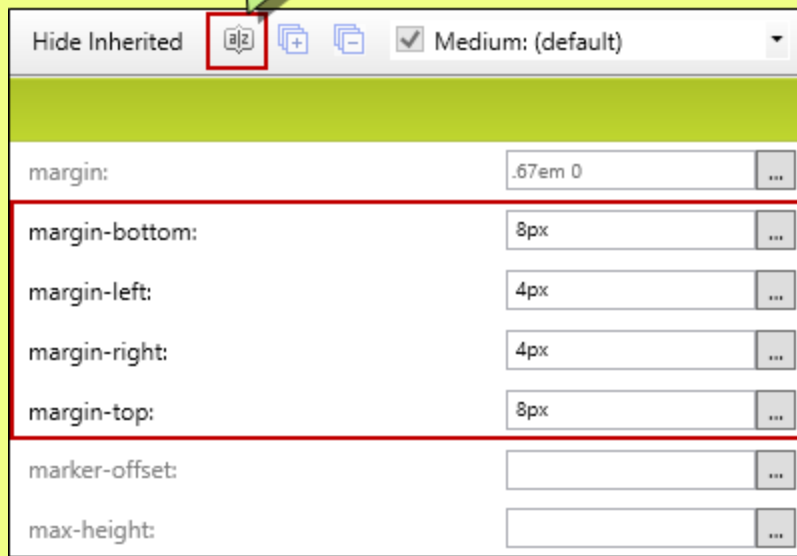
If you are using the alphabetical (grid) view for properties in the Stylesheet Editor, you can multi-select multiple items by holding down the **CTRL** or **SHIFT** key. Once the property items are selected, you can press **Delete** (or right-click and select **Delete**).

This applies only to properties for which you have already set values (i.e., the name of the property shows in darkened font, not light gray). Also, doing this does not completely remove the property, preventing you from editing it in the future. Instead, it removes all values associated with those properties in your stylesheet. So the name of the property will revert from darkened font back to gray, indicating nothing is set on it.

EXAMPLE

Let's say you've set the bottom, left, right, and top margins for a style. The font will be darkened for those properties, because you explicitly set them, as opposed to relying on an inherited value.

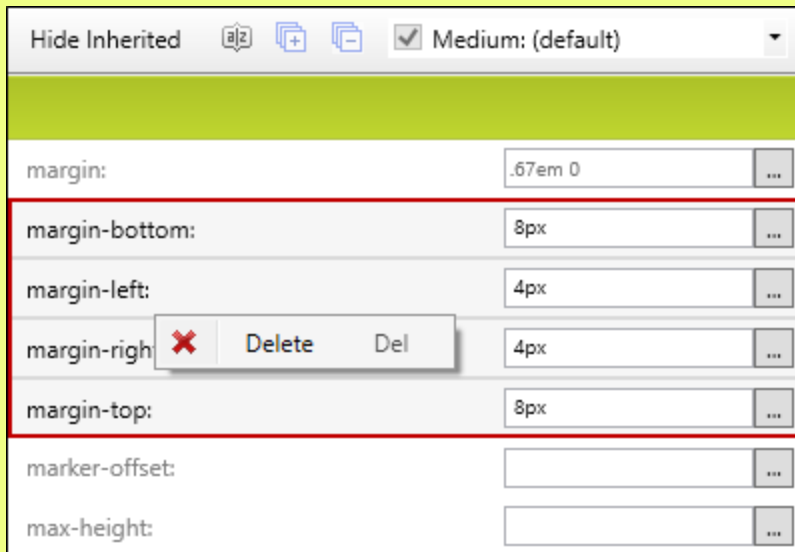
Notice that the properties are shown in alphabetical (or list) view.



The screenshot shows a CSS editor interface. At the top, there is a toolbar with a 'Hide Inherited' button, a 'List View' icon (a document with a list) which is highlighted with a red box, and a '+', '-', and 'Medium: (default)' dropdown. Below the toolbar, a list of CSS properties is displayed. The properties are: 'margin:', 'margin-bottom:', 'margin-left:', 'margin-right:', 'margin-top:', 'marker-offset:', and 'max-height:'. The four margin properties are grouped together and highlighted with a red box. Each property has a text input field to its right. The values are: '.67em 0' for 'margin:', '8px' for 'margin-bottom:', '4px' for 'margin-left:', '4px' for 'margin-right:', and '8px' for 'margin-top:'. The 'marker-offset:' and 'max-height:' properties have empty input fields. The font for the highlighted margin properties is bolded.

Property	Value
margin:	.67em 0
margin-bottom:	8px
margin-left:	4px
margin-right:	4px
margin-top:	8px
marker-offset:	
max-height:	

We click **margin-bottom**, hold down our **SHIFT** key, then click **margin-top**. This selects all of four properties. Then we right-click and choose **Delete**.



Those values are removed from the stylesheet, and the properties are now displayed in light gray, indicating they will be using the inherited default value.

Hide Inherited

az

+

-

☒ Medium: (default)

margin:

.67em 0

...

margin-bottom:

.67em

...

margin-left:

0

...

margin-right:

0

...

margin-top:

.67em

...

marker-offset:

...

max-height:

...

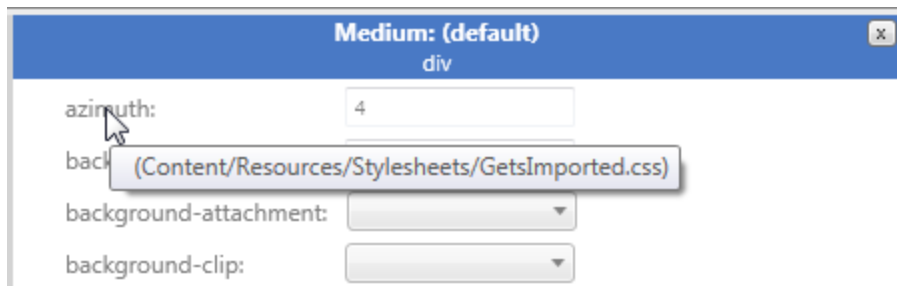
Conditions

Starting with this version of Flare, you can associate a condition with a style by using the `mc-conditions` property. For more information, see "Conditions on Styles" on page 206.

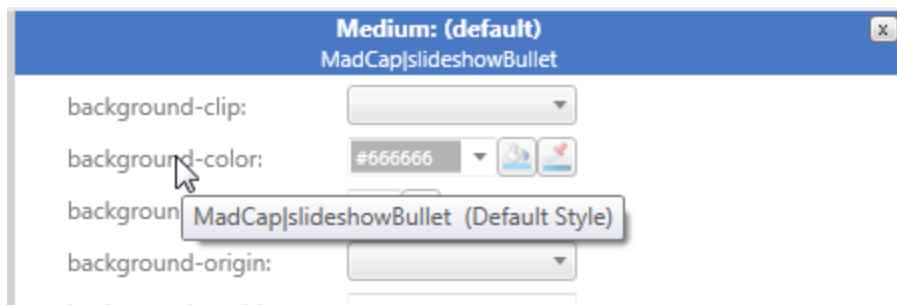
Tooltips Show Inheritance

You can hover over inherited (light gray) properties to see the origin of their inheritance.

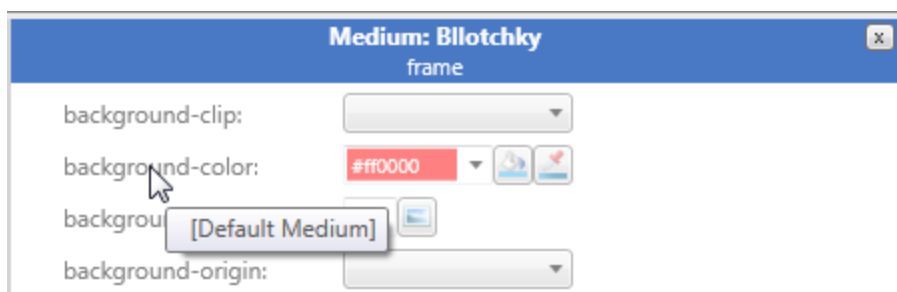
» From another (imported) stylesheet:



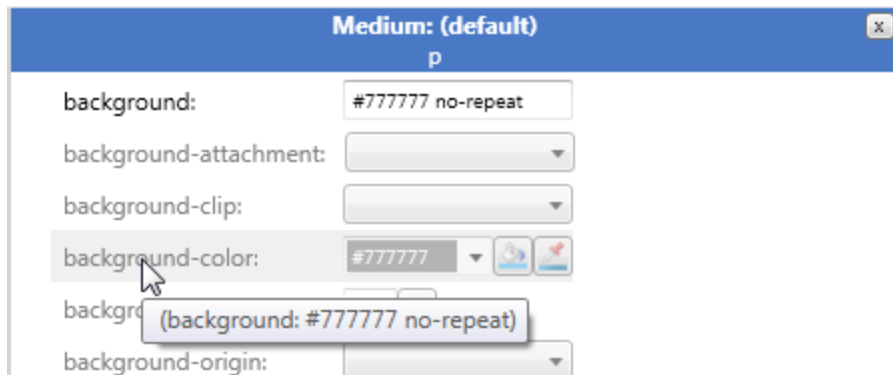
» From a MadCap schema file:



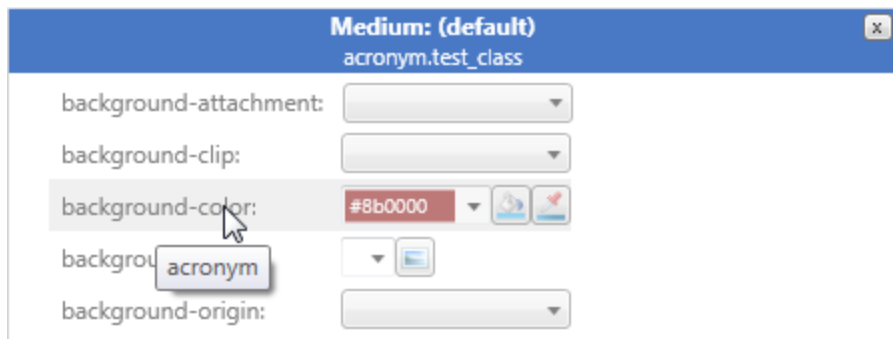
» From a different medium:



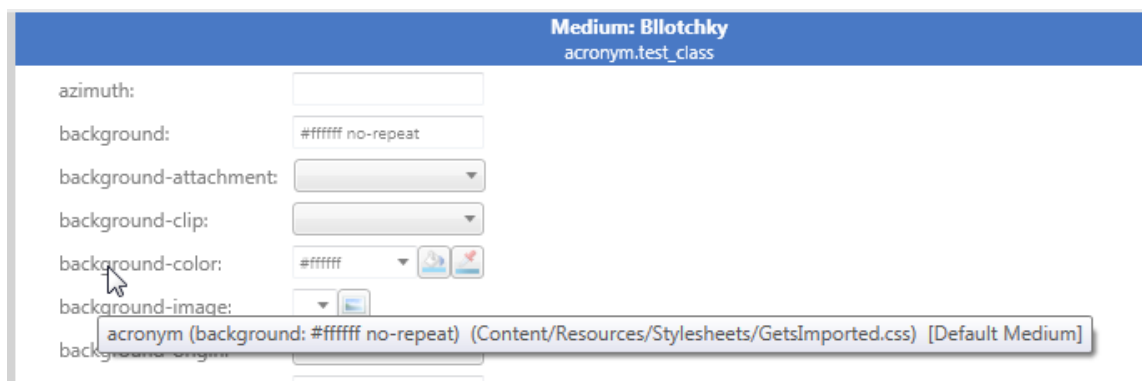
» From a different declaration in the same rule set:



» From a different selector:

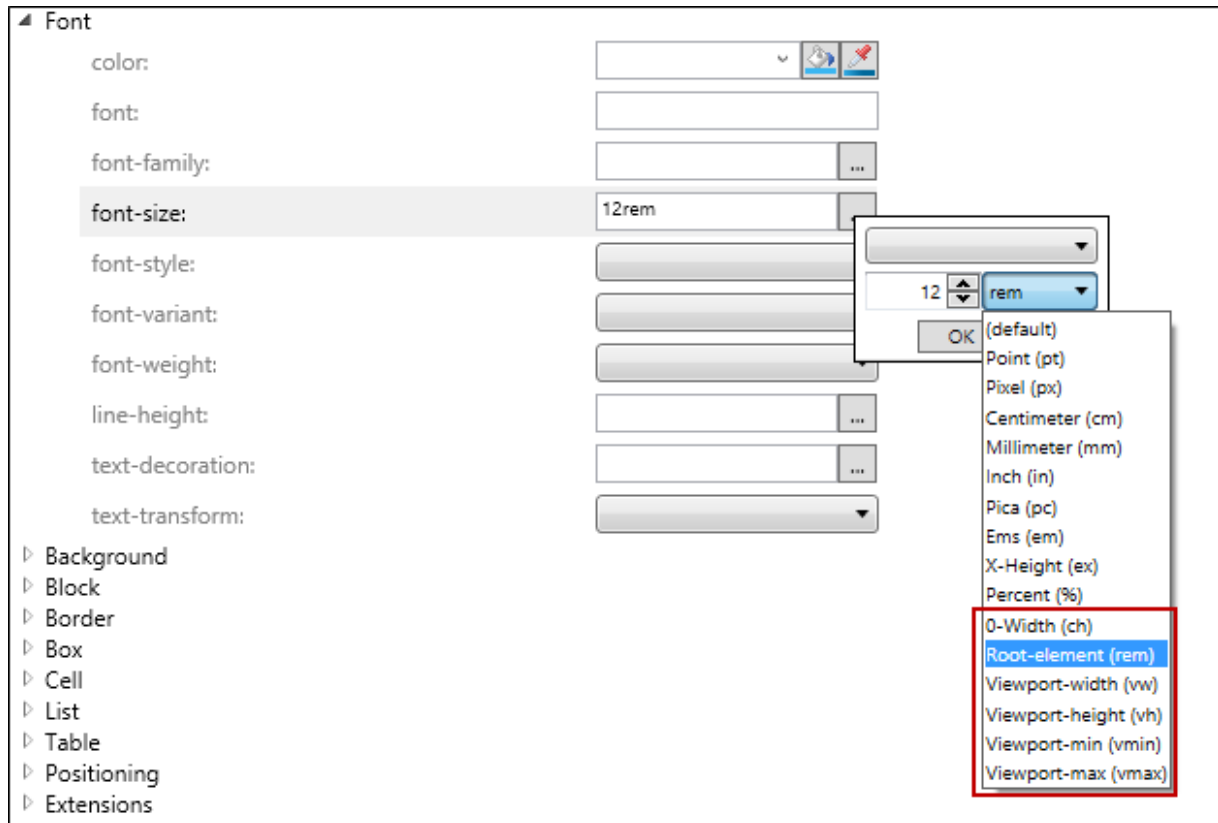


» Combination of the above:



Units of Measurement—New Options

The following relative units of measurement (UOM) can now be selected in various places in Flare's user interface. These new units are intended primarily for web output.



UOM	Description
0-Width (ch)	Relative to the width of the 0 (ZERO, U+0030) glyph in the element's font.
Root-element (rem)	Relative to the font size of the root element.
Viewport-width (vw)	Relative to 1% of the width of the viewport. ¹
Viewport-height (vh)	Relative to 1% of the width of the viewport.
Viewport-min (vmin)	Relative to 1% of the width of the viewport.
Viewport-max (vmax)	Relative to 1% of the width of the viewport.
¹ The viewport is the size of the browser window.	

For more information, see <http://www.w3.org/TR/css3-values/> and http://www.w3schools.com/cssref/css_units.asp.

Tracked Changes

Flare now allows you to preserve your tracked changes in PDF and Word outputs. You can also use change bars in conjunction with tracked changes as a visual indicator of where changes have been made in a document.

You may want to use tracked changes in a PDF or Word document for a number of reasons. The most common reason to use tracked changes is for internal review processes. While you always have the option to use MadCap Contributor for internal reviews, using tracked changes in PDF or Word gives you more options that can help make the review process easier. For example, if you have many reviewers who only review content occasionally, you may want to use tracked changes in PDF or Word so the reviewers can use familiar software. This may help to speed up the review process.

Another reason to preserve tracked changes in PDF or Word is if you need the reader to see changes in a document. By preserving changes in PDF or Word output (and by using change bars, if necessary), the reader can easily locate new material.

This chapter discusses the following:

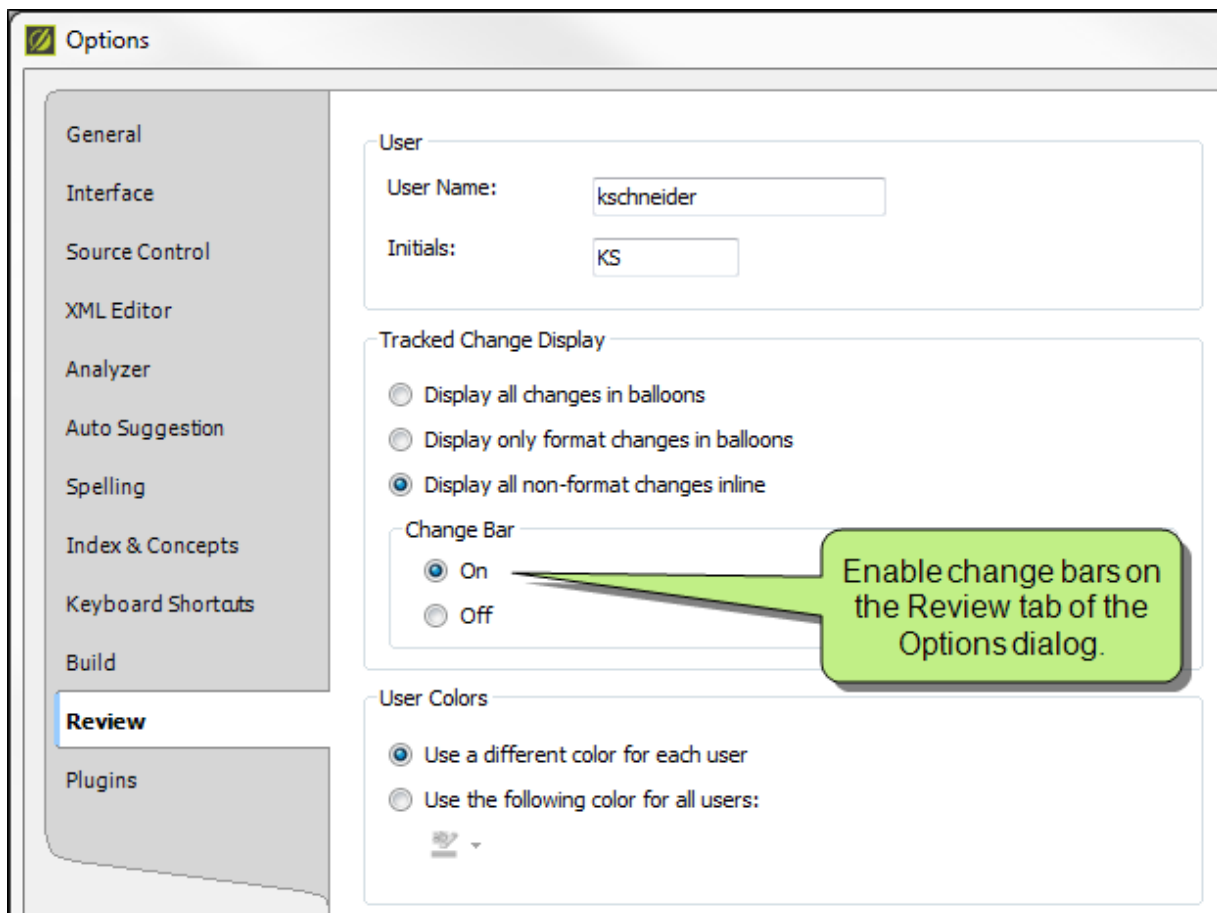
Change Bars	184
Tracked Changes in PDF Output	190
Tracked Changes in Word Output	194



Change Bars

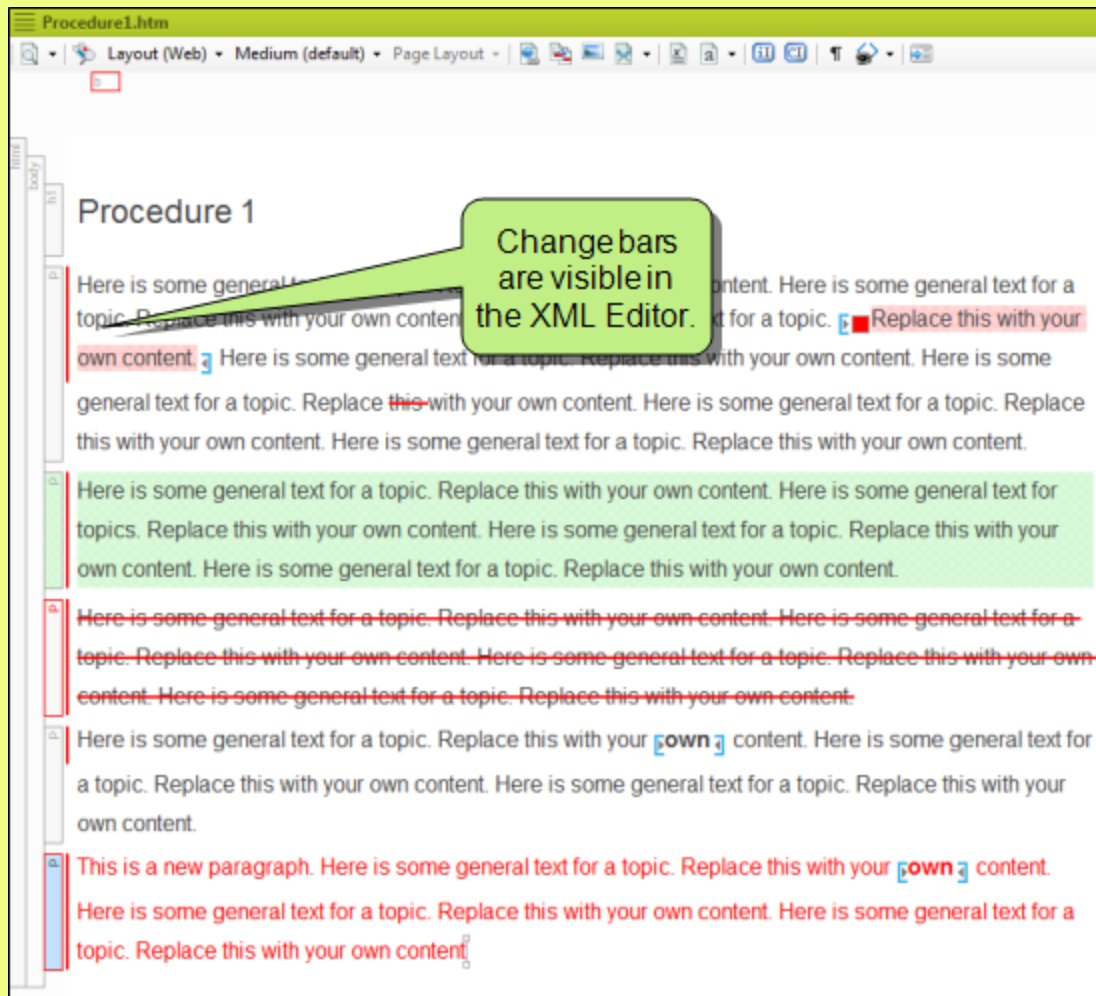
When you track changes in a document, you can now use change bars to identify new, revised, or deleted text. When change bars are enabled and a revision is made, a vertical line appears in the left margin. This visual cue alerts readers that there was a change in that line. Change bars appear in both the XML Editor and PDF outputs.

You can enable change bars from the Review tab of the Options dialog (**Review>Review Options**). Change bars can be enabled in addition to other track changes options (e.g., balloons, inline markup).



EXAMPLE

When you enable change bars, you can see them in both the XML Editor...



...and PDF output.

Procedure 1

Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content.

Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content.

~~Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content.~~

Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content.

This is a new paragraph. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content.

Change bars
are visible in
PDF output.

EXAMPLE

You are editing a document with track changes turned on. Some of your changes are very large (e.g., removed or added words), and others are small (e.g., fixed typos, changes to single words).

Because all of the changes are important, you decide to turn on change bars. By enabling change bars, a red bar appears next to the paragraph whenever there is a change, regardless of whether it is a large change or a small change.

When you publish the PDF output with the changes preserved, readers can scan the document for change bars. When they see a change bar, they know to look in that paragraph for the change. This is much easier and faster than skimming the entire document searching for additions or deletions, at the risk of missing an important change.

November 19, 1863

187

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. ~~We are met on a great battle-field of that war.~~ We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

~~Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.~~

But, in a larger sense, we can not dedicate -- we can not consecrate -- we can not hallow -- this ground. The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember what we say here, but it can not forget what they did here. It is for us the living, rather, to be dedicated to the great task remaining before us -- that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion -- that we here highly resolve that these dead men shall live by the actions of the living -- that the government of this nation, under God, shall have a new birth of freedom -- and that government of the people, by the people, for the people, shall not perish from the earth.

Abraham Lincoln

November 19, 1863

Change bars make it easier to spot all of the edits, even the small changes in the first and last line.

Tracked Changes in PDF Output

If you have tracked changes in Flare, you can now see the changes when producing a PDF output.

HOW TO PRESERVE TRACKED CHANGES

1. From the Targets folder in the Project Organizer, open the PDF target you want to edit.
2. Select the **Advanced** tab.
3. In the **Output Options** section, select **Preserve Tracked Changes**.

If you are using Acrobat or Acrobat Reader to view the output, you can view a list of the changes in the comments pane. You can also hover over any change to see additional information about the change, such as the reviewer who made the change and when it was made.

WHAT CHANGES ARE PRESERVED?

When you produce PDF output from Flare, the following changes are preserved.

- » Annotations
- » Concepts proxy
- » Conditions
- » Cross-references
- » Endnotes proxy
- » Equations
- » Footnotes
- » Glossary proxy
- » Glossary terms
- » Groups/divs
- » Images
- » Index proxy
- » Inline styles
- » Links
- » List-of proxy
- » Lists

- » Mini-TOC proxy
- » Page headers/footers for masterpages
- » QR codes
- » Relationships proxy
- » Snippet conditions
- » Snippet variables
- » Snippets (insertion/deletion)



Note: Inserting or deleting a snippet tracks the entire snippet. Changes within a snippet are not reflected in the output.

- » Style classes
- » Style IDs
- » Tables
- » Text effects
- » TOC proxy
- » Variables



Note: Some changes are tracked using change bars. You can enable change bars in the Options dialog.



Note: Tracked changes may not be visible in all PDF readers.

EXAMPLE

Let's say you tracked changes in Flare.

SanDiegoAttractions.htm

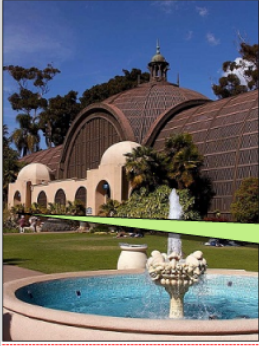
Layout (Mobile) • Medium (mobile) • Page Layout

San Diego Attractions

There are many places to visit in San Diego. Three of these are [Balboa Park](#), the [San Diego Zoo](#), and [Petco Park](#).

BALBOA PARK

Balboa Park is a 1200 acre park in the middle of downtown San Diego. Inside the park are gardens, walking paths, and more than a dozen museums. Every year the park hosts hundreds of events, including concerts, marathons and races, festivals and parades, food and beer tastings, and guest speakers.



SAN DIEGO ZOO

The San Diego Zoo was founded in 1916 and is located inside of Balboa Park. It is home to more than 3700 rare and endangered animals, and more than 700,000 exotic plants. Among the animals at the zoo are giant pandas, gorillas, koalas, and jaguars.

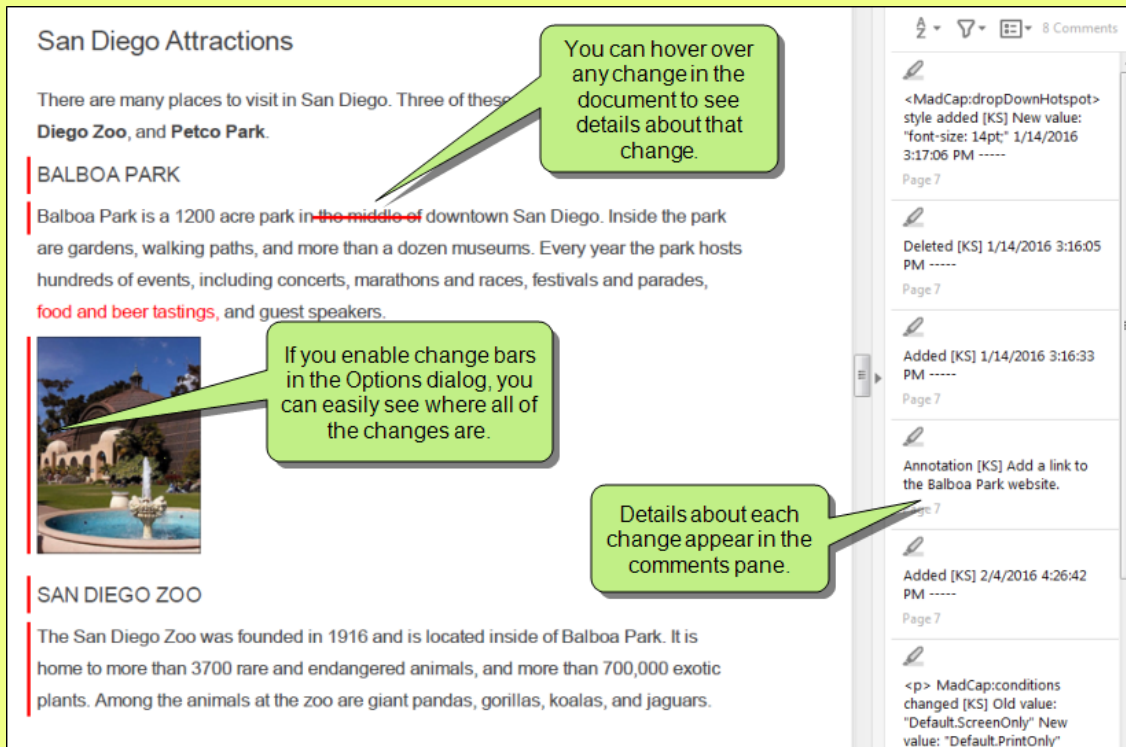
Changes in Flare appear in the sidebar and throughout the document.

Change bars have been enabled in the Options dialog, so a red line in the margin of the XML Editor indicates where your changes are.

Changes in the sidebar:

- <MadCap:dropDownHotspot> style added [KS] New value: "font-size: 14pt;"
- Deleted [KS]
- Added [KS]
- Annotation [KS] Add a link to the Balboa Park website.
- Added [KS]
- <MadCap:dropDownHotspot> style added [KS] New value: "font-size: 14pt;"
- <p> MadCap:conditions changed Old value: "Default: ScreenOnly" New value: "Default: PrintOnly"

If you preserve the tracked changes when you build your output in PDF, the changes will appear like this:



Tracked Changes in Word Output

If you have tracked changes in Flare, you can now see the changes when producing a Word output. You can also enable tracked changes in Word after you produce the Word output from Flare.

HOW TO PRESERVE TRACKED CHANGES

1. From the Targets folder in the Project Organizer, open the Word target you want to edit.
2. Select the **Advanced** tab.
3. In the **Output Options** section, select **Preserve Tracked Changes**.

You can view a list of the changes in the sidebar. You can also hover over any change to see additional information about the change, such as the reviewer who made the change and when it was made.

WHAT CHANGES ARE PRESERVED?

When you produce Word output from Flare, the following changes are preserved.

- » Annotations
- » Concepts proxy
- » Conditions
- » Cross-references
- » Endnotes proxy
- » Equations
- » Footnotes
- » Glossary proxy
- » Glossary terms
- » Groups/divs
- » Image formatting
- » Images
- » Index proxy
- » Inline styles
- » Links
- » List-of proxy

- » Lists
- » Mini-TOC proxy
- » Page headers/footers for masterpages
- » QR codes
- » Relationships proxy
- » Snippet conditions
- » Snippet variables
- » Snippets (insertion/deletion)
- » Style classes
- » Style IDs
- » Tables, including insertion/deletion of rows and columns
- » Text effects
- » TOC proxy
- » Variables



Note: Several items are tracked in the margin (using Word's built-in change bars), including proxies, images, equations, keywords, QR codes, and variables.



Note: If a change consists of multiple attribute changes, only the final attribute change is shown. Likewise, if you have multiple endnote or TOC proxies in your document, only the final one is shown.



Note: Divs are added to Word as tables.



Note: Page numbers in relationship proxies are not tracked.



Note: Running Head variables (Running HFs) are not tracked as changes.



Note: Inserting or deleting a snippet tracks the entire snippet. Changes within a snippet are not reflected in the output.

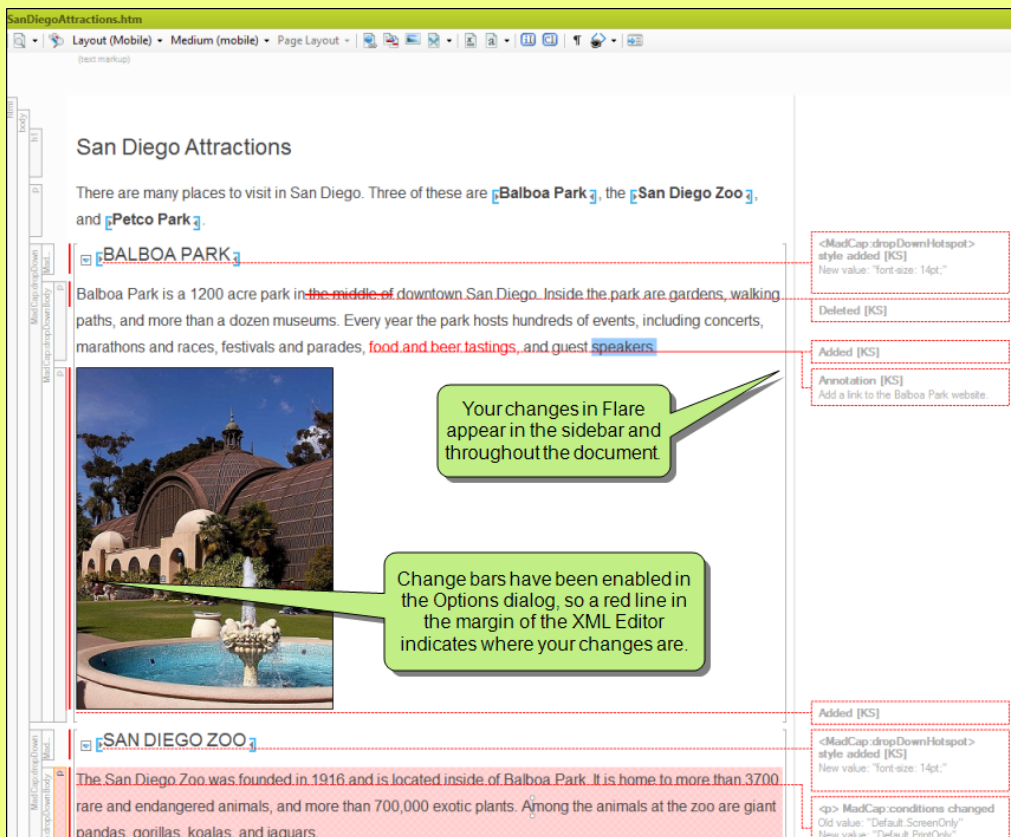
HOW TO ENABLE REVIEW MODE IN WORD

If you want to automatically track any changes made to the Word document after you open it, you can enable review mode in the Word output. This will automatically enable the Track Changes feature in Word.

1. From the Targets folder in the Project Organizer, open the Word target you want to edit.
2. Select the **Advanced** tab.
3. In the **MS Word Output** section, select **Enable review mode in output**.

EXAMPLE

Let's say you tracked changes in Flare.



If you preserve the tracked changes when you build your output in Word, the changes will appear like this:

San Diego Attractions

There are many places to visit in San Diego. Three of these are **Balboa Park**, the **San Diego Zoo**, and **Petco Park**.

BALBOA PARK

Balboa Park is a 1200 acre park in ~~the middle of~~ downtown San Diego. Inside the park are gardens, walking paths, and more than a dozen museums. Every year the park hosts hundreds of events, including concerts, marathons and races, festivals and parades, **food and beer tastings**, and guest **speakers**.

SAN DIEGO ZOO

The San Diego Zoo was founded in 1916 and is located inside of Balboa Park. It is home to more than 3700 rare and endangered animals, and more than 700,000 exotic plants. Among the animals at the zoo are giant pandas, gorillas, koalas, and jaguars.

kschneider
Formatted: Font: 14 pt

kschneider
Formatted: dropDownHead

kschneider
Add a link to the Balboa Park website.

kschneider
Formatted: Font: 14 pt

kschneider
Formatted: dropDownHead

KSchneider
New condition value: Default.PrintOnly. Old condition value: Default.ScreenOnly.

Word adds change bars to the margins for some changes, such as images.

You can hover over any change in the document to see details about that change.

All of the changes appear in the sidebar.

If you enable review mode in the output, Word will automatically track any additional changes you make in the output.

REVIEW | VIEW | ACROBAT

New Comment | Delete | Previous | Next | Show Comments | **Track Changes** | Reviewing Pane | All Markup

The "Enable review mode in output" option in the Word target automatically enables the Track Changes feature in Word.

San Diego Attractions

There are many places to visit in San Diego. Three of these are **Balboa Park**, the **San Diego Zoo**, and **Petco Park**.

BALBOA PARK

Balboa Park is a 1200 acre park in ~~the middle of~~ downtown San Diego. Inside the park are gardens, walking paths, and more than a dozen museums. Every year the park hosts hundreds of events, including concerts, marathons and races, festivals and parades, **food and beer tastings**, and guest **speakers**.

Kate Schneider
Formatted: Font: Cooper Black, Font color: Red

kschneider
Formatted: dropDownHead

kschneider
Add a link to the Balboa Park website.

If you make changes...

...they will appear in the document with any preserved changes from the Flare output.

CHAPTER 7

Additional New Features—March 2016

The following additional new features were added in this version.

This chapter discusses the following:

64-bit Support	201
Build Log Save	202
Byte Order Mark	203
Condition Enhancements	204
Drop-Down and Expanding Text Enhancements	213
Equation Editor	219
Glossary Enhancements	224
Help—Live or Local	238
Indexes—Sorting	242
Link to Styles	245
Macros—Assign Shortcuts in Dialog	247
Master Page Enhancements	251
Output—Choose Specific Browser or Viewer	256
Page Break—Independent of HTML Element	257
Report—Word Count	263
Skins—Runtime	265
Smart Quotes Per Language	278
Source Control Enhancements	279
Tables—Advanced Sorting	289



Variable Enhancements	295
Word Enhancements	302
XML Editor Enhancements	312

64-bit Support

Flare is now a 64-bit application. This is beneficial because 64-bit applications provide better access to your computer's memory (RAM). This is especially useful if you have a large project or often work with multiple files at once, because Flare will run faster and more efficiently.



Note: If you have any mappings to the Flare application (e.g., shortcuts, command line batch files), you will need to update them so they point to the Program Files folder. Previous versions of Flare were native 32-bit applications and were installed in the Program Files (x86) folder.

For example, the new location of the Flare application is **C:\Program Files\MadCap Software\MadCap Flare 12\Flare.app\Flare.exe**.



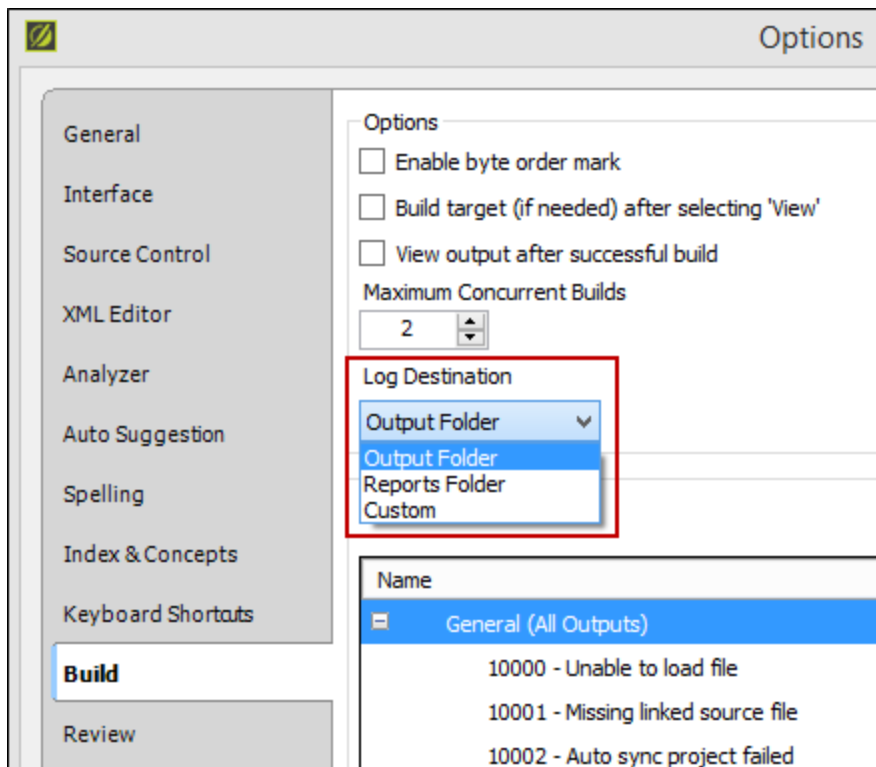
Note: If you want to import Doc-To-Help projects to a 64-bit version of Flare, you will need to download Microsoft Access Database Engine 2010 x64 edition. If you do not already have this add-in, Flare will prompt you to install it from the Microsoft Download Center.

Build Log Save

You can choose where your log file will be stored when you build a target.

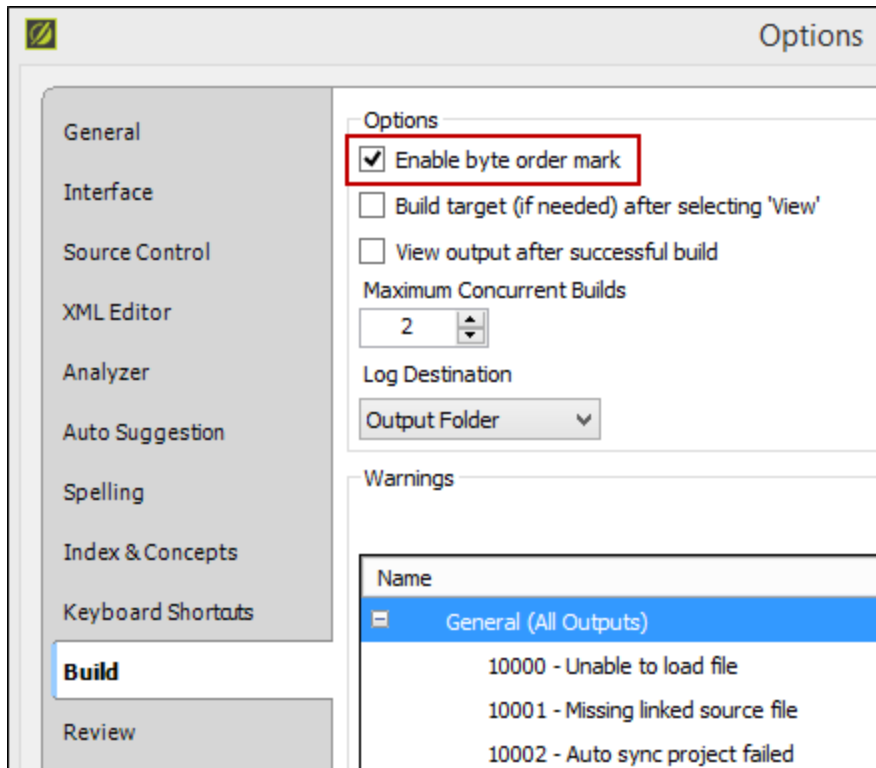
You can choose where your log file will be stored when you build a target .

This can be done on the Build tab of the Options dialog. The default location is the Output folder, but you can also choose to place it in the Reports folder of the Project Organizer, or you can select a custom location.



Byte Order Mark

You can enable the byte order mark (BOM) on the Build tab of the Options dialog. Previously, to enable BOM, you would have to open and edit your system registry.



The BOM is a Unicode character that appears at the start of a text stream. Most authors probably want to disable the BOM. But if you are hosting content on a web server that requires BOM (such as a LINUX machine), you can enable it.


On the first launch of Flare, it will use the preset value of the BOM option from your registry. If there isn't a value in the registry, it will be turned off. After the first launch, you can only enable or disable BOM through the Options dialog. The registry will reflect the changes, but the registry will not be used to control the enabling or disabling of the BOM.

Condition Enhancements






























Some enhancements have been made to conditions in Flare. You can now apply conditions to glossary terms and to styles. In addition, it is now possible to preserve tracked changes to conditions in Word output.

CONDITIONS ON GLOSSARY TERMS

You can apply conditions to terms in the Glossary Editor.

1. Open the Project Organizer.
2. Expand the **Glossaries** folder.
3. Double-click the glossary that you want to open. The Glossary Editor opens to the right.
4. Select the term. If you want to apply the same condition(s) to multiple terms, hold down your **SHIFT** or **CTRL** key and select the terms.
5. In the local toolbar click . The Properties dialog opens.
6. Select the **Conditional Text** tab. The first condition tag set selected and the associated condition tags are shown on the right.
7. If you want to see condition tags for a different condition tag set, select it.
8. For each condition tag that you want to apply to the entry, click the blank square next to the tag. A check mark appears in the square.
9. Click **OK**.

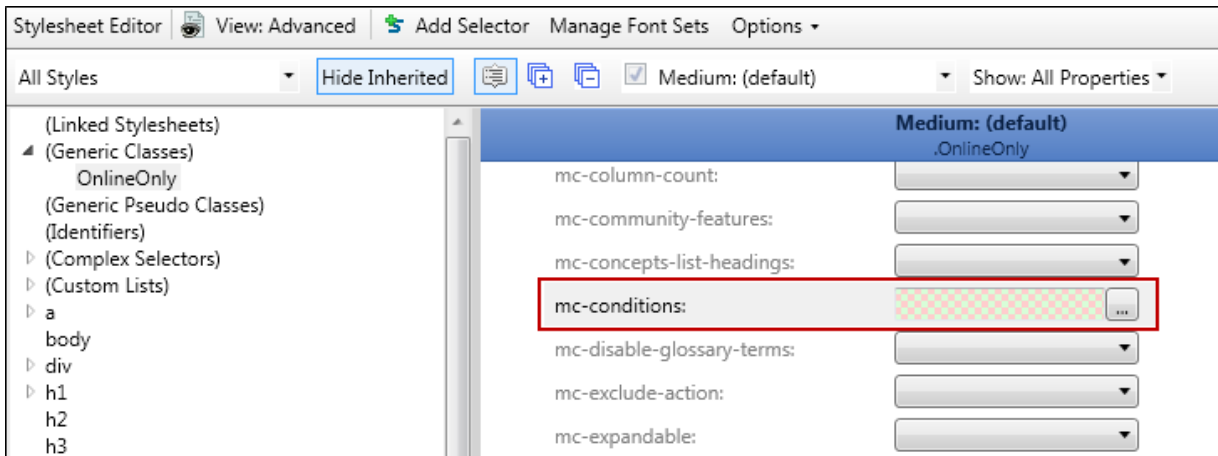
The Condition column indicates whether a term has conditions applied to it.

Glossary Editor       0 Duplicate Term(s) Update			
	Condition	Terms ▲	Definition
		Topic	A chunk of information about a page
		Condition tag	A marker that you can apply to different
		XML Editor	The window in the Flare interface v
		Style	An element that contain pre-set fo
	<input type="checkbox"/>	Single-Sourcing	"Single-Sourcing" is a fancy term t
	<input type="checkbox"/>	Target	One "instance" of an output type.
	<input type="checkbox"/>	Snippet	A pre-set chunk of content that yo
		Variable	A pre-set term or content that you
		Table	A group of intersecting columns a
	<input type="checkbox"/>	Drop-down text	A feature that lets you "scrunch up
	<input type="checkbox"/>	Text snippet	A snippet that is created out of a p
	<input type="checkbox"/>	Block snippet	A snippet that is created out of one
	<input type="checkbox"/>	Span	A tag that is used to group inline e
	<input type="checkbox"/>	Cross-reference	A navigation link that lets you con
	<input type="checkbox"/>	Footnote	A comment that is used to explain
		Batch target	A special target that lets you build

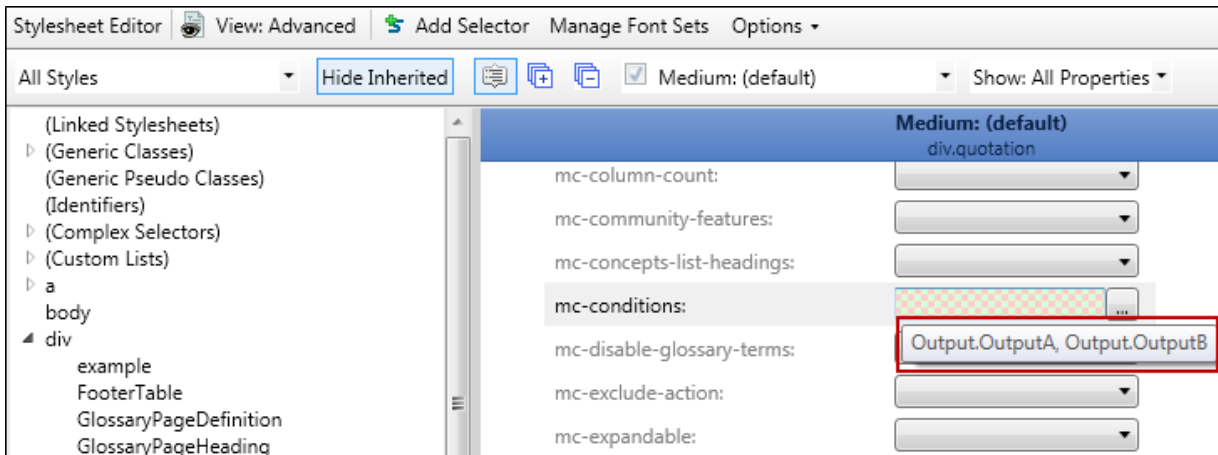
10. Click  to save your work.

CONDITIONS ON STYLES

In the Advanced view of the Stylesheet Editor, you can associate a condition with a style by using the **mc-conditions** property. If you are viewing properties by group (rather than alphabetically) in the Stylesheet Editor, you can find this property in the Unclassified group.



If you hover over this property, a tool tip shows the names of the conditions that have been set on it.



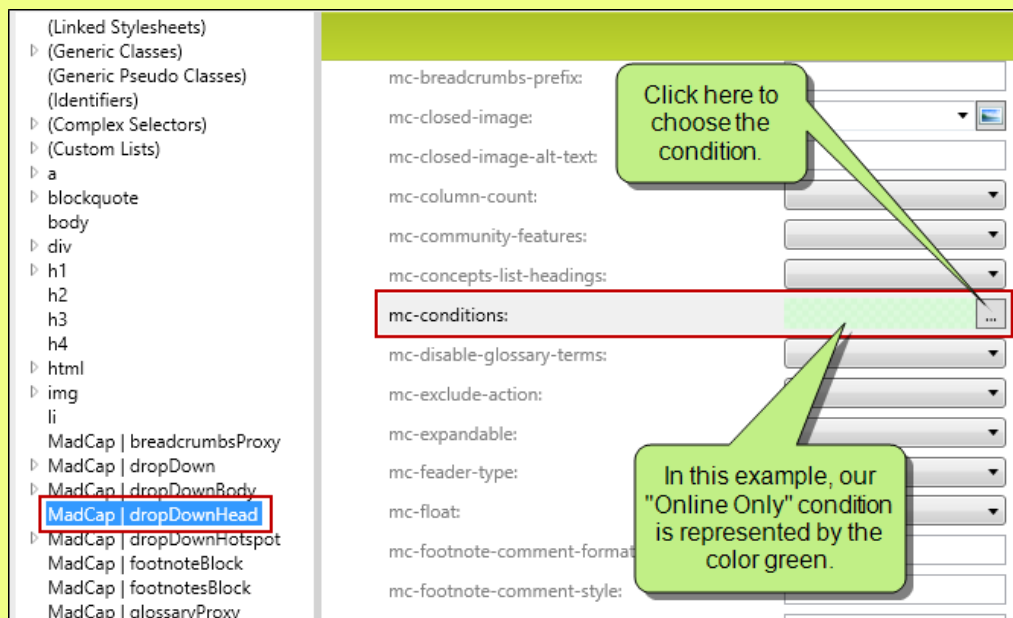
EXAMPLE

Let's say you have created a lot of drop-downs throughout your project. But you want the heading portion of the drop-down to display only in online outputs, not in any print-based outputs.

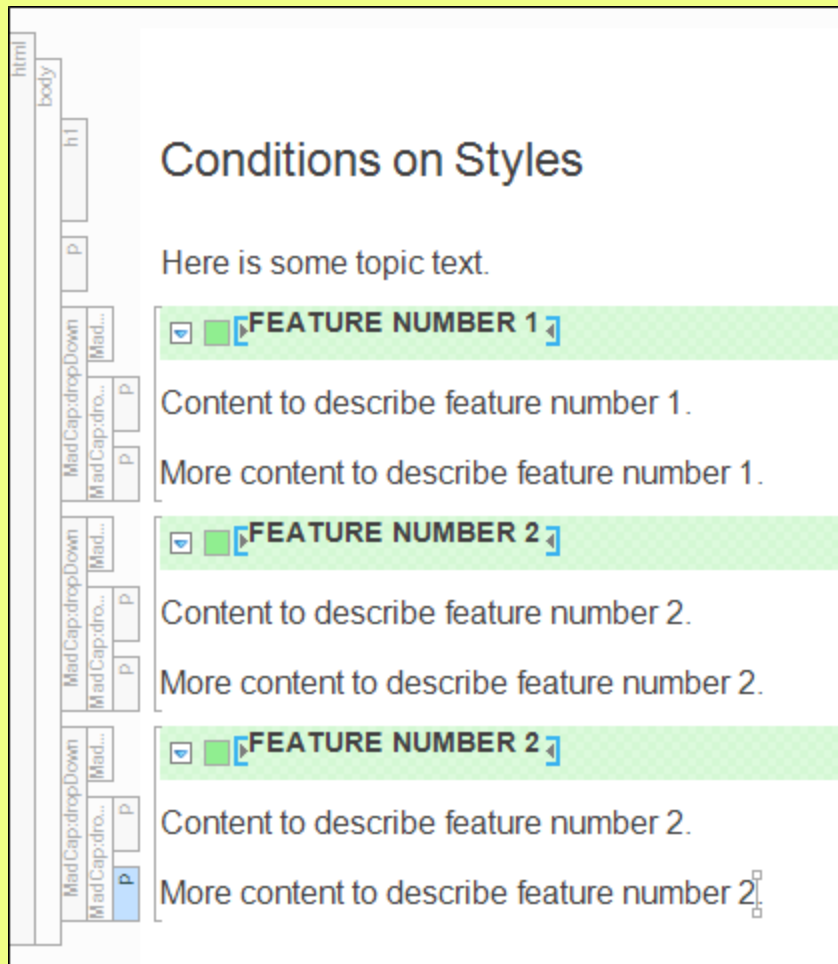
The screenshot shows a web editor interface. On the left is a vertical sidebar with a tree view containing elements like 'html', 'body', 'h1', 'p', and several 'MadCap:dropDown' items. The main workspace displays a document titled 'Conditions on Styles'. The document content includes the heading 'Conditions on Styles', followed by the text 'Here is some topic text.', and then three expandable sections. The first section is titled 'FEATURE NUMBER 1' and contains two paragraphs: 'Content to describe feature number 1.' and 'More content to describe feature number 1.'. The second and third sections are both titled 'FEATURE NUMBER 2' and each contain two paragraphs: 'Content to describe feature number 2.' and 'More content to describe feature number 2.'. A red rectangular box highlights the 'FEATURE NUMBER 1' heading. A green callout bubble points to this box with the text: 'You want this content to be shown only in online outputs.'

Of course you can manually apply conditions to your drop-down headings, but you would have to do that each time you create a drop-down. A better option is to apply a condition to the style used for the drop-down headings.

So you open your stylesheet and select the **MadCap|dropDownHead** style. And then in the **mc-conditions** property you select a condition that you've created for the purpose of online only outputs.



As a result, the online only condition will automatically be applied to any drop-downs you've created in the past, and any that you create in the future.



So when you build online output, you will see this:

Conditions on Styles

Here is some topic text.

▼ FEATURE NUMBER 1

Content to describe feature number 1.

More content to describe feature number 1.

► FEATURE NUMBER 2

► FEATURE NUMBER 2

And when you build print-based output, you will see this:

Conditions on Styles

Here is some topic text.

Content to describe feature number 1.

More content to describe feature number 1.

Content to describe feature number 2.

More content to describe feature number 2.

Content to describe feature number 2.

More content to describe feature number 2.



Note: Conditions that are set locally (i.e., directly on content) will override conditions set on a style. Even if the local condition is empty, it still overrides the conditions set from stylesheet.



Note: When you set a condition tag on a style, you can optionally use the **mc-exclude-action** property to set an exclude action on the tag. For example, you might have the unbind action for a condition tag if you have applied the tag to a hyperlink and want the link to be removed from the text in some outputs, but you still want the text to be shown in those outputs.

TRACKED CHANGES TO CONDITIONS

If you are tracking changes in Flare and make changes to your condition tags, you can preserve the tracked changes when you build Word output. This will let you see the changes you made to your condition tags in your final Word output.

Procedure 1

Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content.

When you preserve tracked changes in Word output, changes to conditional tags are tracked.

New condition value: General.Audience B.

New condition value: General.Audience A.

~~Here is some general text for a topic. Replace this with your own content. Here is some general text for topics. Replace this with your own content. Here is some general text for a topic. Replace this with your own content. Here is some general text for a topic. Replace this with your own content.~~

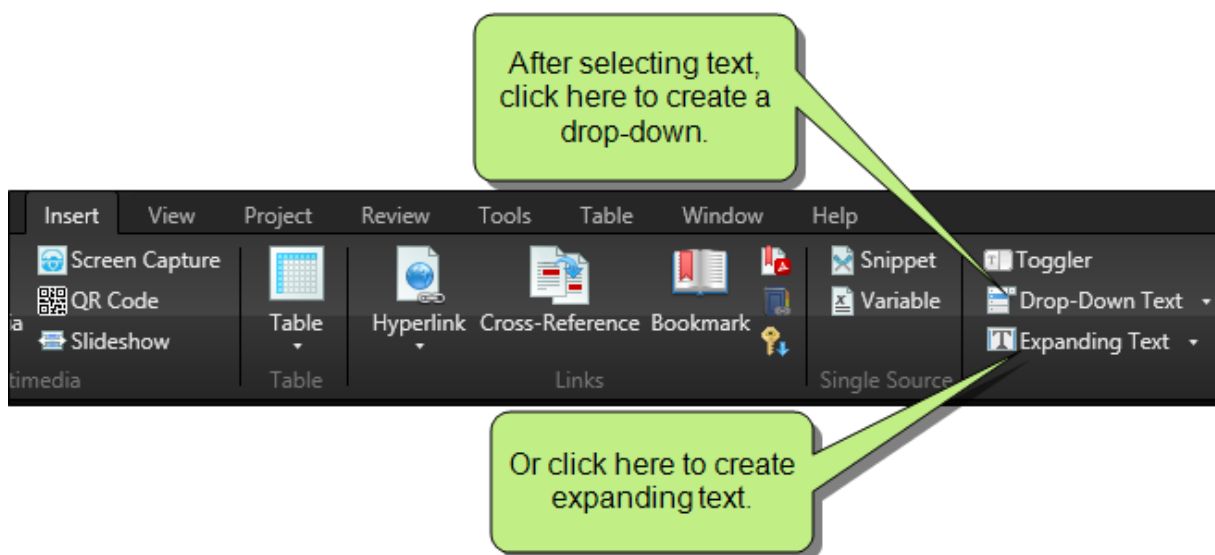
~~Here is some general text for a topic. Replace this with your own content. Here~~



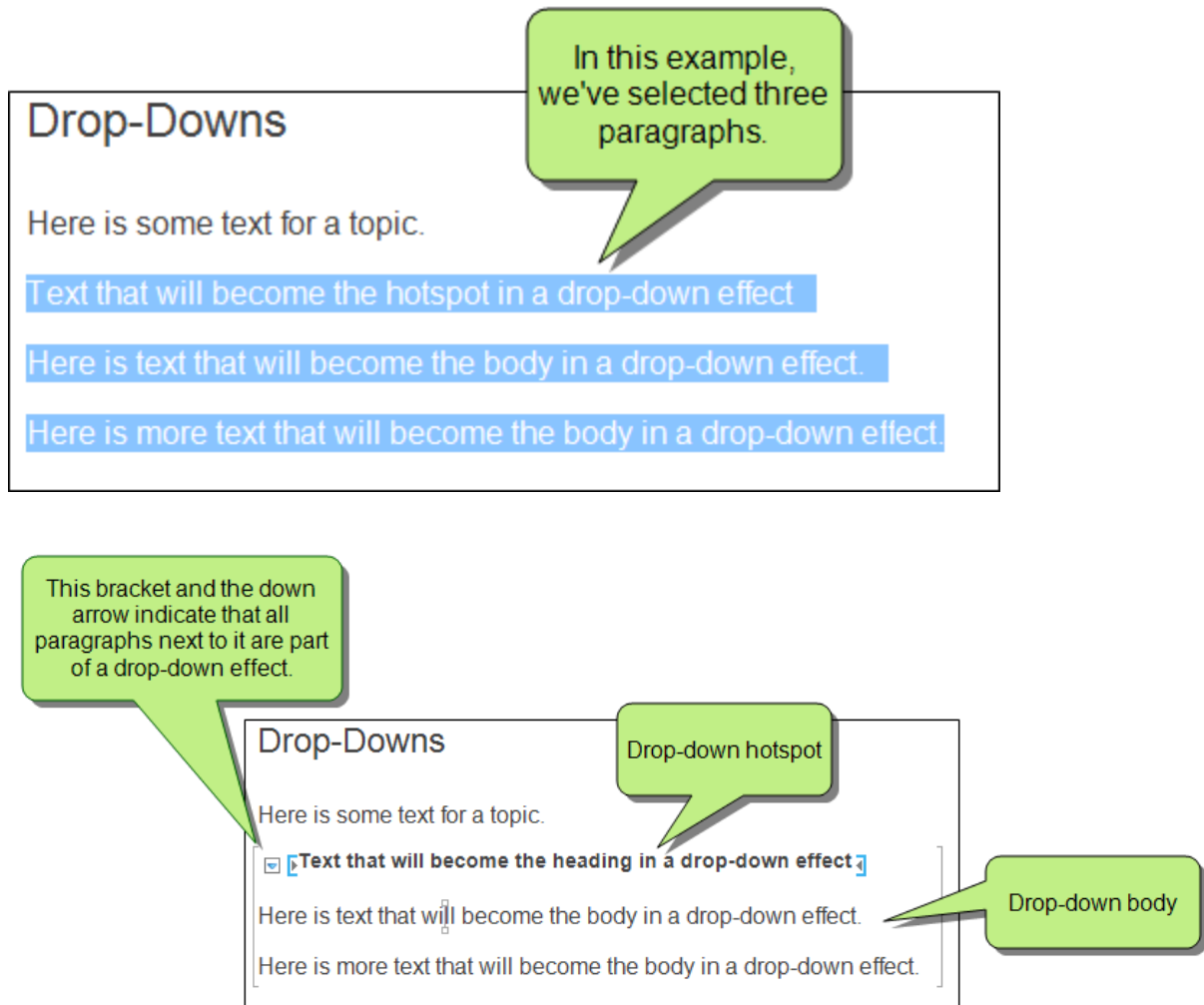
Note: You can see tracked changes to condition tags in PDF output if you enable change bars. However, condition changes are not explicitly marked in the output, and could be missed if there are other changes to the paragraph that would also enable a change bar.

Drop-Down and Expanding Text Enhancements

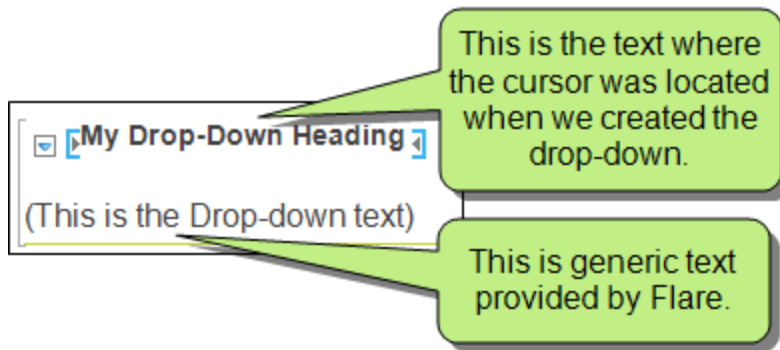
The drop-down and expanding text features have been simplified with the removal of their respective dialogs. Starting with this version, when you select one of these options from the Insert ribbon, the drop-down or expanding text feature is immediately created in the content file.



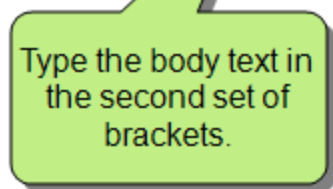
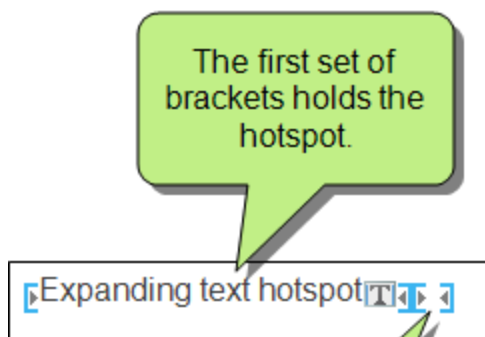
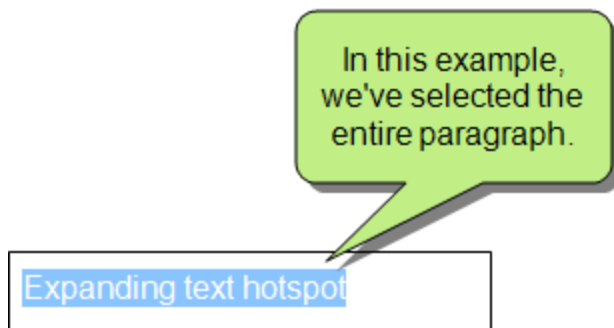
For drop-downs, if you select multiple paragraphs before choosing this option, the first paragraph becomes the hotspot. The paragraphs under it become the drop-down body.



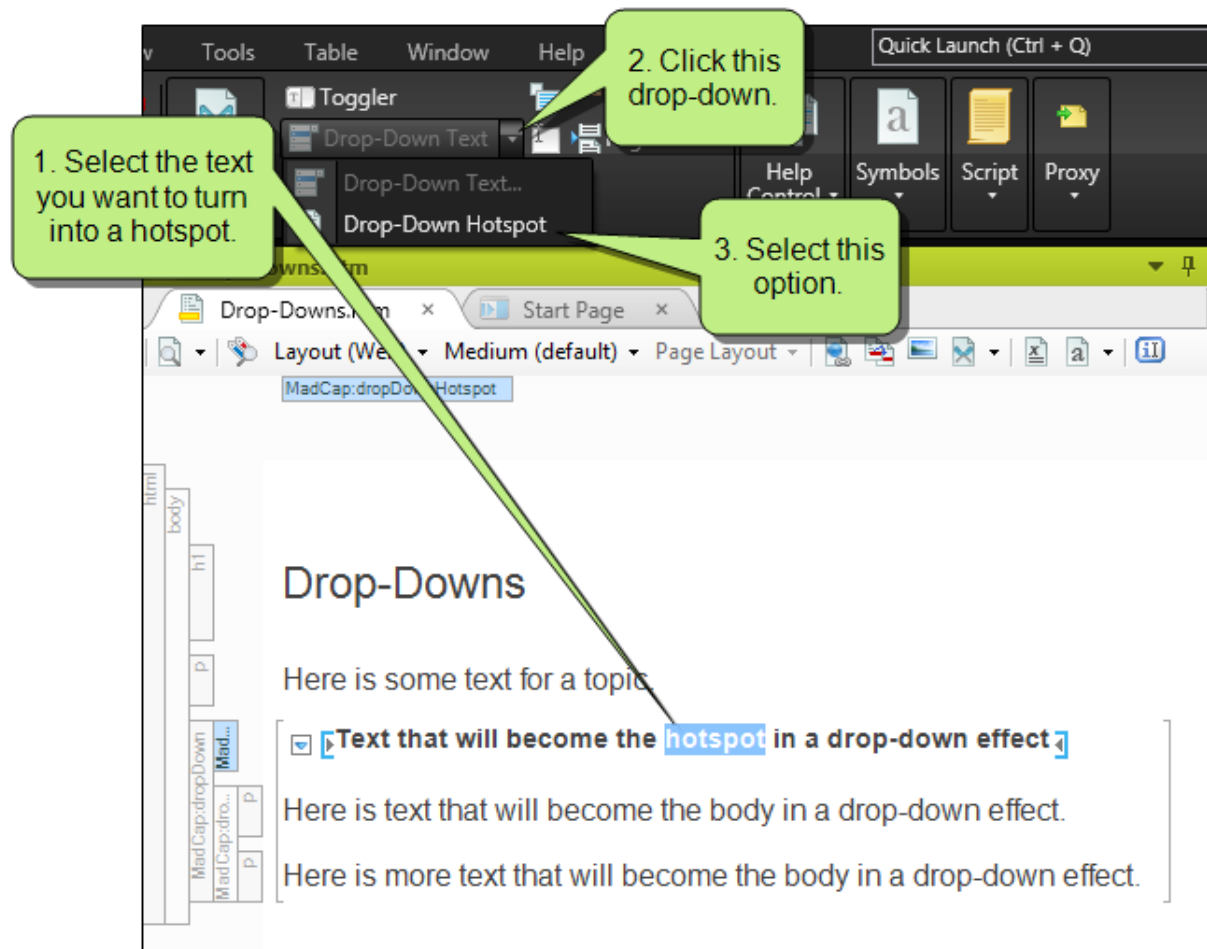
If you do not select multiple paragraphs, Flare adds some generic text after the heading for you to replace.



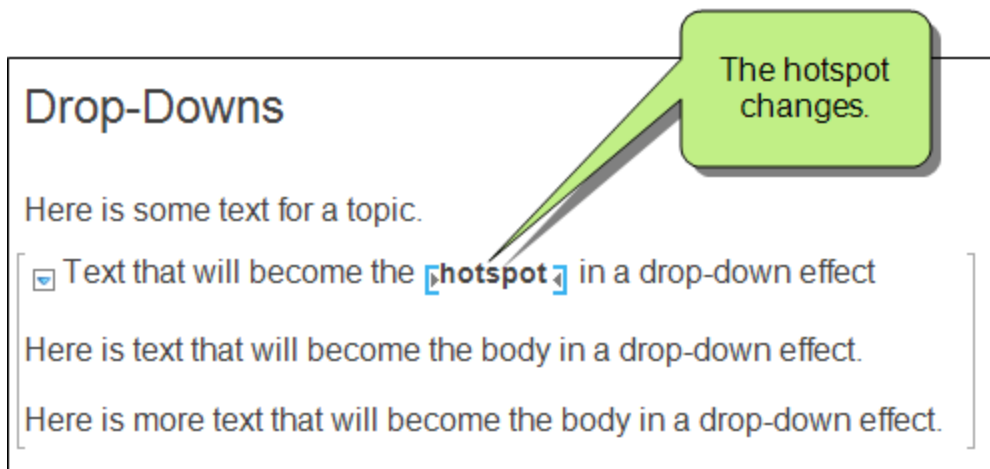
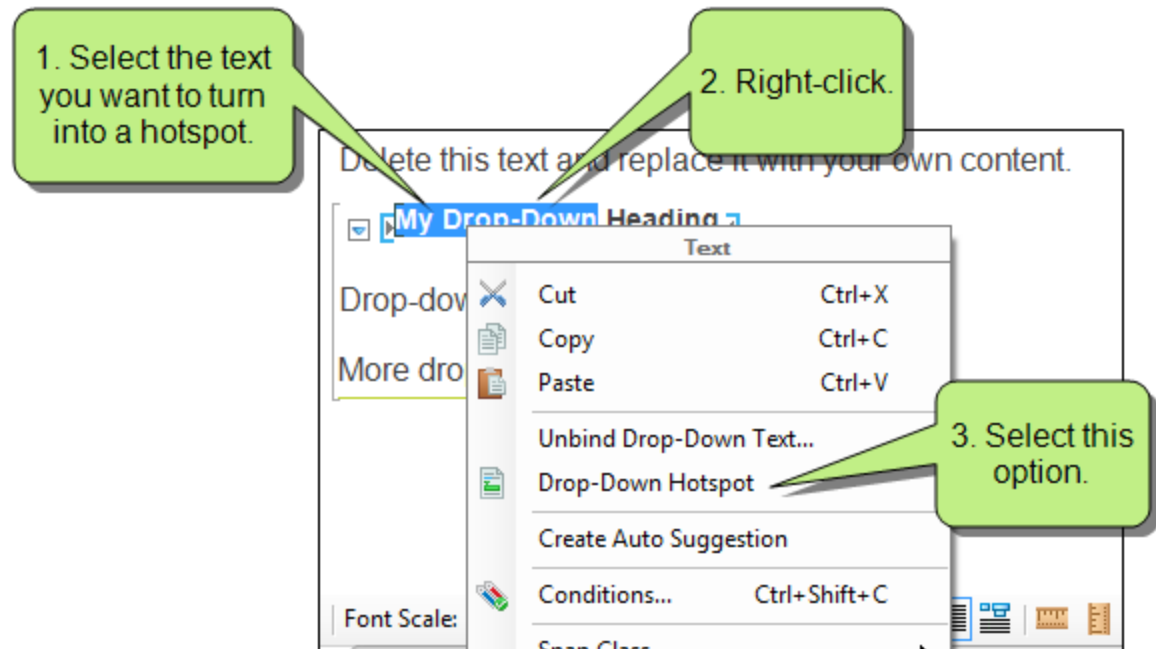
For expanding text, whatever you select in the paragraph becomes the hotspot. If you do not select any text in the paragraph, it all becomes the hotspot. You can then type the expanding text body in the second bracket.



To change the hotspot for either a drop-down effect or expanding text area, you can use the drop-down field from the ribbon button.



Alternatively you can select an option from the context (right-click) menu.



Equation Editor

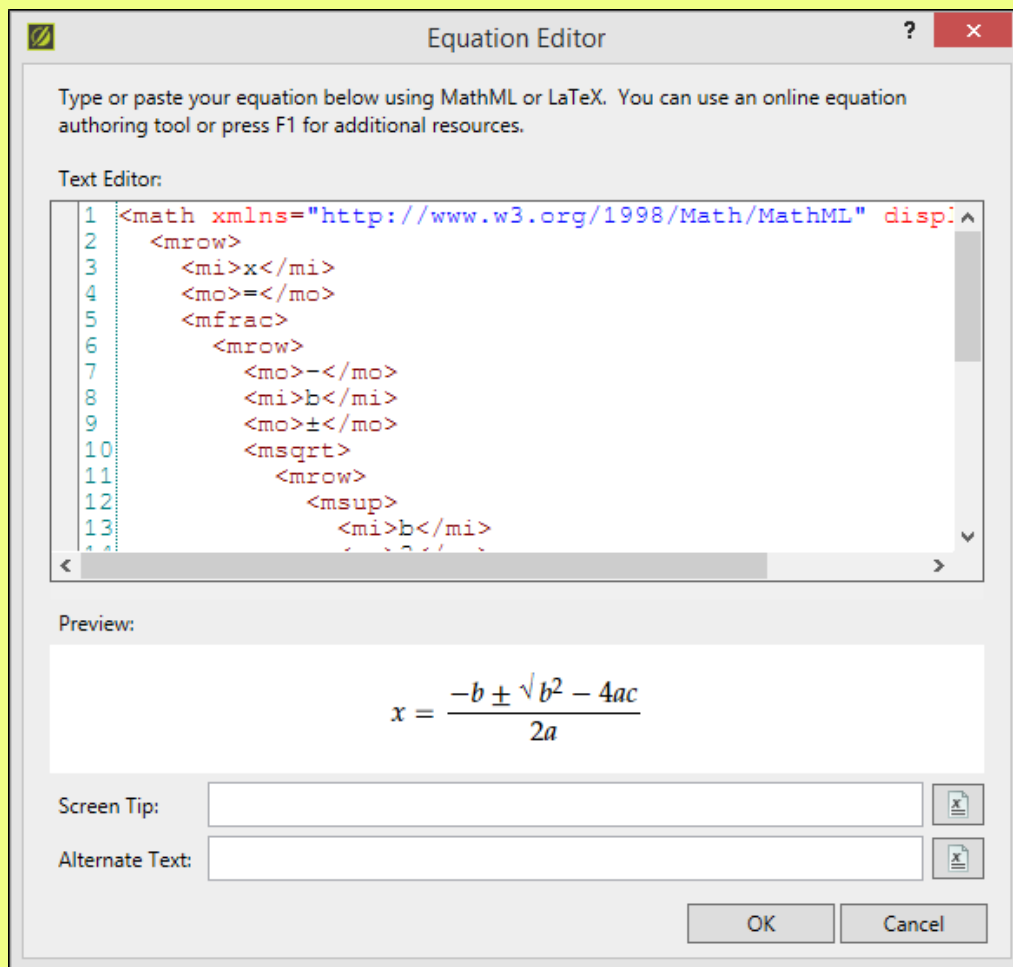
The Equation Editor in Flare has changed. In previous versions, a third-party editor was built into Flare for creating MathML equations. Starting with this version, you can use a third-party editor outside of Flare to create your equations written in MathML or LaTeX. You can perform a Google search for free online resources.

After you've created an equation, you can then copy and paste it into Flare's editor. This gives you more flexibility to use the editor and format you want.

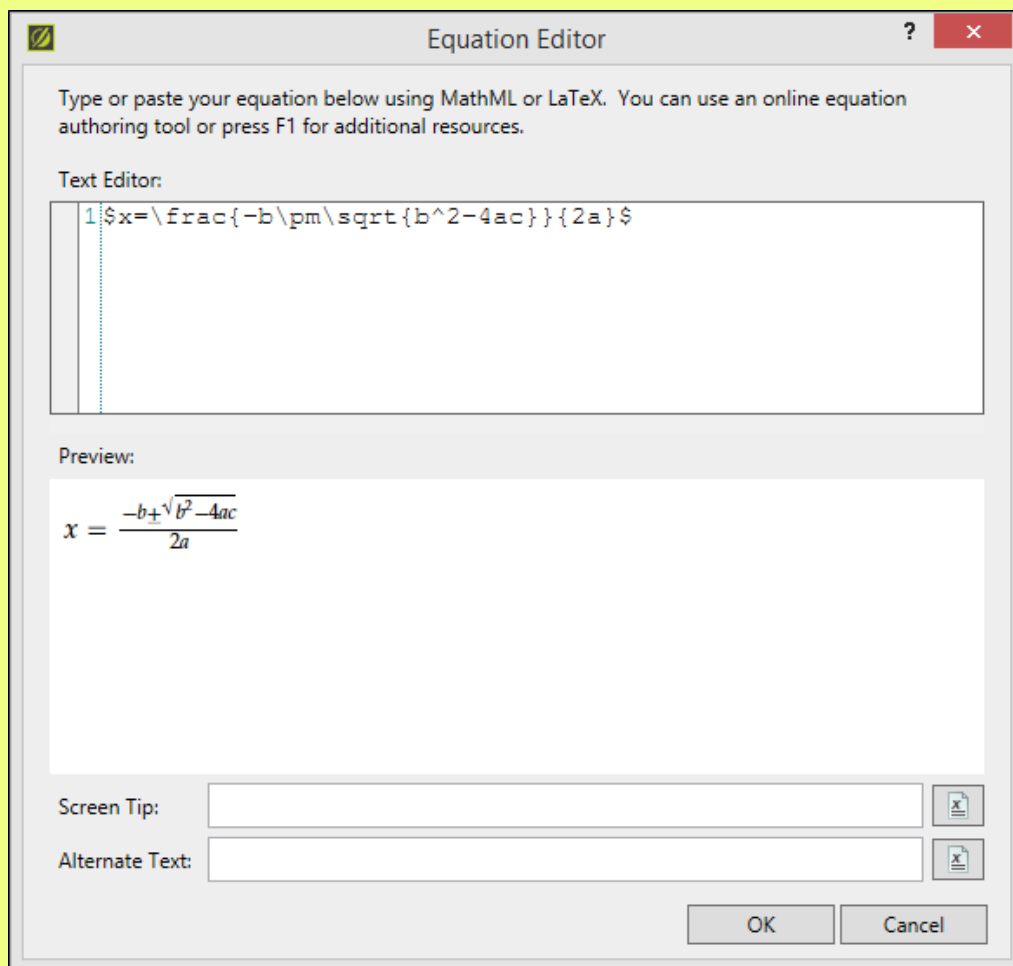
Once you add an equation into your project, Flare uses MathJax to render it. For more about MathJax, see <https://www.mathjax.org/> and <https://docs.mathjax.org>.

EXAMPLES

Here is an example of a MathML equation pasted into Flare's Equation Editor:



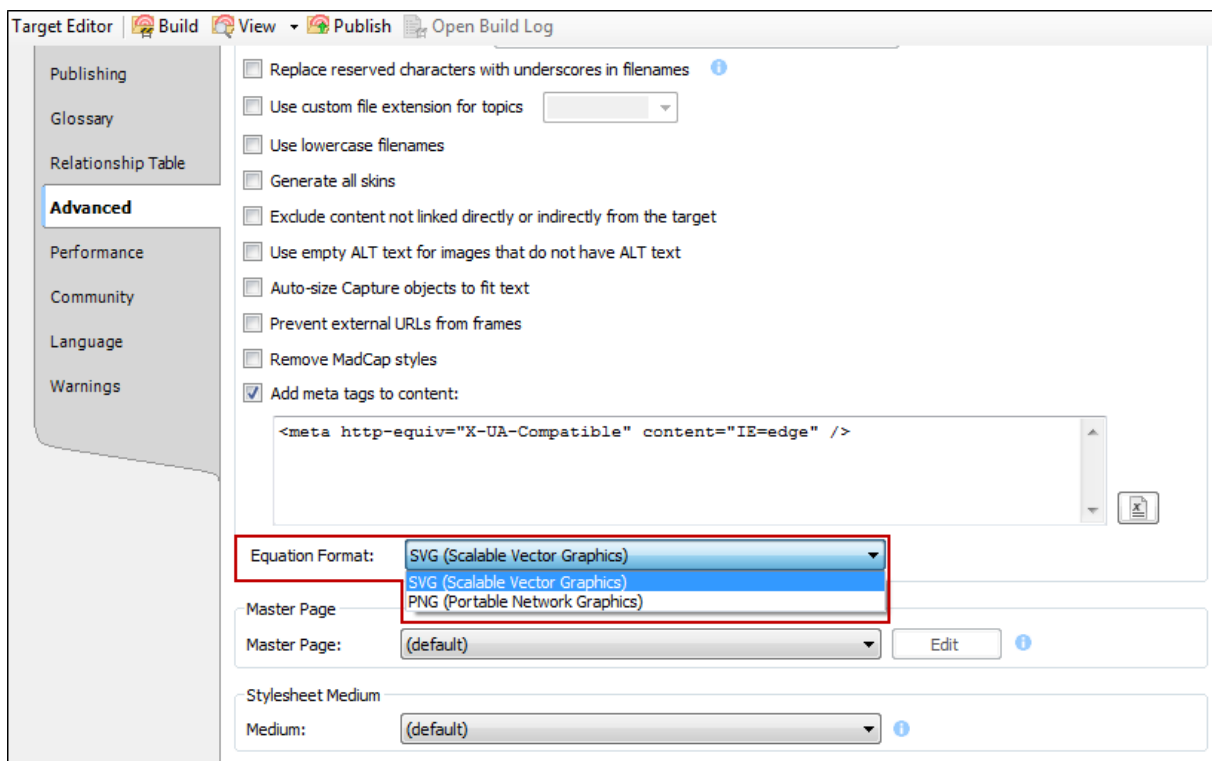
Here is an example of a LaTeX equation pasted into Flare's Equation Editor:



In previous versions, equations were always converted to PNG images in the output. The exception was EPUB, which supported MathML in the output.

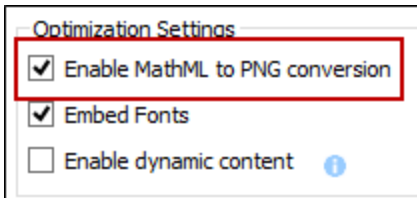
Starting with Flare 12, you can choose the output format for equations when generating certain targets. For HTML5 and the WebHelp formats—WebHelp, WebHelp Plus, WebHelp Mobile, or WebHelp AIR—you can choose between the SVG or PNG as the converted format.

To do this, open the Target Editor, select the **Advanced** tab, and choose the format from the **Equation Format** field. SVG provides a crisper image and is selected by default.



Note: The "Generate 'web-safe' images" option in the Target Editor does not apply to equations. If you want web-safe equations, you should choose PNG from the Equation Format field.

With EPUB targets, you can continue to generate MathML from your equations. However, you now also have the option of converting equations in EPUB to PNG format. To do this, open the EPUB target and select the **EPUB Options** tab. In the **Optimization Settings** area, you can use the check box labeled **Enable MathML to PNG conversion**.



- » If the option *is selected* both MathML and LaTeX equations are converted to PNG.
- » If the option *is not selected* LaTeX equations are converted to MathML. And of course MathML equations remain as they are.

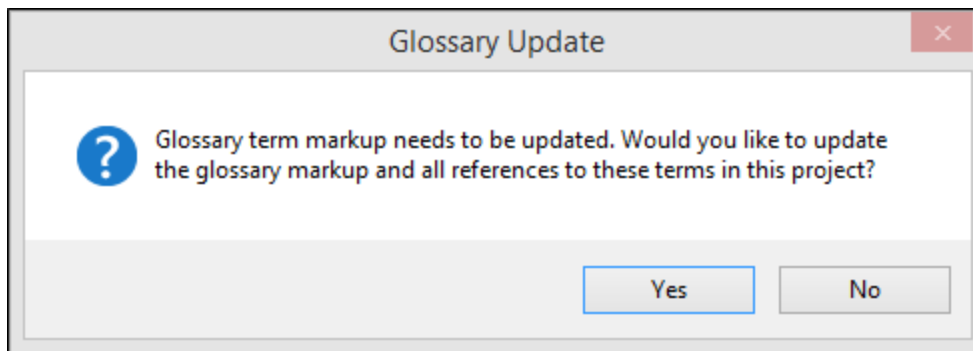
For the rest of the output types in Flare, equations are converted to PNG in the output.

Glossary Enhancements

Some enhancements have been made to glossaries in Flare. This includes the ability to control case-sensitivity and variants in glossary term links. You can also now choose a glossary term for any highlighted text when creating a link. Sorting glossary terms is also now possible. And there are other enhancements related to glossaries in Flare 12.

UPDATING GLOSSARY REFERENCE MARKUP
















If you created a project with a previous version of Flare and then open a glossary in Flare 12, you will be prompted to update the glossary and all content files containing related glossary term links.



This occurs because the markup for glossaries and glossary term links has changed due to new features in Flare 12. This, however, is optional. Flare will continue to support the old markup if you do not update the files.

If you choose not to update a glossary (e.g., you might be connected to source control and don't want to update all of your files at the moment), you will continue to be prompted each time you open that glossary file.

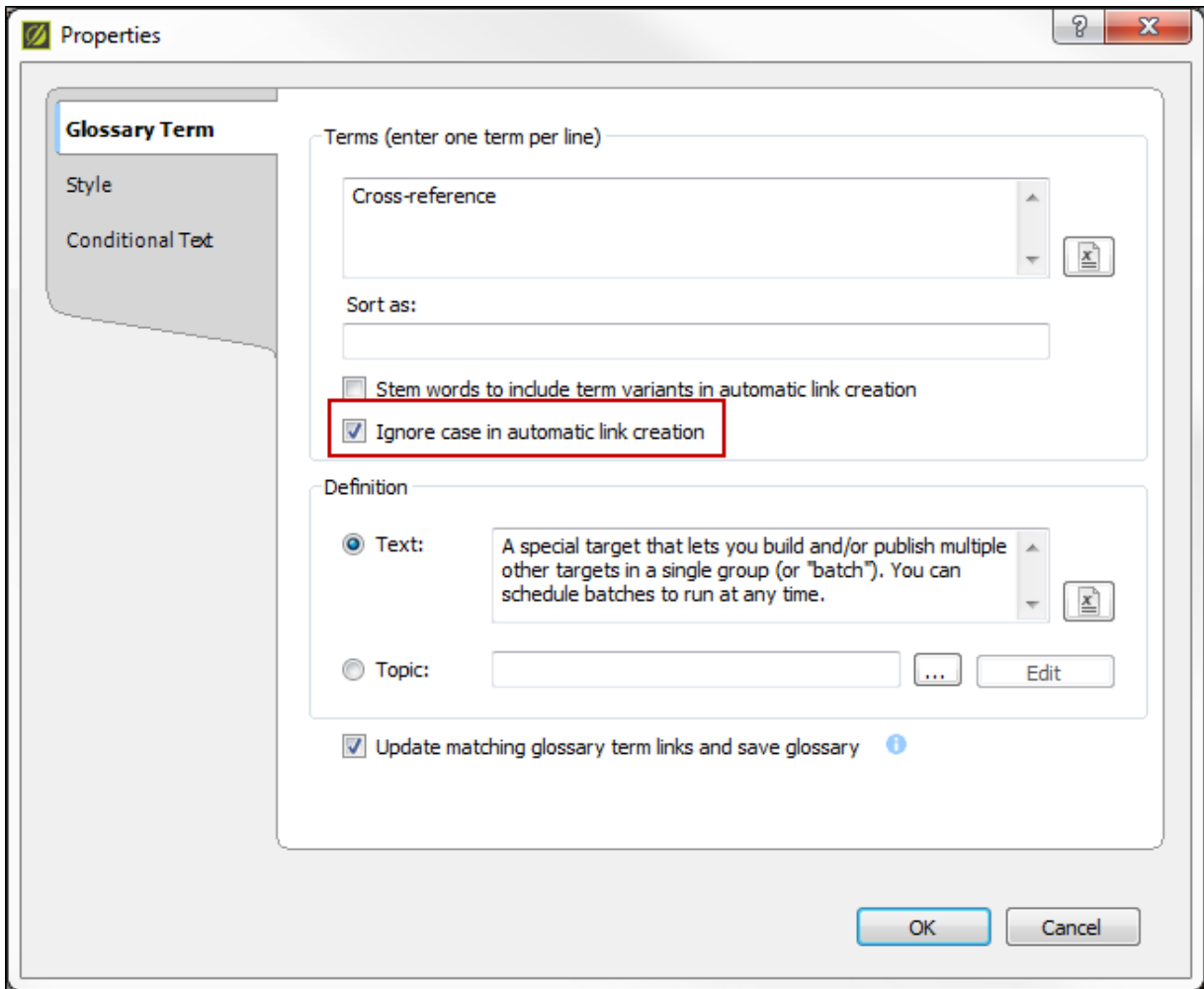
Also, you will notice a button in the local toolbar of the Glossary Editor that is labeled "Update Glossary Reference Markup."

Glossary Editor        0 Duplicate Term(s) Update Glossary Reference Markup 				
	Co...	Terms	Definition	Link
	<input type="checkbox"/>	Batch target	A special target that lets you build ...	
	<input type="checkbox"/>	Block snippet	A snippet that is created out of on...	
	<input type="checkbox"/>	Condition tag	A marker that you can apply to dif...	
	<input type="checkbox"/>	Cross-reference	A navigation link that lets you con...	
	<input type="checkbox"/>	Drop-down text	A feature that lets you collapse co...	
	<input type="checkbox"/>	Footnote	A comment that is used to explain ...	
	<input type="checkbox"/>	Single-Sourcing	Reusing content and producing m...	

For most authors, you probably won't need to use this button. But if you import topics from an older Flare project, you might need to click this button to perform the glossary update for those files as well.

CASE-SENSITIVITY AND VARIANTS

By default, Flare considers the case of text when automatically creating glossary term links. However, in the Properties dialog for a glossary term—which you can open by double-clicking a term in the Glossary Editor—you can tell Flare to ignore the case.



Also, a "Stem" option can be used for a glossary term.

The screenshot shows a 'Properties' dialog box with a 'Glossary Term' tab selected. The left sidebar contains 'Glossary Term', 'Style', and 'Conditional Text'. The main area is divided into 'Terms' and 'Definition' sections. The 'Terms' section has a text area with 'Cross-reference', a 'Sort as:' field, and two checked checkboxes: 'Stem words to include term variants in automatic link creation' (highlighted with a red box) and 'Ignore case in automatic link creation'. The 'Definition' section has a 'Text' radio button selected with the value 'A snippet that is created out of one or more paragraphs.', and a 'Topic' radio button with an empty field and an 'Edit' button. At the bottom, there is a checked checkbox 'Update matching glossary term links and save glossary' with an information icon. 'OK' and 'Cancel' buttons are at the bottom right.

Properties

Glossary Term

Style

Conditional Text

Terms (enter one term per line)

Cross-reference

Sort as:

☒ Stem words to include term variants in automatic link creation

☒ Ignore case in automatic link creation

Definition

☒ Text: A snippet that is created out of one or more paragraphs.

























☐ Topic: ... Edit

☒ Update matching glossary term links and save glossary ⓘ

OK Cancel

This allows you to show only the basic (i.e., stem) format of a term in the glossary, rather than all variations (e.g., singular and plural). However, variations of the term will be included in any glossary term links that are created. (If you select this option, the check box to ignore the case of text is also selected automatically.)

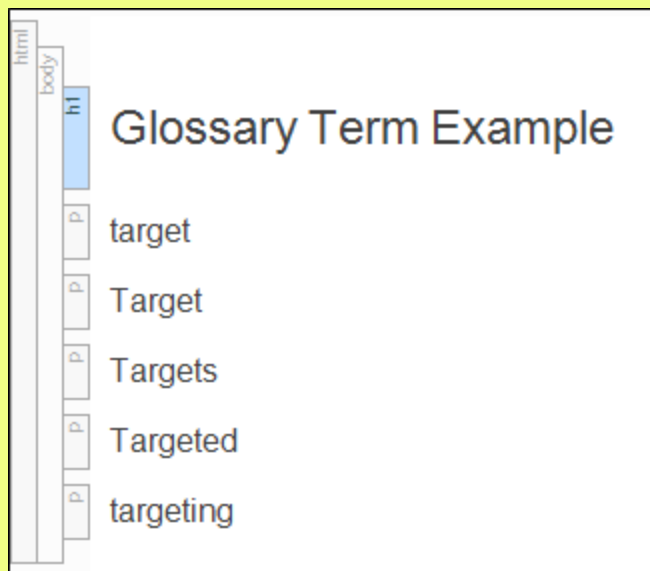
When you use the stem or case fields in the Properties dialog, this is indicated by check boxes in the Glossary Editor.

Glossary Editor       0 Duplicate Term(s) Update Glossary Reference Markup 								
	Co...	Terms	Definition	Link	Class	Sort As	Variants	Ignore Case
	<input type="checkbox"/>	Batch target	A special target that lets you build and/or ...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Block snippet	A snippet that is created out of one or mo...				<input type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	Condition tag	A marker that you can apply to different a...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Cross-reference	A navigation link that lets you connect tex...				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	Drop-down text	A feature that lets you "scrunch up" conte...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Footnote	A comment that is used to explain a speci...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	MyTerm					<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Single-Sourcing	"Single-Sourcing" is a fancy term that me...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Snippet	A pre-set chunk of content that you can u...				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	Span	A tag that is used to group inline element...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Style	An element that contain pre-set formattin...				<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/>	Table	A group of intersecting columns and row...				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	Target	One "instance" of an output type. When y...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Text snippet	A snippet that is created out of a portion ...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Topic	A chunk of information about a particular...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	Variable	A pre-set term or content that you can us...				<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	XML Editor	The window in the Flare interface where y...				<input type="checkbox"/>	<input type="checkbox"/>

EXAMPLE

Let's say you have the word "Target" in your glossary, with the "T" capitalized. Also, in your target you've opened the Glossary tab and told Flare to automatically convert all occurrences of matching terms to glossary links in your topics.

You might have topics that contain variations of the word, something like the following (with much more topic text than this, of course):




Suppose you generate your target without selecting either the "Ignore case..." or "Stem..." options.

- ☐ Stem words to include term variants in automatic link creation
- ☐ Ignore case in automatic link creation

In that case, the topic will look like this in the output, with only one term converted to a glossary term link—the word that matches the one in the glossary perfectly.

Glossary Term Example

target

Target 

Targets

Targeted

targeting

Now suppose you select the option to ignore the case of words.

<input type="checkbox"/>	Stem words to include term variants in automatic link creation
<input checked="" type="checkbox"/>	Ignore case in automatic link creation

In that case, the topic will look like this in the output, with two words converted to glossary term links.

Glossary Term Example

[target](#) ▾

[Target](#) ▾

Targets

Targeted

targeting

And finally, suppose you select both options.

- ☒ Stem words to include term variants in automatic link creation
- ☒ Ignore case in automatic link creation

In that case, the topic will look like this in the output, with all of the words converted to glossary term links.

Glossary Term Example

[target](#) ▾

[Target](#) ▾

[Targets](#) ▾

[Targeted](#) ▾

[targeting](#) ▾



Note: The stem features supports the following languages:

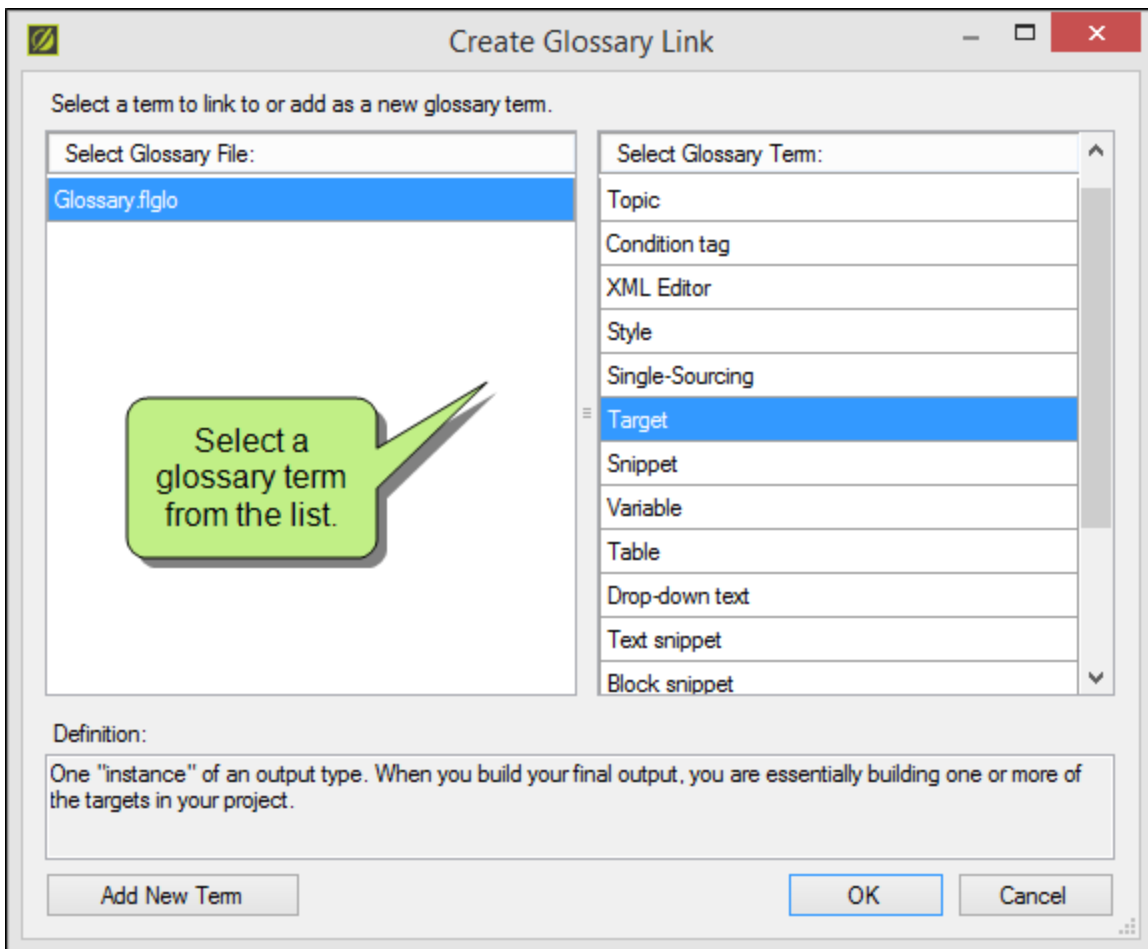
- » Czech
- » Danish
- » Dutch
- » English
- » Finnish
- » French
- » German
- » Hungarian
- » Italian
- » Norwegian
- » Portuguese
- » Romanian
- » Russian
- » Spanish

CONDITIONS

You can apply conditions to terms in the Glossary Editor. For more information, see "Conditions on Glossary Terms" on page 204.

SELECT TERM FOR GLOSSARY LINKS

When you highlight text in a topic and choose to create a glossary term link (e.g., by opening the **Insert** ribbon and clicking **Glossary Term Link**), you can now choose an existing term from the glossary.

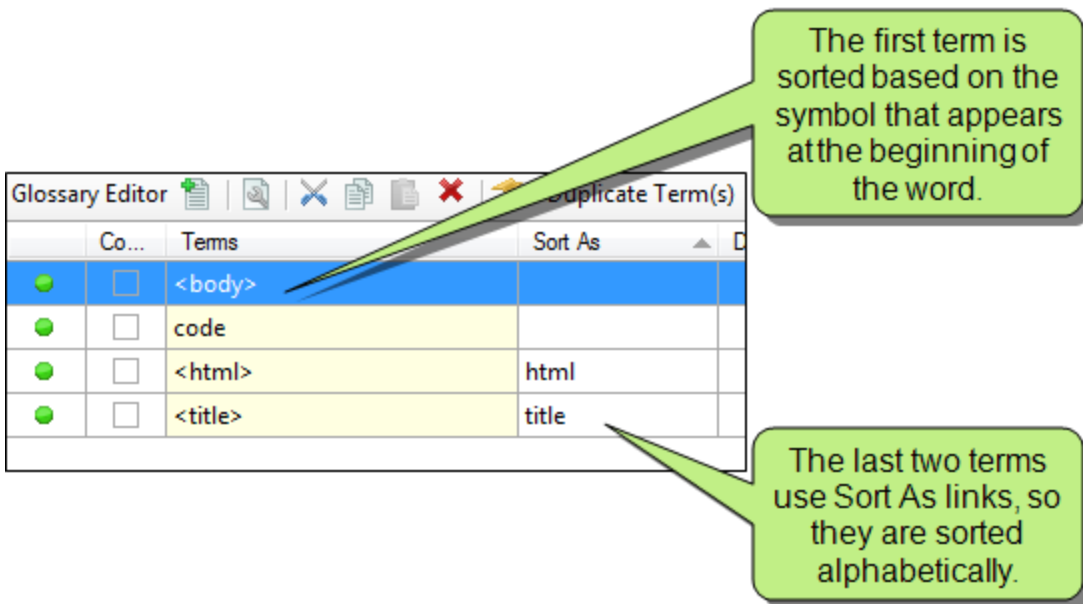


SORTING GLOSSARIES

You can now use "Sort As" links for glossaries.

"Sort As" links can be useful for several reasons. You can use them if a keyword begins with a symbol (e.g., place `<html>` in the "h" section, instead of at the top) or if you want to sort acronyms based on their full spelling. You can also use "Sort As" links when working with multilingual documents, such as when you are using character sets that should be sorted phonetically.

When your glossary is displayed in the output (e.g., the Glossary window pane for Tripane output), by default terms are shown alphabetically. However, you can use the "Sort As" field in the Properties dialog to force a particular term to be displayed where you want it.



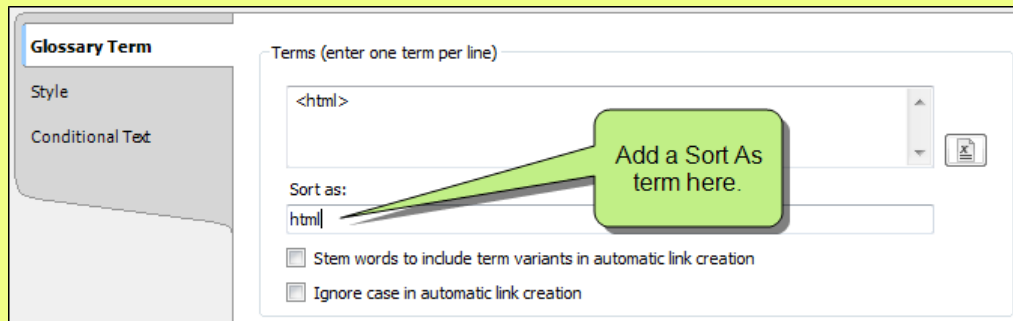
Co...	Terms	Sort As
<input checked="" type="checkbox"/>	<code><body></code>	
<input checked="" type="checkbox"/>	<code>code</code>	
<input checked="" type="checkbox"/>	<code><html></code>	html
<input checked="" type="checkbox"/>	<code><title></code>	title

This is especially useful if you are using "Sort As" links in your index, and you want words that appear in both places to be sorted in the same way.

To set a "Sort As" term for a glossary term, create a new a glossary term or double-click an existing term. A new **Sort As** field has been added to the **Glossary Term** tab in the Properties dialog. Enter the word or phrase you want to use to sort the glossary term. You can see the "Sort As" terms in the Sort As column in the Glossary Editor. When you publish your output, the glossary will be sorted using the "Sort As" terms.

EXAMPLE

Let's say you add the term "<html>" to your glossary. Because this word starts with a symbol, it would normally be sorted using the symbol. However, you want to sort it using the H. After you enter the term in your glossary, you add "html" as a "Sort As" term.



The screenshot shows a web form titled "Glossary Term" with a sidebar containing "Style" and "Conditional Text" links. The main form area has a section titled "Terms (enter one term per line)" with a text input field containing "<html>". Below this is a "Sort as:" label followed by a text input field containing "html". A green callout bubble with a pointer to the "Sort as:" field contains the text "Add a Sort As term here.". At the bottom of the form are two checkboxes: "Stem words to include term variants in automatic link creation" and "Ignore case in automatic link creation", both of which are currently unchecked.

When you publish your output, the term is sorted correctly.

Glossary

<

<body>
The <body> tag defines the document's body. The <body> element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, et

<title>
The <title> tag is required in all HTML documents and it defines the title of the document. The <title> element: defines a title in the browser toolbar, provides a title for the page when it is added to favorites, and displays a title for the page in search-engine results.

C

code
program instructions

H

<html>
The <html> tag tells the browser that this is an HTML document. The <html> tag represents the root of an HTML document. The <html> tag is the container for all other HTML elements (except for the <!DOCTYPE> tag).

Right now these definitions are sorted using the <. You could add Sort As terms to their glossary terms so they would sort alphabetically.

Because it uses a Sort As term, "<html>" is sorted using the H.

VARIABLES IN TERMS AND DEFINITIONS

You can use variables in your glossary terms and definitions. Variable buttons are available in the Properties dialog for glossary terms. For more information, see "Insert Variable Option Added Throughout Interface" on page 295.

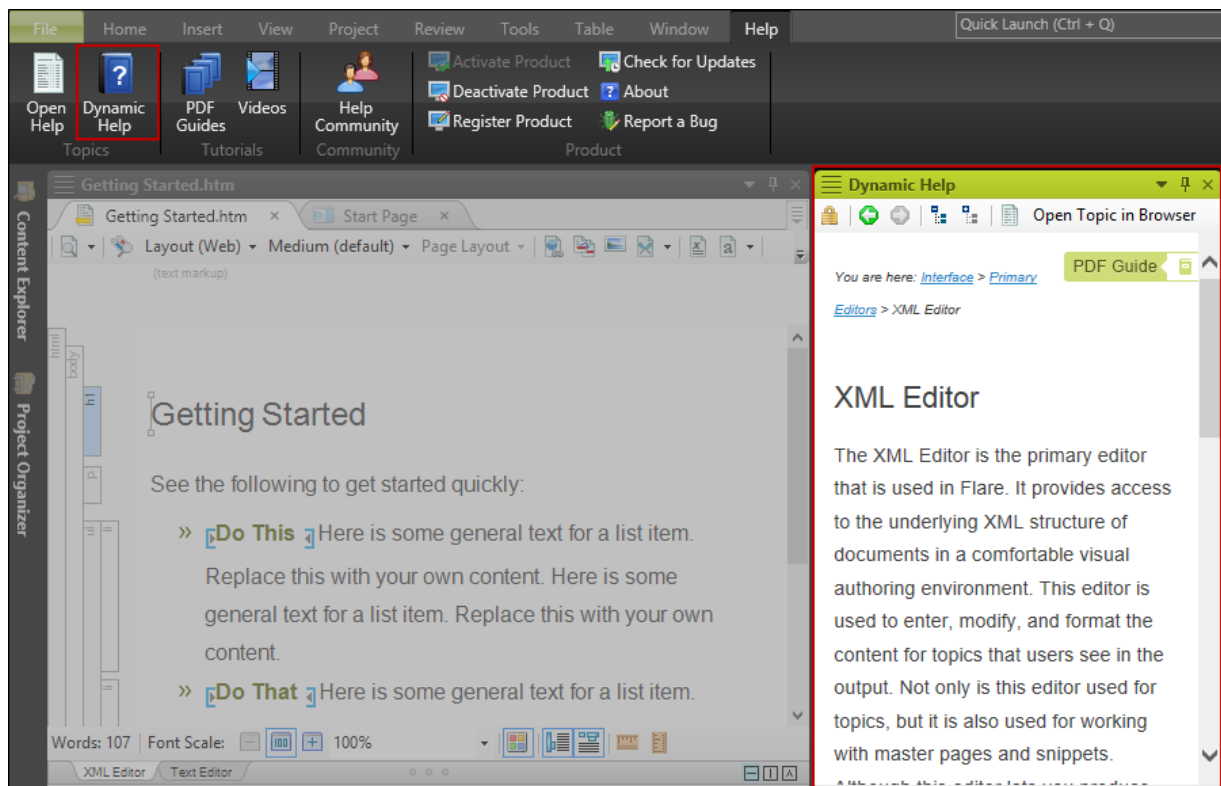
Help—Live or Local

Flare's online and dynamic Help system has changed, pointing to files hosted externally. This allows for live updates to the Help. Also, there is a new local Help option that lets you view Help located on your computer instead.

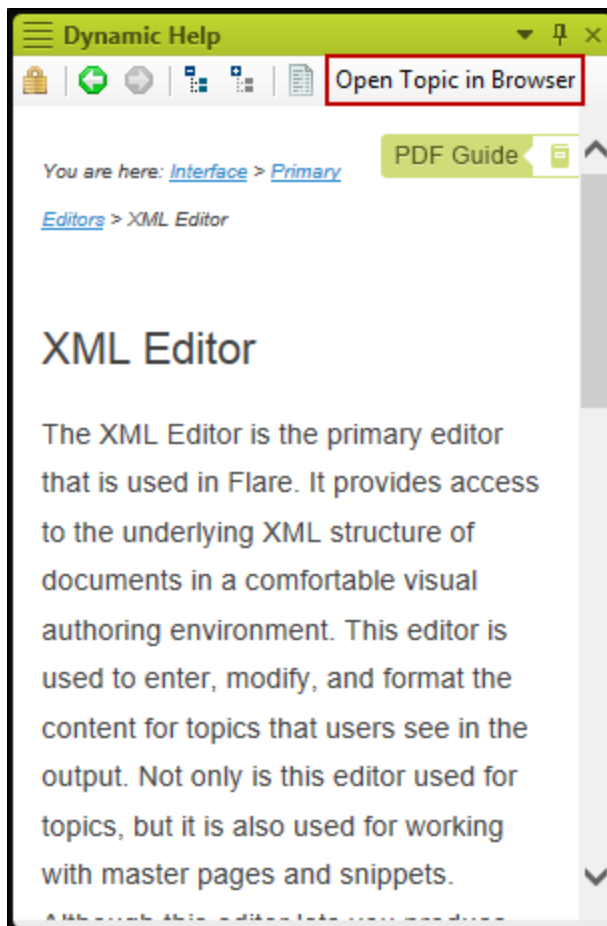
LIVE EXTERNAL HELP SYSTEM

By default, Flare's dynamic Help is now hosted online, rather than inside the application. This allows topics to be updated on an as-needed basis with the most current information. In previous versions of Flare, the dynamic Help was only updated when there was a new product release.

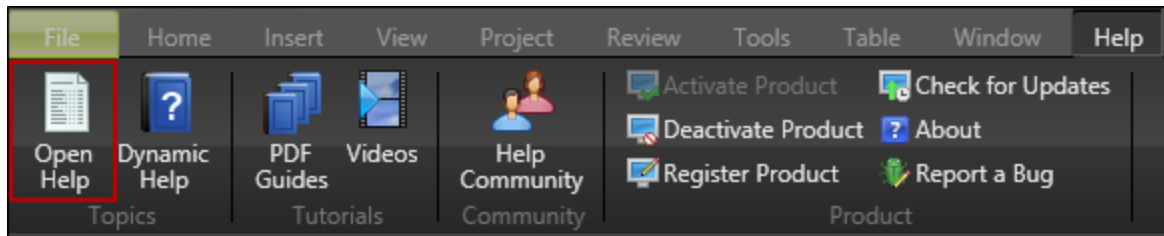
To open the dynamic Help for the active element in the interface, select **Help>Dynamic Help**.



You can also open the full online Help in your browser. To do this, you can click the **Open Topic in Browser** button at the top of the dynamic Help window pane.



You can also select **Help>Open Help**.

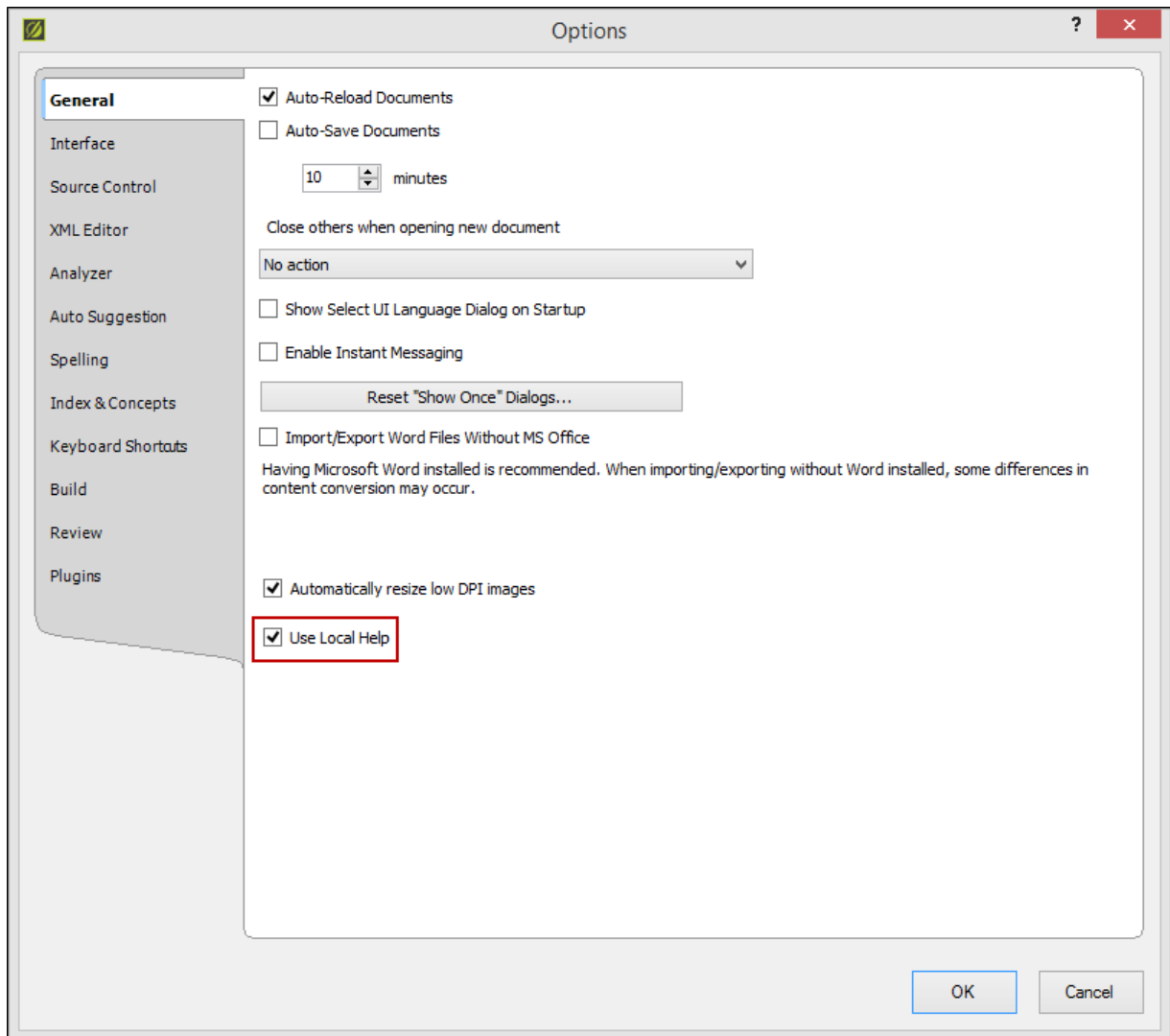


LOCAL HELP SYSTEM

Because the Help system is hosted online, you may not be able to access it if you are behind a firewall or if you have a slow connection. In that case, you can use a local version of the Help. Flare will even switch to the local Help automatically if it detects that you cannot connect to the external Help system.

If you want to use the local Help manually, open the Options dialog (**File>Options**), and on the **General** tab select **Use Local Help**.

Select this option if you want to use the local Help manually.



Note: The local Help system is only updated when the application is updated. If you want to be sure that you have the most current version of the Help, disable the local Help.

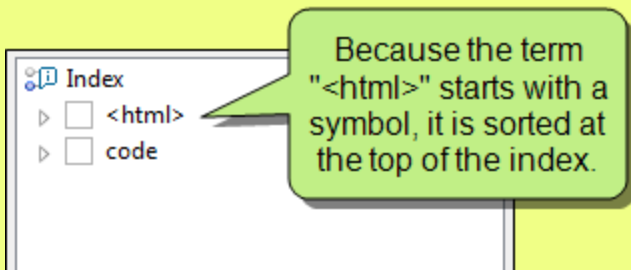
Indexes—Sorting

"Sort As" links can be useful for several reasons. You can use them if a keyword begins with a symbol (e.g., place `<html>` in the "h" section, instead of at the top) or if you want to sort acronyms based on their full spelling. You can also use "Sort As" links when working with multilingual documents, such as when you are using character sets that should be sorted phonetically.

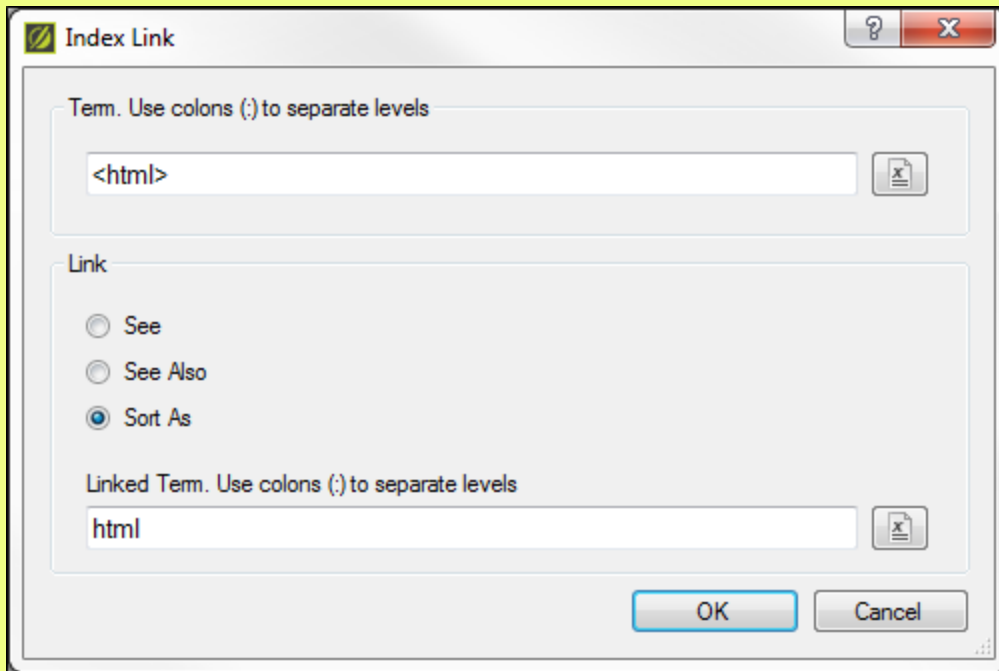
In previous versions of Flare, you could create "Sort As" links for first-level terms. This was helpful, but it could lead to some confusion if that term appeared as a first- and second-level term in your index. For consistency, index sorting has been modified in Flare 12 so it now applies to both first-level terms and second-level terms (e.g., if you sort the first-level term "`<html>`" as "html," the index will also sort the second-level term "code:<html>" as "code:html").

EXAMPLE

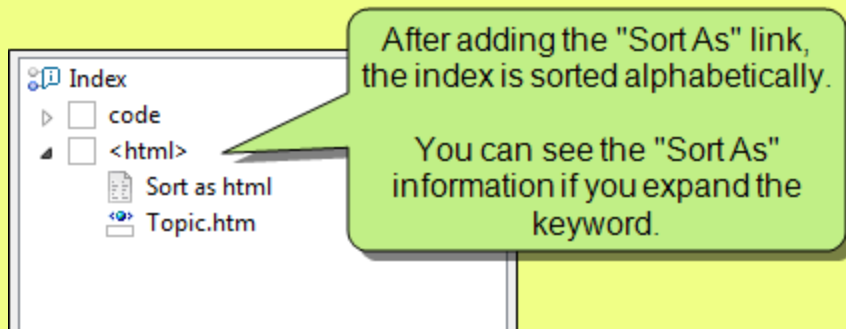
Let's say that you have add the term "`<html>`" to your index. Right now, your index looks like this:



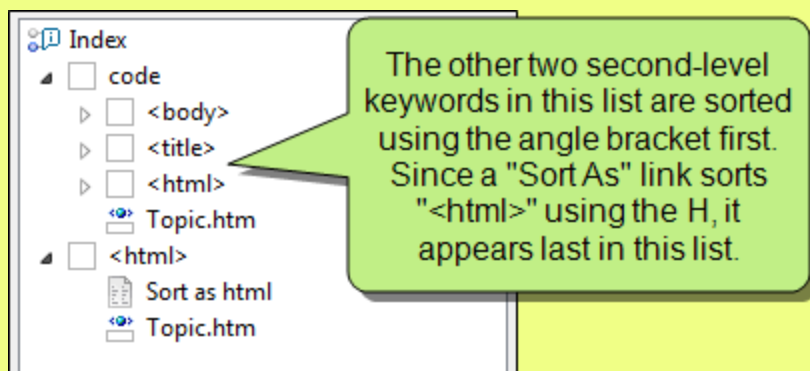
You don't want all of the words beginning with symbols to appear at the top of the index, so you create a "Sort As" link for this term so the index will sort it as "html," using the H.



Now the first-level keyword appears like this in your index:



You also use the term "<html>" as a second-level index keyword, under the keyword "code." Because you created a "Sort As" link, the second-level term is also sorted using the H, it appears like this:



Link to Styles

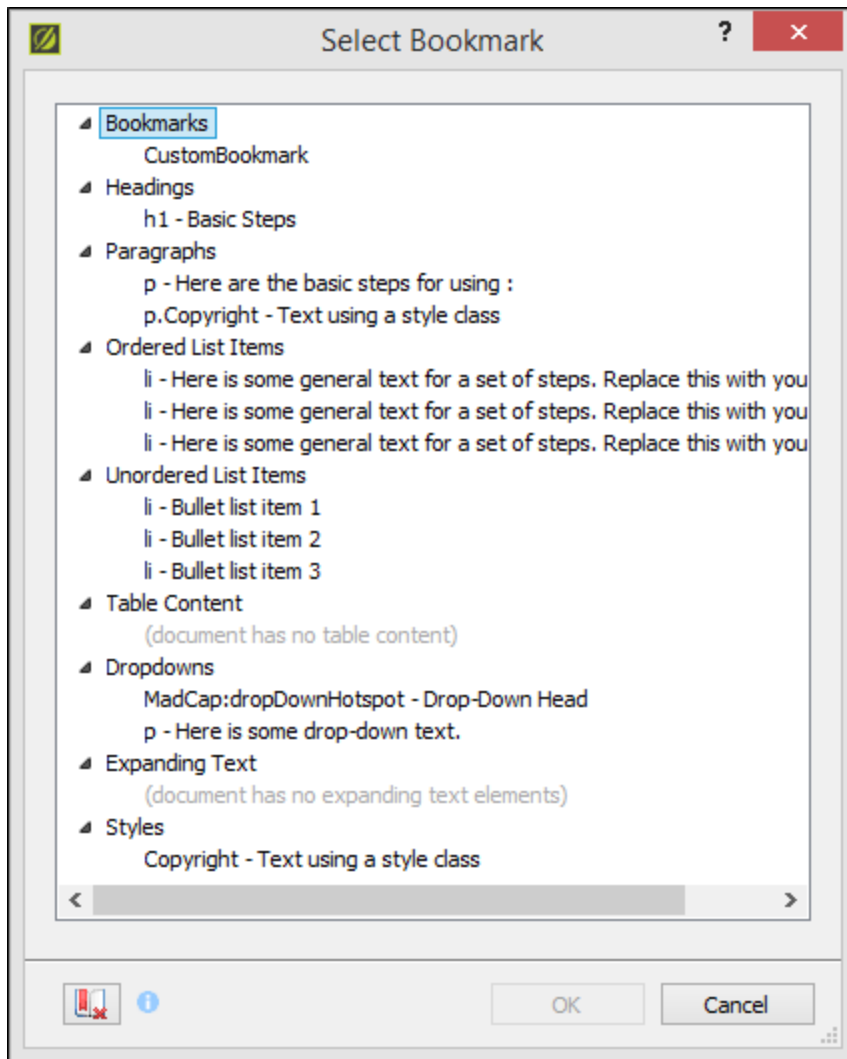
When creating a link, you can point to content using a particular style. This can be done in the Select Bookmark dialog when you are creating any of the following types of links:

- » Cross-references
- » Text hyperlinks
- » Tables of contents
- » Browse sequences

In previous versions, the Select Bookmark dialog displayed only bookmarks that you created manually, as well as h1 through h6 heading styles. But now many more styles are listed (including classes and IDs).

After the Bookmarks and Headings, they are organized as follows: Paragraphs, Ordered List Items, Unordered List Items, Table Cells, Dropdowns, Expanding Text, and Styles (classes, spans, etc.).

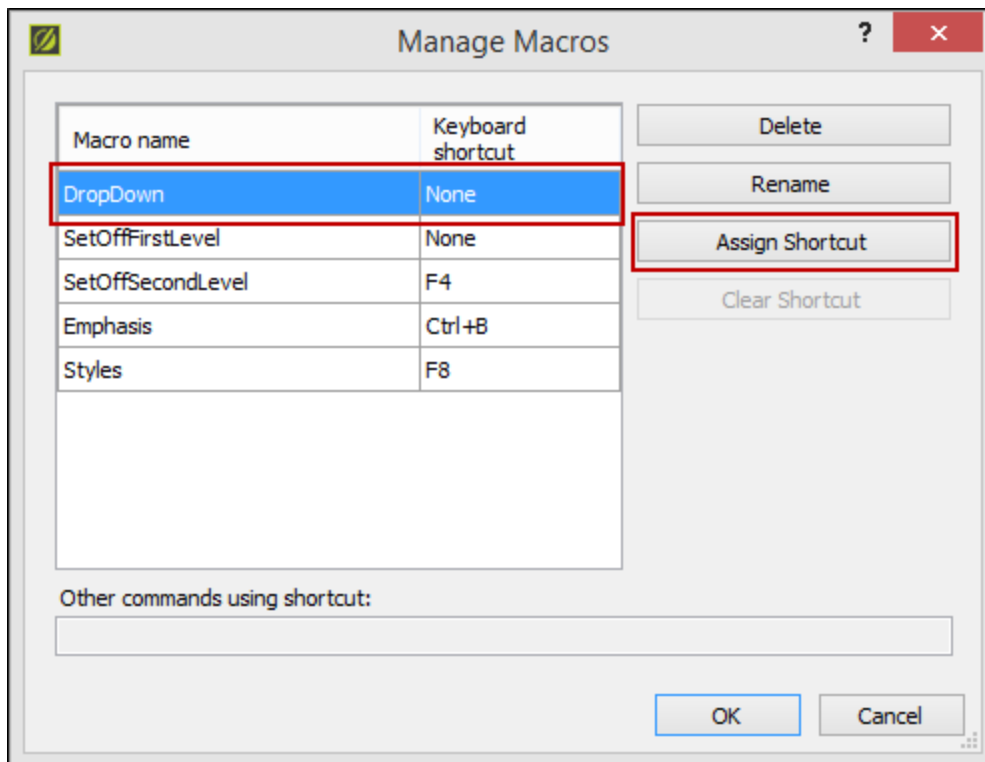
If necessary, a bookmark will be created at that location if one does not exist. If you want to clear a bookmark in the link, select it and click the button at the bottom of the dialog.



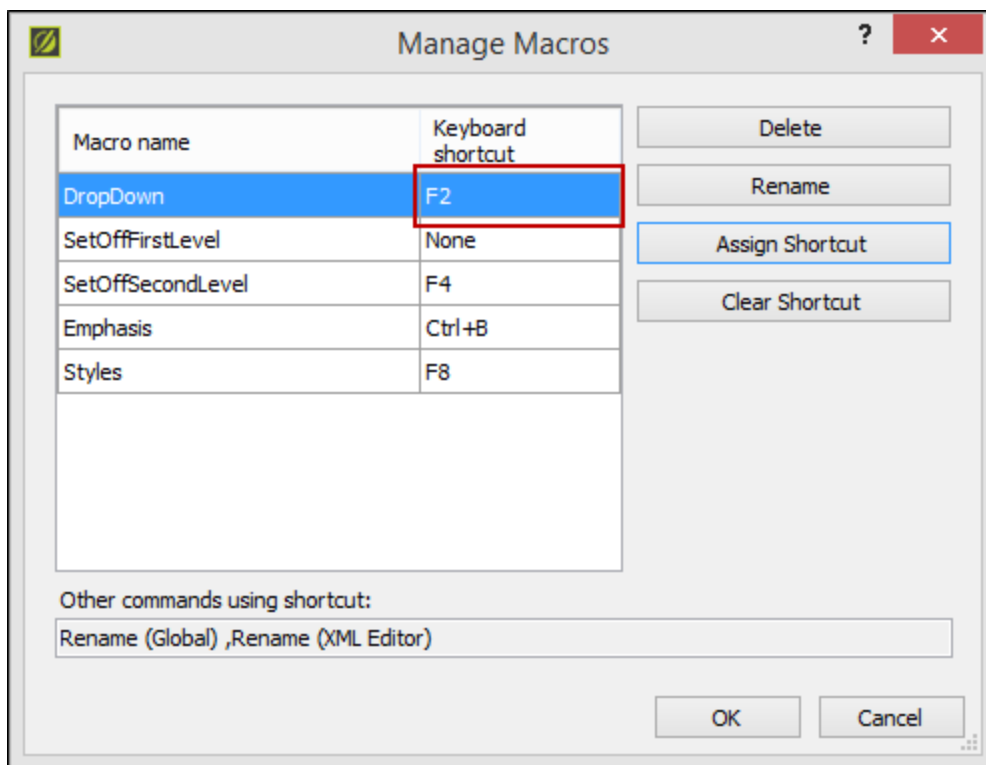
Macros—Assign Shortcuts in Dialog

You can now assign keyboard shortcuts to macros in the Manage Macros dialog. This is an alternative to using the Options dialog.

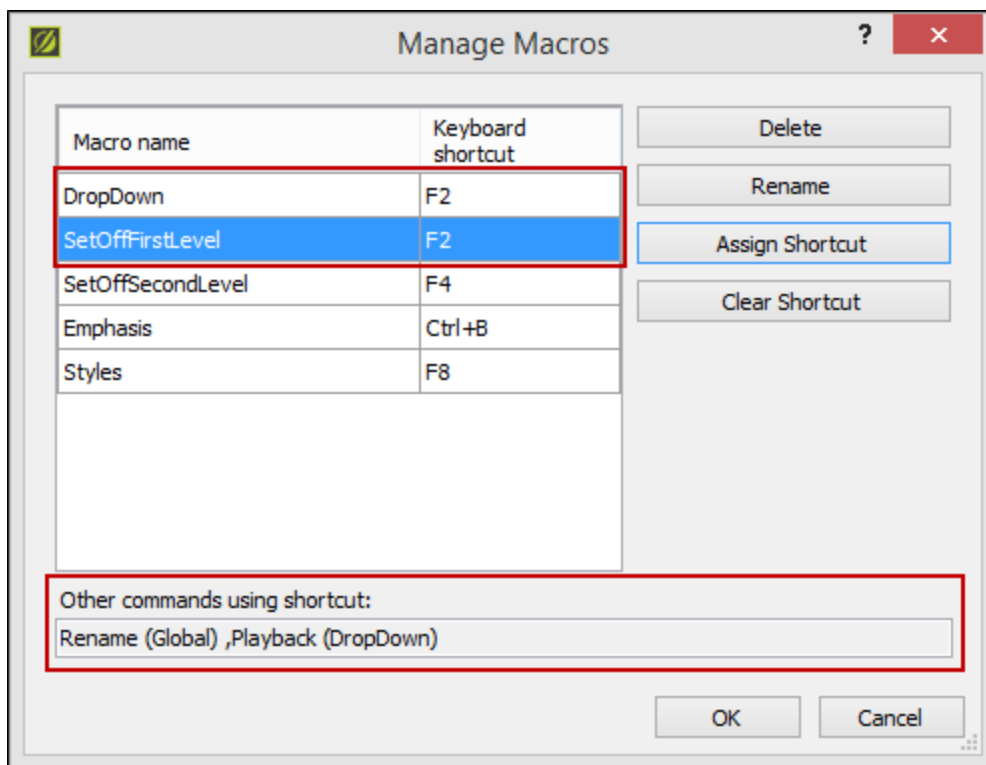
To do this, select the macro in the list, then click **Assign Shortcut**.



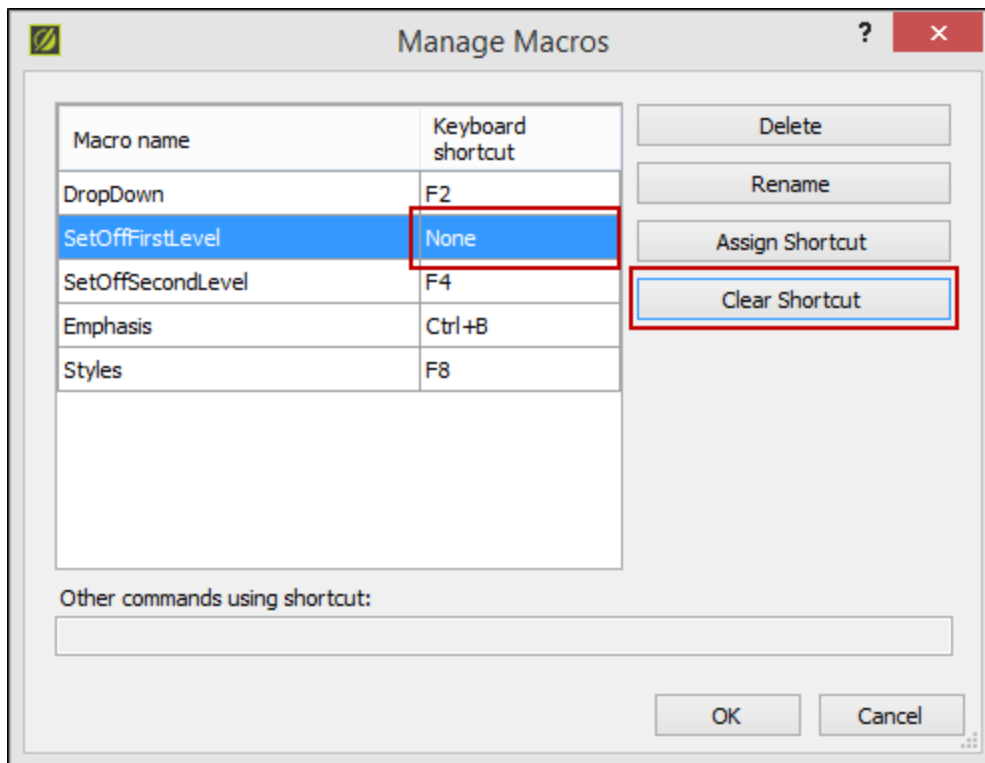
Click **OK** in the message that displays. Then use your keyboard to press the shortcut you want to use. The shortcut is added to the grid on the left.

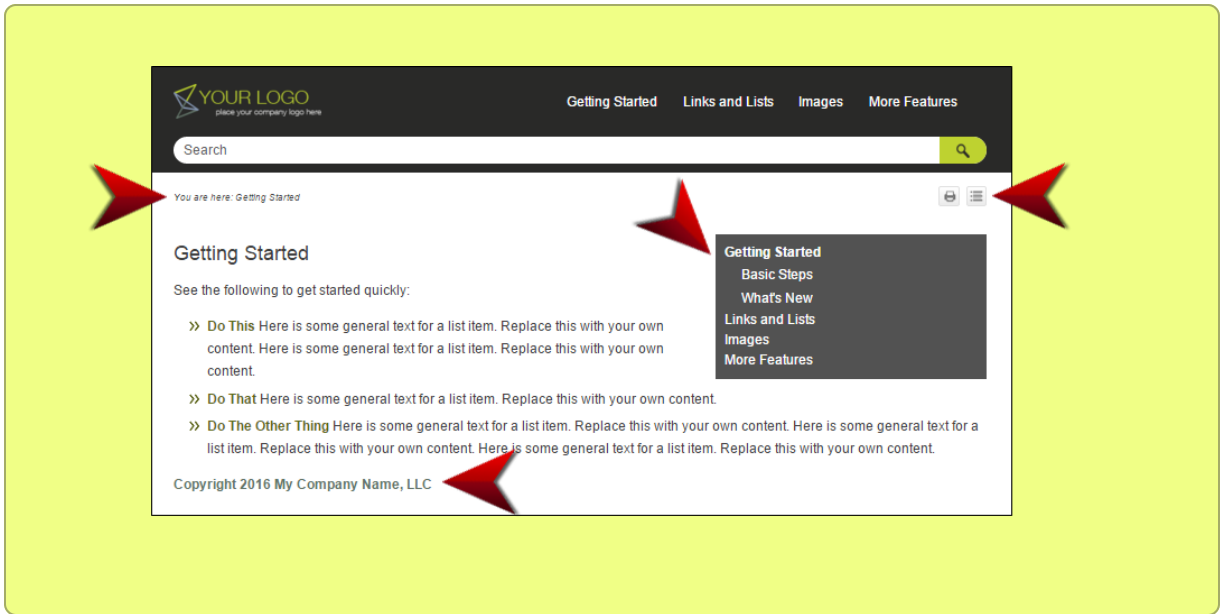


If another command is already using the shortcut, Flare tells you at the bottom of the dialog.

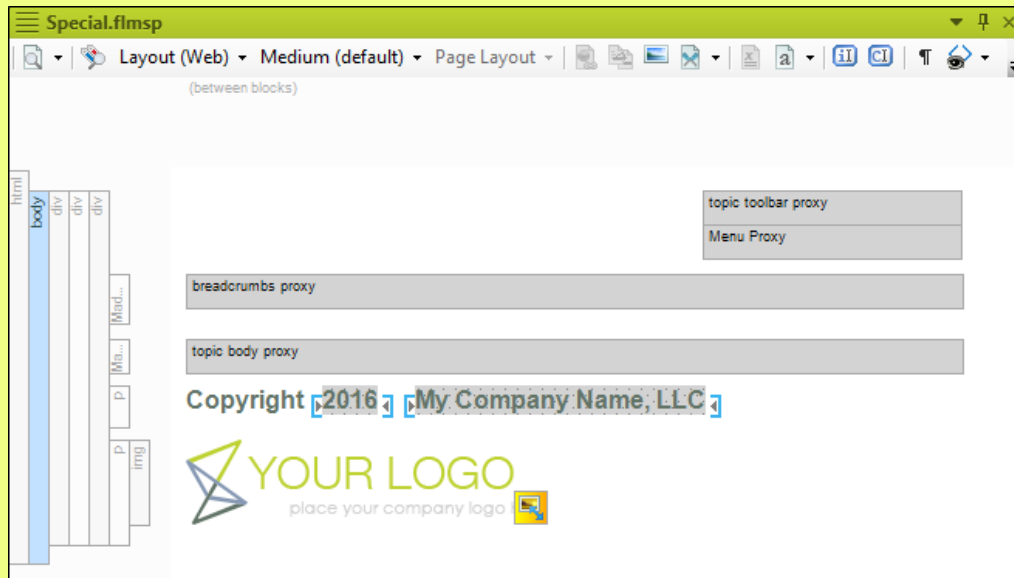


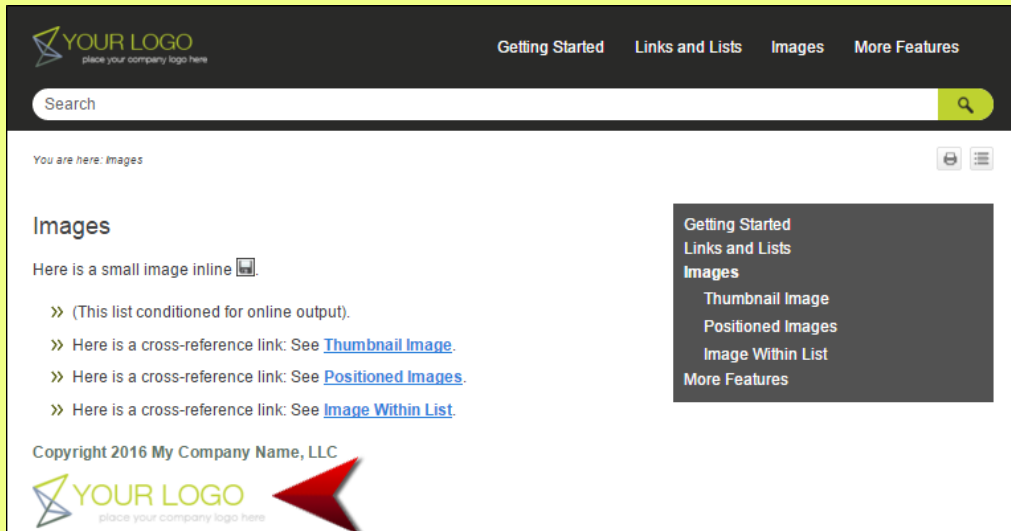
To remove a keyboard shortcut, but keep the macro, select **Clear Shortcut**.



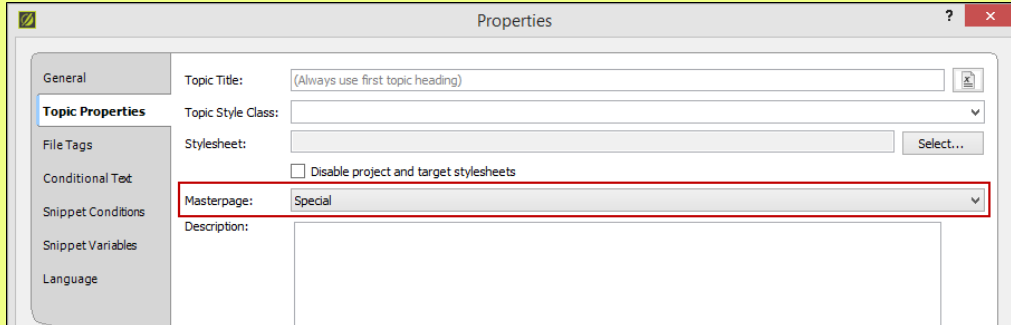


But you also have a secondary master page called "Special." This master page includes the same things as the Primary master page, but you also have your company logo under the copyright text. You want 8 specific topics to use the Special master page.






To accomplish this, you associate the Primary master page with the target. But for the other topics, you can open the Properties dialog and choose the Special master page.

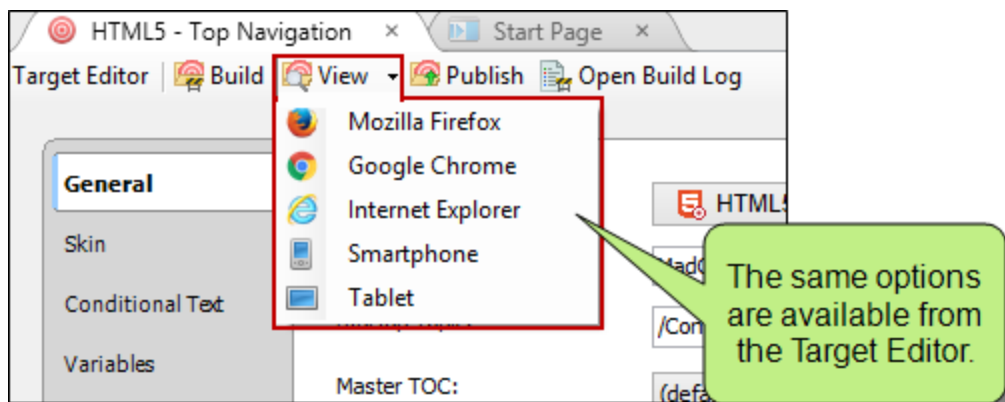
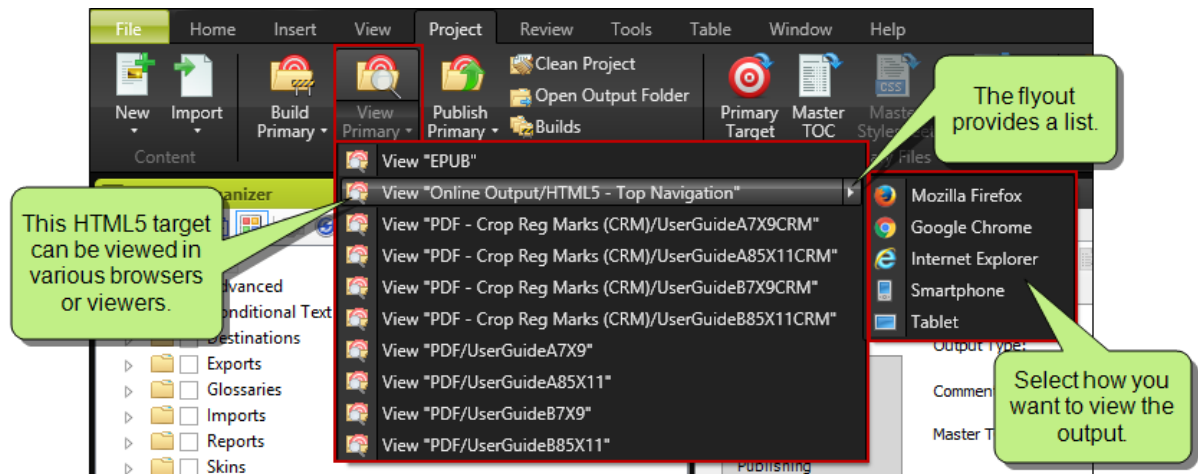


HOW TO ASSOCIATE A MASTER PAGE WITH TOPICS

1. Do one of the following, depending on whether you want to associate a master page with a single topic or multiple topics:
 - » **Single Topic** If you want to associate a master page with a single topic, right-click the topic file in the Content Explorer and choose **Properties**. The Properties dialog opens.
 - » **Multiple Topics** If you want to associate a master page with several topics, do the following:
 - a. Select the **View** ribbon. In the **Explorer** section select **File List**. The File List window pane opens.
 - b. (Optional) From the **Filter** drop-down list in the local toolbar, you can select **Topic Files** to limit the results in the grid to topics only.
 - c. Select the files to which you want to apply a stylesheet. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items.
 - d. In the local toolbar, click . The Properties dialog opens.
2. Click the **Topic Properties** tab.
3. Click the **Masterpage** field and select a master page.
4. Click **OK**.

Output—Choose Specific Browser or Viewer

You can now choose a specific browser or viewer when opening the output for a target. This feature has been available from the Builds window pane, but now you can do the same from the Project ribbon or from the Target Editor. In the ribbon, an extra flyout is provided in the drop-down if a particular target has generated output and your computer is set up to view it in different browsers or viewers.




Page Break—Independent of HTML Element

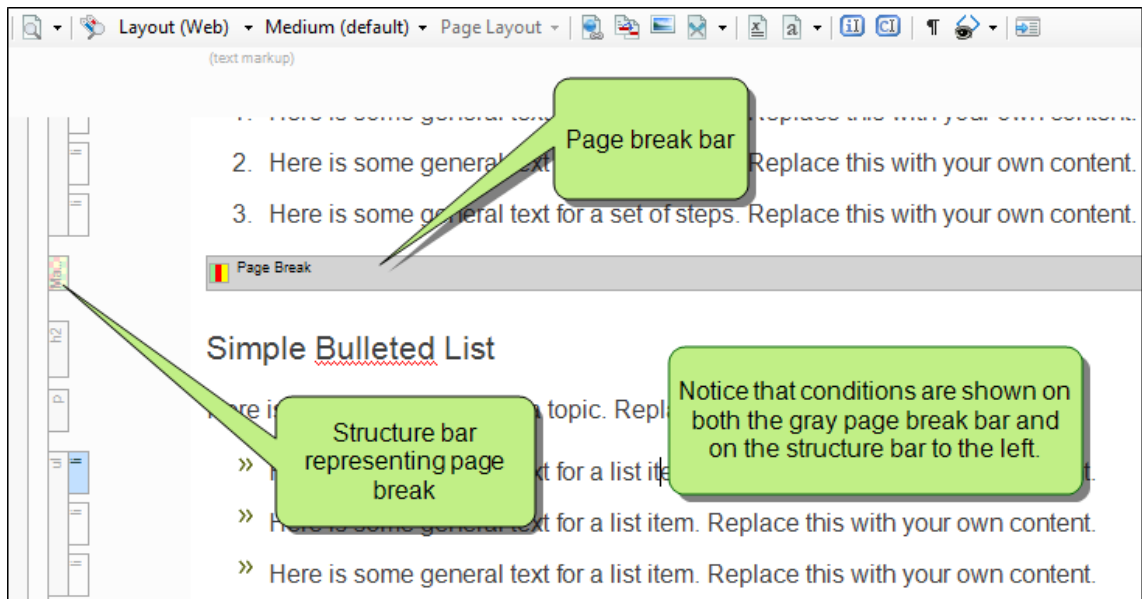
You have always been able to place a page break on a style or locally on a paragraph in Flare, for use in print-based outputs. Starting with this version, you also have the ability to manually insert a page break that's not tied to an HTML element.

Placing a page break on a style may be the preferred method in many cases, but there may be times when you find that it makes more sense to use an independent page break directly in the topic. This might be the case, for example, if you need a page break to occur after a paragraph and just before a heading, but only in some outputs.

HOW TO INSERT AN INDEPENDENT PAGE BREAK

1. Open the content file (e.g., topic, snippet).
2. Place your cursor on an empty line where you want to insert the page break. When you generate print-based output, the break occurs *after* this insertion point.
3. Select **Insert>Page Break**. If your application is resized so that you cannot see the text, look for the **Page Break** icon .

A gray bar is displayed in the XML Editor, and it is also represented by a structure bar on the left side of the XML Editor. You can apply conditions to the break, in case it should be used in some outputs but not others.

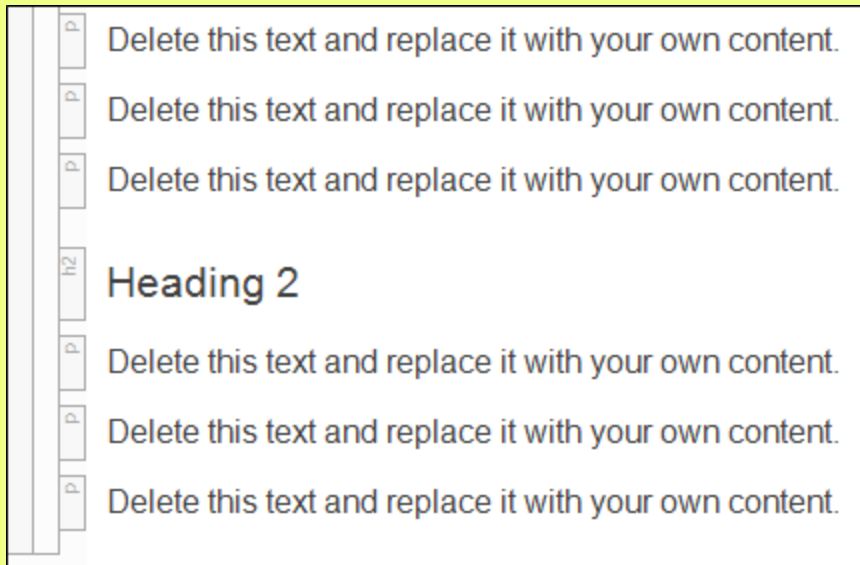


In the code, Flare uses the MadCap:pageBreak element where a page break is inserted.

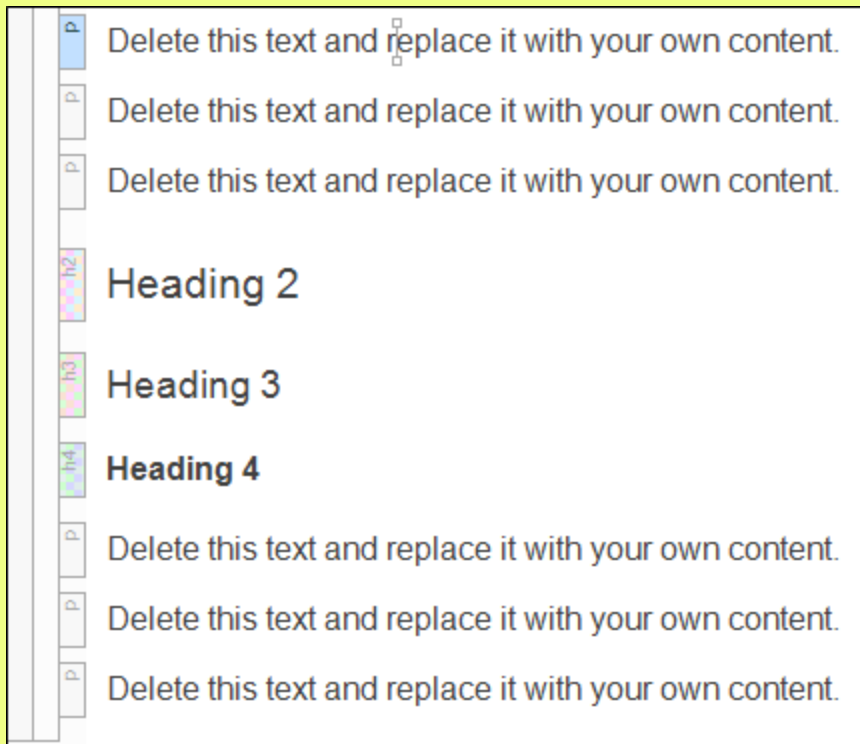
```
5. Replace this with your own content.</p>
  <MadCap:snippetBlock src="../../Resources/Snippets/MiniTOC.flshp" MadCap:condit
  <h2>
    <MadCap:keyword term="Lists:numbered;Numbered lists" />Simple Numbered L
  <p>Here is some general text for a topic. Replace this with your own content
  <ol>
    <li>Here is some general text for a set of steps. Replace this with your
    <li>Here is some general text for a set of steps. Replace this with your
    <li>Here is some general text for a set of steps. Replace this with your
  </ol>
  <MadCap:pageBreak />
  <h2>Simple Bulleted List</h2>
  <p>Here is some general text for a topic. Replace this with your own content
  <ul>
    <li>Here is some general text for a list item. Replace this with your ow
    <li>Here is some general text for a list item. Replace this with your ow
    <li>Here is some general text for a list item. Replace this with your ow
  </ul>
  <h2>Lists With Paragraphs Between Items</h2>
  <p>Here is some general text for a topic. Replace this with your own content
```

EXAMPLE

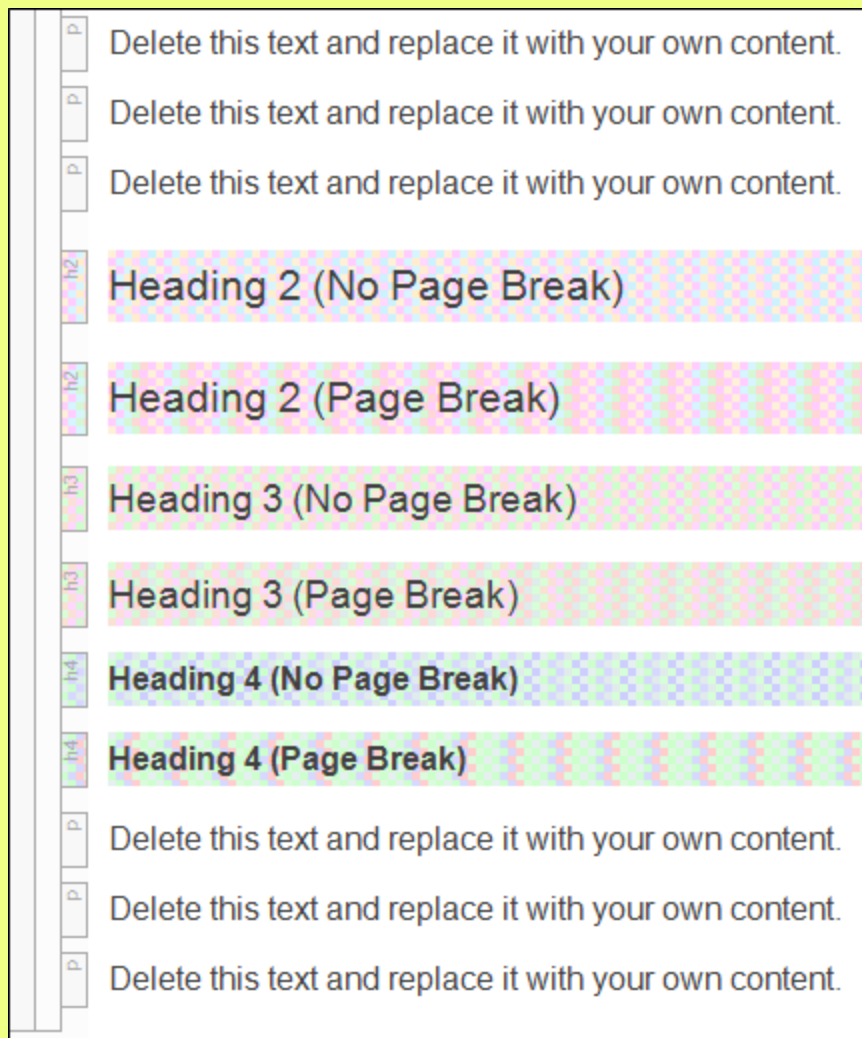
Let's say you have a heading in the middle of a topic that looks like this:



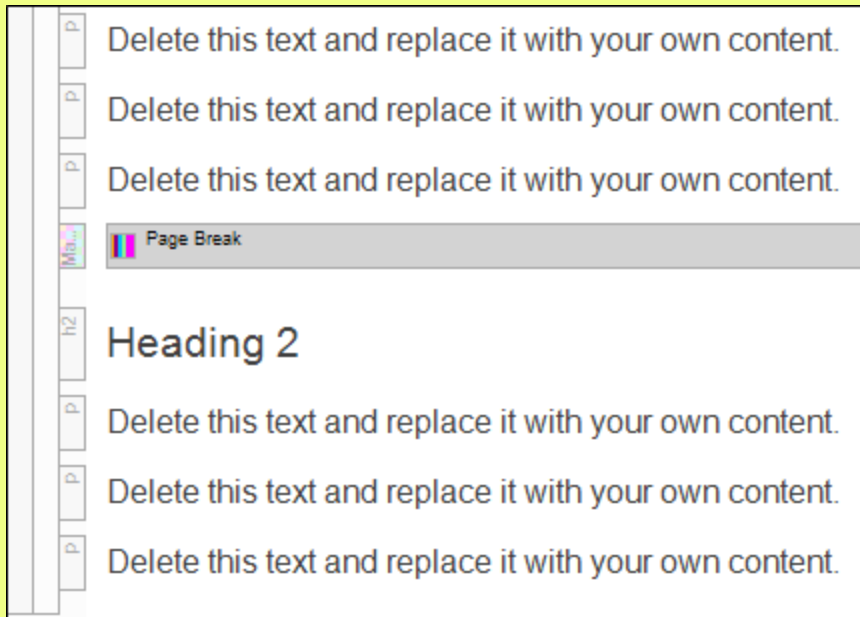
All by itself, this heading is not too difficult to deal with. However, suppose you are producing several outputs that include this content, and in some PDFs this heading should be an H2, in others an H3, and yet in others an H4.



Then, to make things even more complicated, let's say that you want there to be a page break before some of these headings (at each level) and no page break before others. If you relied solely on page breaks placed on styles, you might end up with two styles for each heading level and multiple conditions placed on each heading.



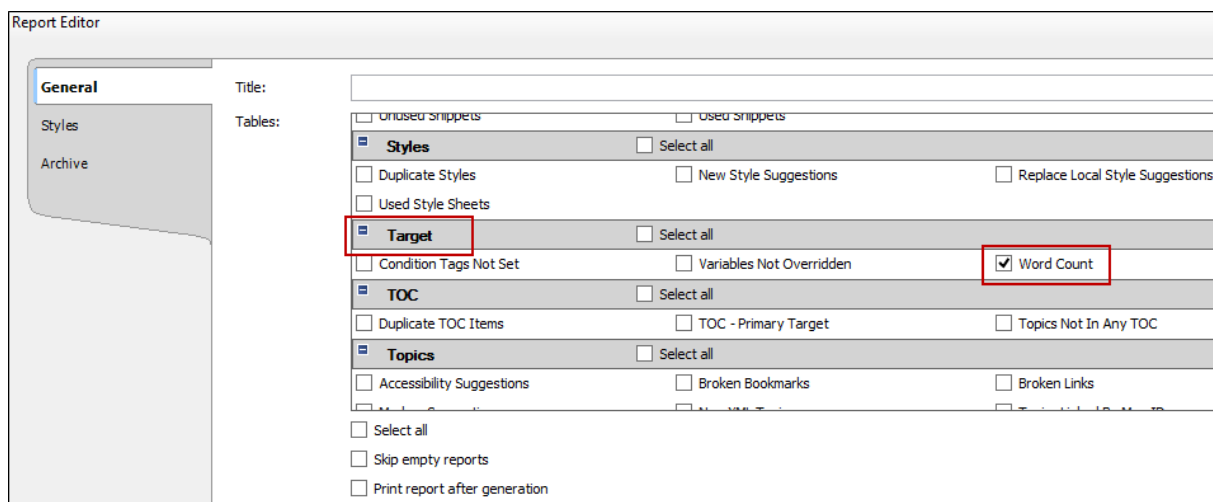
You can see how this might get very difficult to manage over time. So instead of relying on page breaks placed on styles, you might decide to simplify the content by inserting an independent page break and putting conditions on it that are appropriate for your various PDFs. Also, if you use Flare's TOC Depth setting for PDF targets, you only need one heading, and when the output is generated, Flare will automatically convert it to H3 or H4 as necessary.



Note: In online outputs, page breaks are ignored.

Report—Word Count

A new report has been added to Flare that lets you see how many words each target produces. This report is found in the Target section in the Report Editor.



Project: C:\Users\pstoecklein\Documents\My Projects\test\test.flprj
Generated: 1/12/2016 11:43:24 AM
By: pstoecklein

Word Count

Target: HTML5 - Top Navigation.fltar
Word Count: 4155

Target: UserGuideA7X9.fltar
Word Count: 5454

Target: UserGuideA8_5X11.fltar
Word Count: 5454

Target: UserGuideB7X9.fltar
Word Count: 4161

Target: UserGuideB8_5X11.fltar
Word Count: 4161

Target: UserGuideA7X9CRM.fltar
Word Count: 5454

Target: UserGuideA8_5X11CRM.fltar
Word Count: 5454

Target: UserGuideB7X9CRM.fltar
Word Count: 4161


Target: UserGuideB8_5X11CRM.fltar
Word Count: 4161

Skins—Runtime

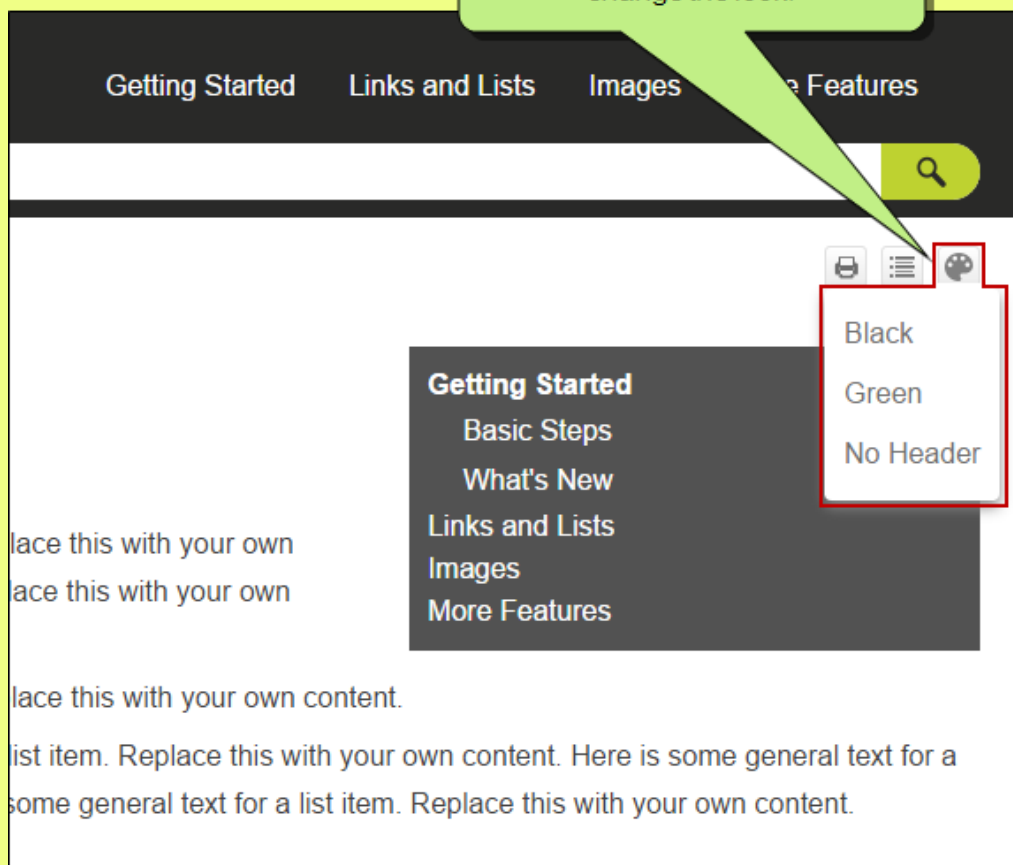
For HTML5 targets, you can let users choose from multiple skins to change the display of the output.

E X A M P L E

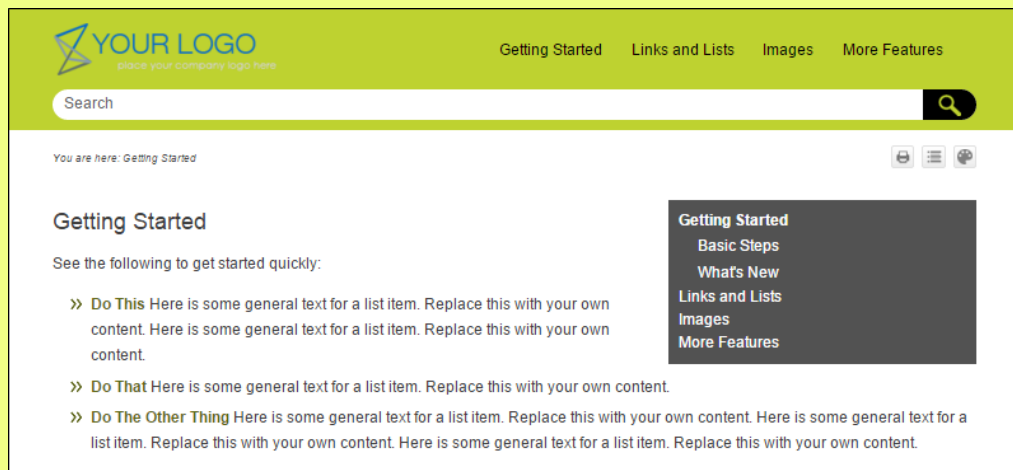
Let's say your main Top Navigation skin has a black header background, but you want users to be able to select a skin with a green header background and a different logo, or even a third skin that does not have a header at all. Therefore, you create the additional skins and edit them to look the way you want. Then you add the Skins button in your Topic Toolbar skin component.

In the output, a user can click the button  in the topic toolbar and choose one of the options, which are based on the skin file name.

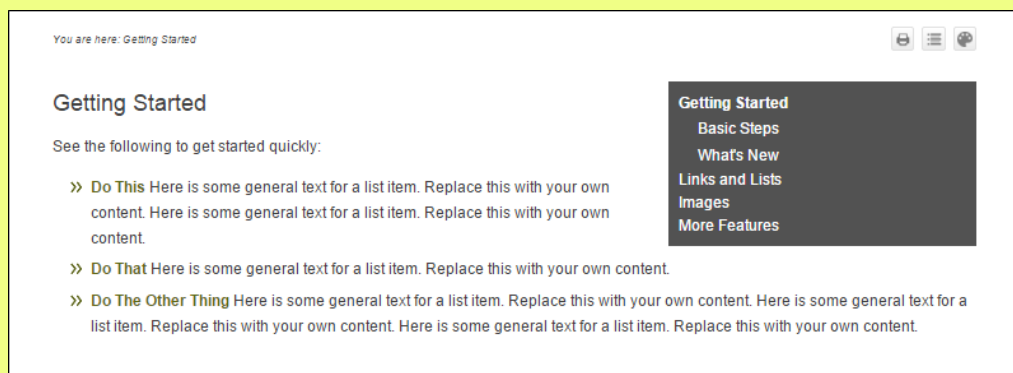
In this example, the Black skin is currently shown around the topic content, but you can click here and select one of the other skins to change the look.



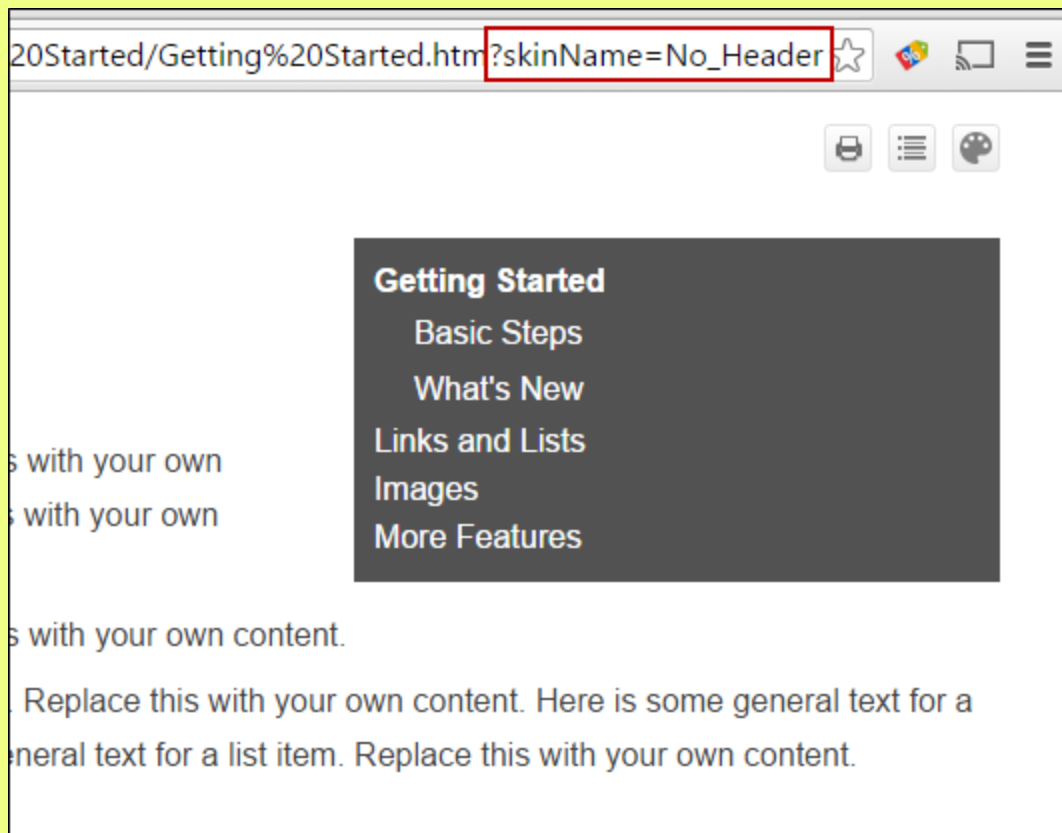
If the user selects **Green**, the output might look something like this:



And if the user selects **NoHeader**, the output might look something like this:



The name of the skin is reflected at the end of the URL in the browser.

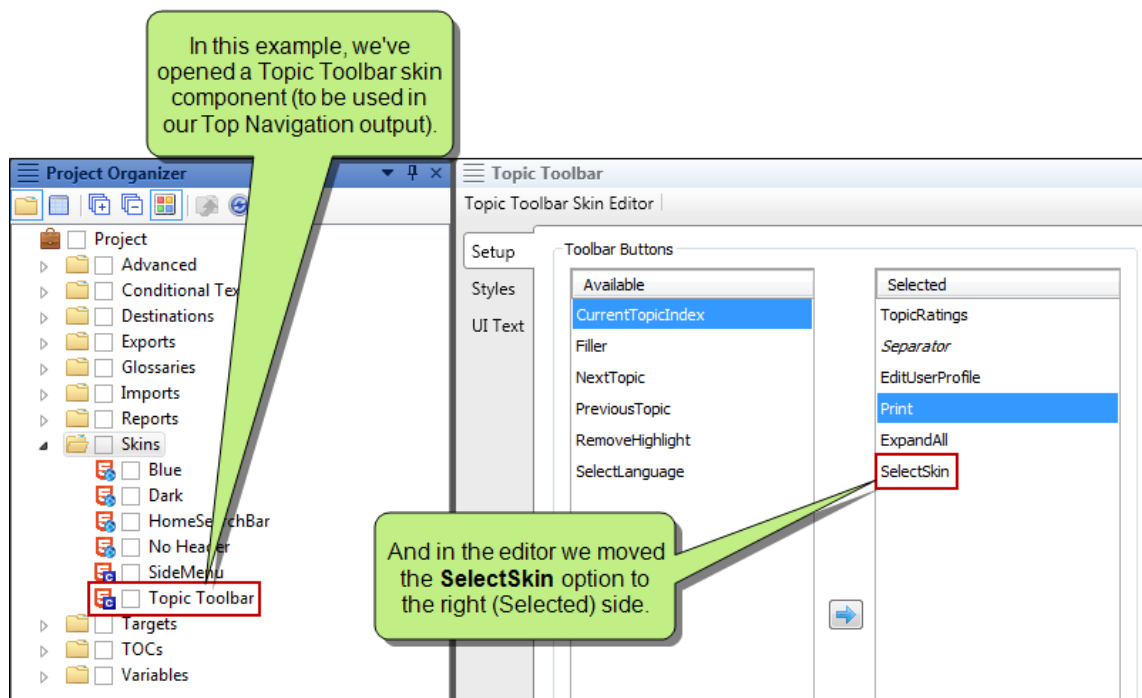


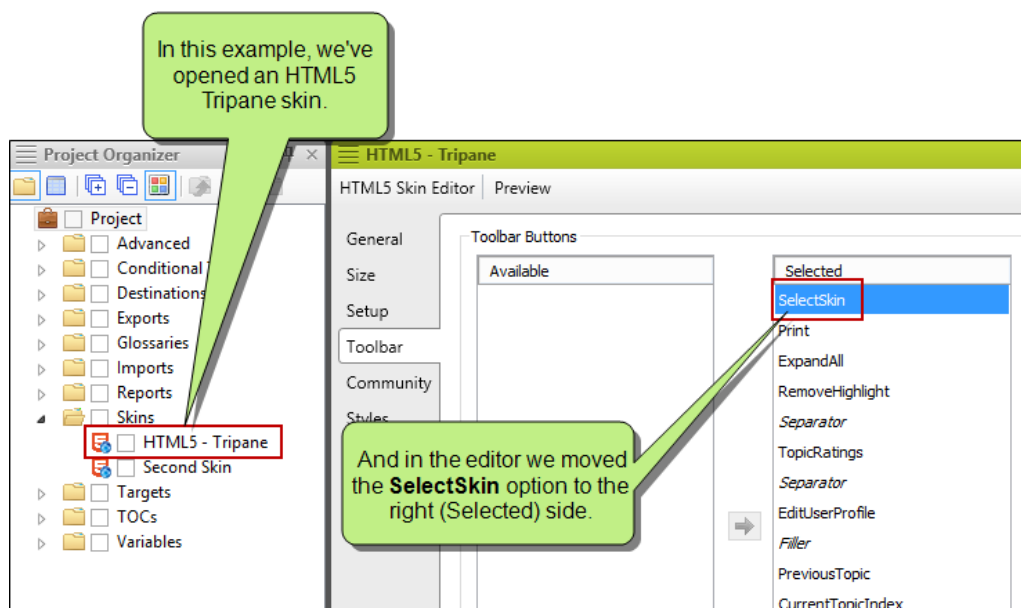
HOW ENABLE RUNTIME SKINS

1. If you are creating Top Navigation output, you should have a Topic Toolbar skin component in your project . Open the skin component, and select the **Setup** tab.

If instead you are creating responsive Tripane output, open the regular skin and select the **Toolbar** tab. This lets you place the skin button in a web toolbar.

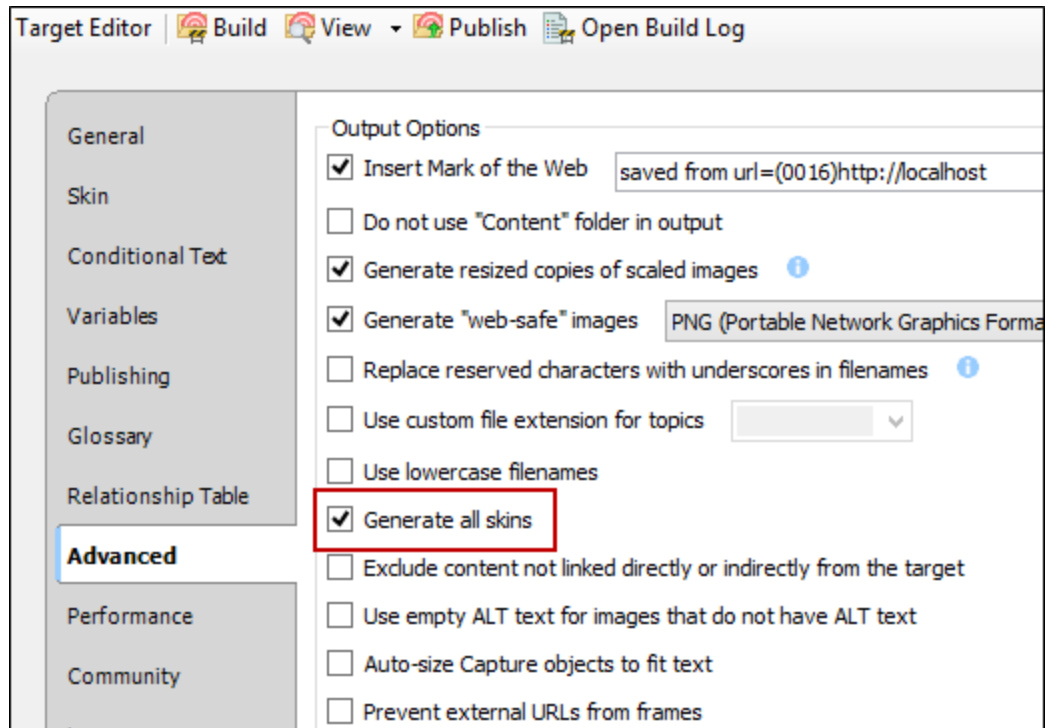
2. If necessary, move the **SelectSkin** button from the **Available** section to the **Selected** section.






3. Open the HTML5 target.
4. In the Target Editor, select the **Advanced** tab.

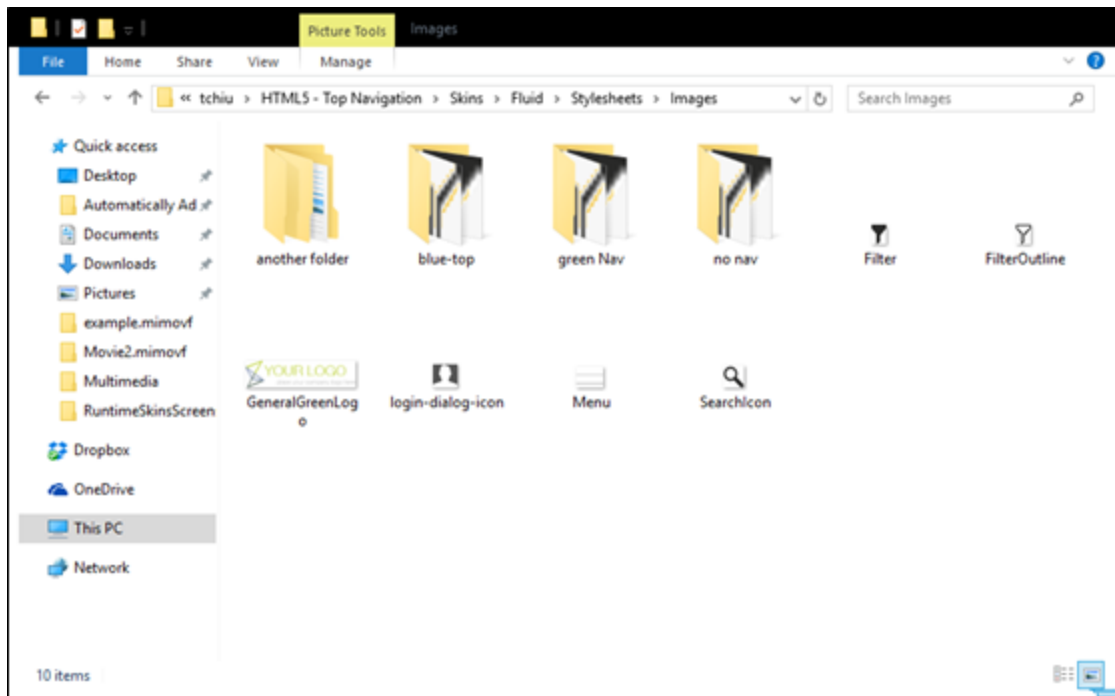
5. Select **Generate all skins**.



6. Click  to save all files.

7. Build the HTML5 target.

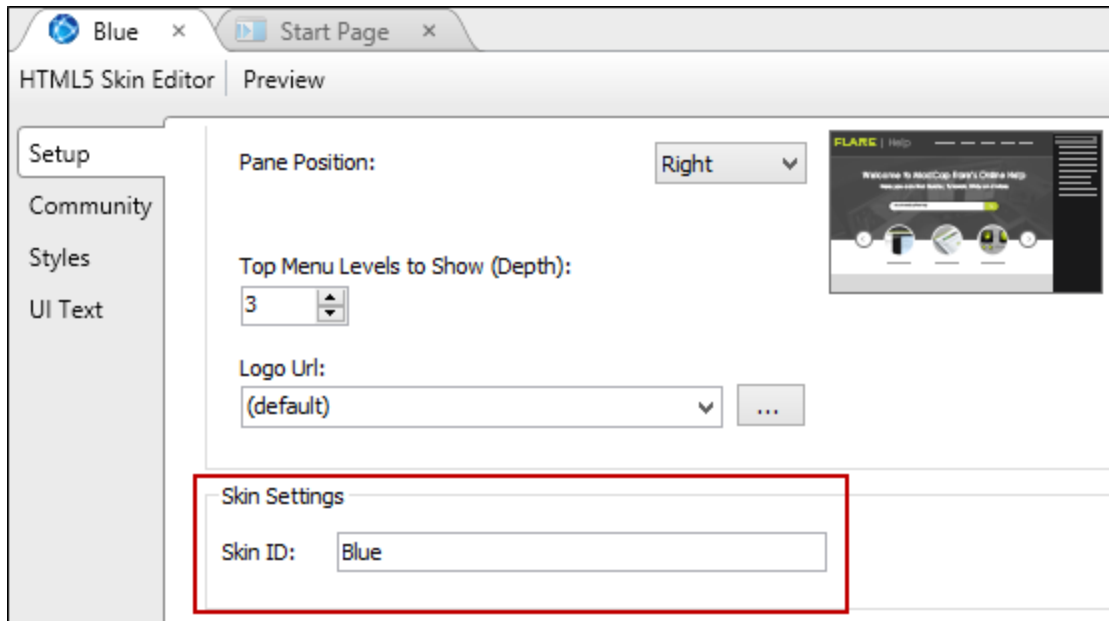
Flare copies all of the project skins of the same type as the selected target (i.e., Tripane output will copy all project Tripane skins, Top Navigation output will copy all Top Navigation skins) and places them in the output folder.



In addition, the CSS styles of those skins will be copied and merged into the stylesheet as its own CSS class.

Another use for runtime skins is to display output differently for context-sensitive Help (CSH), such as Dynamic Help. In a situation such as this, the user does not choose the skin in your output. Instead, your software developer can program the application behind the scenes to use the correct skin depending on the situation.

To help accomplish this, there is a Skin ID field on the Setup tab for both Top Navigation and Tripane skins.



This ID is based on the file name of the skin, but you can change it if you want. Provide your software developer with the appropriate skin ID to be used for CSH.

EXAMPLE

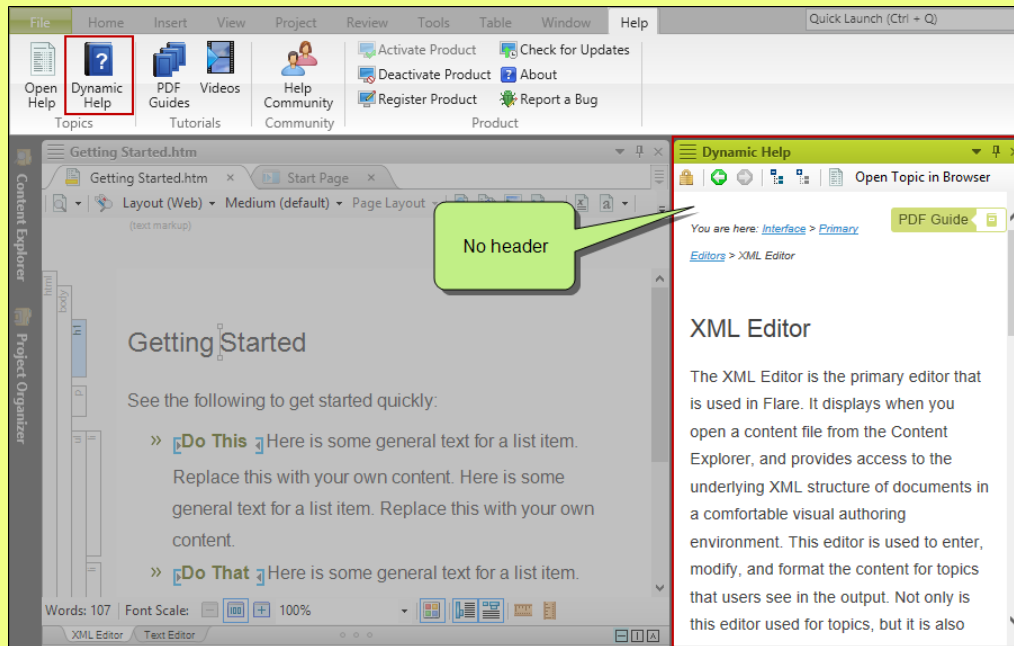
Flare's online Help uses the HTML5 Top Navigation output. If you open the full Help system, you will see a header at the top of every page.



We wanted to use this same output for the Dynamic Help window pane within Flare, but we did not want any page (except for the Home page) to display a header.

So in addition to our regular skin, we created another one and removed the header section from it. We also provided our developers with the IDs for each skin.

Therefore, if you have your cursor in the XML Editor, the Dynamic Help will display the XML Editor topic, without the page header.



But if you open the topic in your browser...



... the topic will look like this, with the header section at the top.

modcap
FLARE
Version 12

Get StartedBasic StepsFeaturesInterfaceTutorials

Search


You are here: [Interface](#) > [Primary Editors](#) > XML Editor

PDF Guide

★★★★★

XML Editor

The XML Editor is the primary editor that is used in Flare. It provides access to the underlying XML structure of documents in a comfortable visual authoring environment. This editor is used to enter, modify, and format the content for topics that users see in the output. Not only is this editor used for topics, but it is also used for working with master pages and snippets. Although this editor lets you produce XML files, you do not need to know anything about XML to use it.



Primary Editors

- Internal Text Editor
- Page Layout Editor
- Skin Editor
- Stylesheet Editor
- XML Editor

▶ HOW TO OPEN THE EDITOR

▶ CONTROLLING HOW CONTENT IS DISPLAYED IN THE EDITOR

▶ EDITING THE XHTML CODE

▶ WHAT YOU CAN DO IN THIS EDITOR

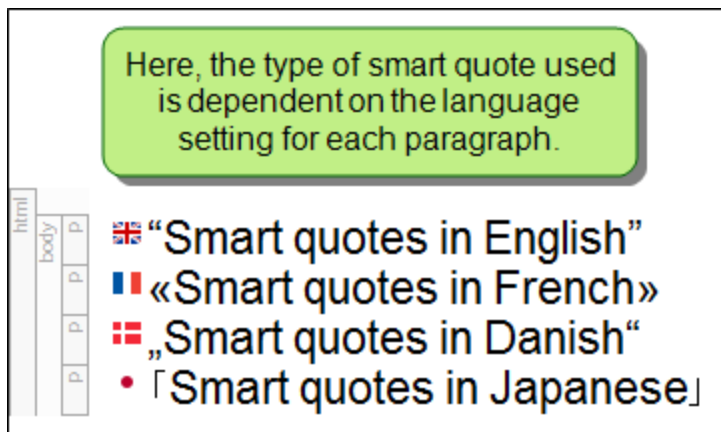
It's the same output file, just using two different skins.



Note: For HTML5 Tripane output that is not enabled for repsonsive output, Flare does not support multiple skins in the output.

Smart Quotes Per Language

Improvements have been made to the way Flare handles smart quotes when adding them to the XML Editor. Previously, Flare would automatically replace straight quotes with curly quotes, regardless of the language being used in the project. However, because different languages use different kinds of quotation marks, Flare now replaces quotes based on your project's current language. Additionally, if you are using a different language in a specific topic or tag, Flare will use the correct smart quote for that language setting.



Note: Selecting the smart quote option does not automatically replace all of the straight quotes that you already have in your project. Likewise, existing smart quotes are not automatically replaced with language-specific smart quotes. If you need to update the quotes in your project, it is recommended that you use the Find and Replace in Files window pane to find all of the files in the project containing straight quotes (or existing smart quotes) and then enter smart quotes in their place.

Source Control Enhancements

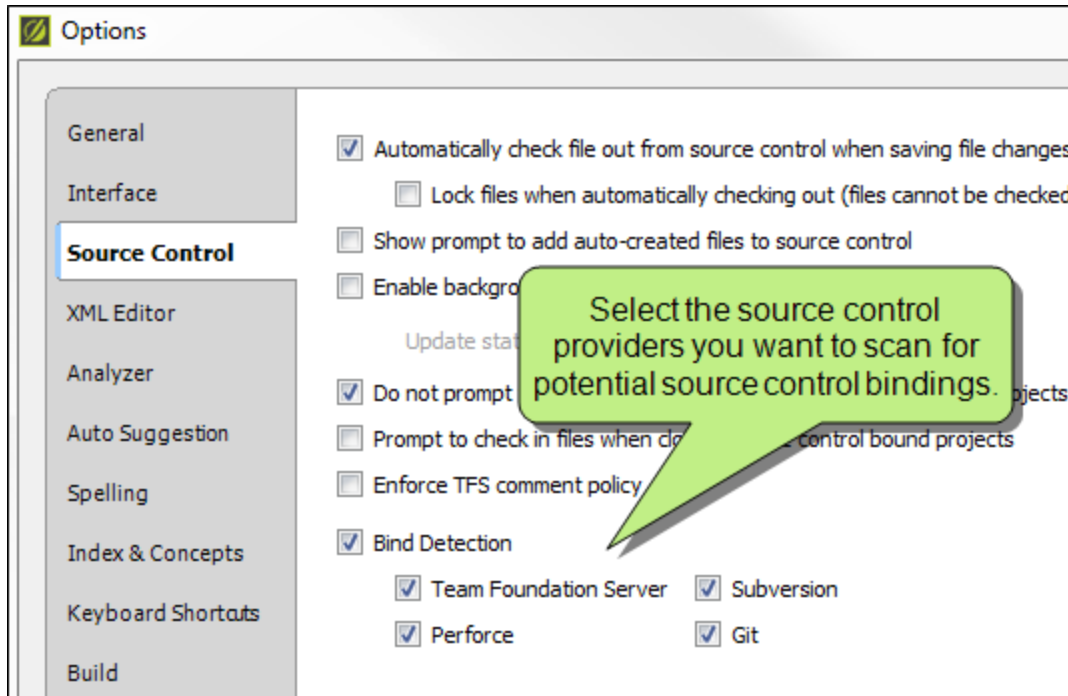
Enhancements have been made to source control functionality. These include source control publishing, offline mode, and bind detection. You can also now make changes to Git and Subversion network settings from within the Flare interface. Git also now supports SSH repositories.

BIND DETECTION

Bind detection has been added to Flare. This functionality scans your project when you load it to see if the project has been previously bound to source control. If a binding is detected, you then have the option of applying the binding and committing the project to source control. Depending on the provider you are using, Flare may search the file system and its artifacts, and contact and query servers, to find potential source control bindings.

You can use bind detection as an alternative to importing a Flare project. If you have received a Flare project file (e.g., by copying it from a server, by opening it from a network location), you can simply open the file and Flare will search for and apply existing source control bindings.

To enable bind detection, open the Options dialog (**File>Options**), and select the **Source Control** tab. Select the **Bind Detection** check box, then select the check boxes next to the source control providers you want to scan for bindings.



Note: Source control providers are scanned in the following order:

1. Git
2. Subversion
3. Perforce
4. Microsoft Team Foundation Server (TFS)



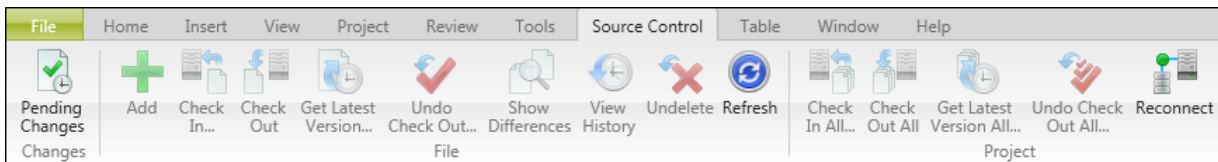
Tip: Detecting source control bindings may take a considerable amount of time. It is recommended that you select only the source control providers that you use to speed up the detection process.

DISCONNECT FROM SOURCE CONTROL

If you use Microsoft Team Foundation Server (TFS), Subversion, or Perforce as your source control provider, you can now disconnect from source control to work in offline mode. To do this, select **Source Control>Disconnect**.

Disconnecting from source control is beneficial because it allows you to modify files when you would otherwise not have access to the source control system (e.g., you are out of the office with your laptop or you do not want to access source control over VPN). It also provides a fallback offline status in the event that the network is disconnected while you are working, so you are able to continue working on the files you have checked out until the network connection is restored.



When you disconnect from source control, it disables the source control network. When the network is disconnected, you are not able to see the source control status of files or access source control functions. Because of this, TFS and Perforce users should check out any necessary files before disconnecting; checked out files can be modified while disconnected from source control. Subversion users who disconnect from source control can work as if their project is unbound without needing to check out files.



When you are disconnected from source control, all source control features are disabled until you reconnect to the network.

You can reconnect to source control at any time. When you are ready to reconnect, select **Source Control>Reconnect**. This will restore your network connection. When you reconnect to source control, you should check in your files to ensure that other users have access to the changes you made while offline.



Note: Your current network connection status is indicated in the lower right corner of the Flare interface. If you are connected you will see **Connected** ; if you are disconnected you will see **Disconnected** .



Note: If you make a change to a file's properties (e.g., delete, rename) while disconnected from source control, your changes may not be preserved when you reconnect to the network. To prevent errors, it is recommended that you do not make these kinds of changes until you reconnect to source control.



Note: If you are using TFS or Perforce while disconnected and you attempt to modify a file that is not checked out, you will receive an error when you try to save the file. To avoid this, be sure you have all necessary files checked out before disconnecting from source control.



Note: Git and Microsoft Visual SourceSafe (VSS) do not support offline mode.

NETWORK SETTINGS

If you use Subversion or Git, you can now change various source control network settings while working in Flare.

HOW TO CHANGE NETWORK SETTINGS

1. Do one of the following, depending on the part of the user interface you are using:
 - » **Ribbon** Select the **Source Control** ribbon. In the **Project** section select **Network Settings**.
 - » **Right-Click** If you have the Content Explorer, Project Organizer, Pending Changes window pane, or File List open, right-click on any file and select **Source Control>Project>Network Settings**.
 - » **Source Control Explorer** From the **View** ribbon, open the Source Control Explorer. From the drop-down, select the **Home** pane. Click **Network Settings**.

The Network Settings dialog opens.

2. In the **Group** field, select the group for which you want to change the settings.



OR

Do one of the following:

- » (Optional) If you want to add a custom group, type its name in the **Group** field, then click **Add**.
 - » (Optional) If you want to remove a group, select it from the **Group** field, then click **Remove**.
3. (Optional) If you entered a custom group name, in the **Remote Host** or **Remote Host/Pattern** field, enter the name of the domain to which the network settings should apply.



Note: If you are using Subversion, you can enter wildcards in this field.

4. In the grid, make changes to the network settings as necessary. Click  to sort the settings by category, or click  to sort them alphabetically.

GIT NETWORK SETTINGS

» Cookies

- » Cookie File
- » Save Cookies

» HTTP Proxy Options

- » Proxy Host

» Identity

- » User Agent

» Performance

- » Enable EPSV
- » Max Requests
- » Maximum High Latency Time
- » Min Sessions
- » Minimum Transfer Speed
- » POST Buffer Maximum

» SSL Configuration

- » SSL CA File
- » SSL CA Path
- » SSL Certificate
- » SSL Key File
- » SSL Password Prompt
- » Try SSL
- » Verify SSL



Note: For more information about each of these settings, refer to:

<https://git-scm.com/docs/git-config>

SUBVERSION NETWORK SETTINGS

» Advanced

- » Chunk Requests
- » HTTP Bulk Update
- » HTTP Compression
- » Maximum Connections

» Caching Options

- » Store Authentication Info
- » Store Passwords
- » Store Plaintext Passwords
- » Store Plaintext SLL Client Cert Passphrase
- » Store SSL Client Cert Passphrase

» Global Options

- » Default User Name

» HTTP Proxy Options

- » Proxy Host
- » Proxy Host Port
- » Proxy Password
- » Proxy Timeout
- » Proxy User Name
- » Site Exceptions

» **SSL Options**

- » SSL Authority Files
- » SSL Client Certificate File
- » SSL Client Certificate Password
- » SSL Trust Default CAs



Note: For more information about each of these settings, refer to:

<http://svnbook.red-bean.com/en/1.7/svn.advanced.confarea.html>

5. Click **Save**.

SOURCE CONTROL PUBLISHING

Source control publishing has been added to Flare 12. Using the Destinations Editor, you can create a source control destination to directly publish your output to source control from Flare, rather than having to upload it to source control provider outside of Flare. Publishing your output to source control is beneficial because you can keep previous versions of your output in source control and access them if necessary.

For steps about how to creating a source control destination, see the online Help.

EXAMPLE

You write online Help for a software company. When you publish your documentation, you always publish a copy of the final release documentation to Git. Rather than uploading your Help files to Git after you finish publishing your Flare HTML5 target to your FTP site, you add a source control destination to your existing FTP destination. By publishing to both destinations at once, you can save time.



Note: Microsoft Visual SourceSafe (VSS) does not support source control publishing.



Note: If you are using Git and you want to push your output to a remote repository, you can select the **Push to Remote** option in the Destination Editor.



Note: Log in credentials are enabled after binding. If you are using Git, they are enabled if you selected the Push to Remote option.

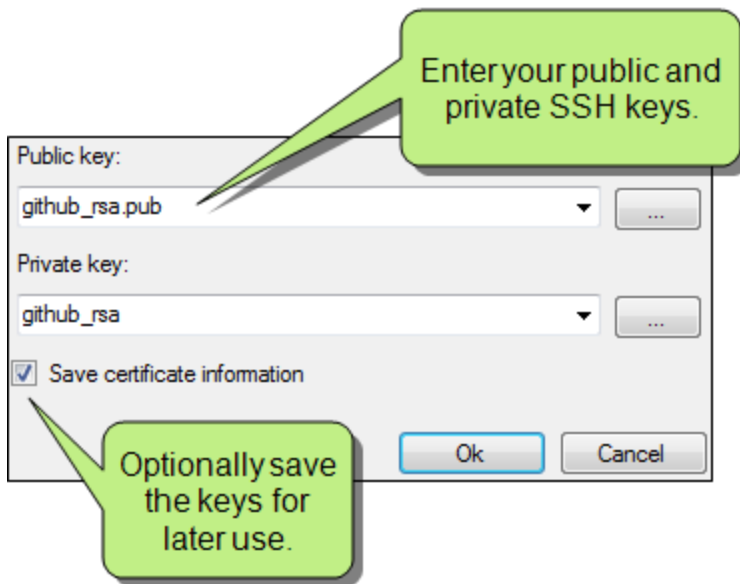


Note: You cannot use Flare to manage previous versions of output stored in source control.

GIT SSH REPOSITORY SUPPORT

Git now supports SSH repositories. Previously only HTTP repositories were supported. This gives you more options when binding your project.

When you are binding your project, enter an SSH URL in the Remote Repository field. After you click OK to bind the project, the Certificate Specification dialog opens. In this dialog, enter your public and private SSH keys. You save your keys so Flare will not request them later (e.g., when pushing or pulling files), and you can save the SSH keys to the SSH folder in your user profile.



SOURCE CONTROL STATUS

When your project is bound to source control, Flare contacts your source control provider prior to opening the project. This extra step allows the source control provider to verify the current status of the files in your project (e.g., checked out, modified).

After the source control provider checks the status of your files, you will be prompted to get the latest files from source control (if this option is enabled).



Tables—Advanced Sorting


You can sort tables using advanced options, choosing which columns to sort first. In addition, you can sort tables in this manner immediately, when the target is generated, or both.

HOW TO SORT TABLES

1. Open the content file (e.g., topic, snippet).
2. Place your cursor inside the table.
3. Do one of the following, depending on the part of the user interface you are using:
 - » **Ribbon** Select the **Table** ribbon. In the **Data** section, select the **Ascending** down arrow and then choose **Advanced Sorting Options**.
 - » **Right-Click** Do one of the following:
 - » Right-click inside the table and from the context menu, select **Sort Rows>Advanced Sorting Options**.
 - OR
 - » Right-click one of the appropriate tag bars to the left of the table, depending on the rows you want to sort (e.g., **table**, **thead**, **tfoot**), and from the context menu, select **Sort Rows>Advanced Sorting Options**.

The Advanced Sorting Options dialog opens.

4. Click . A row is added to the grid on the right.
5. In the **Column** cell, click the down arrow and choose the column to sort by. Then in the **Direction** cell choose **ascending** or **descending**.
6. (Optional) If you want to add another row, telling Flare which column to sort by next, click  again. In the **Column** cell, click the down arrow and choose the column to sort by next. Then in the **Direction** cell choose **ascending** or **descending**. Repeat this for each additional row you want to add to the grid.
7. (Optional) Use the buttons on the right side of the dialog if you want to remove or rearrange the order of any of the rows in the grid.

8. (Optional) If you want Flare to automatically update the table sorting when you generate output, select **Apply at compile time**. This can be useful in case you make changes to the table in the XML Editor and want Flare to automatically make the changes so you don't have to.
9. (Optional) If you want Flare to apply the sorting rules to the table right away in the XML Editor, click **Apply now**.
10. Click **OK**.
11. Click  to save your work.

EXAMPLE

Let's say you have a table listing winners of the New York Marathon for a 10-year period, like this:



Year	Name	Country	Time
2006	Gomes dos Santos, Marilson	Brazil	2:09:58
2007	Lel, Martin	Kenya	2:09:04
2008	Gomes dos Santos, Marilson	Brazil	2:08:43
2009	Keflezighi, Meb	USA	2:09:15
2010	Gebremariam, Gebregziabher	Ethiopia	2:08:14
2011	Mutai, Geoffrey	Kenya	2:05:06
2012	Cancelled	NA	NA
2013	Mutai, Geoffrey	Kenya	2:08:24
2014	Kipsang, Wilson	Kenya	2:10:59
2015	Biwott, Stanley	Kenya	2:10:34

Suppose you want to re-organize the information in this table. Maybe you want to sort first of all by the country of each runner (the third column) in ascending order.

With the third column organized, you then want the rows to be organized once more according to the best times (the fourth column) in ascending order.

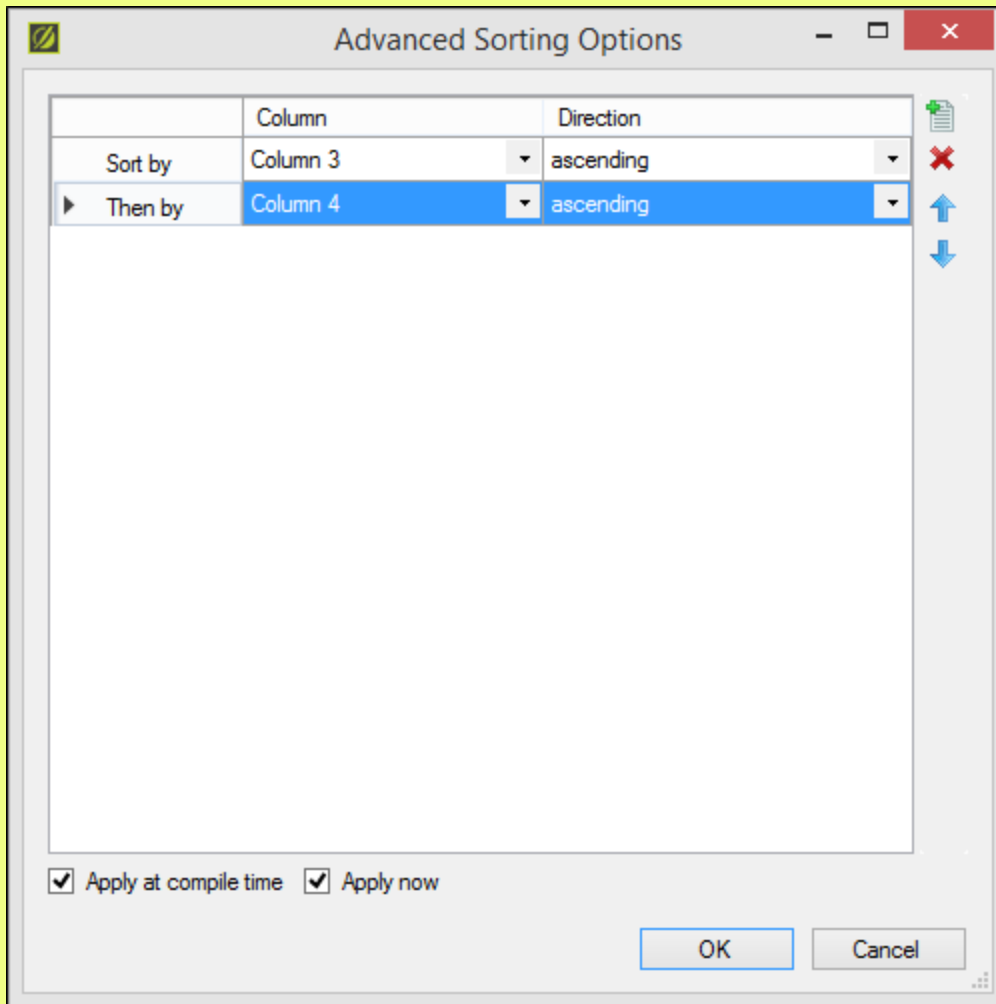
So you right-click the **table** structure bar and choose **Sort Rows**, then **Advanced Sorting Options**.

The screenshot shows a web editor interface with a sidebar on the left containing structure elements: `html`, `body`, `h1`, `p`, `tr`, and `table`. The `table` element is highlighted with a red box. The main content area displays the text "Tables Sorting" and "Here is a table that you can sort." Below this is a table with the following data:

Year	Name	Country	Time
	Santos, Marilson	Brazil	2:09:58
	Martin	Kenya	2:09:04
	Santos, Marilson	Brazil	2:08:43
	ighi, Meb	USA	2:09:15
			08:14
			05:06
	ncelled	NA	NA
	Geoffrey	Kenya	2:08:24
	ng, Wilson	Kenya	2:10:59
	t, Stanley	Kenya	2:10:34

A context menu is open over the table, showing options: Select, Cut, Copy, Paste Before, Paste After, Delete, Sort Rows, Reset Local Cell Formatting, Table Style, Caption, Table Size, Table Properties..., Move, Conditions... (Ctrl+Shift+C), Name..., and Create Snippet. The 'Sort Rows' option is highlighted with a blue bar, and its sub-menu is open, showing 'Ascending', 'Descending', and 'Advanced Sorting Options...'. The 'Advanced Sorting Options...' option is highlighted with a red box.

In the Advanced Sorting Options dialog, your settings would look like this:



As a result, the table would be reorganized to look like this:

The screenshot shows a web browser window with a document titled "index.html". On the left side, there are several panels or tabs labeled "html", "body", "hl", "p", "table", "thead", "tbody", "tr", and "th". The main content area displays the following:

Tables Sorting


Here is a table that you can sort.

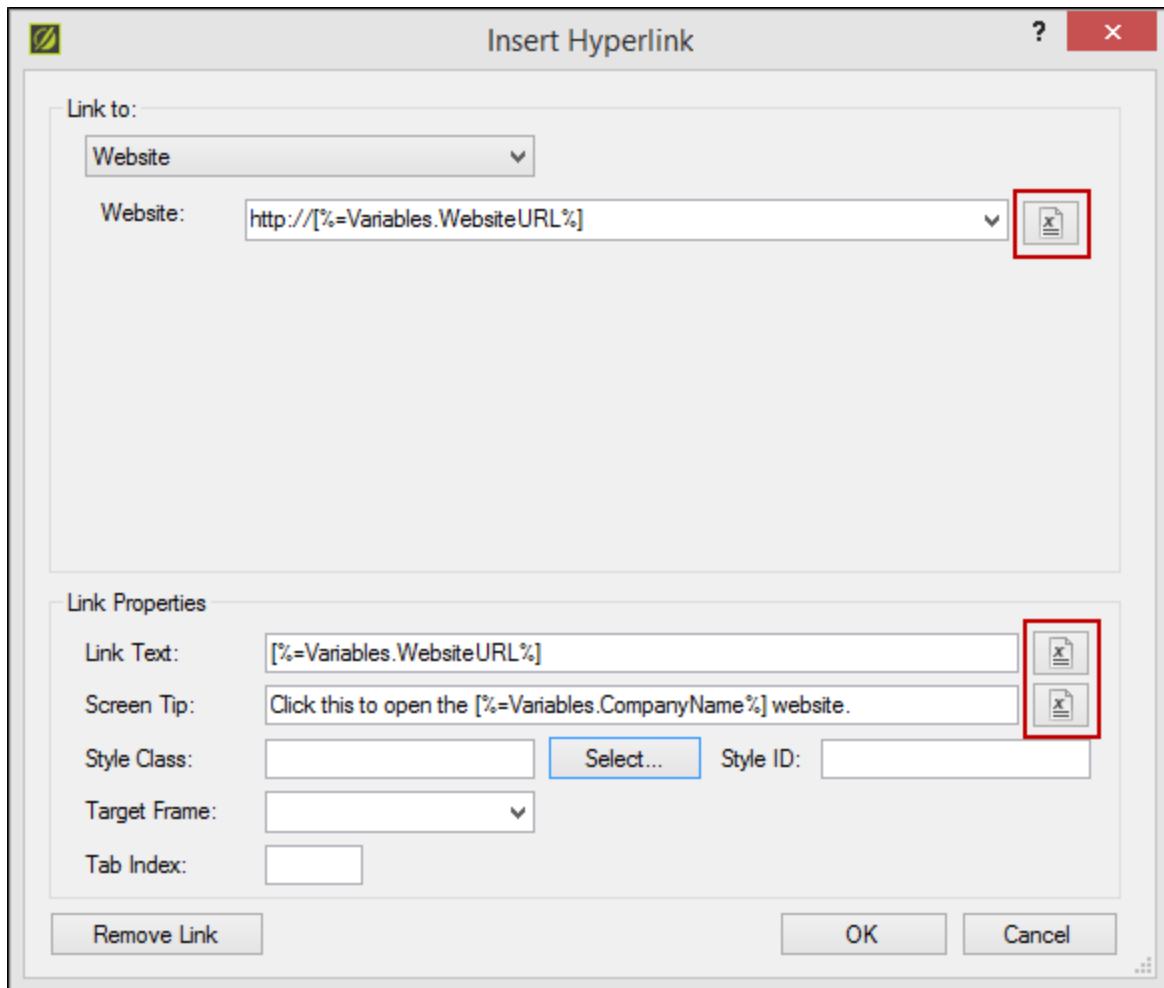
Year	Name	Country	Time
2008	Gomes dos Santos, Marilson	Brazil	2:08:43
2006	Gomes dos Santos, Marilson	Brazil	2:09:58
2010	Gebremariam Gebregziabher	Ethiopia	2:08:14
2011	Mutai, Geoffrey	Kenya	2:05:06
2013	Mutai, Geoffrey	Kenya	2:08:24
2007	Lel, Martin	Kenya	2:09:04
2015	Biwott Stanley	Kenya	2:10:34
2014	Kipsang Wilson	Kenya	2:10:59
2012	Cancelled	NA	NA
2009	Keflezighi, Meb	USA	2:09:15

Variable Enhancements

Several enhancements have been made to variables in Flare. These include the addition of the Insert Variable option to many places throughout the interface, the ability to override variable definitions in snippets, system variables based on the language in Flare rather than the language of the operating system, and more.

INSERT VARIABLE OPTION ADDED THROUGHOUT INTERFACE

In many places throughout the Flare interface, you will find that you can now insert a variable where text can be added. In most cases, this is indicated by the presence of the Insert Variable button .



Location in Flare

Auto-Index Dialog

Create Glossary Term Dialog

Edit Multimedia Dialog

Edit QR Code Dialog

Location in Flare
Edit Slideshow Dialog
Image Properties Dialog (General Tab)
Index Link Dialog
Index Window Pane
Insert Hyperlink Dialog
Insert Multimedia Dialog
Insert QR Code Dialog
Insert Text Popup Dialog
New Cross-Reference Style Class Dialog
Edit Cross-Reference Style Class Dialog
Instead of the Insert Variable button, variables are accessed for cross-references by clicking the Available commands drop-down and choosing Show Variable Commands .
Properties Dialog for Browse Sequences (General Tab)
Properties Dialog for Glossaries (Glossary Term Tab)
Properties Dialog for TOCs (General Tab)
Properties Dialog for Topics (Topic Properties Tab)
Skin Editor (General Tab; Also UI Text Tab for HTML5)
Variables in skin captions are supported for HTML5, WebHelp, WebHelp Mobile outputs.
Target Editor (Multiple Tabs)
For PDF and EPUB options, the Insert Variable button is only shown after you click in the text field.
Variable Set Editor
This means you can have nested variables.

When you insert a variable into these locations, you will see Flare's variable syntax.

The screenshot shows the 'Insert Hyperlink' dialog box with the following fields and values:

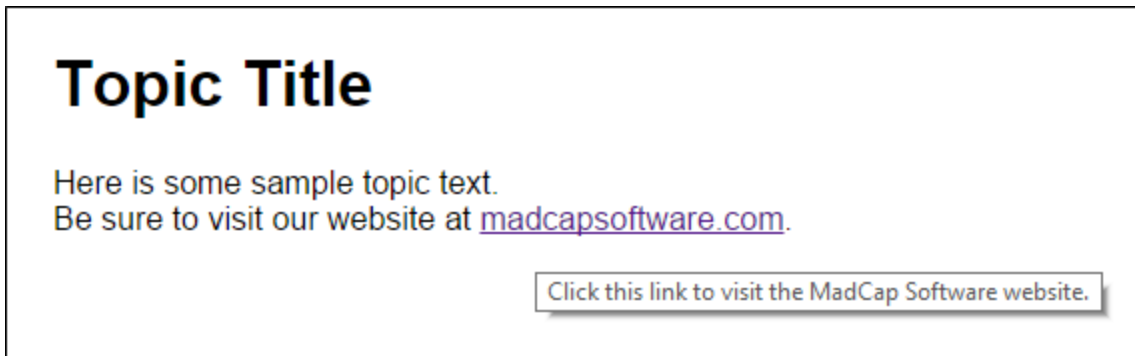
- Link to:**
 - Website (selected in dropdown)
 - Website: `http://[%=Variables.WebsiteURL%]`
- Link Properties:**
 - Link Text: `[%=Variables.WebsiteURL%]`
 - Screen Tip: Click this link to visit the `[%=Variables.CompanyName%]` website.
 - Style Class: (empty) [Select...]
 - Style ID: (empty)
 - Target Frame: (dropdown)
 - Tab Index: (empty)

Buttons at the bottom: Remove Link, OK, Cancel.

You will also see it in the XML Editor and other editors.



But in the output, this syntax will be replaced by the variable definition.



Note: In text boxes, you can insert a variable anywhere you want (before, between, or after text). But in a field such as a drop-down, the variable will always be inserted at the beginning, before any text. Therefore, you will need to copy and paste the variable syntax where you want it.








SHOW/HIDE EVALUATED VARIABLE TERMS AND DEFINITIONS


There is a toggle button to show or hide variable definitions in the following editors:

- » Auto-Index Editor
- » Index Links Editor
- » Glossary Editor
- » Variable Set Editor








EXAMPLE

In the following glossary, a variable has been used in the second row, both for the term and its definition. Initially, only the variable syntax is shown.

Glossary Editor       0 Duplicate Term(s) Update Glossary Reference Markup 				
Condition		Terms	Definition	Link
	<input type="checkbox"/>	My Term	My definition	
	<input type="checkbox"/>	<code>[%=Variables.Variable1%]</code>	This is the definition for <code>[%=Variables.Variable1%]</code>	

After clicking  in the local toolbar, two more columns are displayed. These columns display the actual term and definition that will display in the output, in place of the syntax.

Click this button to show or hide the variable definitions.

Glossary Editor       0 Duplicate Term(s) Update Glossary Reference Markup 					
Condition		Terms	Evaluated Terms	Definition	Evaluated Definition
	<input type="checkbox"/>	My Term	My Term	My definition	My definition
	<input type="checkbox"/>	<code>[%=Variables.Variable1%]</code>	My Variable Definition	This is the definition for <code>[%=Variables.Variable1%]</code> .	This is the definition for My Variable Definition.

SNIPPET VARIABLES













Now you can override definitions for variables in snippets. You can do this at the topic level or the snippet level. This gives you more ways to customize your snippets.

SYSTEM VARIABLES USE LANGUAGE IN FLARE

System variables no longer use the language of the operating system. Instead, the language set in Flare is used, whether the language is set in a tag, in the target, or at the project level. For example, you might have an English operating system, but in a Flare target you have French set as the language. If you insert a date/time variable, the day and month will display in French when you generate that target. Flare will also adjust the format of the variable if necessary.

WARNING FOR INVALID SYNTAX

If there is a variable inserted within another variable in the Variable Set Editor, a warning will show if it is invalid. Also, the cell containing the invalid syntax is shown in pink.

VariableSet Editor         				! Invalid syntax for variable definition(s) (1)	
	Name	Definition	Comment		
	CompanyName	MadCap Software [=Variables.Variable1%]			
	PhoneNumber	858 123 4567			
	PhoneNumber	858 234 5678			
	Variable1	My Variable			

Word Enhancements

Several changes have been made to Flare's Word output. This includes new supported Flare elements, image transparency, and navigation improvements.

DIV IMPROVEMENTS

Changes have been made to how groups (or divs) are supported in Word output.

SINGLE-CELL TABLES

In previous versions of Flare, if you used divs in your content and then generated Word output, they appeared in Word as regular text. In Flare 12, divs appear in the output as single-cell tables. This allows you to move the div in your output, resize it, or even add additional rows or columns. Any styling (such as background color) that you have applied to your div will appear in the output.

Additionally, you can nest divs within tables or within other divs. Because each div appears as a single-cell table, you can resize or edit each nested element individually.

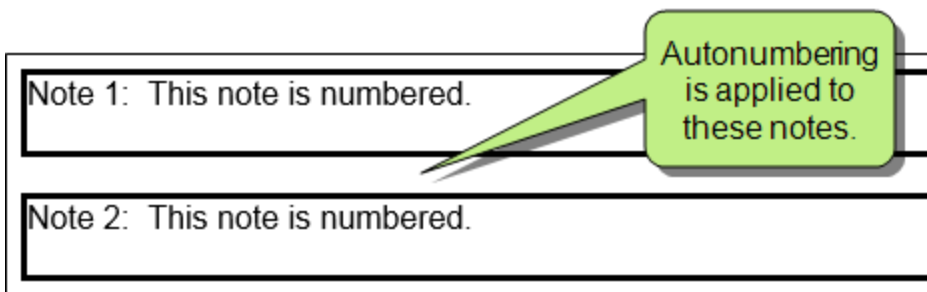
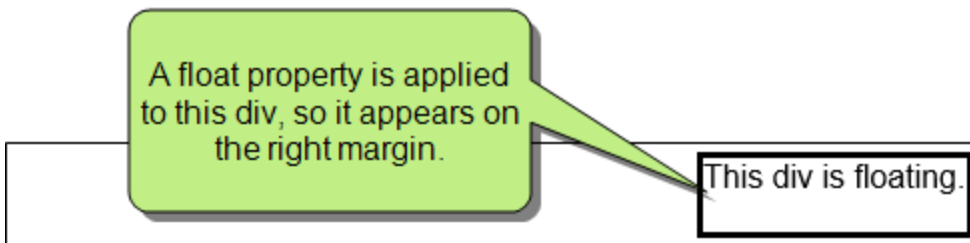
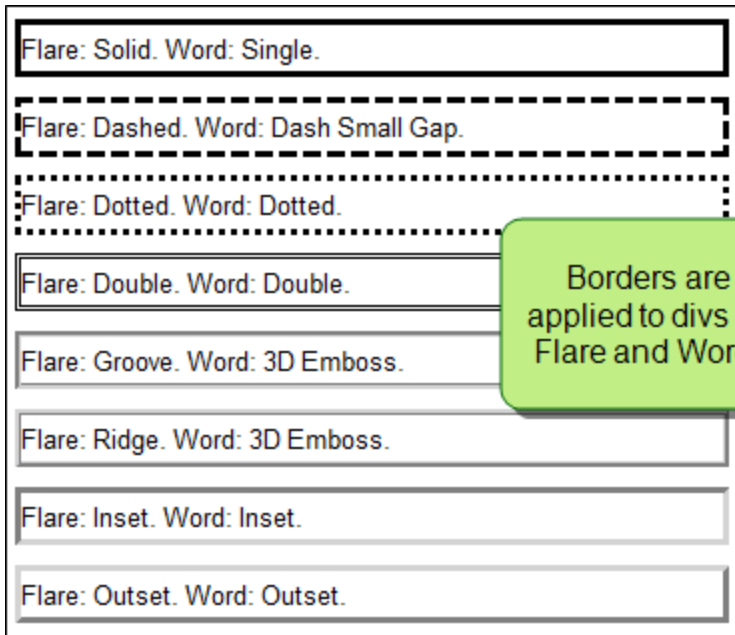
APPEARANCE

Border styles, float properties, and autonumbering are now supported on divs in Word output.

If your divs use a CSS border style in Flare (such as the solid style), they will use the corresponding Word border style when you generate your Word output.

Float properties are also preserved when they are applied to a div. You can create a div style that uses a CSS float property, and when you generate Word output, the div will be placed in the appropriate location. However, you do not need to use a float property to achieve this effect; because divs are created as single-cell tables when you generate Word output, you can move the table cell to any location on the page.

If your div uses autonumbering, it will be maintained when you generate your Word output. This is especially useful if you use divs to create numbered note styles or numbered lists.





Note: Styling, such as borders or background colors, must be applied to the div (not to the paragraph tag within the div) to appear in the generated Word output.



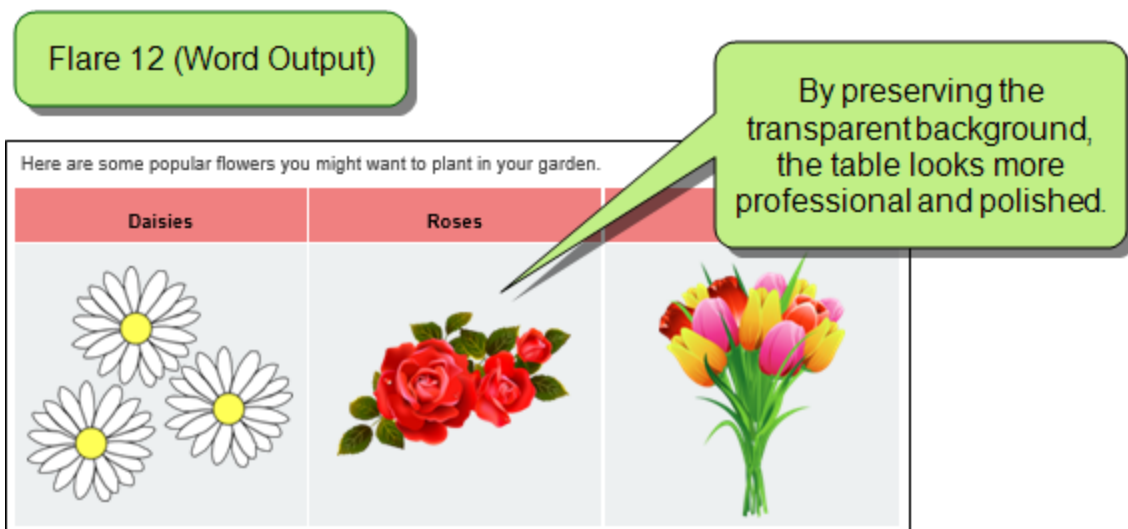
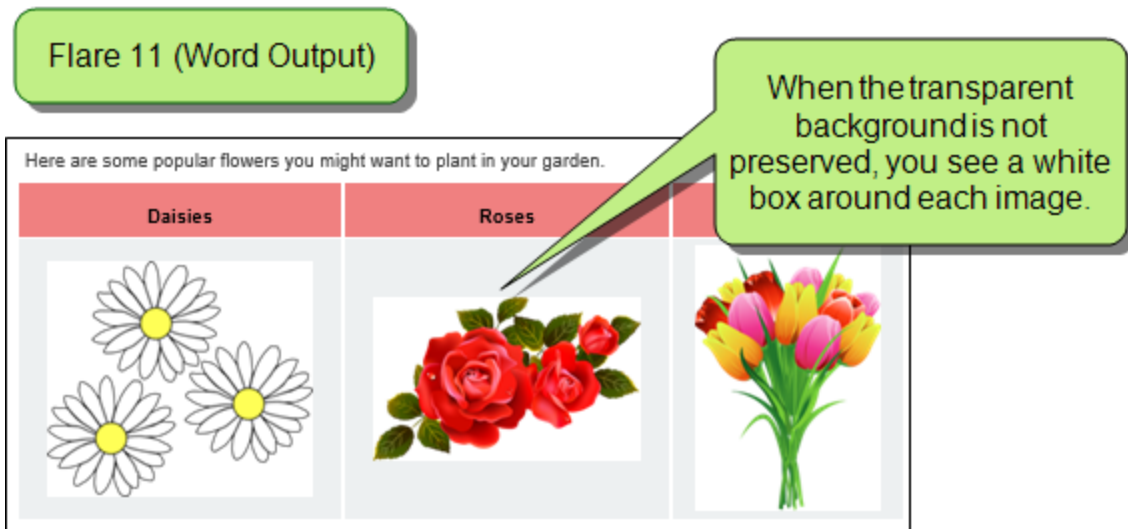
Note: Flare does not support background images in generated Word output. If you use background images (e.g., in a note div, in a cover page layout), you can apply them in Word after generating the output from Flare.



Note: Autonumbering is also maintained for keywords, concepts, empty tags, and other non-text elements that are not part of divs.

IMAGE TRANSPARENCY AND BORDERS

When you generate Word output, images in Flare with a transparent background now retain their transparency. This gives you a seamless look when you use images on a colored background in Word.



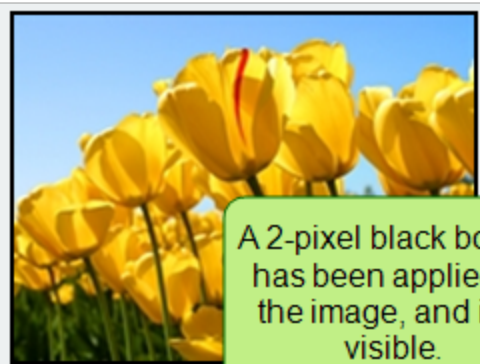
Additionally, borders are also now supported on images.

Flare 11 (Word Output)



A 2-pixel black border has been applied to the image, but it is not visible.

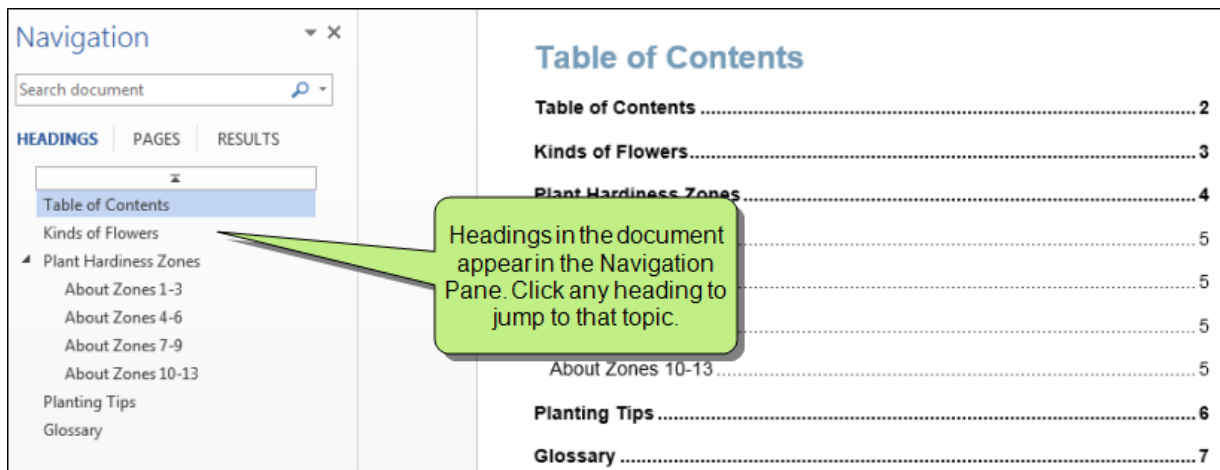
Flare 12 (Word Output)



A 2-pixel black border has been applied to the image, and it is visible.

NAVIGATION PANE IMPROVEMENTS

It is now easier to navigate Word output. You can use Word's Navigation Pane (select the **View** ribbon, then select the **Navigation Pane** check box), to quickly jump from one section of your document to another. This is because the headings in your Word output now appear in the Headings section of the Navigation Pane. In previous versions of Flare, the Navigation Pane was not supported.

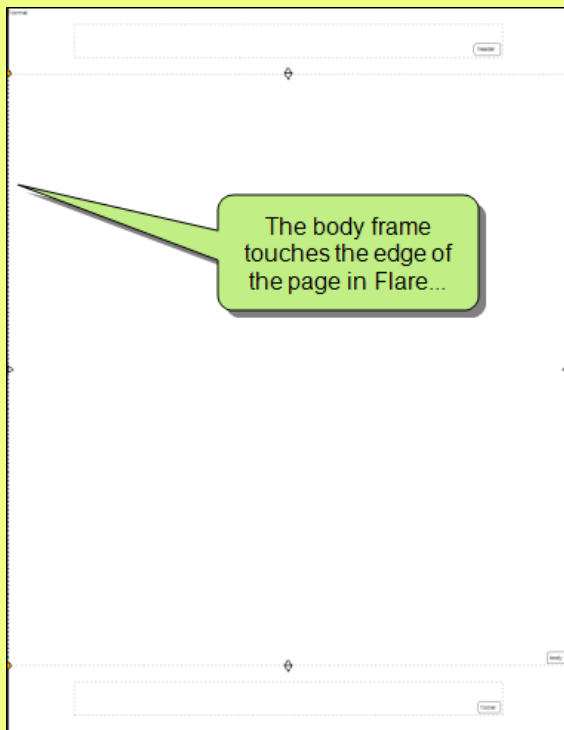


PAGE LAYOUT BUILD LOG WARNING

A new build log warning has been added if your Flare project uses a header, footer, or body frame that falls outside of the page layout boundaries supported by Word. If you try to print in Word, some of the text may be clipped. To resolve this error, you can modify your page layouts in Flare so they are within Word's page layout boundaries.

EXAMPLE

When you are creating a Word output, you set your body frame so it touches the very edge of the Flare page layout.



Plant Hardiness Zones

Use the following chart to determine your USDA plant hardiness zone.

Zone	Average Annual Minimum Temperature Range (F)
1a	-60 to -55
1b	-55 to -50
2a	-50 to -45
2b	-45 to -40

...so the text in Word touches the left and right margins.

When you view the generated Word output, the text is outside of the printable area, so it is clipped when you try to print.

Plant Hardiness Zones

Use the following chart to determine your USDA plant hardiness zone.

Zone	Average Annual Minimum Temperature Range (F)
1a	-60 to -55
1b	-55 to -50
2a	-50 to -45
2b	-45 to -40

Because Flare's body frame is outside of Word's page layout boundaries, the text is clipped when you try to print.

TITLE PAGE IMPROVEMENTS

Flare 12 now supports title page layouts for Word. Pages that use a title page layout always appear at the beginning of your generated Word output.

Title page layouts affect auto-generated TOCs. If your Flare project includes a title page, and you selected the "Generate TOC proxy" option in the Target Editor, the auto-generated TOC will appear after the title page. If title page is not specified, the auto-generated TOC will appear at the beginning of the output. Additionally, although the auto-generated TOC is considered a chapter break, it will use the same page layout file that is used for the title page.

Flare 11 (Word Output)




Table of Contents

Table of Contents	1
Kinds of Flowers	2
Plant Hardiness Zones.....	3
About Zones 1-3.....	4
About Zones 4-6.....	4
About Zones 7-9.....	4
About Zones 10-13.....	4
Planting Tips.....	5
Glossary.....	6

Planting Tips and Tricks

Kinds of Flowers

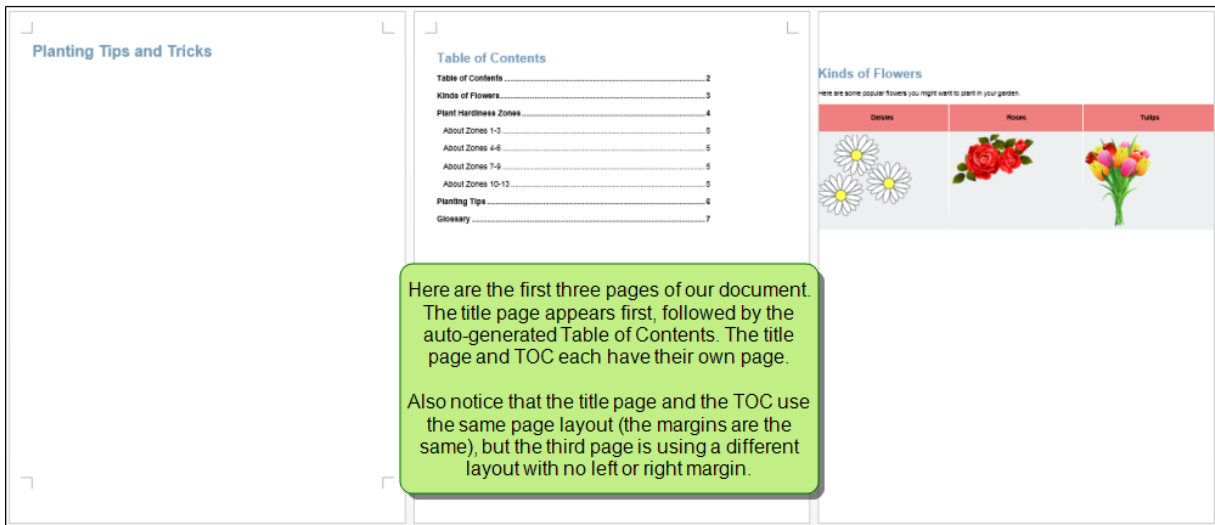
Here are some popular flowers you might want to plant in your garden.

Daisies	Roses	Tulips
		

Here are the first two pages of our document. The auto-generated Table of Contents appears first, and the title page text appears after the TOC. It does not get its own page (because it uses a paragraph text style and the title page layout is not recognized).

Since the title page layout is not recognized, both pages use the same layout.

Flare 12 (Word Output)



TRACKED CHANGES

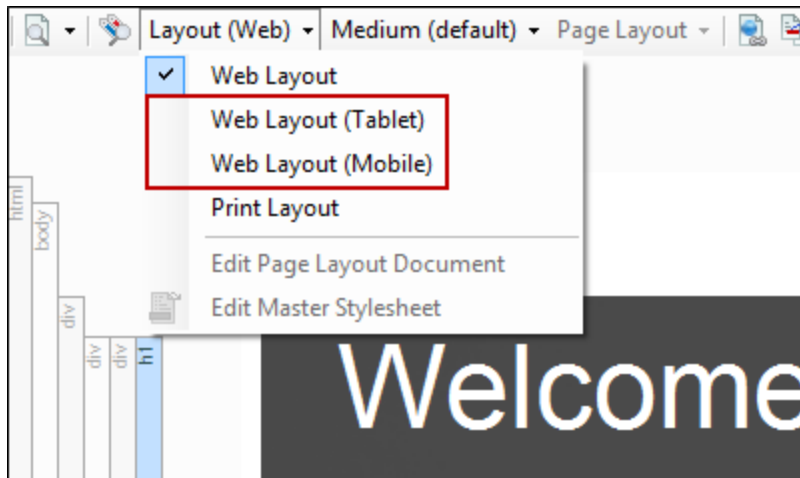
If you have tracked changes in Flare, you can now see the changes when producing a Word output. You can also enable tracked changes in Word after you produce the Word output from Flare.

XML Editor Enhancements

Some enhancements have been made to the XML Editor. There are now layout views for Tablet and Mobile. Also, you can now select style IDs, applying them to content.

LAYOUT VIEWS FOR TABLET AND MOBILE

The Layout option in the local toolbar of the XML Editor now allows you to edit topics not only in web and print view, but also in tablet and mobile view.

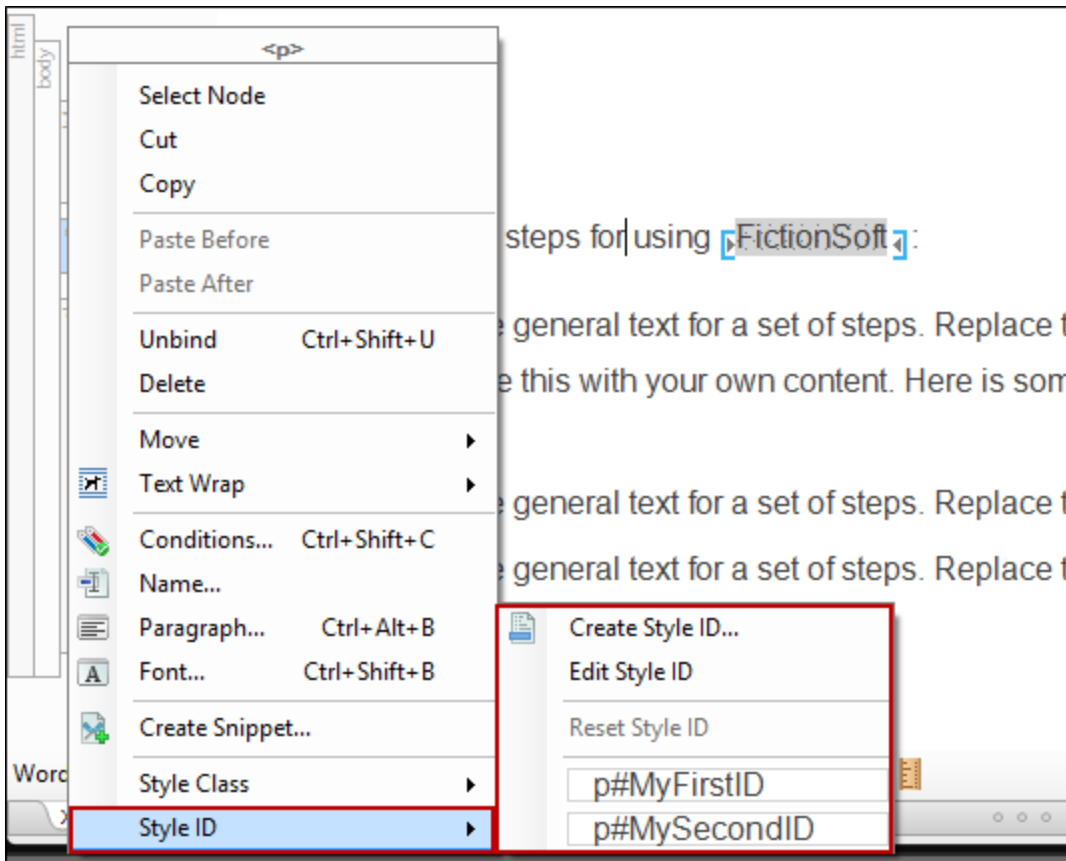


The default and print mediums are tied to the Web Layout and Print Layout modes, respectively. The tablet and mobile media queries are tied to the Web Layout (Tablet) and Web Layout (Mobile) modes, respectively. However, you can choose any stylesheet medium or media query when you are in a particular layout mode to show the content with certain styles applied.

Keep in mind that the tablet and mobile media queries are associated with your responsive skin settings on the Skin tab of HTML5 targets. This means that the Web Layout (Tablet) and Web Layout (Mobile) modes will show content in the XML Editor at the maximum width that is set for each in the Target Editor. Therefore, when viewing content in one of those modes, you might see a horizontal scroll bar at the bottom of the editor. If you adjust the breakpoints for these media queries in the Target Editor, the XML Editor will respond accordingly.

STYLE ID SELECTION

You can now apply a style ID to content in the XML Editor). If you right-click on a structure bar, you will see a submenu labeled "Style ID" that lets you choose an ID from your stylesheet. And similar to the Style Class option, you can also create, edit, and reset style IDs from this submenu.



CHAPTER 8

New Features—May 2016

A few new features and enhancements were added in this version.

This chapter discusses the following:


Linking to Lingo Projects	316
Sync to Lingo Project Updates	323



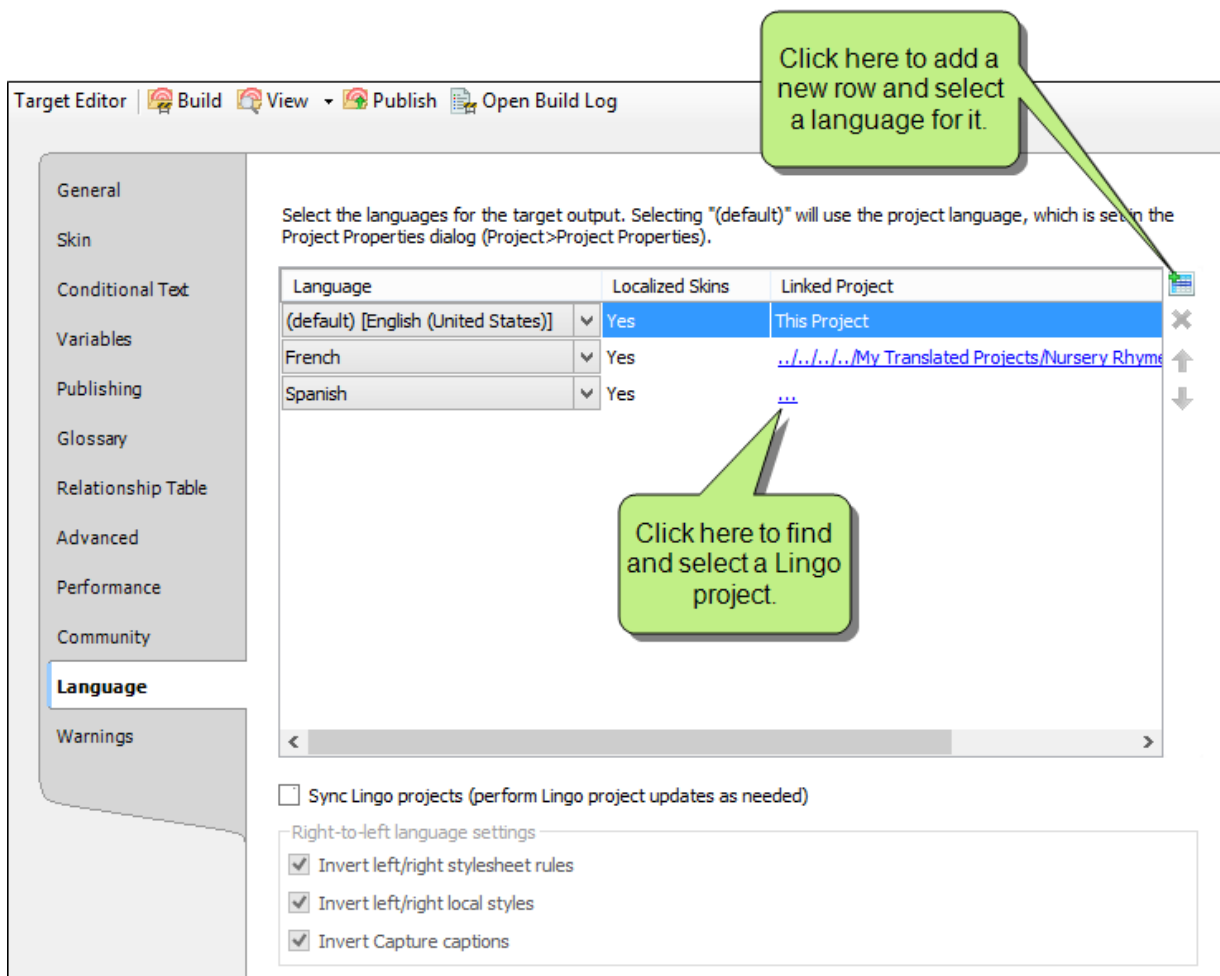
Linking to Lingo Projects

Previously, we added the ability to create links to other Flare targets that have been translated. This allows you to generate a single output containing multiple languages.

In this version, there is another dimension to this feature. You now also have the option to link directly to Lingo projects. This allows you to skip the step in Lingo of exporting translated content out to additional Flare projects.

To use this feature, open a target in Flare and select the **Language** tab. You can then click  for each language you want to add. After that, you can click the ellipsis (...) link to find and select the Lingo project containing the translated files.

You can link to many separate Lingo projects, or you can link multiple rows in this tab to the same multilingual Lingo project, or you can do both.



The language associated with the selected Lingo project is automatically selected in the drop-down on the left side of the row. However, if you link to a multilingual Lingo project, you can select the language you want. Only the remaining languages from a multilingual project can be chosen from the drop-down; in other words, if you've already selected one of the languages for a previous row, it is no longer available when you add more language rows.

EXAMPLE—LINKING TO LINGO PROJECTS

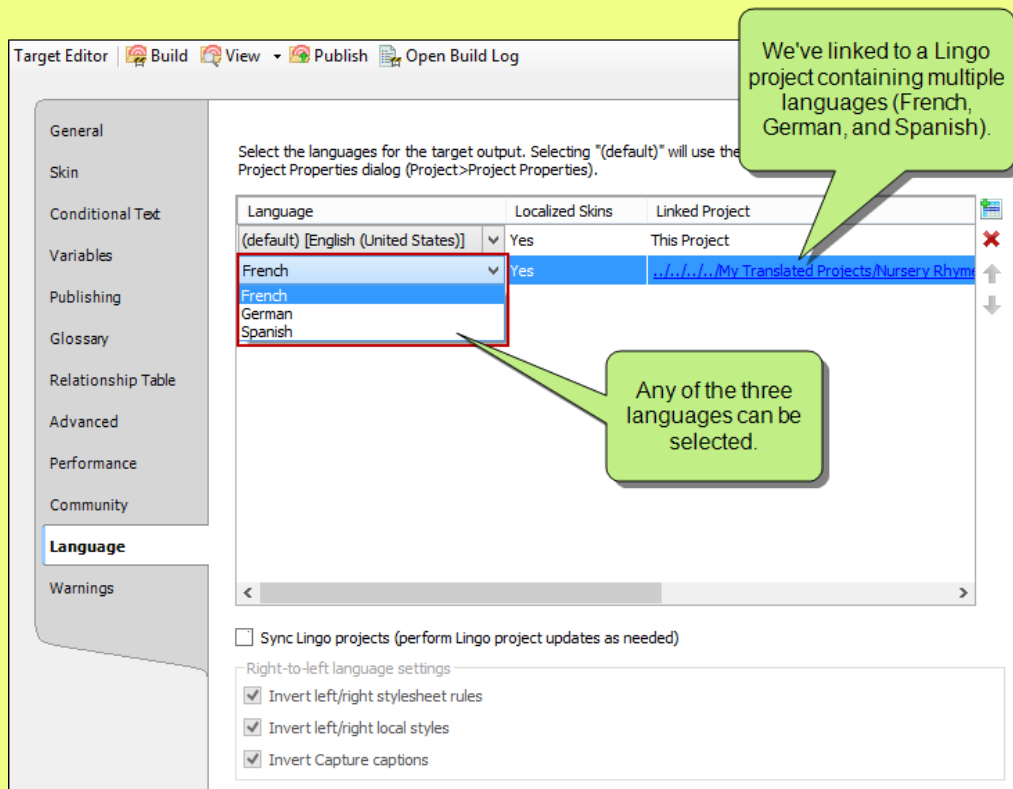
Let's say you want your Flare project to be translated into Arabic, French, German, and Spanish. You have one translator who knows French, German, and Spanish, and you have a second translator who knows Arabic.

Because the first translator knows three languages, there is no reason to have multiple Lingo projects for each language. Instead, the translator adds all three languages to a single Lingo project and translates the files.

Meanwhile, the second translator creates a second Lingo project and uses it to translate the Flare project into Arabic.

Both translators keep their Lingo projects on a server where you have access to them. When the translation work is finished, you open your target in Flare, and on the **Language** tab, you add a row for each language.

In the first row, you link to the Lingo project that was used for the French, German, and Spanish translations. After you add the first row, you notice that you can select any of the three languages.



For the first row, you select **French**. Then you add a second row. This time, you can only select German or Spanish (because French has already been used).

Select the languages for the target output. Selecting "(default)" will use the project's default language. Selecting "(default)" will use the project's default language. Selecting "(default)" will use the project's default language.

Language	Localized Skins	Linked Project
(default) [English (United States)]	Yes	This Project
French	Yes	.../My Translated Projects/Nursery Rhymes
German	Yes	.../My Translated Projects/Nursery Rhymes
Spanish	Yes	.../My Translated Projects/Nursery Rhymes

From the second row, we can select either German or Spanish.

The first row has French selected.

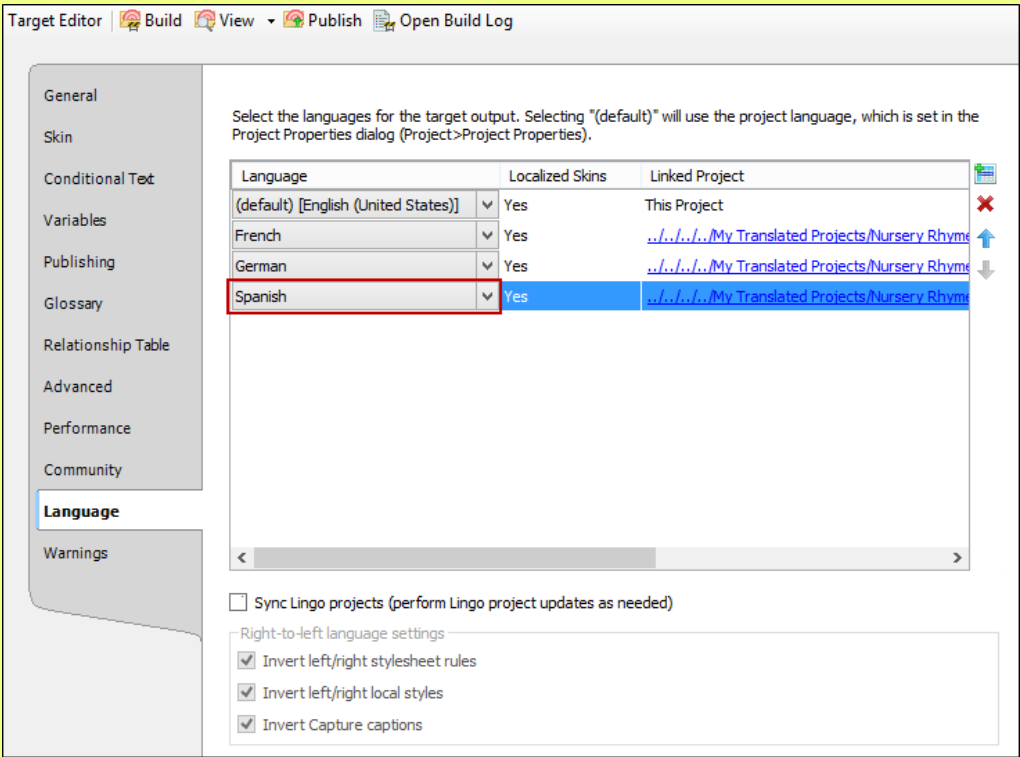
We've added a second row, linking to the same Lingo project.

☐ Sync Lingo projects (perform Lingo project updates as needed)

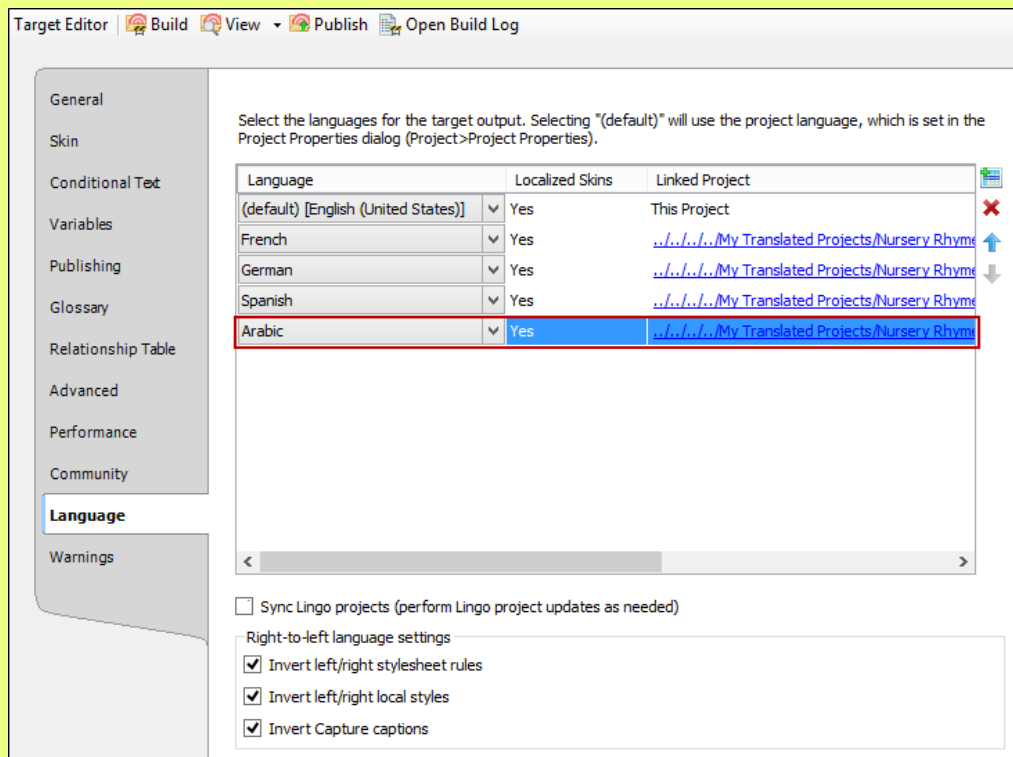
Right-to-left language settings

- ☒ Invert left/right stylesheet rules
- ☒ Invert left/right local styles
- ☒ Invert Capture captions

For the second row, you choose **German**. Then you add a third row. **Spanish** is automatically chosen for that row, because it is the only one left.



Finally, you create a fourth row and link it to the Lingo project used for the **Arabic** translation.



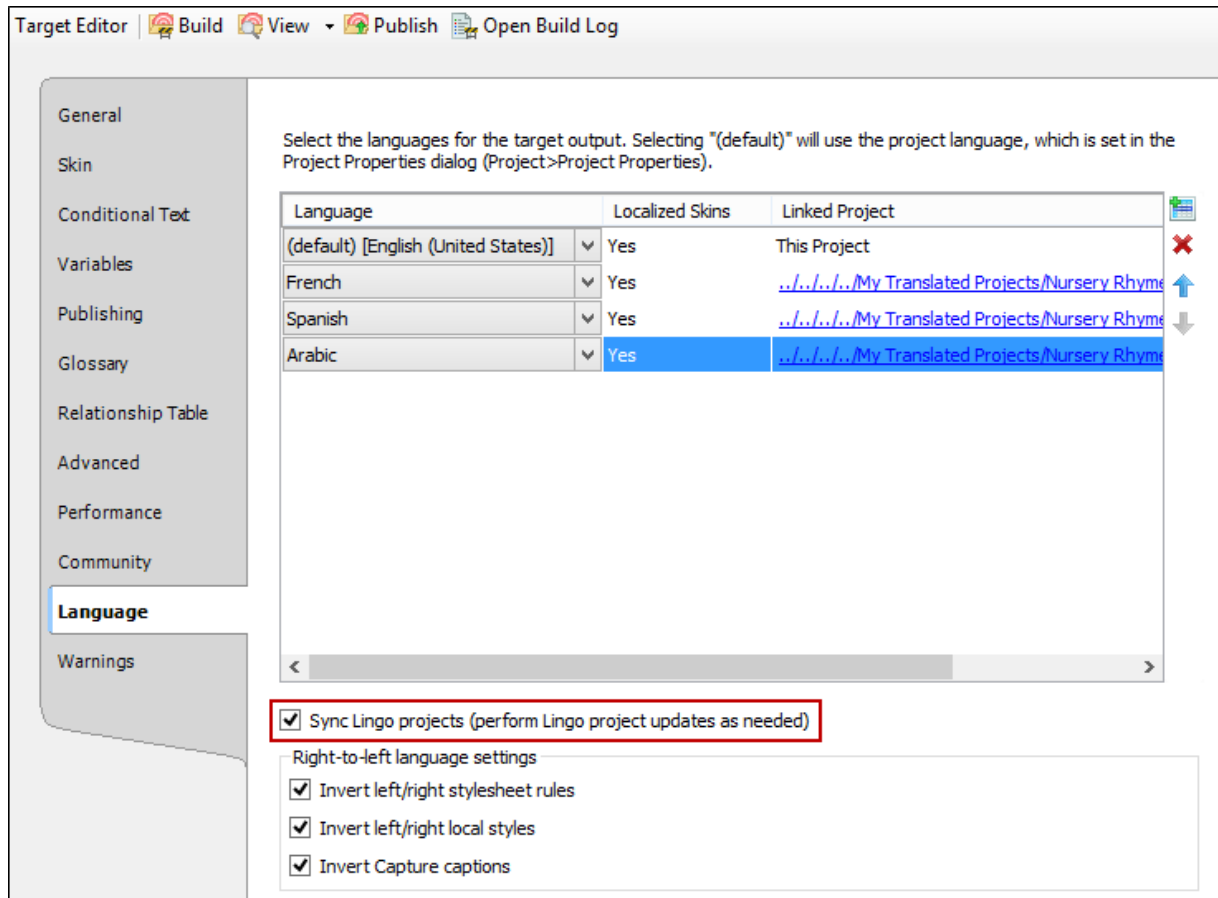
Note: In order to build output that links directly to multilingual Lingo projects, the Flare user must have Lingo 10 installed on the same computer.



Note: When you build a target that is set up for multilingual output and you link directly to a Lingo project, the export process runs automatically in Lingo so that the master Flare project can grab the necessary translated Flare projects. If the Lingo export process encounters warnings, these will not display with the other build warnings in Flare's interface. Instead, you must open the build log to find any such warnings.

Sync to Lingo Project Updates

On the Language tab of the Target Editor, a check box has been added that lets you synchronize with Lingo projects.



If this option is enabled, the application detects whether any of the Lingo and Flare source files are out of sync. If they are, the Lingo project is automatically updated and these changes are also brought into the master Flare project. This is different from the usual process, where the translator would normally update the Lingo project manually and translate the changed or new files.



Warning: You might use this feature if you want to quickly see any updated files in your master Flare project, including non-translated content such as images. However, enabling this option is typically not recommended, because there is always the risk of updating the Lingo project (and therefore also the master Flare project) with content that has not yet been translated.

APPENDIX

PDFs

The following PDFs are available for download from the online Help.

TUTORIALS

Getting Started Tutorial

Product Foldout Tutorial

Top Navigation Tutorial

Tripane and PDF Tutorial

USER GUIDES

Accessibility Guide

Analyzer Guide

Autonumbers Guide

Condition Tags Guide

Context-sensitive Help Guide

DotNet Help Guide

Eclipse Help Guide

Getting Started Guide

Global Project Linking Guide

HTML Help Guide

HTML5 Guide

Images Guide

Importing Guide

Index Guide

Key Features Guide

Language Support Guide



Movies Guide

Navigation Links Guide

Print-based Output Guide

Project Creation Guide

Pulse Guide

QR Codes Guide

Reports Guide

Reviews & Contributions Guide

Search Guide

SharePoint Guide

Skins Guide

Snippets Guide

Source Control Guide: Git

Source Control Guide: Perforce

Source Control Guide: Subversion

Source Control Guide: Team Foundation Server

Source Control Guide: Visual SourceSafe

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

Transition From FrameMaker Guide

Variables Guide

WebHelp Outputs Guide

What's New Guide

CHEAT SHEETS

Folders and Files Cheat Sheet

Print Output Cheat Sheet

Shortcuts Cheat Sheet

Structure Bars Cheat Sheet

Styles Cheat Sheet