

**MADCAP FLARE 2024 r2**

# HTML5

Copyright © 2024 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  
1660 17th Street, Suite 201  
Denver, Colorado 80202  
858-320-0387  
[www.madcapsoftware.com](http://www.madcapsoftware.com)

**THIS PDF WAS CREATED USING MADCAP FLARE.**

# CONTENTS

---

## CHAPTER 1

Introduction .....	7
--------------------	---

## CHAPTER 2

General Information for HTML5 .....	8
HTML5 Key Features .....	9
HTML5 Skin Option Comparison .....	37

## CHAPTER 3

HTML5 Side Navigation Output .....	39
Side Navigation Skin .....	41
Options for Creating Side Navigation Output .....	42
Before You Begin .....	43
Producing Side Navigation Output .....	45

## CHAPTER 4

HTML5 Top Navigation Output .....	64
Top Navigation Skin .....	66
Options for Creating Top Navigation Output .....	67
Before You Begin .....	69
Producing Top Navigation Output .....	83

## **CHAPTER 5**

HTML5 Tripane Output .....	107
Tripane Skins .....	108
Setting Tripane Output .....	110
Setting Responsive Options for Tripane Output .....	111

## **CHAPTER 6**

HTML5 Skinless Output .....	112
How To Set Skinless Output in the Target .....	114
How to Set Skinless Output in the Skin .....	115

## **CHAPTER 7**

Skin Components and Proxies .....	117
Adding Skin Components .....	118
Inserting Proxies .....	123
Working With Skin Components and Proxies .....	124
What's Noteworthy? .....	149

## **CHAPTER 8**

Responsive Skins .....	150
Examples .....	151
Responsive Skins and Mediums .....	155
How to Enable Responsive Tripane Skins .....	158
Device Width Media Queries .....	159

## **CHAPTER 9**

Responsive Content .....	165
Responsive Conditions .....	166
Responsive Layouts .....	179
Responsive Features Per Output Type .....	222



## **CHAPTER 10**

HTML5 Server-Based Output .....	223
Enabling HTML5 Server-Based Output .....	224
HTML5 Server-Based Output Scenario .....	241
Viewing HTML5 Server-Based Output .....	246

## **APPENDIX A**

PDFs .....	250
Tutorials .....	250
Cheat Sheets .....	251
User Guides .....	252

## **APPENDIX B**

Output Type Comparison Tables .....	253
Accessibility Features Per Output Type .....	255
Context-Sensitive Help Features Per Output Type .....	257
eLearning Content Features Per Output Type .....	258
Generated Content Features Per Output Type .....	260
Glossary Features Per Output Type .....	264
Image Features Per Output Type .....	265
Index Features Per Output Type .....	266
Link Features Per Output Type .....	268
List Features Per Output Type .....	270
MadCap Central Features Per Output Type .....	271
Merging Features Per Output Type .....	272
Micro Content Features Per Output Type .....	273
Miscellaneous Features Per Output Type .....	275
Multimedia Features Per Output Type .....	278
Responsive Features Per Output Type .....	283
Search Features Per Output Type .....	284

Skin Features Per Output Type .....	291
TOC and Mini-TOC Features Per Output Type .....	297
Translation and Localization Features Per Output Type ...	299
Variable Features Per Output Type .....	301

## CHAPTER 1

---

# Introduction

HTML5 is the recommended online format in Flare. This output type supports the HTML5 specification developed by the World Wide Web Consortium (<http://w3.org>) and the Web Hypertext Application Technology Working Group (<http://whatwg.org>). Therefore, the HTML5 format results in better markup and offers additional features not found in other output formats.

### General Information


- "HTML5 Key Features" on page 9
- "HTML5 Skin Option Comparison" on page 37

### Output Formats

- "HTML5 Side Navigation Output" on page 39
- "HTML5 Top Navigation Output" on page 64
- "HTML5 Tripane Output" on page 107
- "HTML5 Skinless Output" on page 112

### Other Information

- "Skin Components and Proxies" on page 117
- "HTML5 Server-Based Output" on page 223

 **NOTE** The following browsers support HTML5: Firefox 10 or later, Google Chrome 13 or later, and other browsers that support the HTML5 standard.

**CHAPTER 2**

---

# General Information for HTML5

There are various pieces of general information you should know if you plan to use this feature.

This chapter discusses the following:

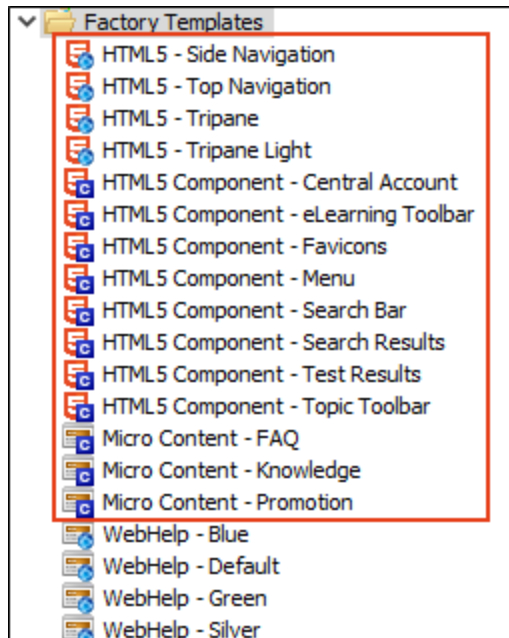
- HTML5 Key Features ..... 9
- HTML5 Skin Option Comparison .....37

# HTML5 Key Features

Following are some of the key features of the HTML5 format.

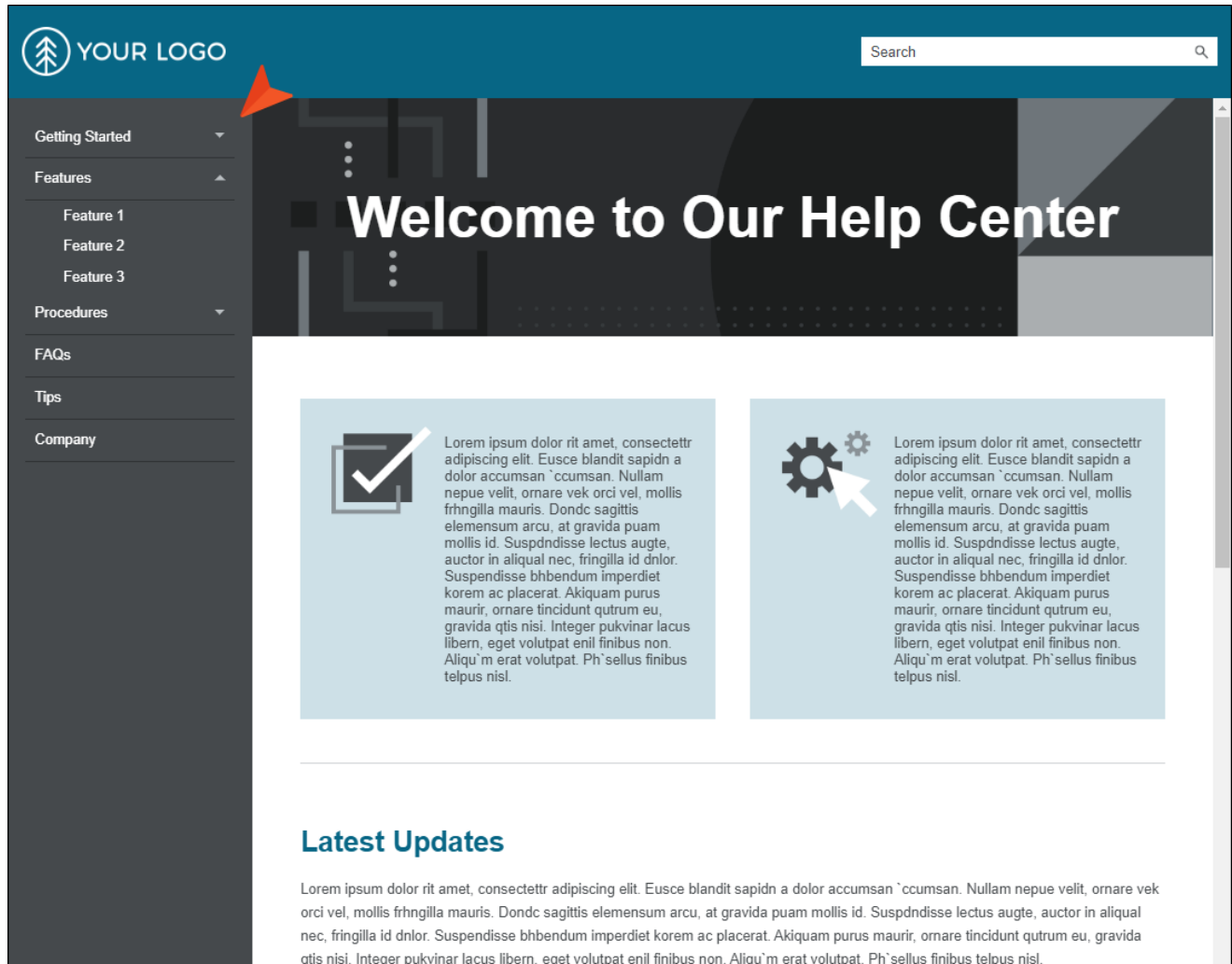
## Skins—Multiple Options

Because HTML5 output is quite different from the other online formats, it requires its own skin. In fact, you have more than one skin from which to choose.



# Side Navigation Skin

One popular option is the Side Navigation skin, in which the main navigation is positioned at the left or right. This lets you use a flexible, frameless output while continuing to have a more traditional configuration that looks like the Tripane format.



# Welcome to Our Help Center






Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eiusmod blandit sapien a dolor accumsan `ccumsan. Nullam nepue velit, ornare vek orci vel, mollis fringilla mauris. Dondc sagittis elemensum arcu, at gravida puam mollis id. Suspdndisse lectus augte, auctor in aliquat nec, fringilla id dnlor. Suspendisse bhhendum imperdiet korem ac placerat. Akiquam purus maurir, ornare tincidunt qutrum eu, gravida qtis nisi. Integer pukvinar lacus libern, eget volutpat enil finibus non. Aliqu`m erat volutpat. Ph`sellus finibus telpus nisi.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eiusmod blandit sapien a dolor accumsan `ccumsan. Nullam nepue velit, ornare vek orci vel, mollis fringilla mauris. Dondc sagittis elemensum arcu, at gravida puam mollis id. Suspdndisse lectus augte, auctor in aliquat nec, fringilla id dnlor. Suspendisse bhhendum imperdiet korem ac placerat. Akiquam purus maurir, ornare tincidunt qutrum eu, gravida qtis nisi. Integer pukvinar lacus libern, eget volutpat enil finibus non. Aliqu`m erat volutpat. Ph`sellus finibus telpus nisi.

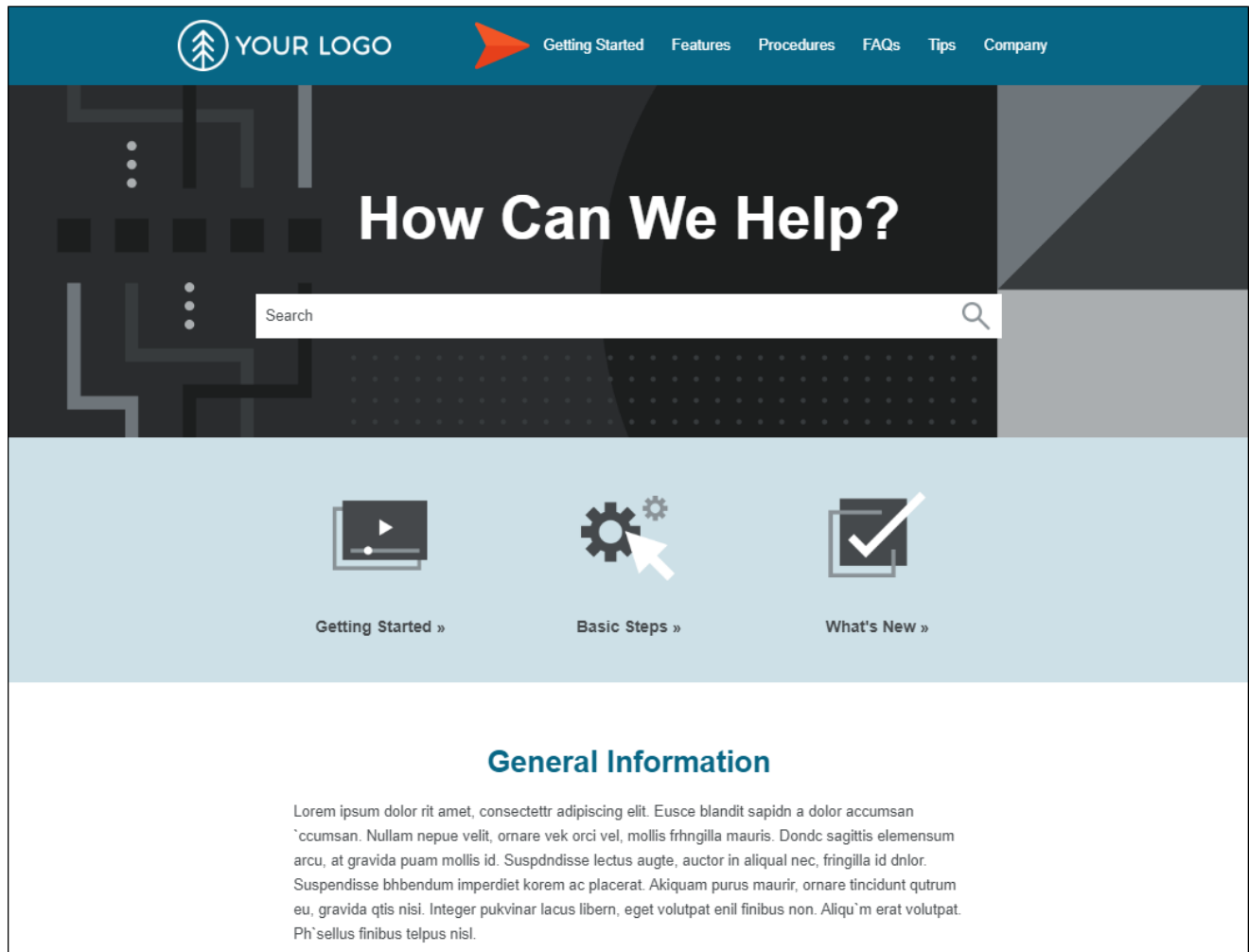
## Latest Updates

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eiusmod blandit sapien a dolor accumsan `ccumsan. Nullam nepue velit, ornare vek orci vel, mollis fringilla mauris. Dondc sagittis elemensum arcu, at gravida puam mollis id. Suspdndisse lectus augte, auctor in aliquat nec, fringilla id dnlor. Suspendisse bhhendum imperdiet korem ac placerat. Akiquam purus maurir, ornare tincidunt qutrum eu, gravida qtis nisi. Integer pukvinar lacus libern, eget volutpat enil finibus non. Aliqu`m erat volutpat. Ph`sellus finibus telpus nisi.

- Getting Started 
- Features 
- Procedures 
- FAQs
- Tips
- Company

# Top Navigation Skin

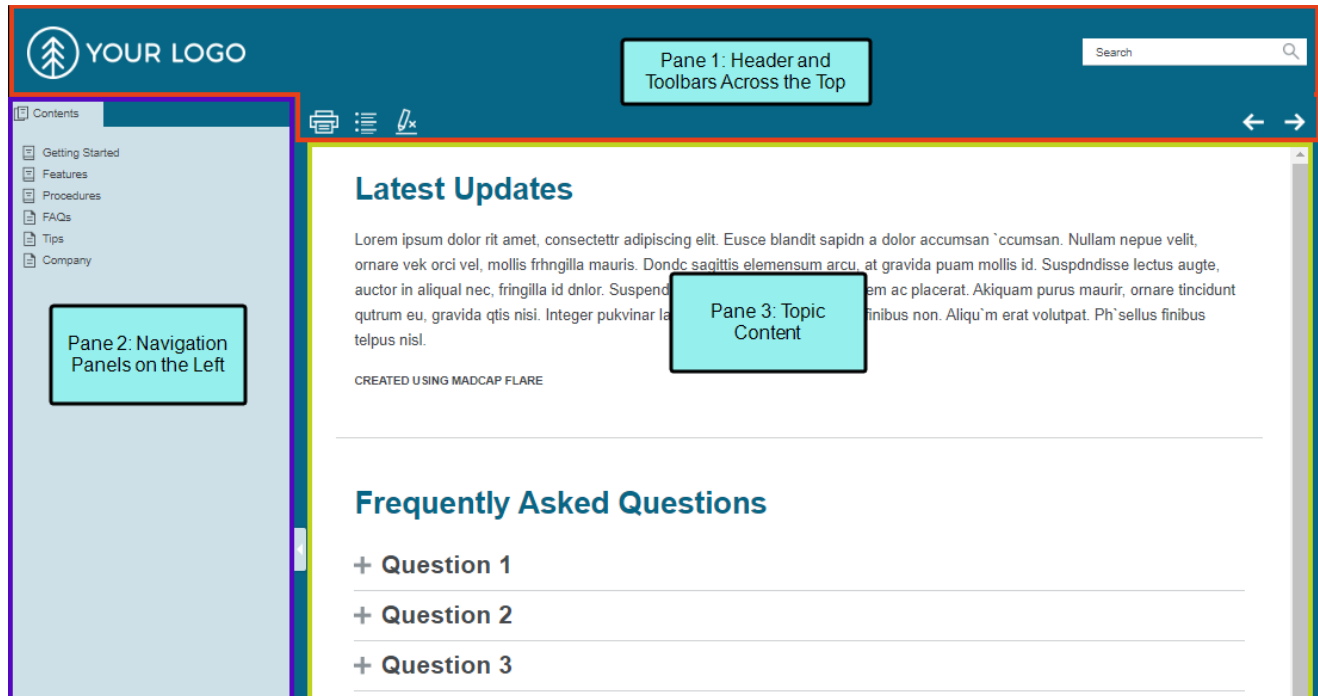
Then there is the Top Navigation skin. This lets you generate a flexible, frameless output like a modern website, with the main navigation items at the top of the screen.





# Tripane Skin

Another alternative is to use a Tripane skin, which lets you generate output in a traditional format with three distinct frames (a navigation pane on the left, a toolbar and search on the top, and the main body pane).



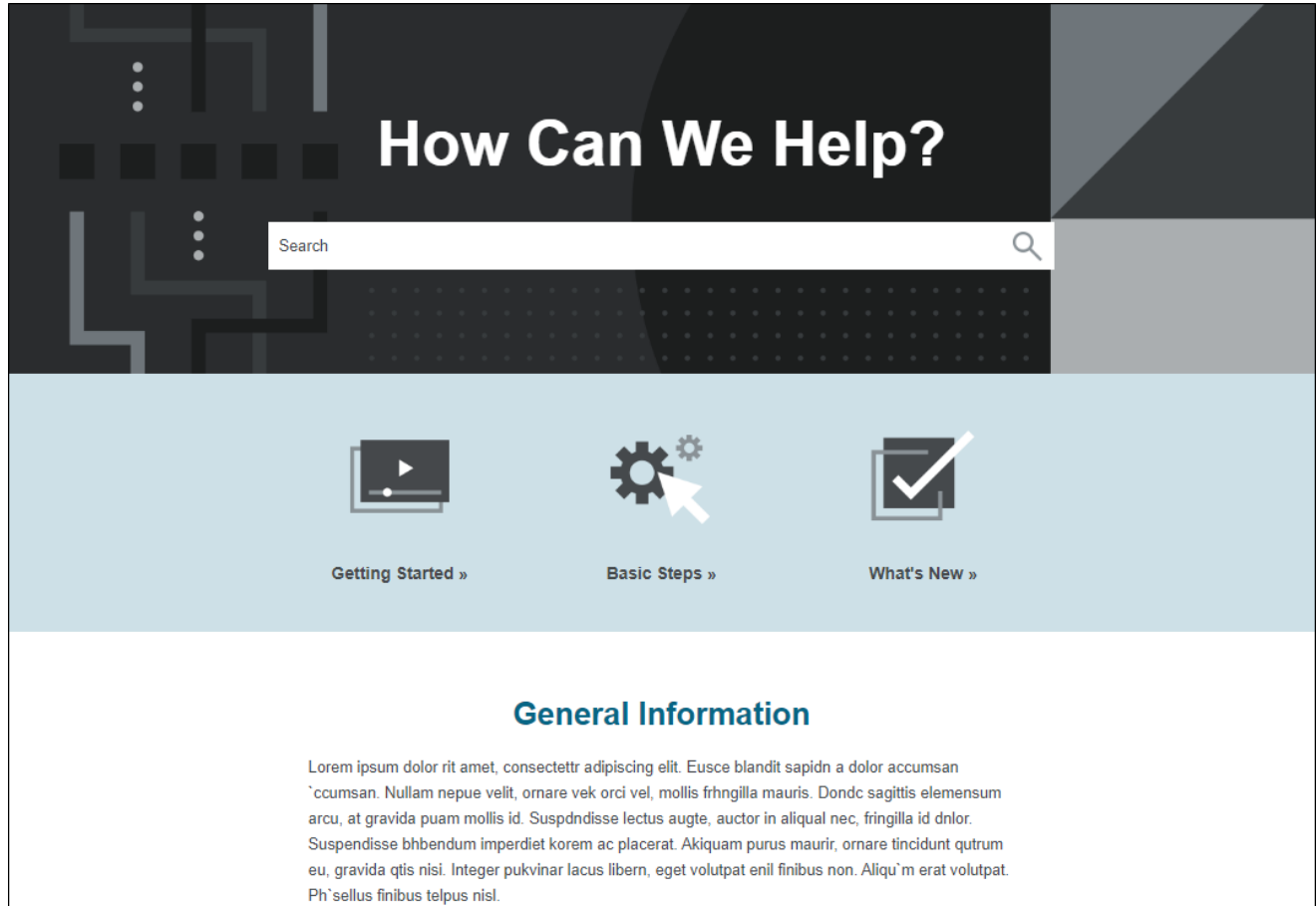
# Differences Between HTML5 Skins

Aside from the obvious structural characteristics, HTML5 Side Navigation and Top Navigation skins are different from Tripane skins in the following ways:

- **Tabs in the Skin Editor** When you open a Side Navigation or Top Navigation skin, you will notice that they do not have the General, Size, or Toolbar tabs. Those tabs contain fields and features that are pertinent only to Tripane output. For example, with Side or Top Navigation output you do not select navigation elements to include (e.g., TOC, glossary, index), because those types of elements are more prominent in Tripane output. Instead, Side or Top Navigation output puts a premium on search and menu items instead. (Menus in Side or Top Navigation output are populated from your TOC file.)
- **Responsive Output** You do not need to turn responsive output on or off in Side or Top Navigation, because it is always enabled for those skins, which is not the case for Tripane skins. However, there are some additional settings you can provide for responsive output on the Skin tab of the Target Editor.
- **Different Styles** Because the Side and Top Navigation skins have fewer elements in them, there are fewer fields in the Styles tab of the Skin Editor than there are for Tripane skins. Also, there are some styles for menus that are unique in each type of skin. For more information see the online Help.
- **UI Text Fields** With fewer elements involved in Side and Top Navigation, there are fewer fields to be concerned about in the UI Text tab. Tripane skins have more UI Text fields. For more information see the online Help.
- **Project Merging** The Side and Top Navigation skins do not support project merging, whereas Tripane does.

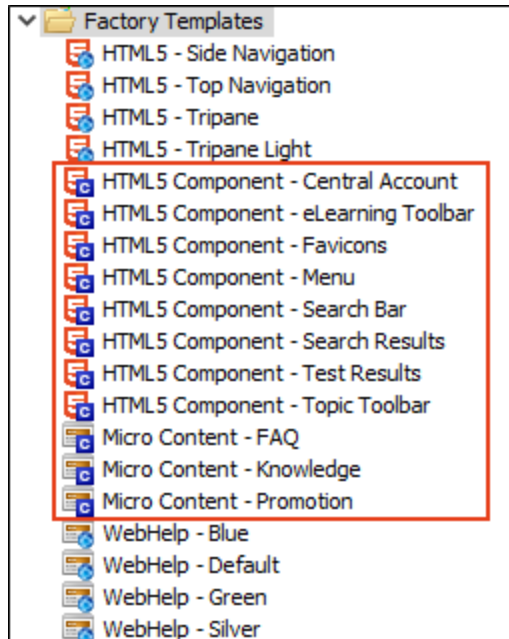
# Skinless Output


You can even tell Flare not to use any skin at all for HTML5 output. In this situation, you would rely on proxies or links inserted directly into your content, or you would have no links or navigation at all. See "HTML5 Skinless Output" on page 112.



# Skin Components and Proxies

You can also use smaller skin components that work with related proxies. With these, you can add and design menus and toolbars, inserting them in your content anywhere you like. For Side Navigation, Top Navigation, and skinless outputs, you can also create search bars and customized results pages. In addition, there are skin components to work with micro content containers. See "Skin Components and Proxies" on page 117.



 **NOTE** If you want to translate interface elements in an HTML5 skin, you can do this using a language skin or using the UI Text tab located in the Skin Editor. This tab lets you perform all of the localization tasks that you would normally accomplish in a language skin. Be aware that translations in the Skin Editor are prioritized over translations in a language skin, so you should try to perform your localization tasks in one place to avoid conflicts.

# Frameless

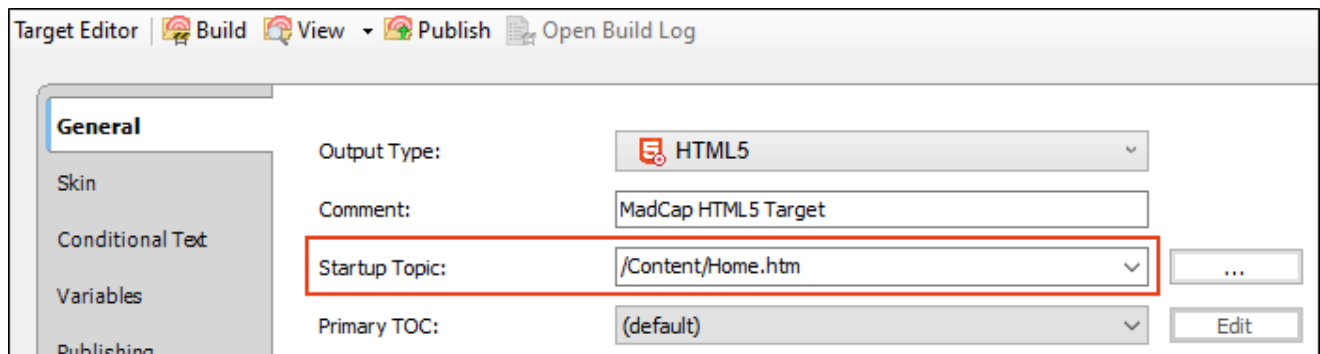
Pages in HTML5 output do not use <frameset> or <frame> tags, like they do in regular WebHelp output. In addition, HTML5 Side and Top Navigation outputs are completely frameless, meaning they also do not contain iframes, which offers the following benefits:

- **Better Search Engine Optimization** Side or Top Navigation means better search engine optimization (SEO). This is thanks in part to the absence of iframes. In addition, the output is not dynamically loaded in div tags, but rather the content is flattened, which makes it easier for web crawlers to locate.
- **Navigation Displays with External Search Results** Side or Top Navigation provides a better experience with external searches and navigation. For example, if you have Tripane output and click on a Google search result for a specific page, that page opens without the surrounding navigation (e.g., TOC) included in that Help system. But with Side or Top Navigation output, that same page would display with its intended navigation.
- **Improved Scrolling and Zoom in Mobile Devices** When Side or Top Navigation output is viewed on a mobile device, scrolling and zoom features are typically better than they are for Tripane output.

# Home Topic

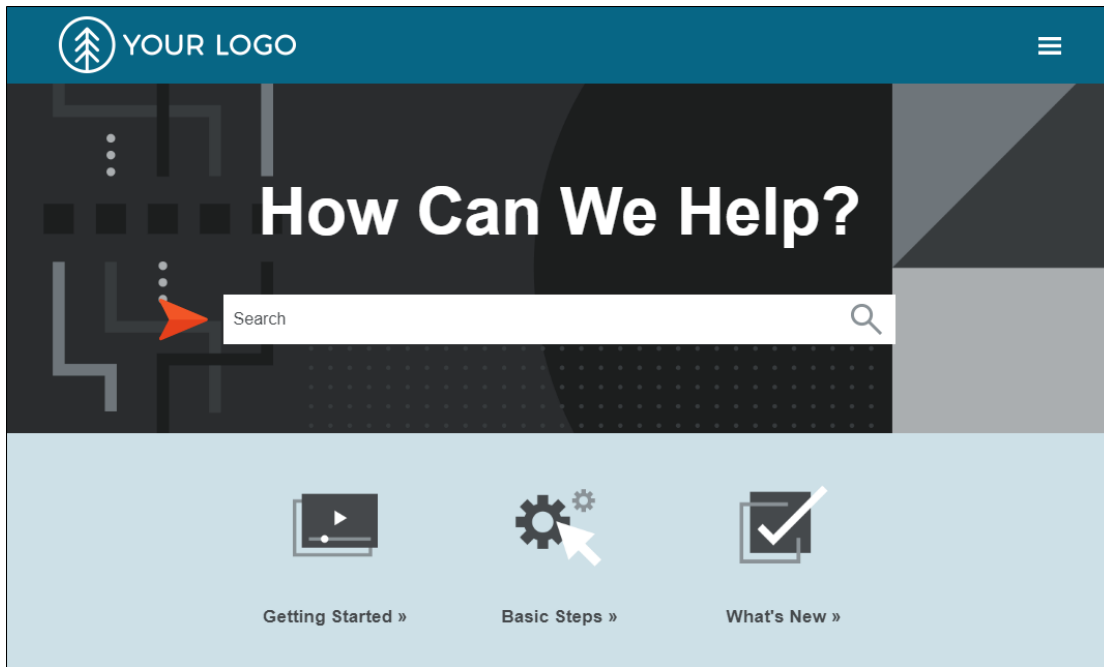
A Home topic is often used in HTML5 outputs. However, for Side and Top Navigation outputs, you usually do not add this topic to your TOC (whereas you do add it to the TOC when creating the older Tripane output). Instead, it is standard practice to link to this page from the logo you provide in your Side or Top Navigation skin.

A Home topic is the one that you specify as the startup topic in the Target Editor.



The Home page is just a regular topic, but because a Side and Top Navigation skins are designed to resemble a modern website, you may want this topic to stand out with a different appearance. You might accomplish this in the following ways, all of which are optional:

- **Unique Search Bar** The Side and Top Navigation skins are designed to show a search bar at the top of all topics. However, you might want to emphasize the search bar on your Home page, especially since search is the most popular and effective way to find specific information.



To do this, you can insert a Search Bar proxy in a more prominent place in the Home topic. See "Skin Components and Proxies" on page 117.

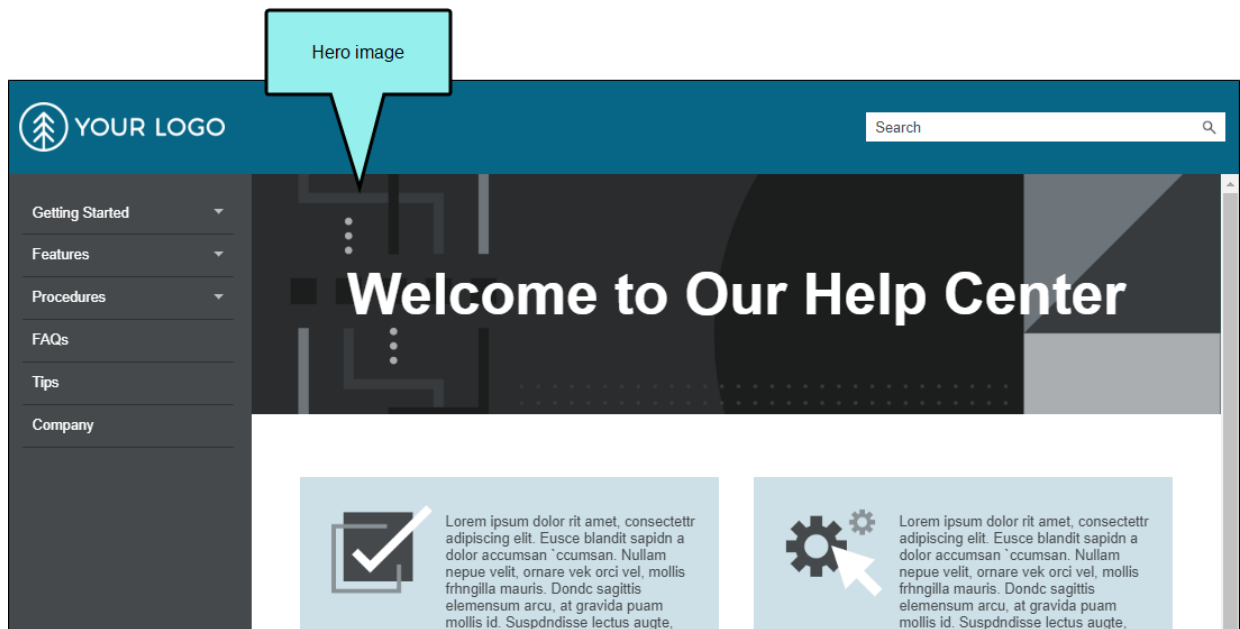
Create a separate stylesheet for the Home topic and open it in the Internal Text Editor. Then, copy and paste the following to hide the top search bar that is included with the skin.

```
.nav-search
{
  display: none;
}
```

You also need to associate the new stylesheet with the Home topic.

For more details, see the online Help.

- **Unique Hero Image** One easy way to make a Home page stand out from the other topics is to place a background image (or watermark) on it.



In the world of web design, this is often referred to as a “hero image.” You can set a background image in a stylesheet. It is recommended that you use a hero image that is quite wide. In the Flare Top Navigation project templates, this image has a width of 1903 pixels. In Flare’s Side Navigation project templates, this image is 3456 pixels wide. A big hero image helps to account for even the largest of monitors.

- **Unique Template Page** By using one template page for your Home topic and another for the rest of your topics, it is easier to give them different looks, as well as different headers and footers.



# Responsive Skins

Responsive web design (RWD) is a way to construct your HTML5 output so that the display and content are adjusted automatically depending on the device. herefore, on tablets and mobile devices such as smart phones, users will see a condensed look nd perhaps different text that is more appropriate for those devices, compared with larger monitors. You can get the same effect if you shrink your browser to a smaller size. Responsiveness is automatically built in to HTML5 skins. See "Responsive Skins" on page 150.

# Responsive Conditions

For HTML5 targets, conditions can be used to make content responsive (e.g., content changes automatically according to the size of the screen, rather than simply based on "include" or "exclude"). Associating conditions with factory media queries (Web, Tablet, Mobile) can help make some content more appropriate or easier to read on different devices. See "Responsive Conditions" on page 166.

# Responsive Layouts

Styles and media queries can be used to make your content responsive in HTML5 output. This lets you present information—both its substance and structure—differently depending on the size of the screen or device. Flare provides a Responsive Layout window pane that helps you create this kind of content more easily. You also have the option of using third-party solutions (e.g., Zurb Foundation grid system) by adding the appropriate styles in your stylesheets and topics. See "Responsive Layouts" on page 179.

# Specific Page Addresses

With the regular WebHelp output, you can only give readers a single URL path, which opens the starting page for the output. With HTML5 output, you can point end users to a specific page in the output by giving them the exact URL path. Because HTML5 output also allows for pagination in search results, you can also direct users to specific page addresses in search results.

# Choosing a Search Engine

For HTML5 targets, you can choose the type of search engine you want people to use—MadCap Search, Google Search, or Elasticsearch (for Side Navigation, Top Navigation, or skinless output). There are additional steps that you can follow and features you can select, depending on the search engine you choose. For MadCap Search and Elasticsearch, you can include micro content in the output, which can especially enhance your search results. For more information see the online Help.

There are benefits to any of the search methods offered by Flare. Some users might prefer the search results returned by Google. However, others might prefer the results returned by MadCap. Or some users might want all of the configurable options provided by Elasticsearch.

✔ **TIP** To determine the type of search that is best for you, we recommend that you generate different outputs using each method. Then perform several different searches and see which results you like best.

	MadCap Search	Elasticsearch	Google Search
<b>Less Time and Effort to Set Up</b>	✔ Setting up MadCap Search in a Flare project takes only a few seconds.	Setting up Elasticsearch is done within the Target Editor and Destination Editor in Flare. Web servers also need to be configured to support Flare, along with the Elasticsearch service.	Setting up a Google custom search engine (CSE) and performing other necessary tasks requires a combination of steps in both Google and Flare. This can take a bit of time to complete. However, it is usually just a one-time setup.

	MadCap Search	Elasticsearch	Google Search
<b>Public vs. Local Output Flexibility</b>	✓ Output can be searched whether it is available publicly or locally.	Output must be published on a server in order to use the full capability of Elasticsearch.	Output must be available publicly so that you can use Google. Output cannot be behind a firewall.
<b>Multiple Outputs</b>	Searches will include results for only the Flare output that is open.	✓ You can manage multiple HTML5 targets that use Elasticsearch through the Content Services Portal.	✓ You can add multiple folders/sites to be included in your Google custom search engine (CSE). Each of these sites can be a separately published Flare output. When users search from one of the published Flare outputs, results will display for all of the outputs.
<b>Other File Types</b>	In order to include file types such as PDF in your search results, you need to generate and publish server-based output.	✓ Most file types are included in the search results.	✓ If you have added other types of files (e.g., PDF) to the site (s) associated with your CSE, they may be included in search results.

	MadCap Search	Elasticsearch	Google Search
<b>Search Tailored to Flare Projects</b>	<p>✔ MadCap Search rankings are specifically tailored for Flare users. They take into account not only H1-H6 headings, but also index keywords, glossary terms, etc.</p>	<p>Elasticsearch does not provide any direct mapping of search criteria to the structure of a Flare project.</p>	<p>Google does not recognize the unique characteristics of a Flare project (e.g., index keywords, glossaries). Therefore, those elements will have no effect on how Google ranks pages in the results.</p>
<b>Micro Content in Results</b>	<p>✔ MadCap Search lets you include micro content in your search results, which can especially improve the experience for users.</p>	<p>✔ Elasticsearch lets you include micro content in your search results, which can especially improve the experience for users.</p>	<p>Google does not include micro content (created in Flare) in your search results.</p>
<b>Glossary Definitions in Results</b>	<p>✔ MadCap Search will fetch your project's glossary terms and feature them at the top of search results.</p>	<p>Elasticsearch does not provide an option to include glossary terms in your search results.</p>	<p>Google will not fetch your project's glossary terms and feature them at the top of search results.</p>

	MadCap Search	Elasticsearch	Google Search
<b>Results for Incorrectly Typed Searches</b>	Incorrectly typed search terms will not return results.	✔ The partial-word search and fuzzy match settings allow for numerous searches to be returned when these settings are enabled.	✔ If an end user incorrectly enters a search term (e.g., types “Tble” instead of “Table”), Google will often guess the correct term and show results anyway.
<b>Better Search and Indexing Speed</b>	✔ Search results might be faster because the output is immediately indexed once it is generated. You do not need to spend any extra time indexing your pages or waiting for search results to appear.	✔ The administrator portal allows for the search indexes to be managed closely. This will assist in keeping search result times optimal.	Search results might lag at times, depending on the size of the output and whether Google is finished with the indexing (i.e., making your output available for searches).  Google will index your pages automatically over time, but it can take awhile, sometimes weeks.

	MadCap Search	Elasticsearch	Google Search
<b>Mobile Devices</b>	Search results are “chunked” and then collapsed so that they can be displayed when a search is performed. This is usually okay on PCs, but if you try to perform a search on a mobile device, the results might be slow or even crash because the device cannot handle the number of search results.	✔ Search results always come from a server, so you do not need to be concerned about the ability of the mobile device to handle the information.	✔ Search results always come from a server, so you do not need to be concerned about the ability of the mobile device to handle the information.
<b>No Advertisements</b>	✔ No advertisements are included in search results.	✔ No advertisements are included in search results.	Users might see advertisements in search results.
<b>Customize Auto-Complete Results</b>	There are no options available for auto-complete results.	✔ You can enable multiple options to include a mix of personal and global searches on your output. There is also an option to include project content information such as topic titles and the file path in your project.	While Google Search does provide auto-complete results, it does not provide the ability to customize these results.

	MadCap Search	Elasticsearch	Google Search
--	---------------	---------------	---------------

**Customize  
Predictive Search**

MadCap Search does not offer any predictive methods in obtaining search results.

✔ There are multiple options that allow you to obtain predictive results when searching, especially when using auto-complete options.

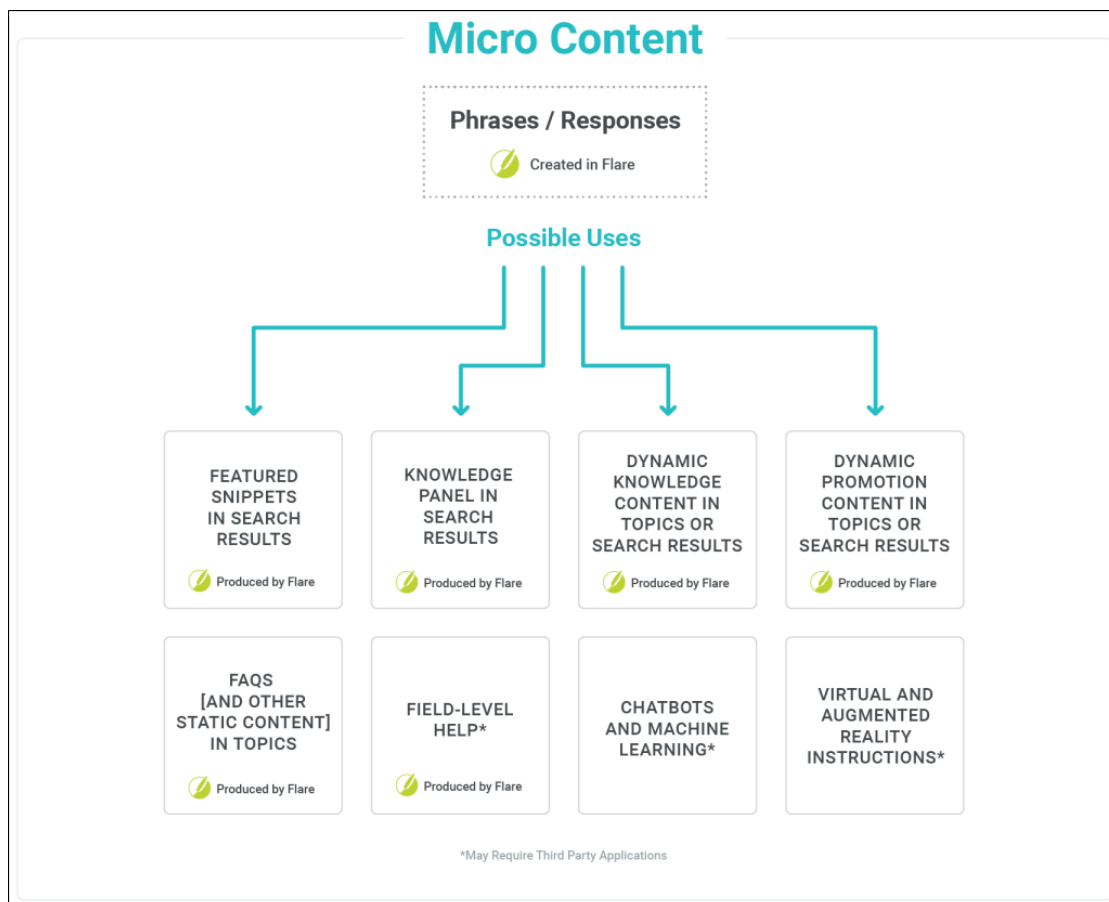
Google Search provides a measure of predictive results, although it is not customizable within Flare.

# Sitemap/Search Engine Optimization

For web-based targets, you can generate a sitemap when compiling your output. This helps with search engine optimization (SEO), making it easier for search indexing services (i.e., spiders, crawlers, or bots) to find your output. Therefore, the entire output is indexed and search engine results are improved.

## Micro Content

Micro content is short, concise information that stands alone and is easily consumable. In Flare, it begins with the creation of a collection of brief *phrases* and corresponding *responses*, such as questions and answers. After generating HTML5 output, these phrase/response combinations can be used in different ways as users interact with your output. For example, it can greatly improve the user experience when people search in your HTML5 output. It can also be used to create field-level context-sensitive Help. For more information see the online Help.





Your search for "csh" returned 69 result(s).

## Video



[Context-Sensitive Help](#)

April 9, 2021

39:27



## CSH Calls for HTML5 Output

Use the following information if you are producing HTML5 and want to incorporate context-sensitive Help (CSH) into the software application. There are two methods you can use: Method 1—JavaScript Using this method requires calling a JavaScript function that Flare provides. Method 2—URL Using this...

Method 2—URL Using this...

[Flare/CSH/Process/CSH-Calls/CSH-Calls-HTML5-Developers.htm](#)

## Excluding JavaScript for CSH Calls in HTML5 Output

If you are using context-sensitive help (CSH) in your HTML5 output, you can either use JavaScript or a URL to open the output. See [CSH Calls for HTML5 Output](#) . In the Target Editor for HTML5 targets, there is an option on the Advanced tab to include JavaScript for CSH calls. This option is enabled...

## Context-Sensitive Help

Context-sensitive Help (CSH) is a way to tie your topics or micro content with links or specific areas of a software interface. For example, when users click a particular part of the interface (e.g., a Help button), the topic or micro content pertaining to it opens.

> [General Information](#)

> [Process \(If Author Provides Header File\)](#)

> [Process \(If Developer Provides Header File\)](#)

[Context-Sensitive Help](#)

[Flare/CSH/CSH.htm#microcontent1](#)

# eLearning

It is possible to generate HTML5 output with eLearning components for learning and development (L&D) courses. You can use Flare to leverage single-sourced content from a project for creating interactive knowledge checks and gradable quizzes. It provides the capability to design courses with linear workflows, form questions, and build online output with learning management system (LMS) standards for sharing. Use the Target Editor to set output options for eLearning material.

## Question 2

Which are not official slogans for Austin? (Select all answers that apply.)

- "Center for Urban Bat Colonies"
- "The Live Music Capital of the World"
- "Best Barbecue in the World"

← Previous

Complete →

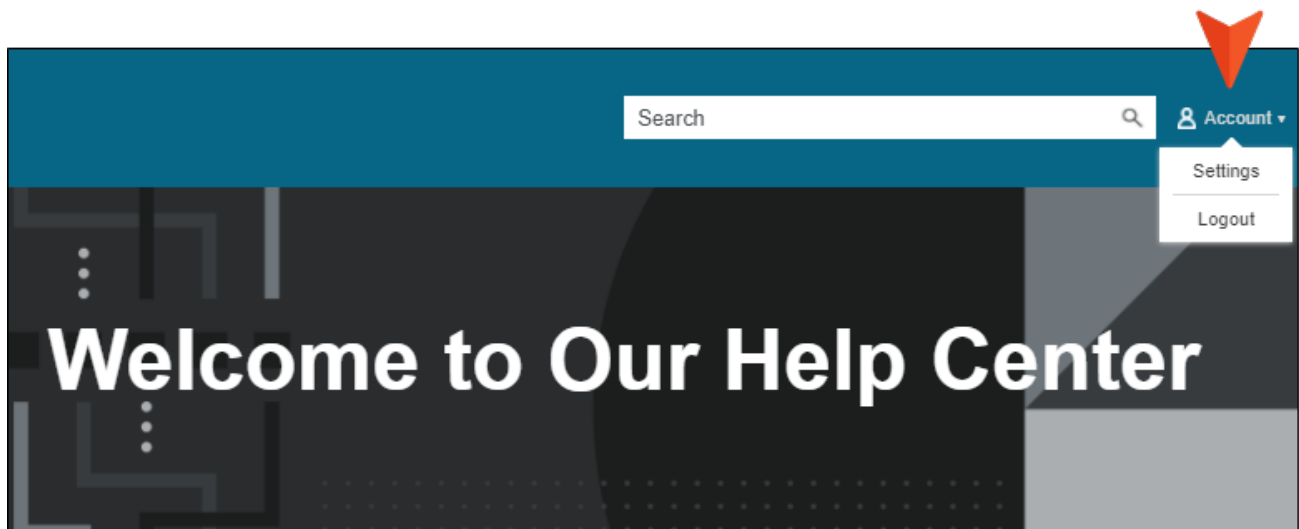
Page 9 of 9

# Private Output

You can produce private output that requires a user to log in with an email and Central password. Without credentials, people cannot see the output. It is possible to set private output for all output types supported by Central. Also, accessing private output is not limited to people in your company. Central has a free viewer user type that you can use to add any of your customers to your Central license, and then give them access to live private output.

After users log in to see private output, an account link displays in the output interface for HTML5 targets (not for other output types). If the output is not set to private, this link is not shown. When users hover over the link, a menu shows two options.

For more details about making output private, see the Central online Help.



# Analytics

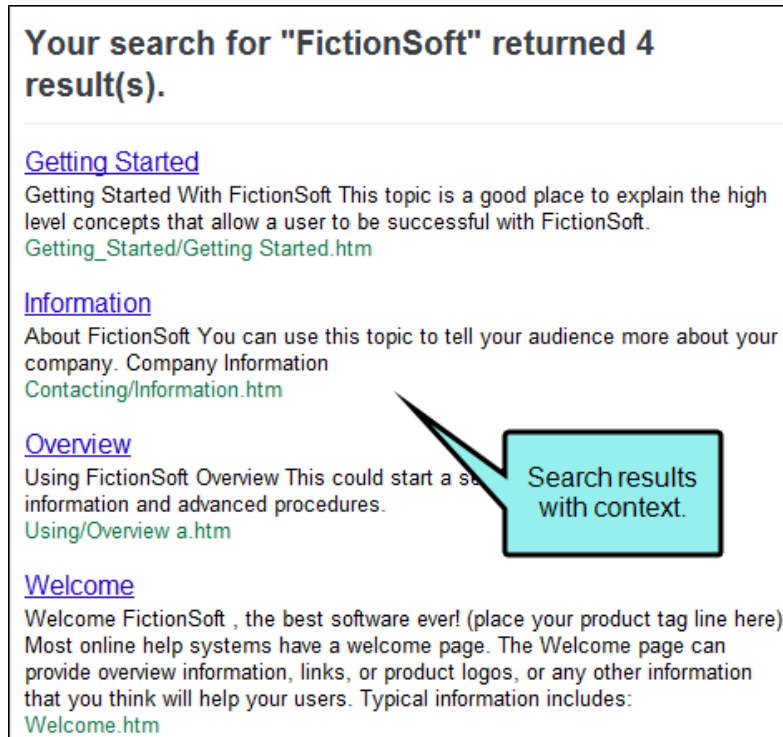
If you have a MadCap Central license, you can view analytics on published Flare HTML5 output. This includes search phrases used, search phrases with no results, topics viewed, context-sensitive Help calls, and demographic statistics (browsers and operating systems).



Central analytics works on Flare projects uploaded to Central, or you can host output on your own servers. If you host the output outside of Central, you still need to use Central with a key to view the analytics data. Also, the server where the output is hosted must be able to communicate with Central (e.g., not be behind a firewall).

# Search With Context

If end users perform a search in your output, they will see context next to each result, rather than terms only.



**Your search for "FictionSoft" returned 4 result(s).**

---

[Getting Started](#)  
Getting Started With FictionSoft This topic is a good place to explain the high level concepts that allow a user to be successful with FictionSoft.  
[Getting\\_Started/Getting Started.htm](#)

[Information](#)  
About FictionSoft You can use this topic to tell your audience more about your company. Company Information  
[Contacting/Information.htm](#)

[Overview](#)  
Using FictionSoft Overview This could start a section on getting started information and advanced procedures.  
[Using/Overview a.htm](#)

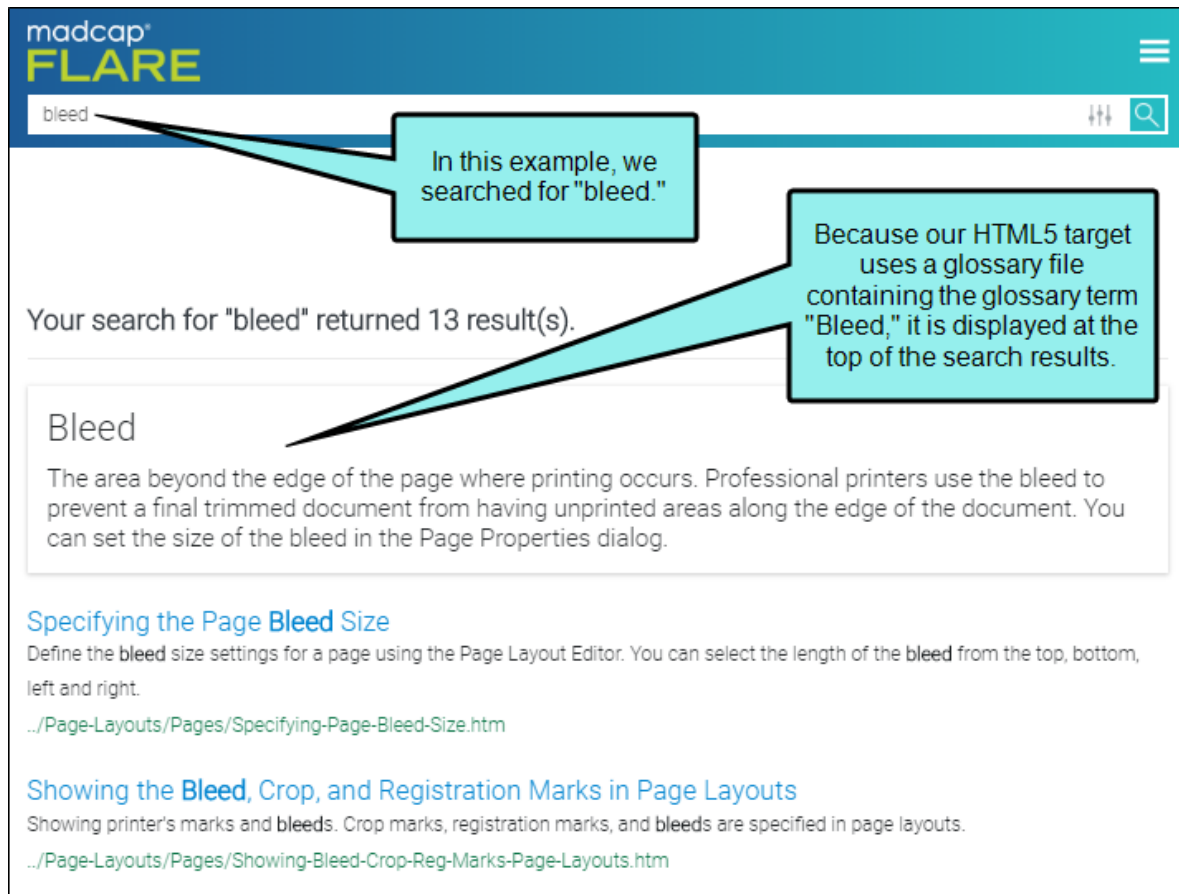
[Welcome](#)  
Welcome FictionSoft , the best software ever! (place your product tag line here)  
Most online help systems have a welcome page. The Welcome page can provide overview information, links, or product logos, or any other information that you think will help your users. Typical information includes:  
[Welcome.htm](#)

Search results with context.

In addition, you can create topic meta descriptions, which will be used in the search results abstract. Also, the meta description itself is searchable.

# Glossary Terms in Search Results

For HTML5 output, search results display glossary terms if they are used as the search text.



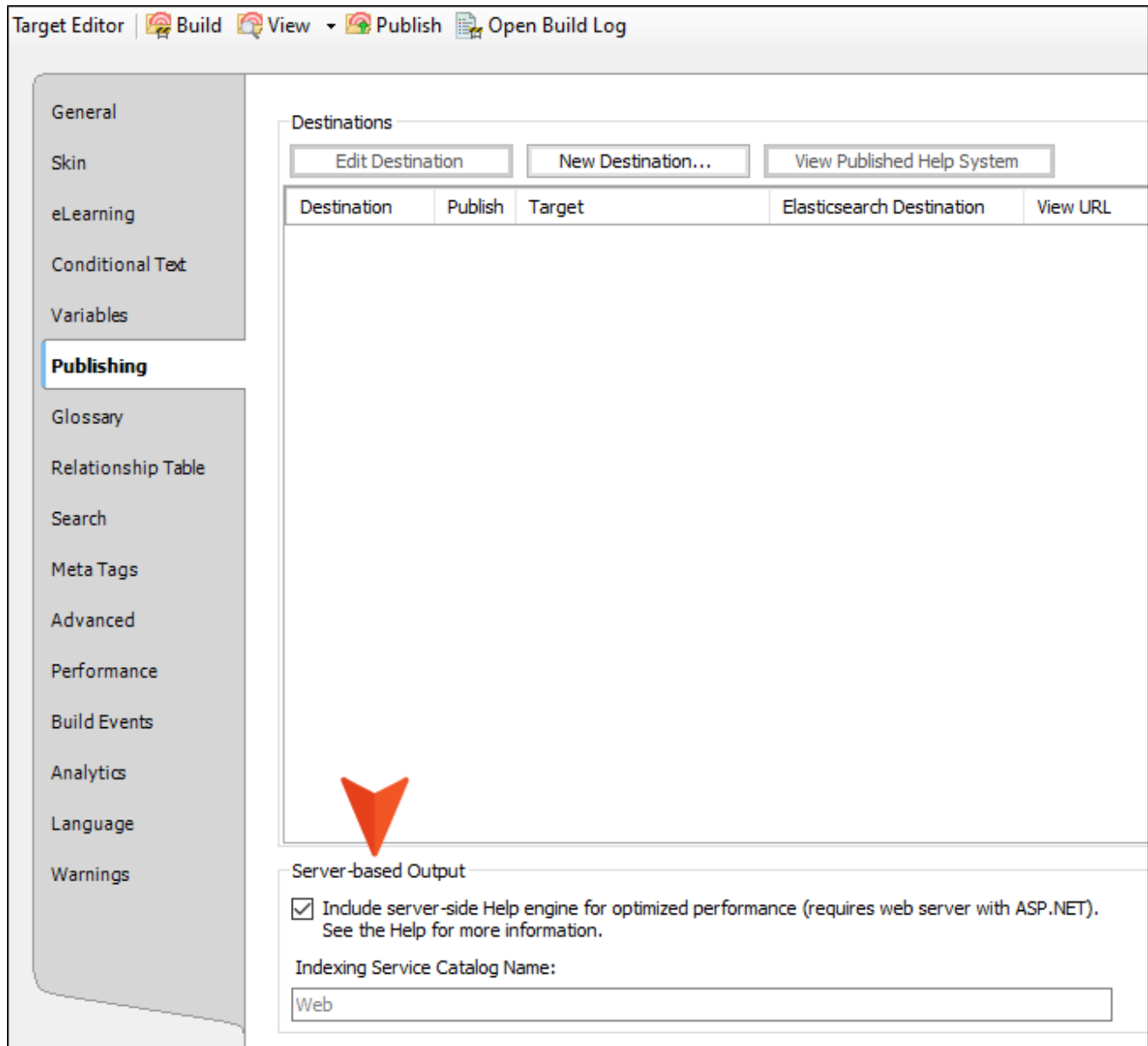
This option to display glossary terms in search results is enabled by default. However, it can be turned off in the HTML5 Target Editor.

## Smoother Dynamic Effects

Features such as drop-down text, expanding text, and popups have a smoother appearance when they are activated.

# Server-Based Output

You can create HTML5 output in its regular state, or you can select an option in the Publishing tab of the Target Editor to enable server-based functionality. This lets you accomplish the same results as WebHelp Plus output—searching of non-XHTML content, server-side search, and automatic runtime merging. See "Enabling HTML5 Server-Based Output" on page 224.



The screenshot shows the Target Editor interface with the Publishing tab selected. The left sidebar contains a list of settings categories: General, Skin, eLearning, Conditional Text, Variables, Publishing (highlighted), Glossary, Relationship Table, Search, Meta Tags, Advanced, Performance, Build Events, Analytics, Language, and Warnings. The main content area is divided into two sections. The top section, titled "Destinations", contains three buttons: "Edit Destination", "New Destination...", and "View Published Help System". Below these buttons is a table with the following headers: "Destination", "Publish", "Target", "Elasticsearch Destination", and "View URL". The table body is currently empty. The bottom section, titled "Server-based Output", contains a checked checkbox with the text "Include server-side Help engine for optimized performance (requires web server with ASP.NET). See the Help for more information." Below this is a label "Indexing Service Catalog Name:" followed by a text input field containing the word "Web". A red arrow points to the "Server-based Output" section.

Following are benefits of server-based HTML5 output:

- **Multiple Platforms** You can publish to a machine running Windows Server 2003, Windows Server 2008, or Windows 7.
- **Searching of Non-XHTML Content** When end users perform a search in your online output, you can ensure that non-XHTML files (e.g. PDF, DOC, XLS) are included in that search. When you build HTML5 output, a subfolder named "AutoSearch" is created and placed in the generated output folder. You can place non-XHTML files within the published AutoSearch subfolder (whether the non-XHTML files are linked to content from your Flare project or not). When users perform a search, those non-XHTML files will also be accessible to the users. For more information see the online Help.
- **Microsoft IIS Search Engine** Another benefit of generating and publishing HTML5 output to a web server is that it takes advantage of Microsoft's IIS search engine to help power your search. This is especially useful if you have a very large Help system.
- **Automatic Runtime Merging of Flare Projects** This is an easy way to merge the output from multiple HTML5 server-based Flare targets into one Help system. These targets can be originated from the same Flare project or from different Flare projects. You simply place the output files in the correct location on the server (i.e., within your parent project's AutoMerge folder). Flare then automatically merges the output from all of the targets when users access the Help. From the end user's perspective, the results are seamless, appearing as one large Help system. All of the TOCs, browse sequences, indexes, glossaries, and search capabilities for the projects are merged. For more information see the online Help.



# HTML5 Skin Option Comparison

Following is a comparison of the HTML5 Side Navigation, Top Navigation, and Tripane skins, as well as the skinless option.

	Side Nav	Top Nav	Tripane	Skinless
eLearning Interactivity	✓	✓	✓	✓
Frameless—No Framesets	✓	✓	✓	✓
Frameless—No iframes <sup>1</sup>	✓	✓	✗	✓
Glossary Terms in Search Results	✓	✓	✓	✓
Micro Content in Search Results	✓	✓	✓	✓
Navigation Panes—Browse Sequence, Glossary, Index, Pulse Community, TOC	✗	✗	✓	✗
Project Merging	✗	✗	✓	✗
Pulse Integration	✓	✓	✓	✓
Responsive	✓	✓	✓	✓
Responsive Skin Always Enabled	✓	✓	✗	✗
Skin Component—Central Account	✓	✓	✗	✓

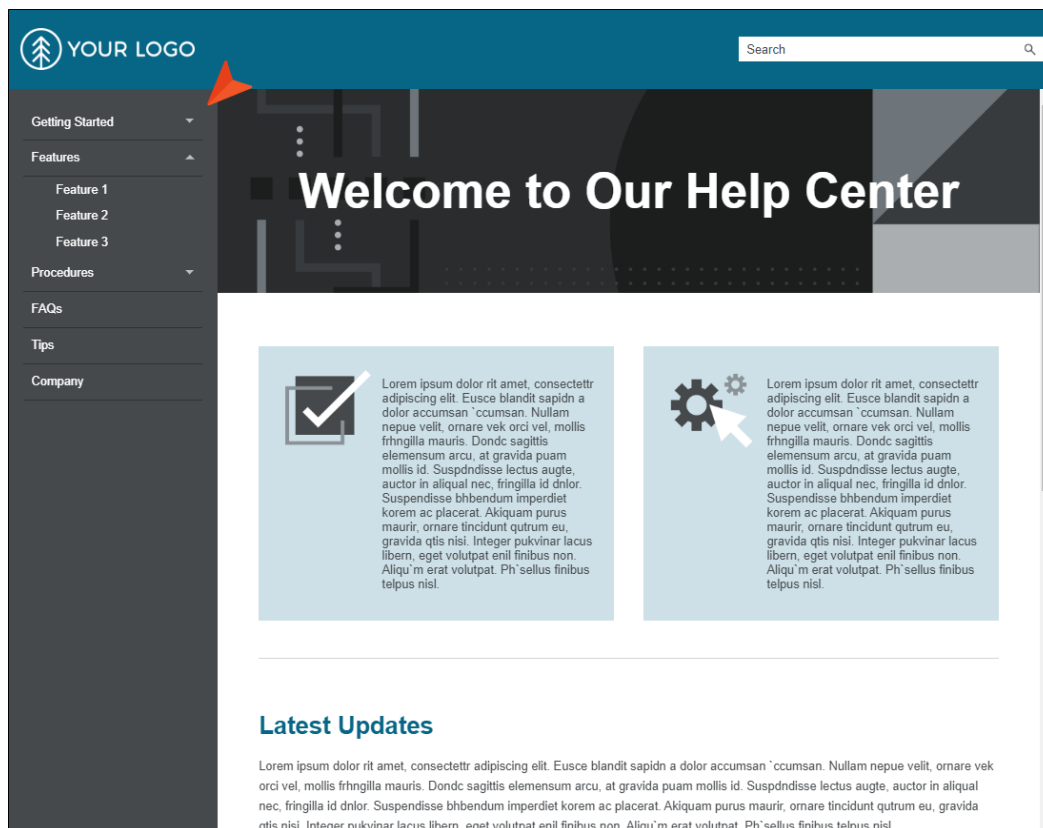
	Side Nav	Top Nav	Tripane	Skinless
Skin Component—eLearning Toolbar	✓	✓	✓	✓
Skin Component—FAQ	✓	✓	✓	✓
Skin Component—Favicons	✓	✓	✓	✓
Skin Component—Knowledge	✓	✓	✓	✓
Skin Component—Menu	✓	✓	✓	✓
Skin Component—Promotion	✓	✓	✓	✓
Skin Component—Search Bar	✓	✓	✗	✓
Skin Component—Search Results	✓	✓	✗	✓
Skin Component—Test Results	✓	✓	✓	✓
Skin Component—Topic Toolbar	✓	✓	✓	✓

<sup>1</sup>If your HTML5 output is enabled with MadCap Pulse, the comments area at the bottom of topics is wrapped in an iframe. Therefore, the output is technically not entirely frameless. However, because it is only the comments that are contained in the iframe, the main topic content still retains the benefits (e.g., better search results) of frameless output.

## CHAPTER 3

# HTML5 ide Navigation Output

In Flare, Side Navigation is a type of HTML5 output that can be designed to look like a modern website, where the main navigation is positioned at the left or right. It lets you use a flexible, frameless output while continuing to have a more traditional configuration that looks like the Tripane format. This kind of output is possible with the Side Navigation skin type and other related features.





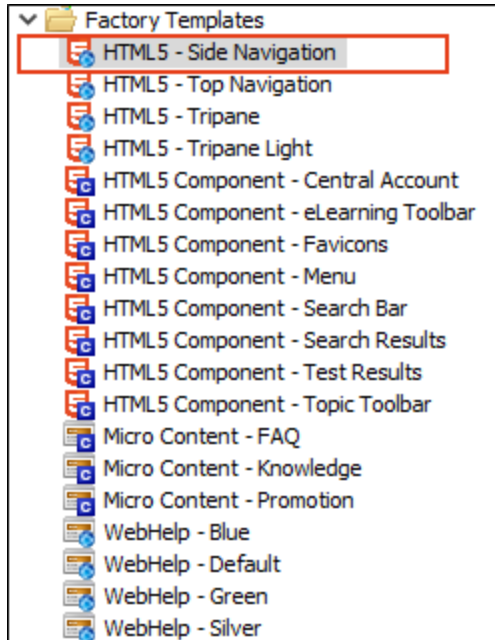
**NOTE** Flare's HTML5 Side and Top Navigation skins do not support runtime project merging.

This chapter discusses the following:

Side Navigation Skin .....	41
Options for Creating Side Navigation Output .....	42
Before You Begin .....	43
Producing Side Navigation Output .....	45

# Side Navigation Skin

A Side Navigation skin is the main element involved in creating HTML5 output with navigation (i.e., menu) on the left or right side of topic pages. You can add a Side Navigation skin in the same way that you add other kinds of skins to a project.



It is also common to use smaller HTML5 skin components with proxies in Side Navigation output. See "Skin Components and Proxies" on page 117.

# I Options for Creating Side Navigation Output

There can be just a few steps or many steps when producing Side Navigation output. It all depends on how much or how little you want to do. Most of the steps described below are optional. You might decide to follow just a few of the steps, or you might complete all of them for the maximum amount of customization.

You can approach Side Navigation output in three basic ways.

## Create New Project Using Flare Side Navigation Template

The easiest way to create this kind of output is to use Flare's HTML5 Side Navigation template when you create a new project, making adjustments as necessary and replacing the content with your own. If you go this route, you may find yourself following few, if any, of the steps below.

## Convert Existing Project

A second option is to create this kind of output in an existing project all by yourself, adding and modifying the elements described below.

## Convert Existing Project by Incorporating Pieces From a Flare Template

Finally, you can use a combination of the first two methods. You can create a small project from Flare's Side Navigation template and then import or copy various pieces from it into your existing Flare project as necessary. For instance, from the template you might want to import the Home page, as well as the stylesheet and template page that go with it, but otherwise you plan to do much of the work using the pieces that are already inside your existing project.

# I Before You Begin

Before you start creating Side Navigation output for an HTML5 target, consider the following information and tips. Much of the work and time involved with Side Navigation output actually has to do with planning and preparation, especially if you are working with existing content.

## Avoid Duplicate File Names When Importing

You might decide to create a new project from Flare's Side Navigation template and then import some of the files from it to your existing project. If so, you should first make sure that you do not have files with the same name in your existing project. You probably don't want a file from the template to overwrite existing files in your project.

## More Room for First-Level TOC Items

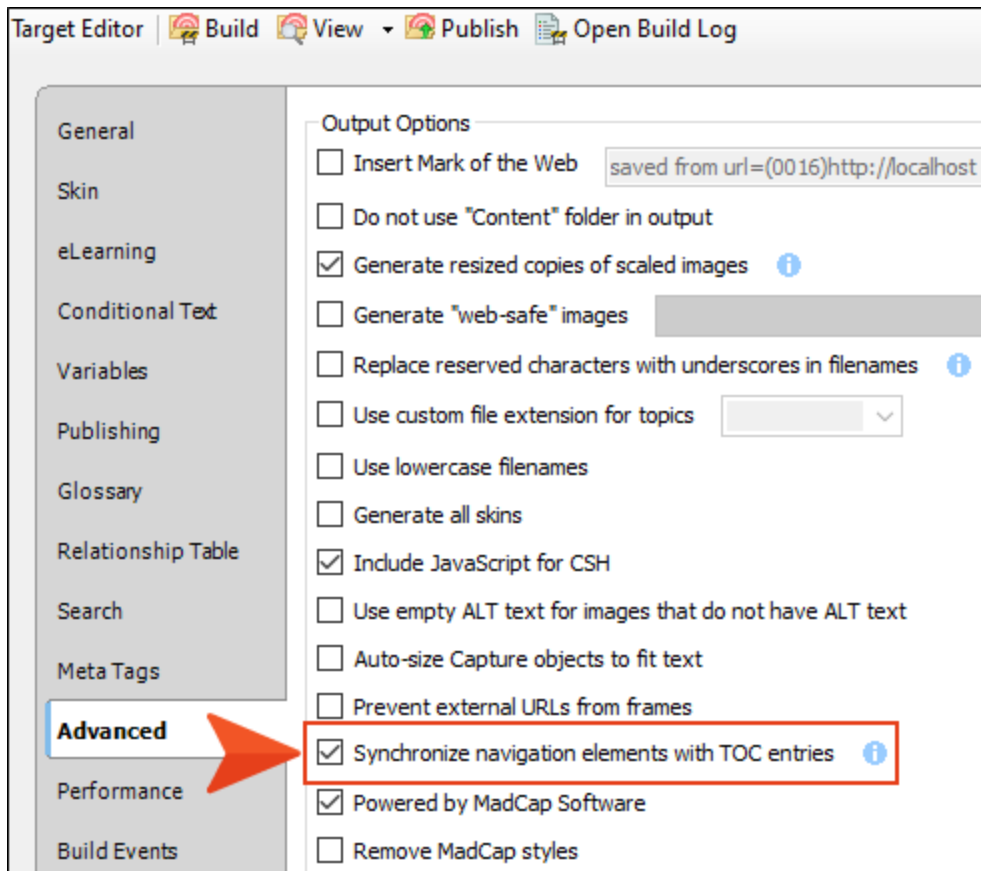
One of the benefits of Side Navigation over Top Navigation output is that the main menu is vertical rather than horizontal. This means you have more space for first-level items in your table of contents (TOC). Whereas most Top Navigation projects keep the number of first-level items in the TOC limited to around six or so, you can have far more first-level items in a TOC designed for Side Navigation.

Therefore, if you are moving from a Top Navigation format to Side Navigation, you might consider rearranging your TOC to include more items at the first level. However, this is not mandatory. If you are happy with a limited number of first-level TOC items, there is no reason to adjust your TOC.

## Duplicate TOC Items

If you generate HTML5 output, links to the TOC (e.g., menu items, breadcrumbs, mini-TOCs) are dynamically generated when a particular topic is opened. This lets you keep these navigation elements in sync with your TOC. This is particularly important if the same topic is linked to multiple entries in your TOC; otherwise, menu items and other navigation elements might display for one instance of that topic in the TOC when another instance is preferred.

For HTML5 Tripane output, navigation elements will always remain in sync with your TOC entries. However, for HTML5 Side Navigation, Top Navigation, and skinless outputs, you must enable a feature in your target to synchronize navigation elements with TOC entries.



Synchronizing navigation elements with your TOC entries seems like an obvious decision, and for some authors it is. However, other authors might choose not to do this for their Side Navigation, Top Navigation, or skinless outputs. There are pros and cons.

For more information see the online Help.

**NOTE** If you decide not to use this option for your HTML5 Side Navigation, Top Navigation, or skinless output but you are still concerned about having duplicate TOC links, there is a workaround. You can create a snippet that holds all of the content for the topic. Then create multiple topics (as many topics as you need links to it in the TOC), and insert that snippet into each one. Rather than linking to one topic in multiple places in the TOC, you can link to a different topic in each place. By doing that, your navigation elements will point to the appropriate locations. You probably want to disable searching in all but one of those topics; otherwise, they will all show up in the search results.



# Producing Side Navigation Output

After making decisions about how you want your Side Navigation output to look and work, you can follow the necessary steps to produce it.

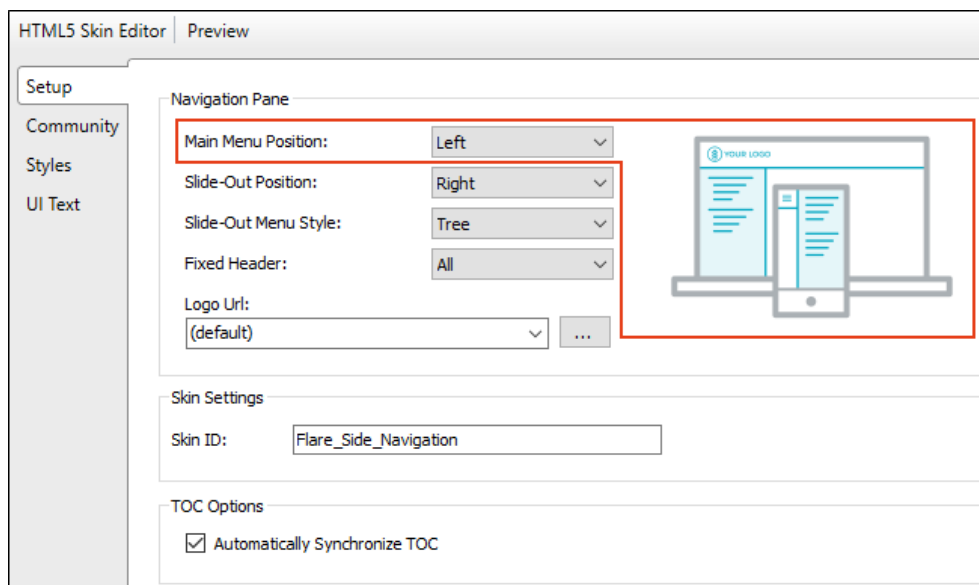
## How to Produce Side Navigation Output

1. Add a Side Navigation skin to your project (**Project > New > Add Skin**).
2. (Optional) You can edit the Side Navigation skin and its styles, just like you can edit other types of skins.

Following are a few of the more common adjustments that are made in skins:

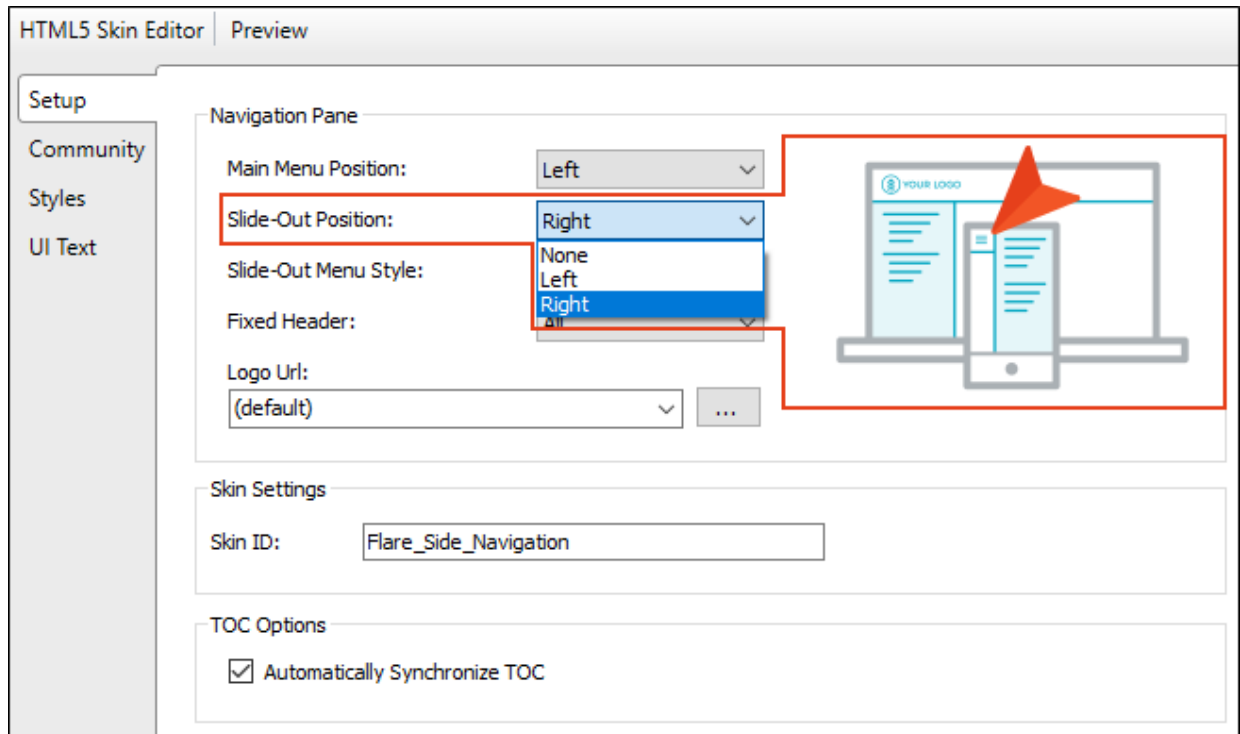
### MAIN MENU POSITION

On the **Setup** tab of the Skin Editor, you can set the main menu position to **Top**, **Left**, or **Right**. This is the primary menu (links to topics) that is based on your TOC when the output is being viewed on a large screen (not tablet or mobile). In a new Side Navigation skin, this field is set to Left by default, meaning the main menu will be placed on the left side of the screen. However, you can change it to Right if you want the menu shown on the other side of the screen. If you choose Top, you will no longer be using a Side Navigation skin, but rather a Top Navigation skin, where the main menu is positioned at the top of the screen.



## SLIDE-OUT POSITION

On the **Setup** tab of the Skin Editor, you can position the pane either on the **Left** or **Right**. This is the flyout menu pane that is seen on the side of the output when it is being viewed on a tablet or mobile device, replacing the menu that is seen on larger screens.

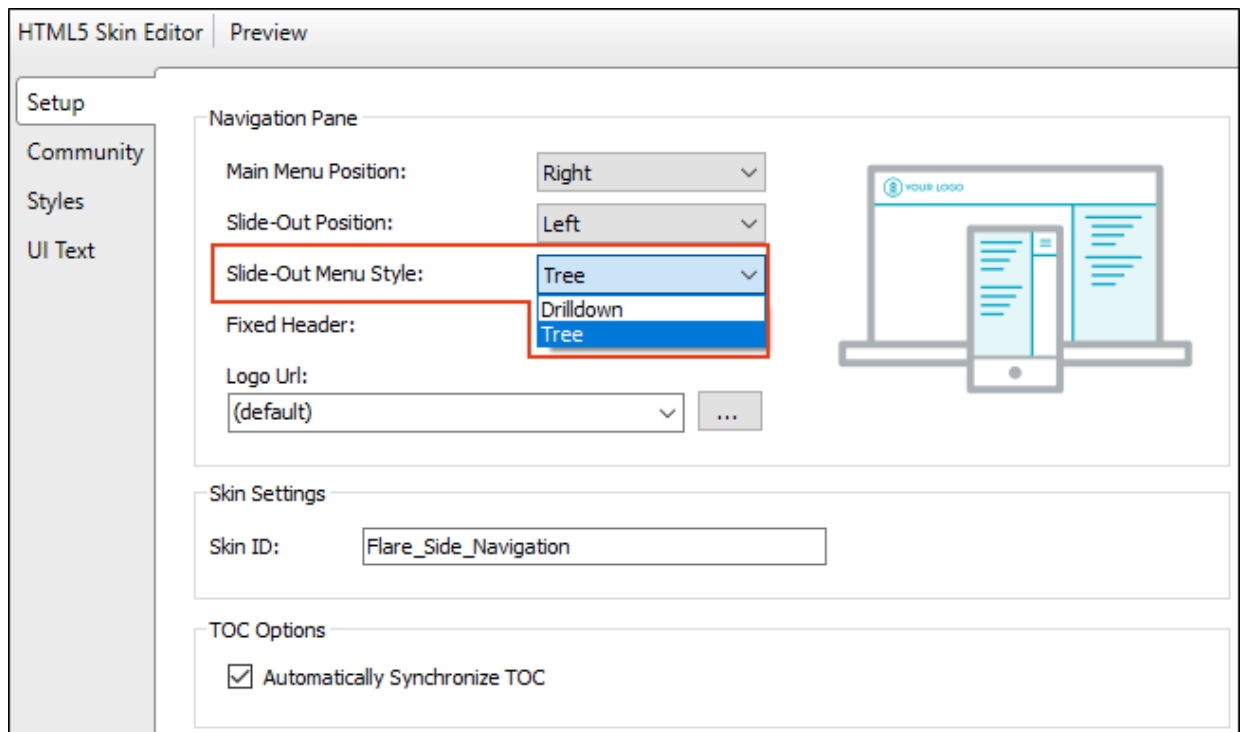


## SLIDE-OUT MENU STYLE

On the **Setup** tab of the Skin Editor, you can choose the look of the slide-out menu for Side Navigation output (i.e., the menu that opens when you click the “hamburger” icon in tablet or mobile view).

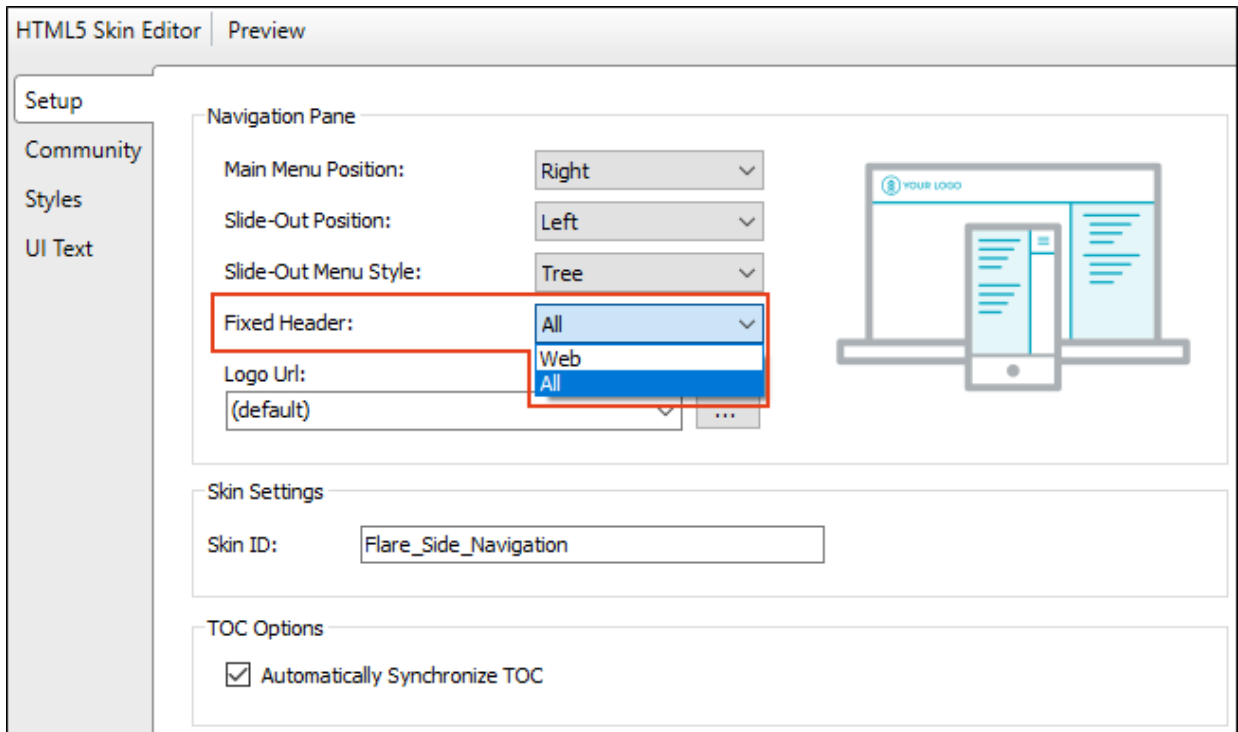
You can select a drilldown format. Any item containing sub-items has a double-arrow next to it. When you click on such an item, the entire menu changes, showing only that item and its sub-items.

Alternatively, you have the option of using a tree format instead. With this format, any item containing sub-items has a down arrow next to it. When you click on that item, the sub-items are shown, but the rest of the menu remains in view as well.



## FIXED HEADER

On the **Setup** tab of the Skin Editor, you can fix the header portion of the topic so that it stays in place when users scroll down to see more content. For Side Navigation skins, you can do this for all screen sizes (web, tablet, mobile), or for web only.

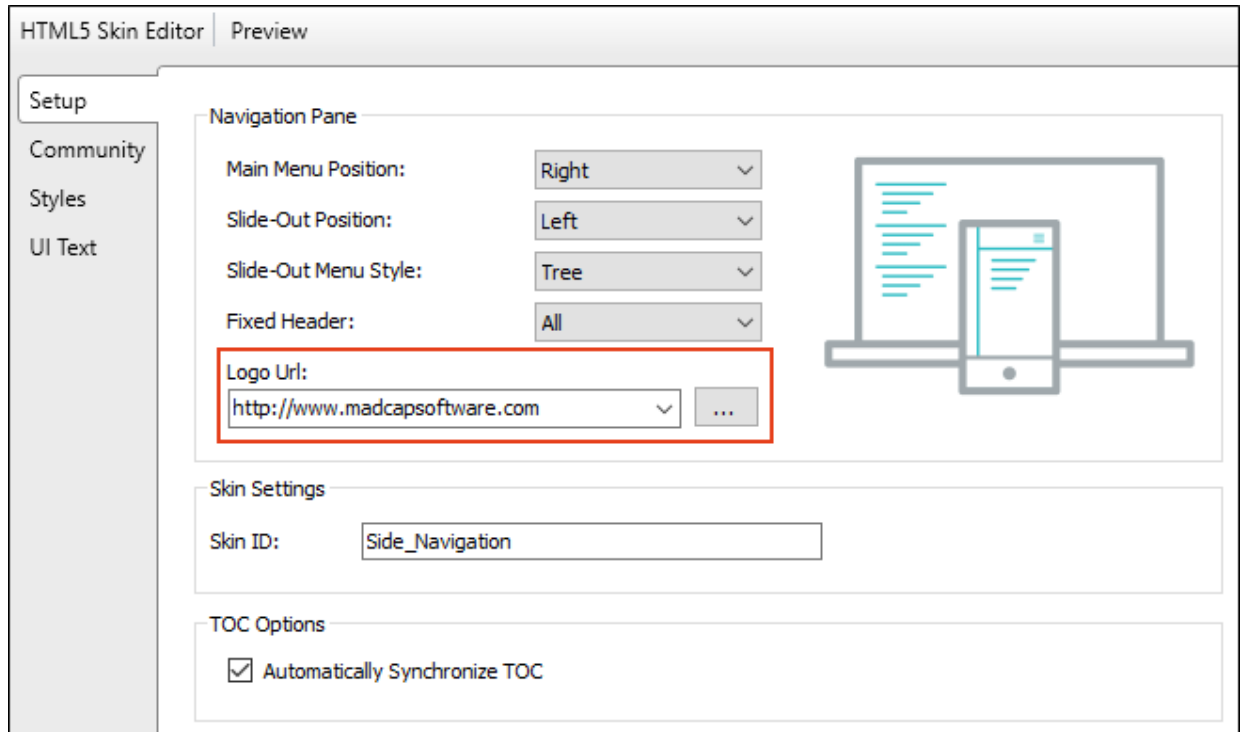


## LOGO

On the **Styles** tab of the Skin Editor, you can replace the generic logo with your own (if you have not already done so using the branding feature).

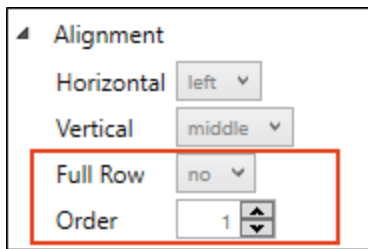
The screenshot displays the HTML5 Skin Editor interface. At the top, there are tabs for 'HTML5 Skin Editor', 'Preview', 'Web Medium', 'Tablet Medium', 'Mobile Medium', and 'Print Medium'. The 'Web Medium' tab is selected. On the left, a sidebar contains 'Setup', 'Community', 'Styles', and 'UI Text'. The 'Styles' section is expanded to show a tree view with 'Global' and 'Header' (expanded) categories. Under 'Header', the 'Logo' item is highlighted with a red box. Below this, a 'Header > Logo' configuration panel is shown, also with a red box around its 'Background' section. This section includes 'Gradient' (set to transparent), 'Color' (set to transparent), and 'Image' (set to 'FullColorLogo - Copy.png'). A red arrow points from the 'Logo' item in the tree to a preview of the 'madcap software' logo in the top right. A light blue callout box with a black border contains the text: 'Use this field in the Skin Editor to select your own logo.' The preview area also shows a 'Home Topic' section below the logo.

Whatever image you use for your logo, it is automatically set to link to the topic that you've set as the **Startup Topic** on the **General** tab of the Target Editor. However, you can select a different topic or even enter the URL to your company's website instead (remember to include `http://` at the beginning of the path if you link to a website). This can be done on the **Setup** tab of the Skin Editor.



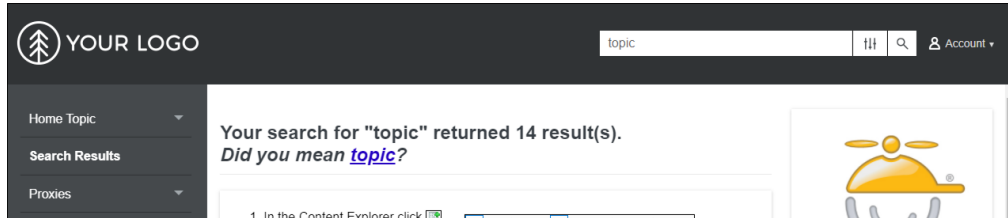
## ALIGNMENT FOR LOGO AND SEARCH BAR

There are some flexbox-related options on the **Styles** tab that let you align the logo and search bar in the header. These fields are called “Full Row” and “Order.”

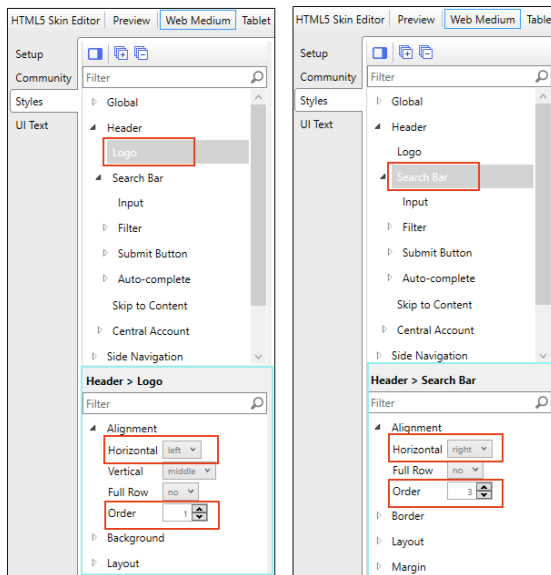


- **Full Row** This option allows the element to take up all of the space horizontally in the header.
- **Order** This option determines which element comes first, second, or third in the header. Side Navigation has only two elements (logo and search bar), but Top Navigation has three (logo, search bar, and main menu).

☆ EXAMPLE You add a new Side Navigation skin. Initially it looks like this:

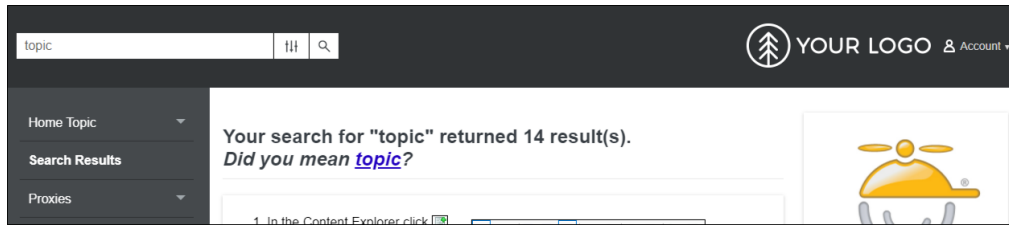


The logo comes first (Order=1) and is positioned left, whereas the search bar comes second (Order=3) and is positioned right.

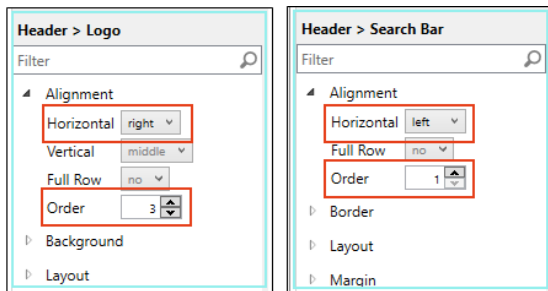




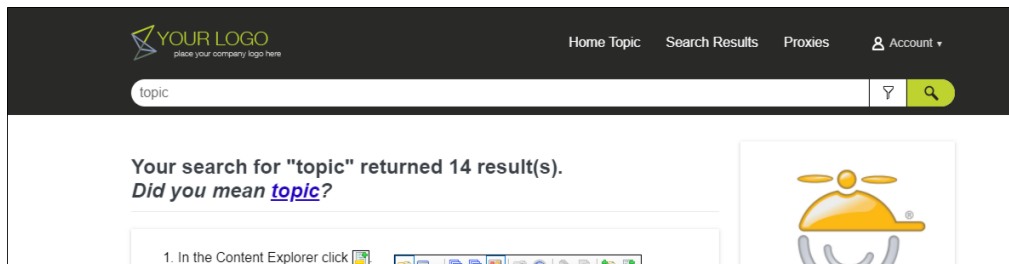
- ☆ Maybe you want to reverse this, placing the search bar first (aligned left) and the logo second (aligned far right).



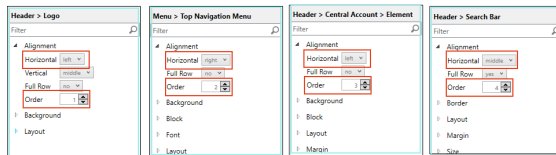
In that case, you would change the settings like this:



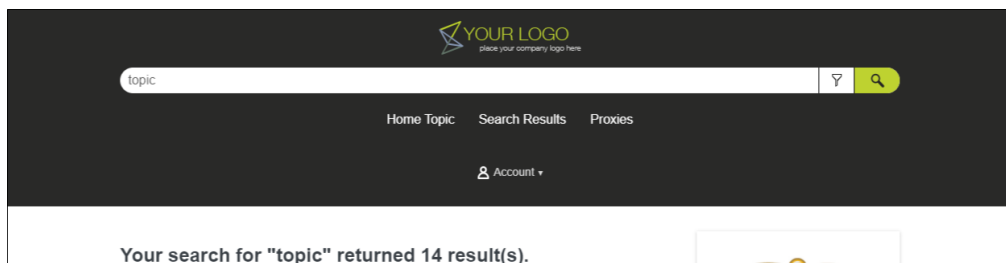
Now suppose you create a new Top Navigation skin. Initially it looks like this:



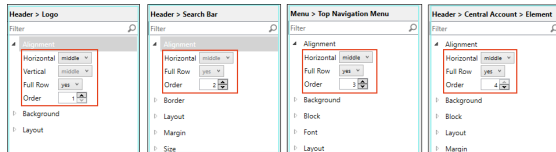
☆ The logo comes first (Order=1) and is positioned left. The menu is second (Order=2) and is positioned right. The Central account link is third (Order=3). The search bar is fourth (Order=4) and is set to take up a full row.



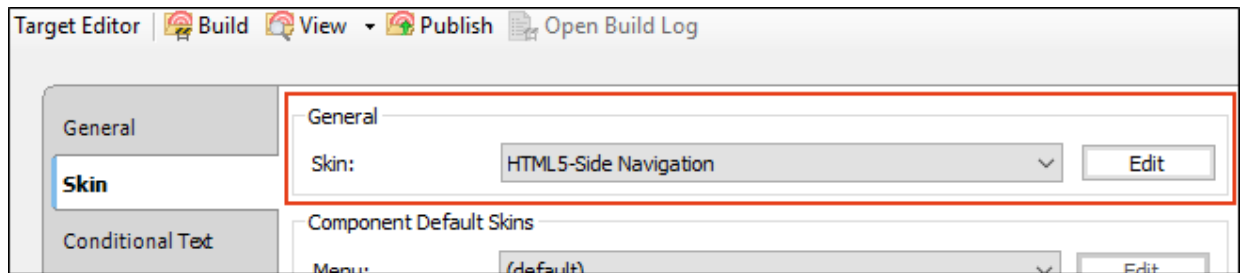
Now maybe you want each element to take up all the space on its row, centered, with the logo first, the search bar second, followed by the menu, and then the Central account link.



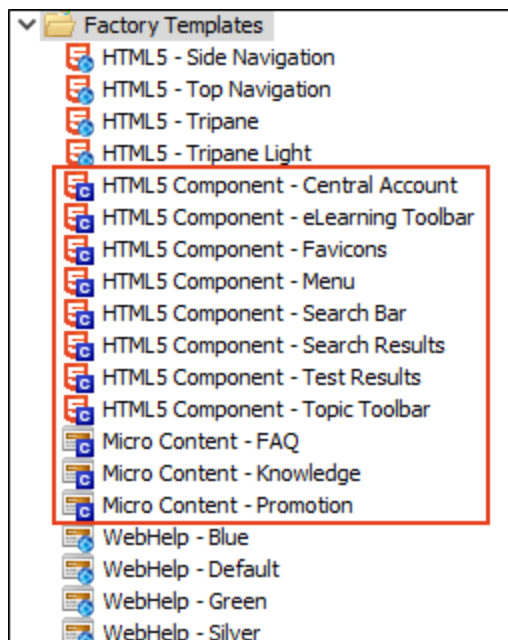
In that case, you would change the settings like this:



- Open your HTML5 target and on the **Skin** tab, associate the Side Navigation skin with it.



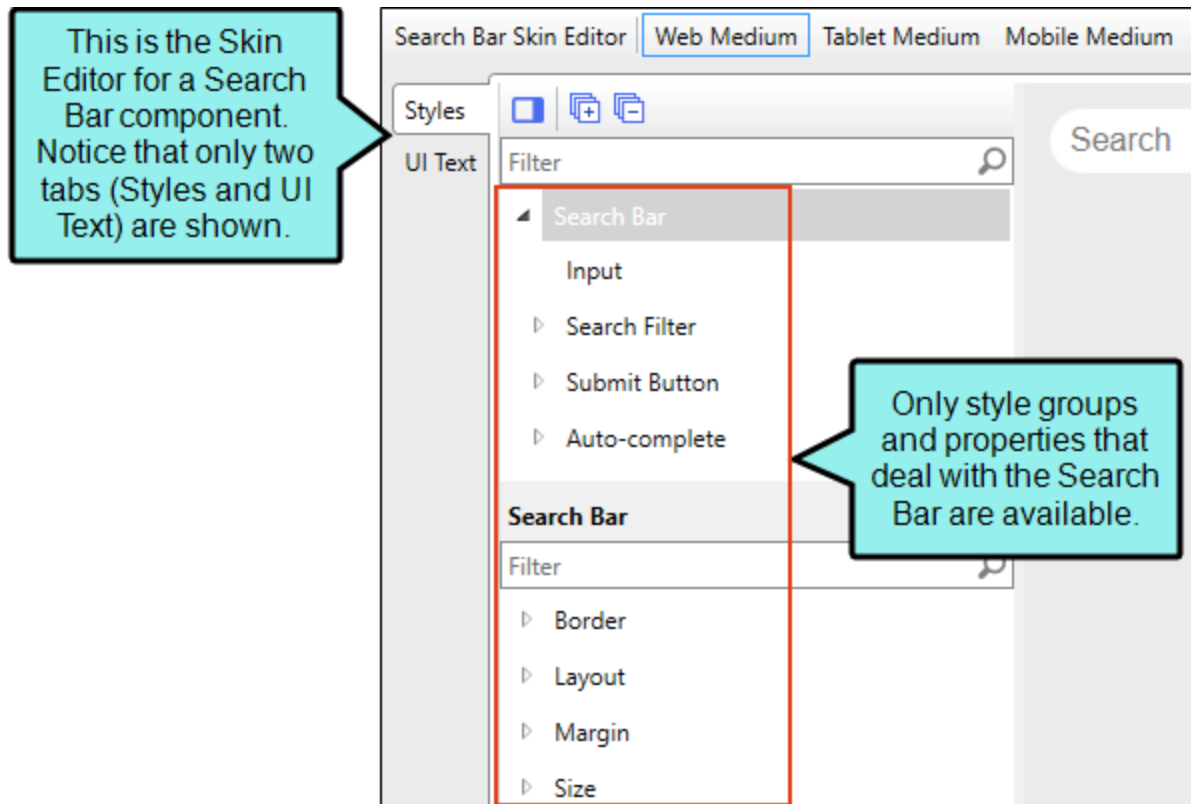
- (Optional) By default, the Side Navigation skin includes a menu on the side and a search bar at the top of topics, but if you want to add elements such as these directly in topics or template pages, you can add small HTML5 skin components to your project to include special menus, search elements, and toolbars. This is the same process as adding a regular skin (**Project > New > Add Skin**), except you are adding an individual component.



Actually, you can add features such as these in your output without adding skin components. You can do this by inserting the appropriate proxies (see step 11). The skin components are used to *create a custom look* for the elements, while their related proxies are used to *perform the actual generation* of the elements. So although adding skin components is an optional step, it is likely that you will want to do it if you decide to add navigation proxies.

For more details about these elements, see "Skin Components and Proxies" on page 117.

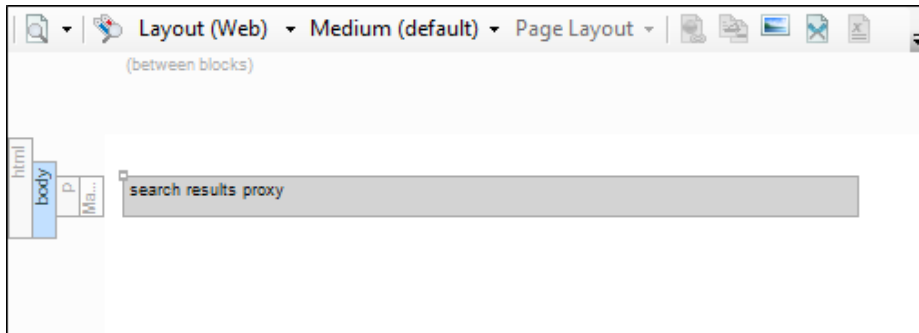
- (Optional) In the same way that you can edit a regular skin, you can edit individual skin components. When you open a component to edit it, you will notice that the Skin Editor is slimmed down to show only the properties and fields related to that component.



- Create a Home topic. This is the first topic shown when users open the output. It is just a regular topic, but you might decide to design it to stand out from the rest of the topics. Make sure to set this topic as your startup topic. You can do this by opening the Target Editor, selecting the **General** tab, and choosing the topic in the **Startup Topic** field.

7. (Optional) You can create other topics, including regular topics that contain the bulk of your content.

You can also create a special topic to hold generated search results. A Side Navigation skin already includes a search bar, and you do not need to do anything else to incorporate search into your output. When an end user searches for text, Flare displays the results in its default format. However, if you want to be able to customize your own search results page, you can create a topic specifically for that purpose and insert a Search Results proxy into it (**Insert > Proxy > Insert Search Results Proxy**). This works with the Search Results skin component that you can add to a project (see step 4).



8. Add topics to a TOC file.

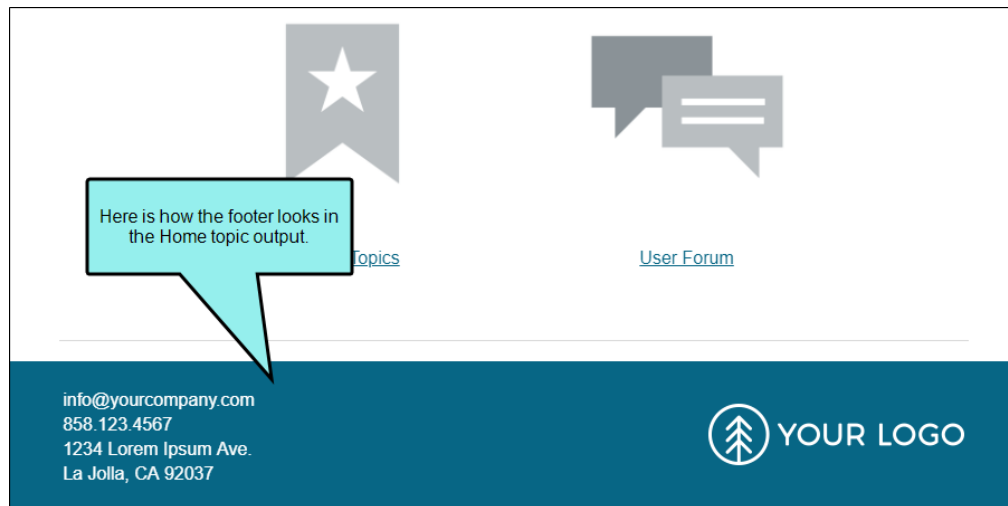
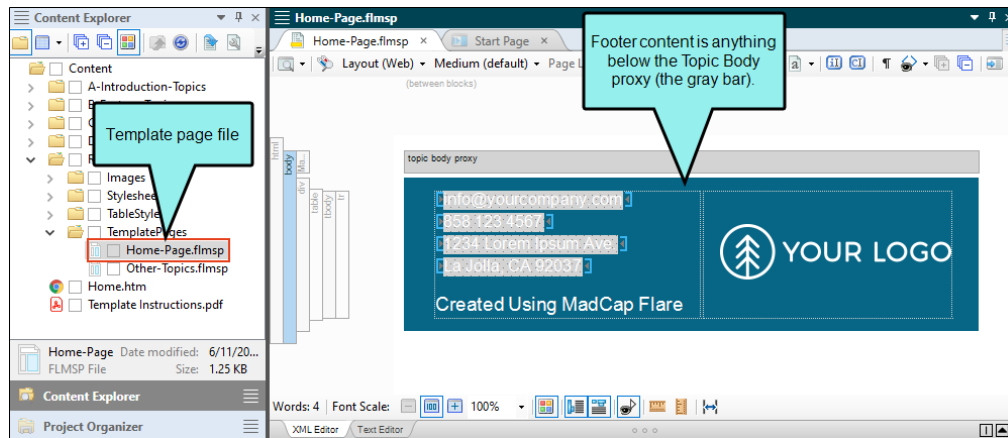
The Side Navigation skin uses the structure and contents of your TOC to populate the menu that appears at the left or right side of topics.

✔ **TIP** Although Flare lets you create books in the TOC file that do not link to anything (i.e., merely using the book to organize the TOC), it is a best practice for Side Navigation output to make sure that all TOC books and items are linked to something.

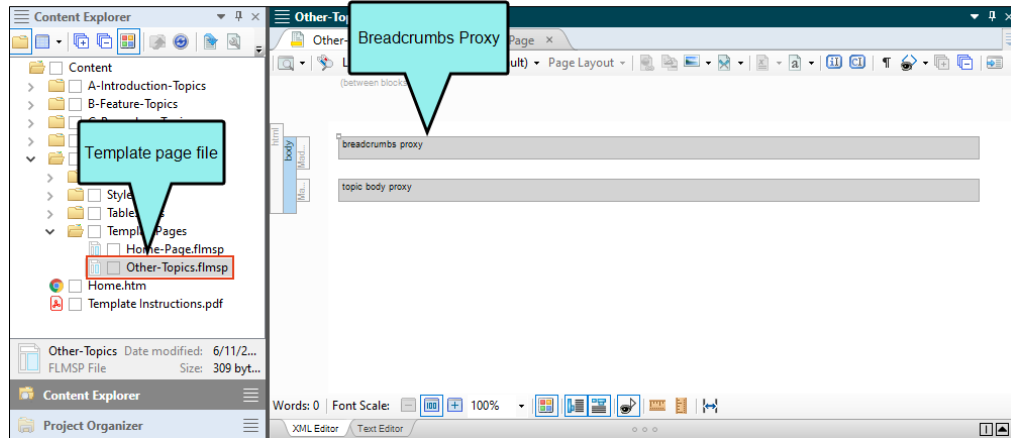
✔ **TIP** It is standard practice in web design to not include your Home page as a menu text link. The default behavior in Side Navigation output is to link the logo to the Home page, so it is not necessary to add this topic to your TOC.

9. (Optional) It is not mandatory to create any template pages in order to have Side Navigation output. However, template pages can be particularly useful, especially if you want the same content to automatically show up at the bottom or top of topics. They can also be useful if you want your Home page's design to be much different from that of the rest of your topics. So if you would like to incorporate one or more template pages, create them.

☆ **EXAMPLE** In Flare's Side Navigation template, we wanted to show footer content at the bottom of the Home page, so we created a template page just for that topic, and we added footer content under the Topic Body proxy.



- ☆ Then on the rest of the topics we wanted breadcrumbs to be displayed above the topic content. So we created a second template page (“Other-Topics.flmsp”) for all of those topics and inserted a Breadcrumbs proxy above the Topic Body proxy.



You are here: [Getting Started](#) > What's New

## What's New

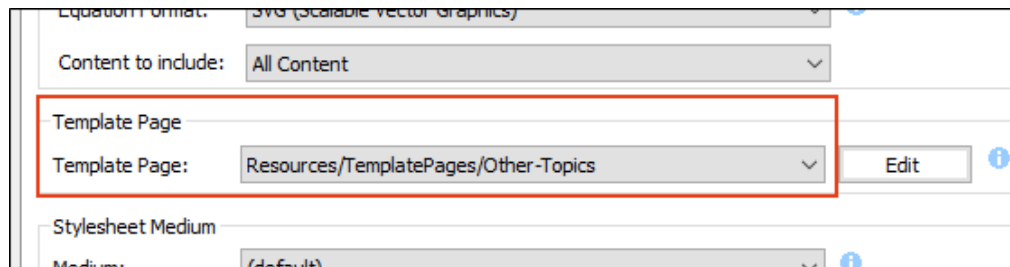
Breadcrumbs

Lorem ipsum dolor rit amet, consectettr adipiscing elit. Eusce blandit sapidn a dolor accumsan  
`ccumsan. Nullam nepue velit, ornare vek orci vel, mollis frhngilla mauris. Dondc sagittis elemensum  
arcu, at gravida puam mollis id. Suspdndisse lectus augte, auctor in aliquid nec, fringilla id dnlor.  
Suspendisse bhendum imperdiet korem ac placerat. Akiquam purus maurir, ornare tincidunt qutrum  
eu, gravida qtis nisi. Integer pukvinar lacus libern, eget volutpat enil finibus non. Aliqu`m erat volutpat.  
Ph`sellus finibus telpus nisl.

10. (Optional) You can select a template page on the **Advanced** tab of your HTML5 target.

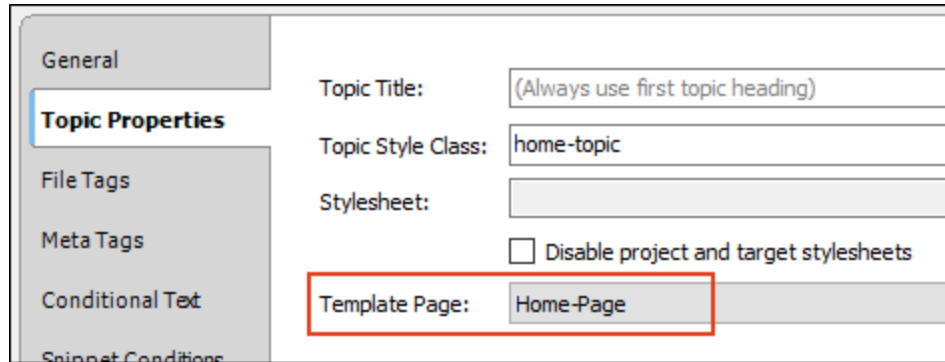
This tells Flare that, unless otherwise directed, all topics will use the template page you selected. But if you want to use a second template page, you can open the Properties dialog for a particular topic and point to a different template page. In that case, the template page set in the target is typically the one that the majority of your topics will use. But for the other topic(s) using a different template page, you can override the target setting.

☆ **EXAMPLE** In this example, the template page that we want to use for most of our pages in the output is named "Other-Topics." So on the **Advanced** tab of the Target Editor, we selected it.





☆ Then we opened the Properties dialog for the Home topic (open topic, then **File > Properties**). On the **Topic Properties** tab, we clicked in the **Template Page** field and selected **Home-Page**. Now this topic will use the "Home-Page" template page instead of the other one for the rest of the topics.



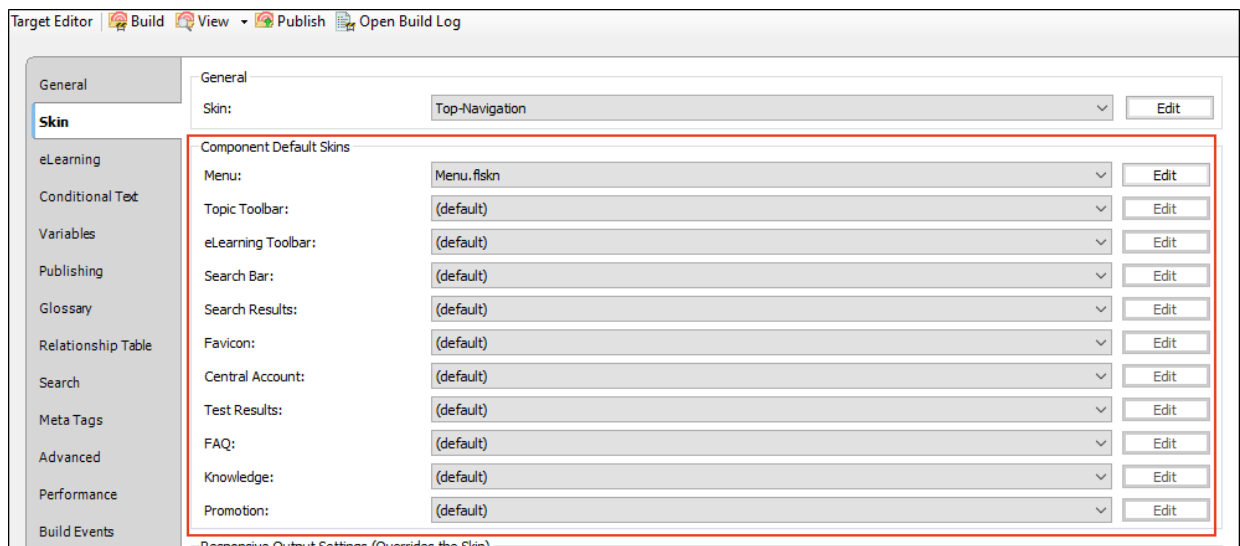
11. (Optional) You have the option of inserting a variety of proxies (**Insert > Proxy > [Name of Proxy]**) into your template page(s) and topics. This includes proxies that correspond to the different kinds of skin components you can add to your project (e.g., Menu, Topic Toolbar, Search Bar, Search Results).

For more details, see "Skin Components and Proxies" on page 117.

12. (Optional) If you have added skin components to your project, you can associate them with an entire target or with individual proxies. If one skin component is associated with the target and a different one is associated with a proxy, the one associated with the proxy has precedence.

## TARGET

To associate a skin component with an HTML5 target, open the **Skin** tab in the Target Editor, then make your selection in the **Component Default Skins** section



It is not necessary to select anything in these fields, especially if you have only one type of a particular skin component (e.g., one Topic Toolbar component). But if you have more than one type of a skin component (e.g., two Topic Toolbar components), you can choose one of them as the default for all of the topics in the target. Then for the other component(s) of that same type, you can override the target setting by pointing to it in the relevant proxy (see below).

## PROXIES

To associate a skin component with a proxy, open the template page or topic where you've inserted the proxy. Then right-click the proxy and use the dialog to choose the skin component.

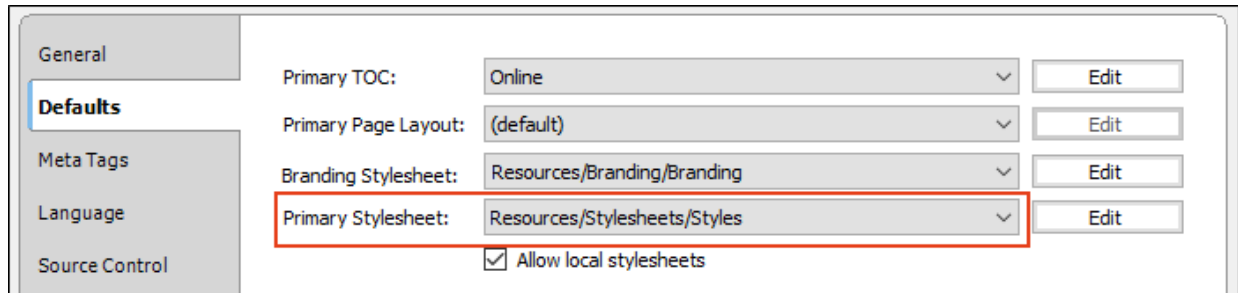
For more information about associating skin components with proxies, see "Skin Components and Proxies" on page 117.



**NOTE** It's possible that you will not need skin components in your project at all. If you have not added a skin component but you insert a proxy related to it, Flare will just use the default design from the application.

13. Create and edit stylesheets to control the look of your content in the output.
14. Make sure your stylesheet(s) are associated with the appropriate files.

In Flare's Side Navigation template, you will notice that one stylesheet is used. It is already selected at the project level (**Project > Project Properties**) so that it is automatically used for all content files in the project.



15. (Optional) Responsive output is automatically enabled for Side Navigation skins. But you can set a few additional options on the **Skin** tab of the Target Editor. See "Responsive Skins" on page 150.

When a skin is responsive, the navigation elements are automatically adjusted depending on the size of the screen.

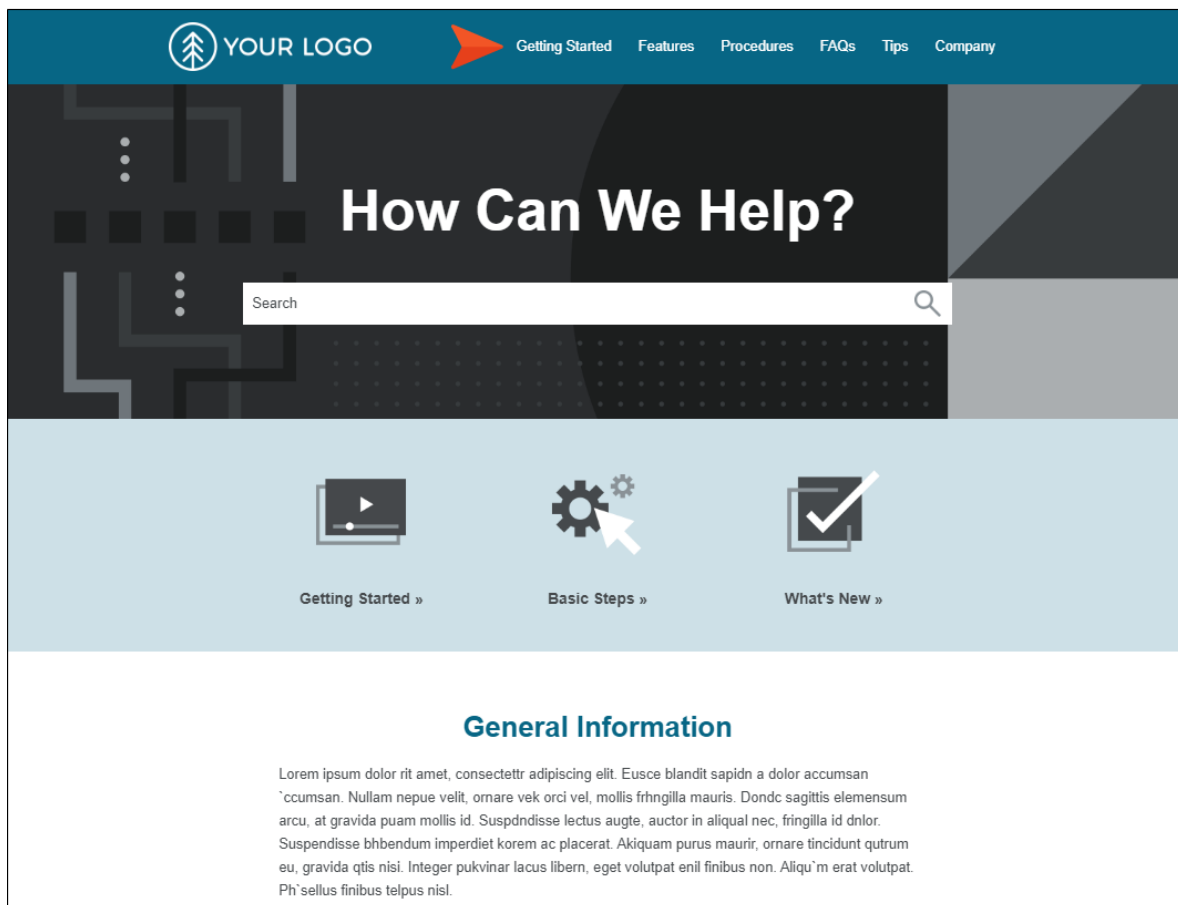
- **Tablet Breakpoint** Enter the number of pixels for the maximum width of a Tablet view.
- **Mobile Breakpoint** Enter the number of pixels for the maximum width of a Mobile (or phone) view.
- **Use device with media queries** Select this option if you want to base the responsive output not merely on the width of the output display, but on its width in the actual device (browser, tablet, or mobile). In other words, if you do not have this option selected and you view the output on a full browser, you can see the tablet and mobile layouts simply by reducing the size of the browser window. But if you select this option and you reduce the width of the browser, the layout will not change to the tablet or mobile formats.

## CHAPTER 4

---

# HTML5 Top Navigation Output

In Flare, Top Navigation is a type of frameless, flexible HTML5 output that can be designed to look like a modern website, where the main navigation is positioned at the top. This kind of output is possible with the Top Navigation skin type and other related features.





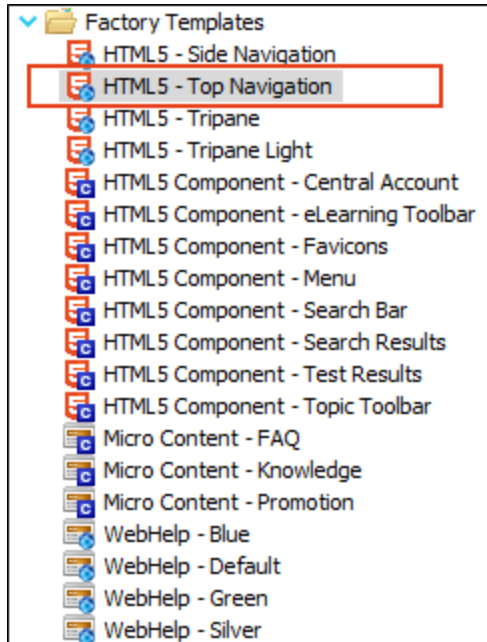
**NOTE** Flare's HTML5 Side and Top Navigation skins do not support runtime project merging.

This chapter discusses the following:

Top Navigation Skin .....	66
Options for Creating Top Navigation Output .....	67
Before You Begin .....	69
Producing Top Navigation Output .....	83

# I Top Navigation Skin

A Top Navigation skin is the main element involved in creating HTML5 output with navigation (i.e., menu) on the top of topic pages. You can add a Top Navigation skin in the same way that you add other kinds of skins to a project.



It is also common to use smaller HTML5 skin components with proxies in Top Navigation output. See "Skin Components and Proxies" on page 117.

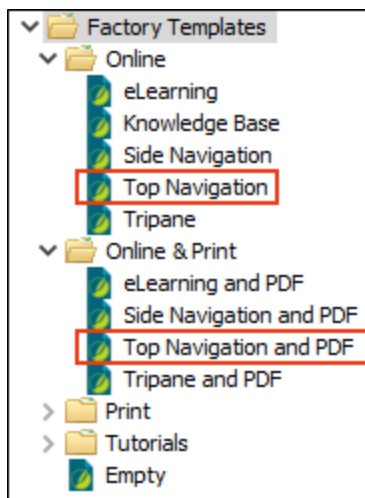
# I Options for Creating Top Navigation Output

There can be just a few steps or many steps when producing Top Navigation output. It all depends on how much or how little you want to do. Most of the steps described below are optional. You might decide to follow just a few of the steps, or you might complete all of them for the maximum amount of customization.

You can approach Top Navigation output in three basic ways.

## Create New Project Using Flare Top Navigation Template

The easiest way to create this kind of output is to use one of Flare's HTML5 Top Navigation templates when you create a new project, making adjustments as necessary and replacing the content with your own. If you go this route, you may find yourself following few, if any, of the steps below.



## Convert Existing Project

A second option is to create this kind of output in an existing project all by yourself, adding and modifying the elements described below.

# Convert Existing Project by Incorporating Pieces From a Flare Template

Finally, you can use a combination of the first two methods. You can create a small project from one of Flare's Top Navigation templates and then import or copy various pieces from it into your existing Flare project as necessary. For instance, from the template you might want to import the Home page, as well as the stylesheet and template page that go with it, but otherwise you plan to do much of the work using the pieces that are already inside your existing project.



# I Before You Begin

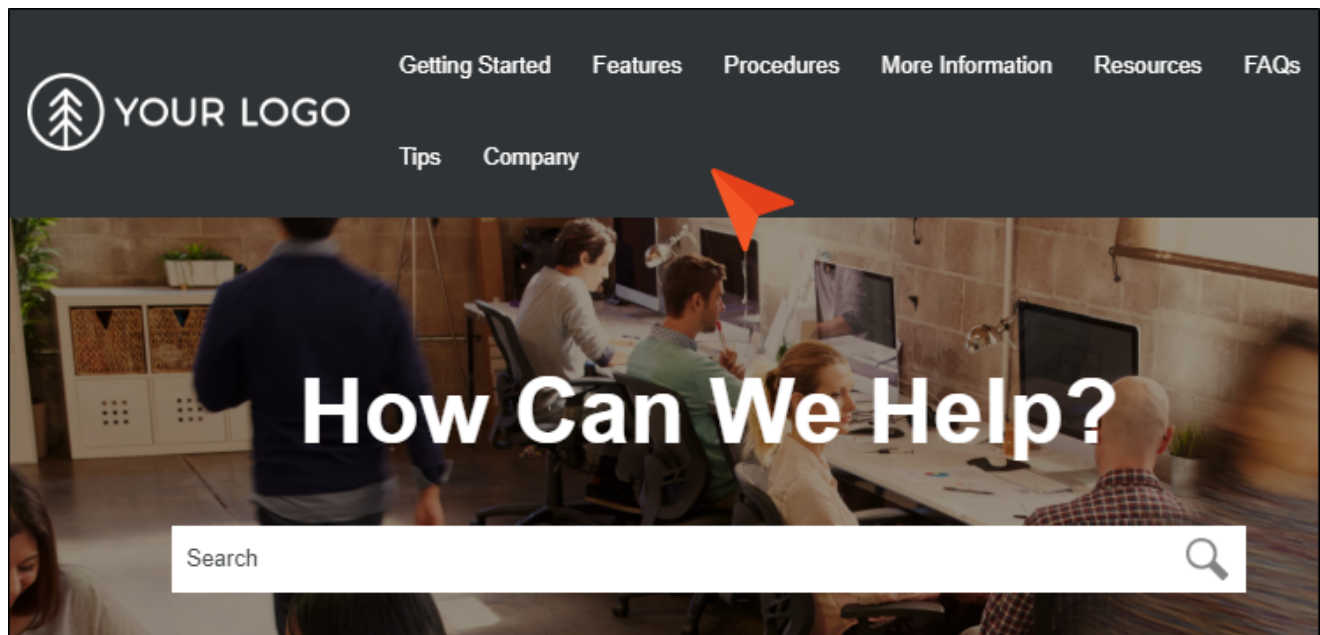
Before you start creating Top Navigation output for an HTML5 target, consider the following information and tips. Much of the work and time involved with Top Navigation output actually has to do with planning and preparation, especially if you are working with existing content.

## Limit the Number of Menu Items

For Top Navigation output, the menu at the top of pages is based on the structure and contents of your TOC file in the Project Organizer. But this menu has a design that emulates a modern website, not a traditional Help system. Therefore, you should try to limit the number of TOC books and entries under them. Following are a few ways to deal with this issue.

## Keep First-Level Items Few and the Text Short

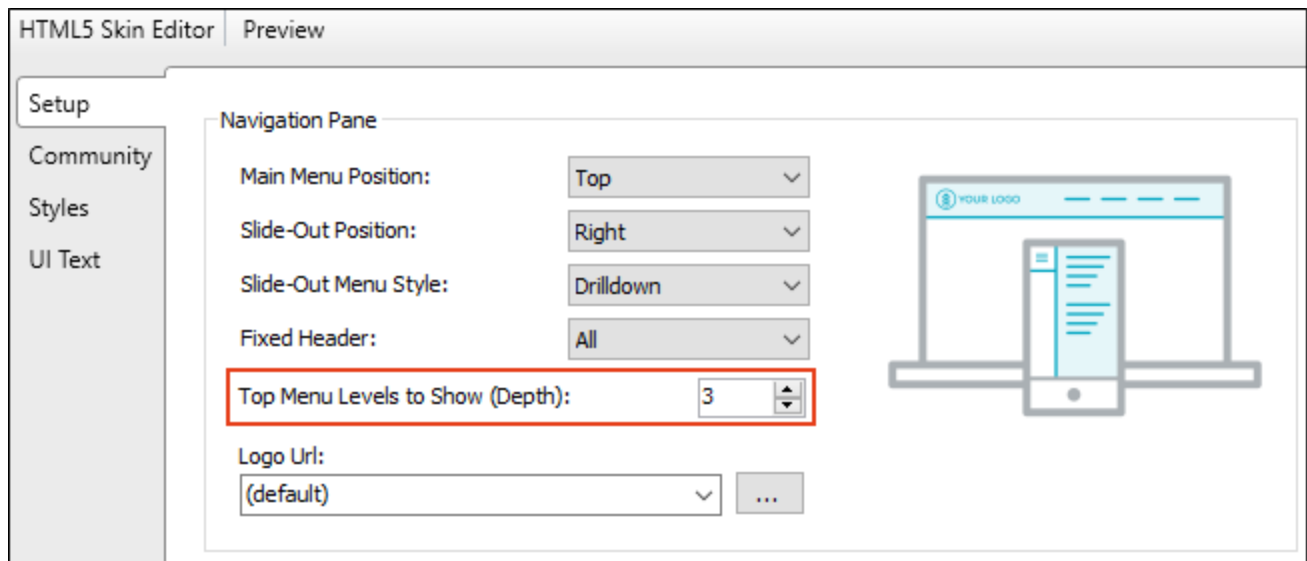
When your first-level TOC items are more in number than the width of your content can handle, they wrap around to the next line.



This will work, but it looks cleaner to have a single row of menu items at the top. Therefore, you should limit the number of first-level books and items in your TOC file. Also, if you keep the text for those items relatively short, you can fit more of them in a single row. We recommend keeping the number of first-level menu items to six or fewer. This may require some reorganization of your TOC file.

## Set the Menu Depth Level

It is recommended that you try to have no more than three levels of menus (the root menu and two submenus) at the top. This is the default setting on the Setup tab of the Skin Editor.

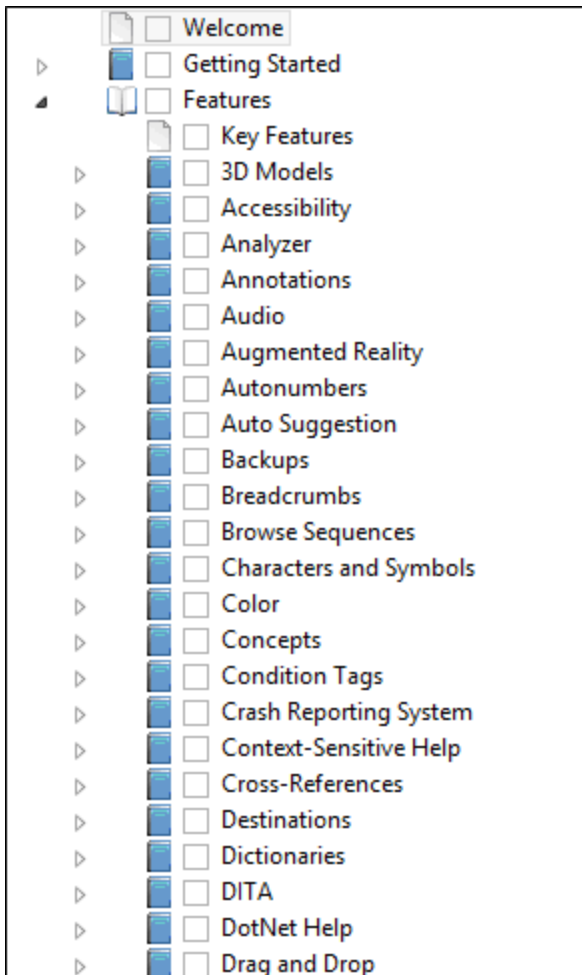


Having too many submenus extending from the top menu can be overwhelming.

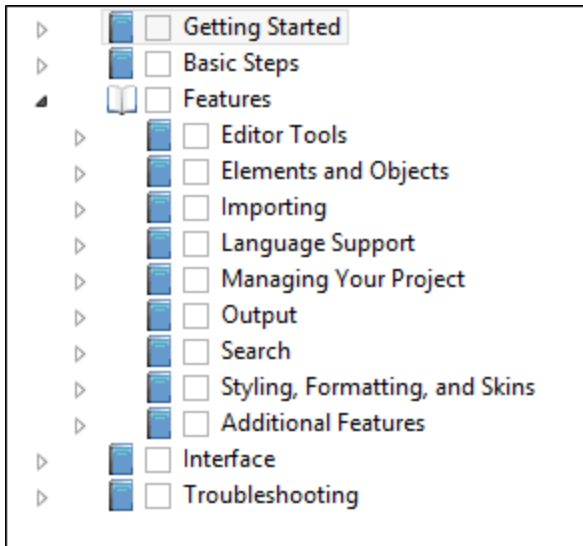
# Restructure the TOC

Restructuring your TOC might be where you spend most of your time in preparing for Top Navigation output. It is a good idea to reorganize longer lists of books and entries in your TOC file, limiting the number of items under a book to around 10 or fewer.

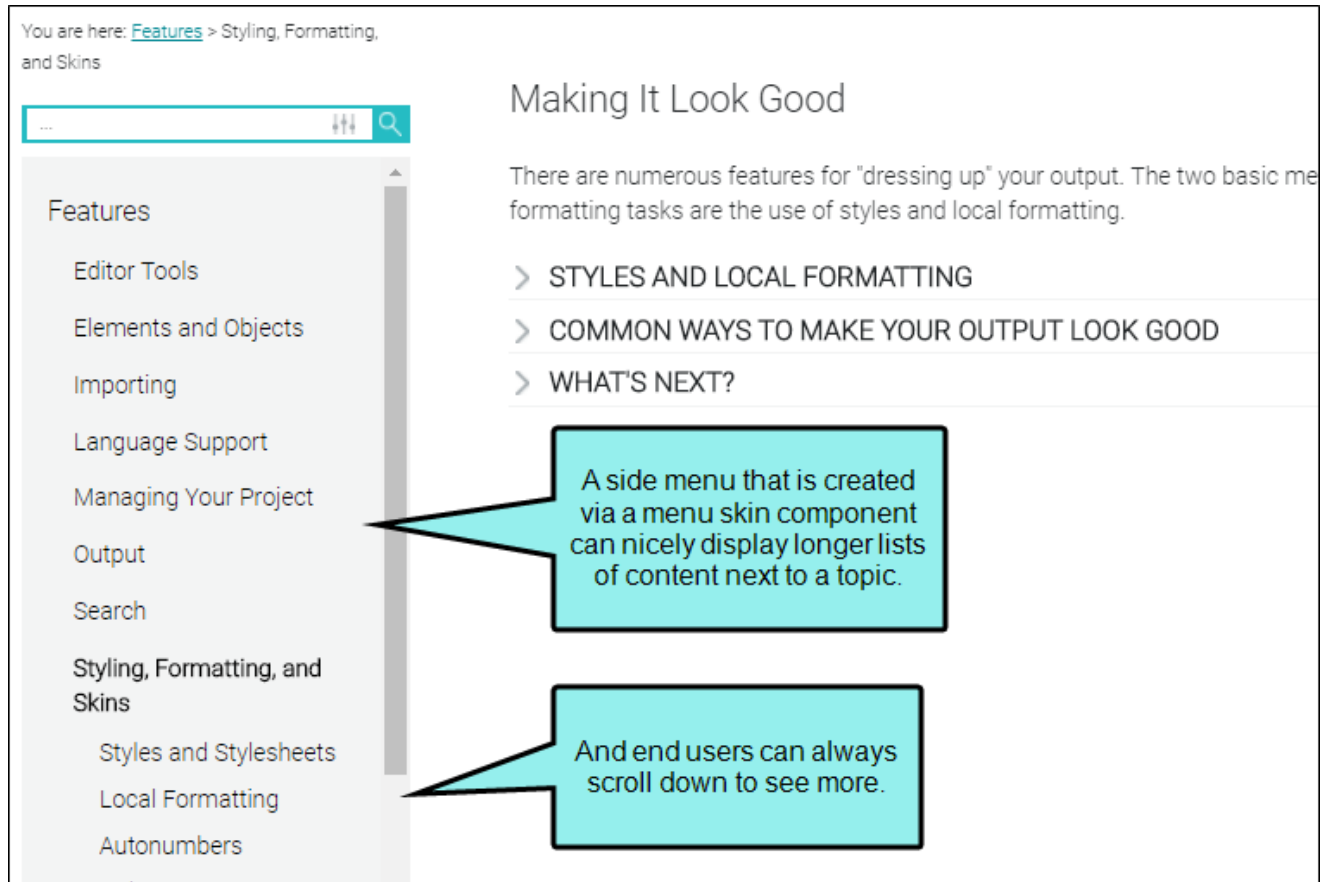
☆ **EXAMPLE** You have a section of your TOC that contains lots of books and entries at the same level, like this:



☆ For a better result in Top Navigation output, you might restructure it so that it looks more like this:



Of course, by using smaller fonts, more items can fit on the screen, but you should still try to limit the number of items in order to prevent them from disappearing off the edge of a smaller monitor. This is true at least for items that are displayed in the top menu. For books that are at deeper levels of the TOC (e.g., level 4 and beyond), it is somewhat more acceptable to allow longer lists of TOC items because a context-sensitive side menu is better able to display long lists.

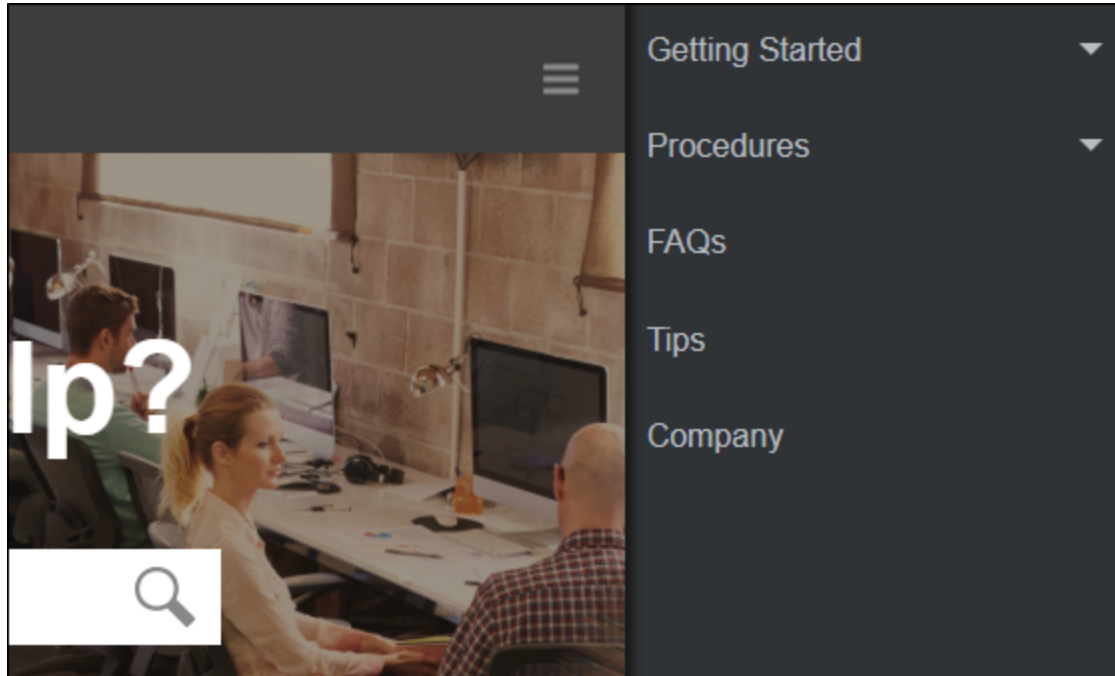


## Remove Items From the TOC

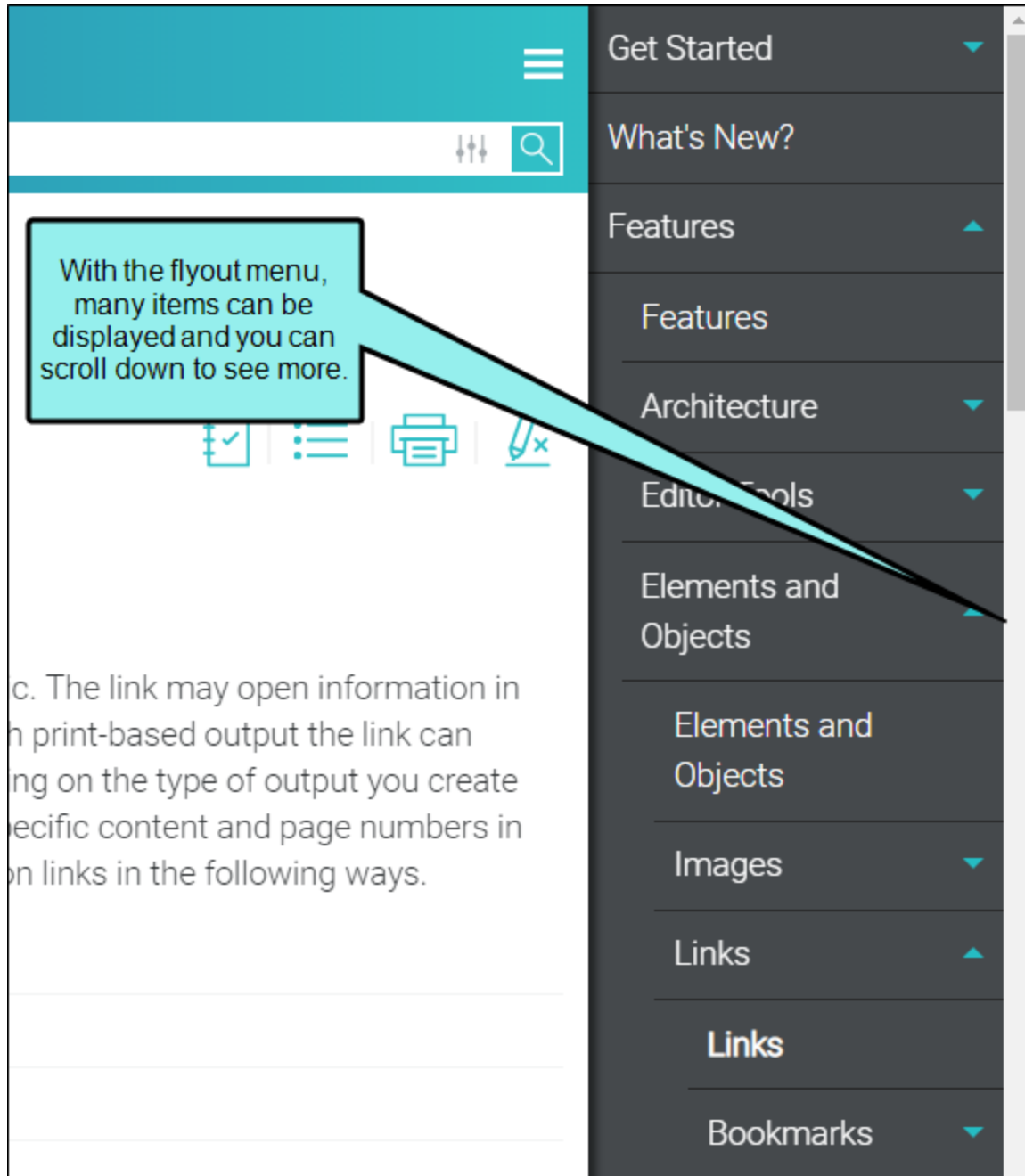
When looking at your TOC file, you might find that you have several topics that do not need to be included in it. Perhaps you decide to keep only the most important topics in the TOC, removing the others. When you generate your output, the most important items will be accessible in the top menu and context-sensitive menus (if you include them). As for the lesser topics you removed, end users can still find those by using search or from links found in other topics, which is what they will most likely do anyway.

# Turn Off Top Navigation

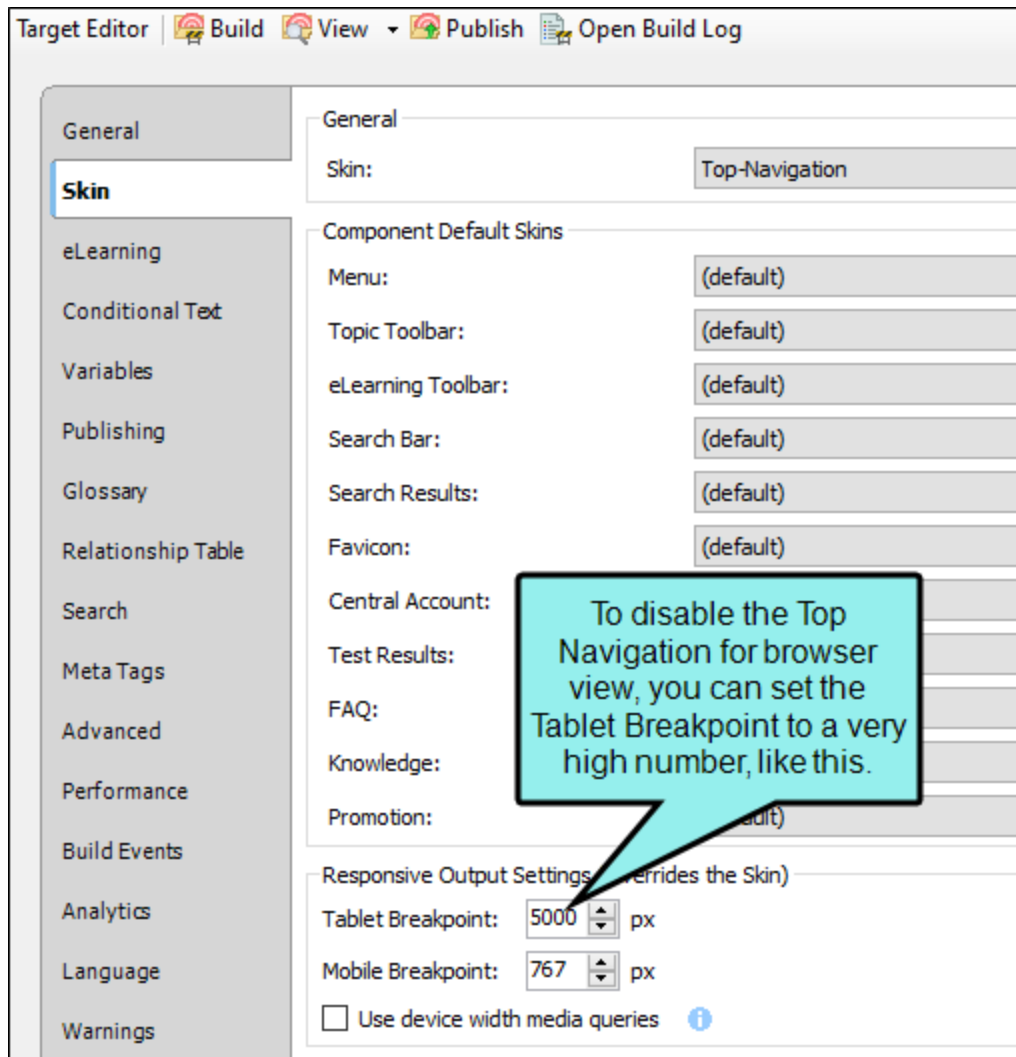
Although this kind of output is called "Top Navigation," it is designed to work responsively, changing its layout if it is being viewed on smaller device such as a tablet or smart phone. When this occurs, the top menu is replaced with a flyout menu on the side.



This kind of layout is better able to display longer lists of TOC items that would not look as good in a top menu.



If you would like this kind of layout for larger browser windows—as well as for smaller tablets and mobile devices—you can turn off the top navigation altogether via the responsive output settings. To do this, open the Target Editor, select the **Skin** tab, and set the **Tablet Breakpoint** to a very high number. Doing this displays the output on extremely large monitors the same way that it looks on small tablets (i.e., with the side flyout menu instead of the top navigation menu). This workaround is probably the easiest way to deal with a long TOC without having to make changes to it.





# Use Side Navigation

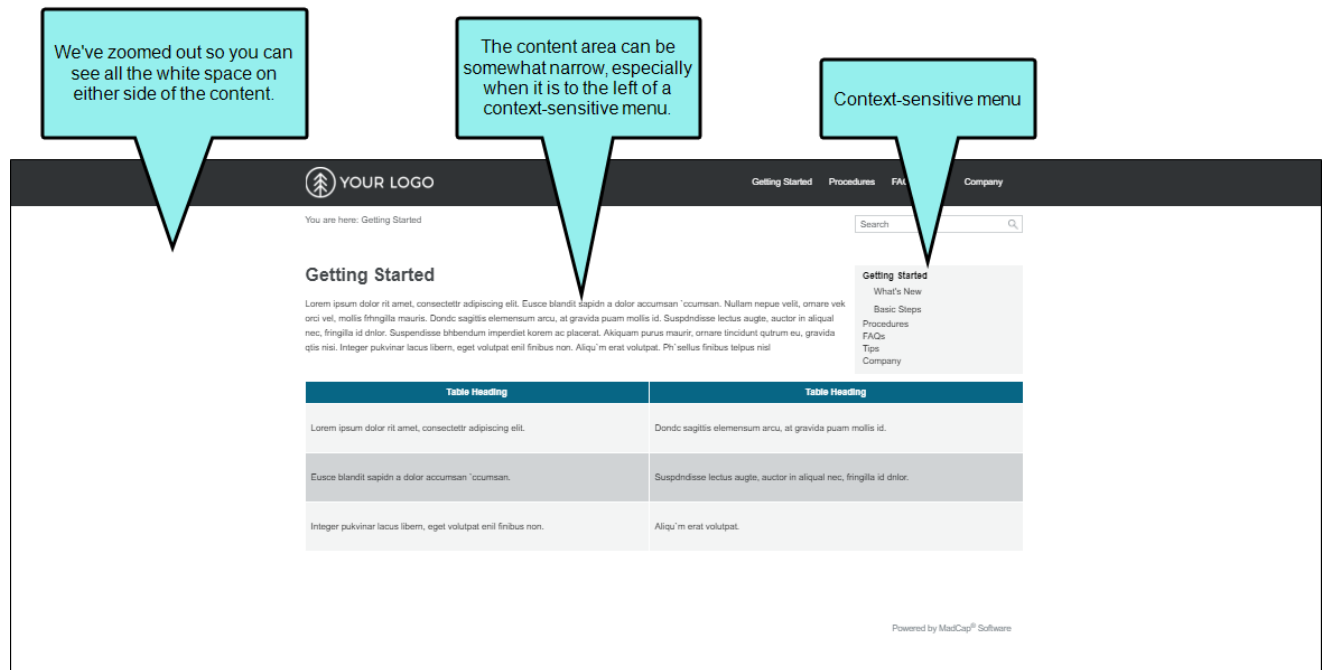
If you decide that you simply do not want to adjust your TOC, you might consider using Side Navigation instead of Top Navigation output. With Side Navigation, your TOC can be as long as you want and it shouldn't make any difference because the navigation will be on the left or right side of the screen, instead at the top. See "HTML5 ide Navigation Output" on page 39.

# Avoid Duplicate File Names When Importing

You might decide to create a new project from one of Flare's Top Navigation templates and then import some of the files from it to your existing project. If so, you should first make sure that you do not have files with the same name in your existing project. You probably don't want a file from the template to overwrite existing files in your project.

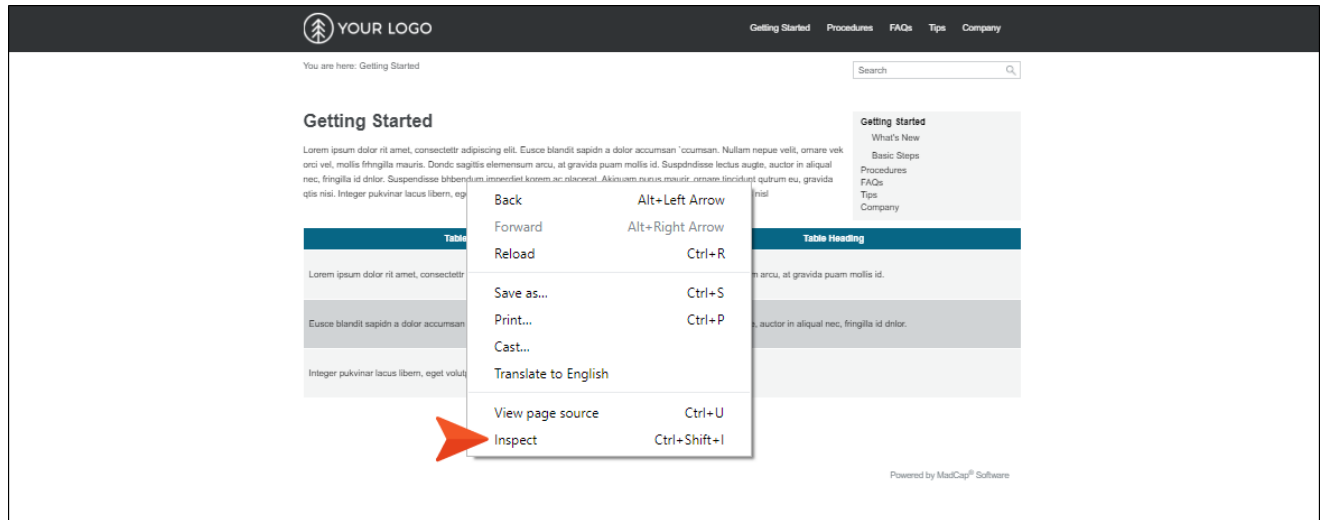
# Consider the Width of Content

If you decide to use Flare's project templates as a basis for your new Top Navigation output, you will notice that we've made the main content area somewhat narrow on the left side of the context-sensitive menu, with content wrapping under the menu.



It's a nice look, but it also means that you might experience issues if you have extra wide content, such as big tables. You probably won't have an issue if the wide content in question wraps under the topic menu. But if you have a long topic menu on a particular page due to the number of related links in the TOC, you could have a challenge displaying the wide content properly.

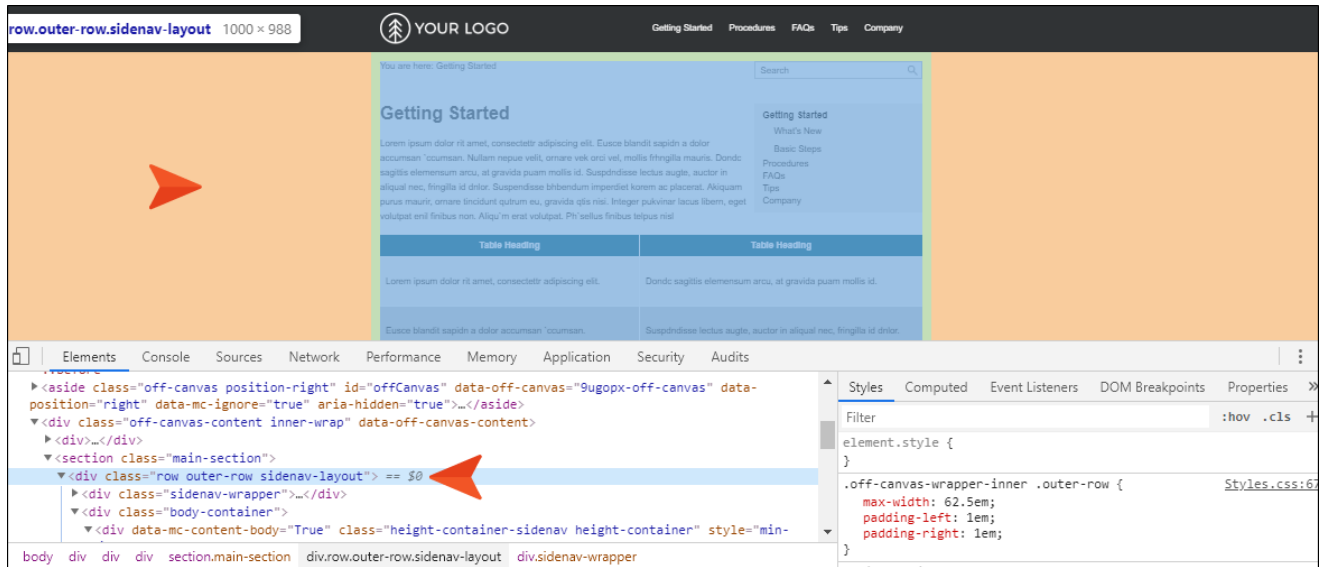
If this is the case, you might consider either changing your content so it does not require so much horizontal space or making the content area wider. If you want to change the width, you can use the "Inspect" feature in your browser to see which style is influencing that look. With a topic in the output open, right-click in the browser and select **Inspect** (the option might be somewhat different depending on your browser).



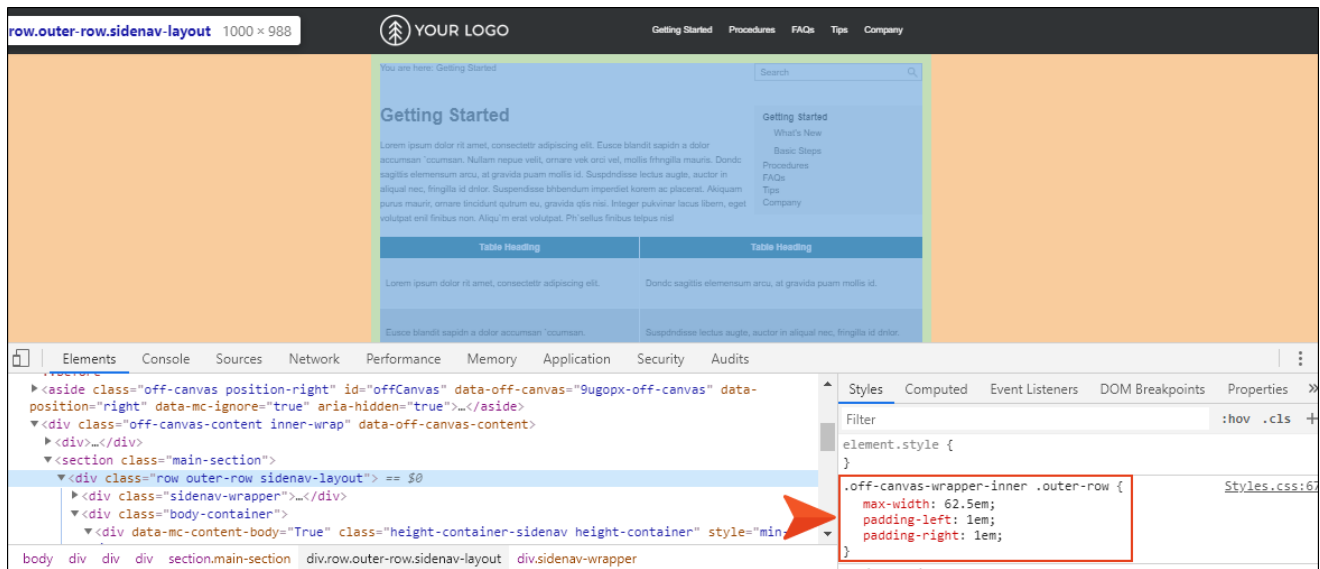
As you hover over styles in the code that is shown, different parts of the output will be highlighted. If you hover over this:

```
<div class="row outer-row sidenav-layout"> == $0
```

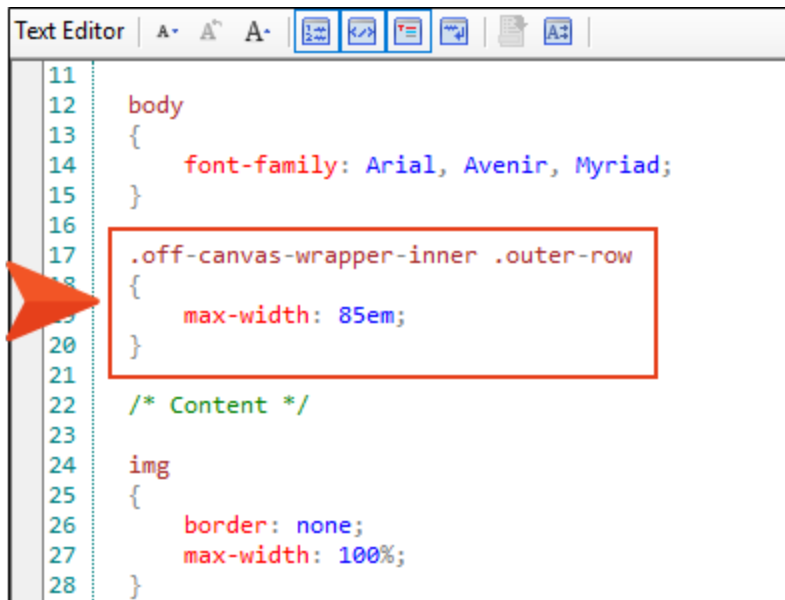
You will notice that the output is highlighted like this:



Then, on the right side, you will see this:

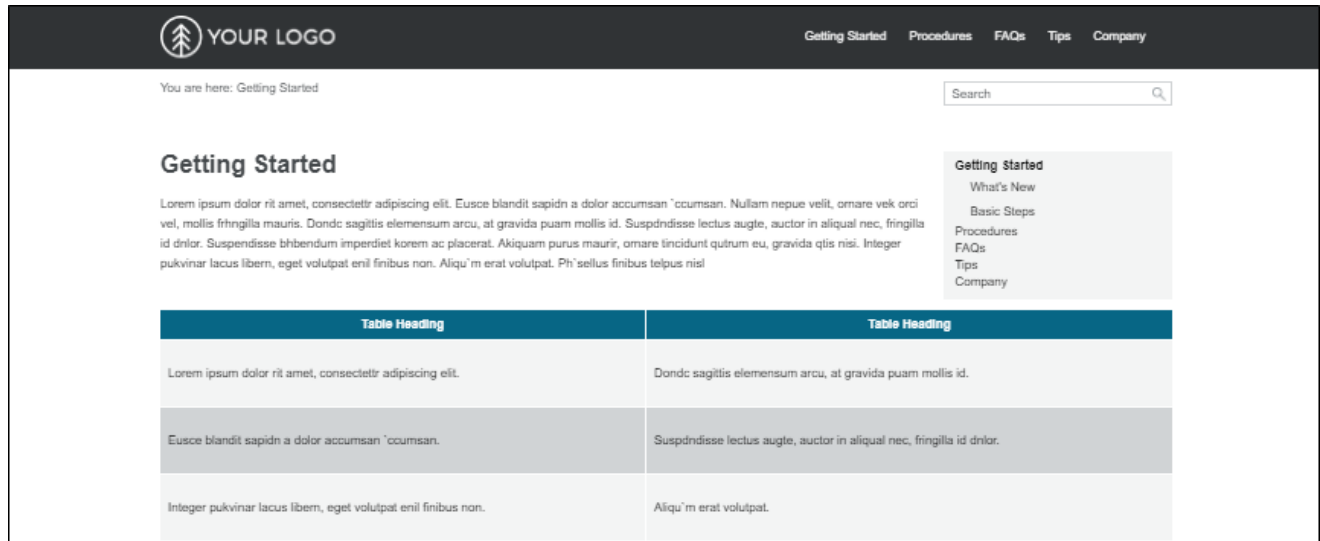


So you can open your stylesheet in the Internal Text Editor and add the following, increasing the value of the max-width property:



```
11
12  body
13  {
14      font-family: Arial, Avenir, Myriad;
15  }
16
17  .off-canvas-wrapper-inner .outer-row
18  {
19      max-width: 85em;
20  }
21
22  /* Content */
23
24  img
25  {
26      border: none;
27      max-width: 100%;
28  }
```

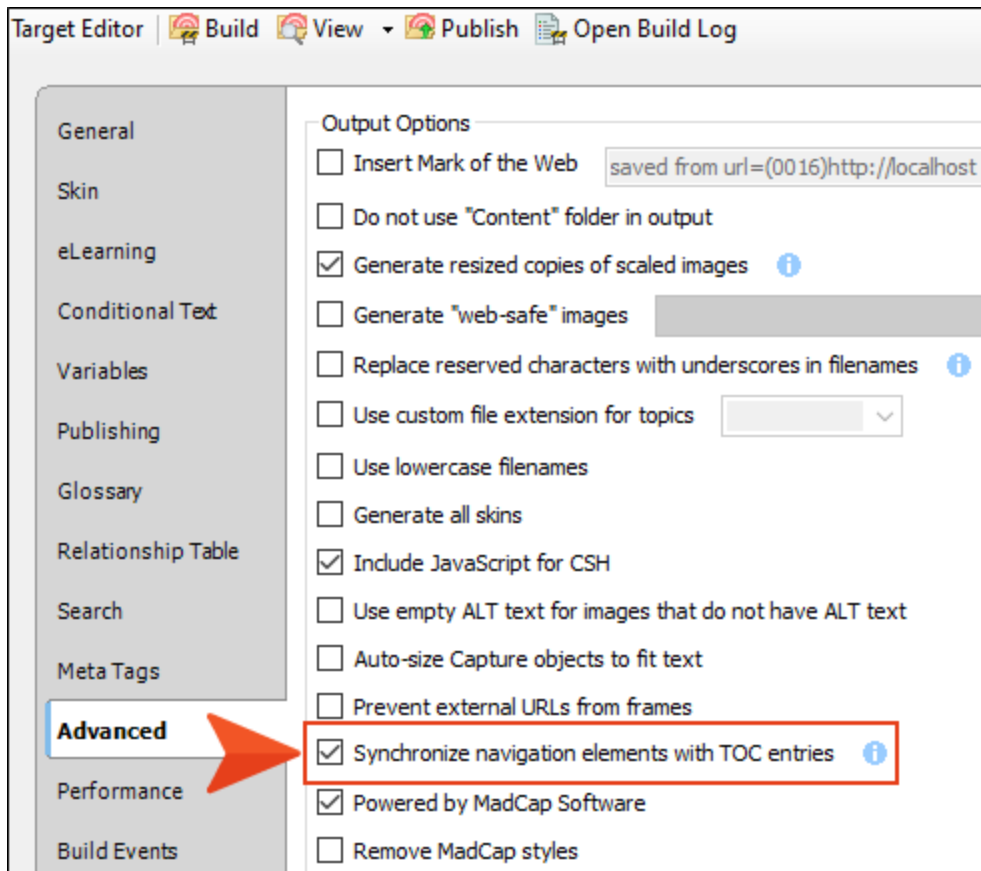
In this example, we increased the max-width to 85em, and as a result the content in our output now looks like this:



## Duplicate TOC Items

If you generate HTML5 output, links to the TOC (e.g., menu items, breadcrumbs, mini-TOCs) are dynamically generated when a particular topic is opened. This lets you keep these navigation elements in sync with your TOC. This is particularly important if the same topic is linked to multiple entries in your TOC; otherwise, menu items and other navigation elements might display for one instance of that topic in the TOC when another instance is preferred.

For HTML5 Tripane output, navigation elements will always remain in sync with your TOC entries. However, for HTML5 Side Navigation, Top Navigation, and skinless outputs, you must enable a feature in your target to synchronize navigation elements with TOC entries.



Synchronizing navigation elements with your TOC entries seems like an obvious decision, and for some authors it is. However, other authors might choose not to do this for their Side Navigation, Top Navigation, or skinless outputs. There are pros and cons.

**NOTE** If you decide not to use this option for your HTML5 Side Navigation, Top Navigation, or skinless output but you are still concerned about having duplicate TOC links, there is a workaround. You can create a snippet that holds all of the content for the topic. Then create multiple topics (as many topics as you need links to it in the TOC), and insert that snippet into each one. Rather than linking to one topic in multiple places in the TOC, you can link to a different topic in each place. By doing that, your navigation elements will point to the appropriate locations. You probably want to disable searching in all but one of those topics; otherwise, they will all show up in the search results.

# Producing Top Navigation Output

After making decisions about how you want your Top Navigation output to look and work, you can follow the necessary steps to produce it.

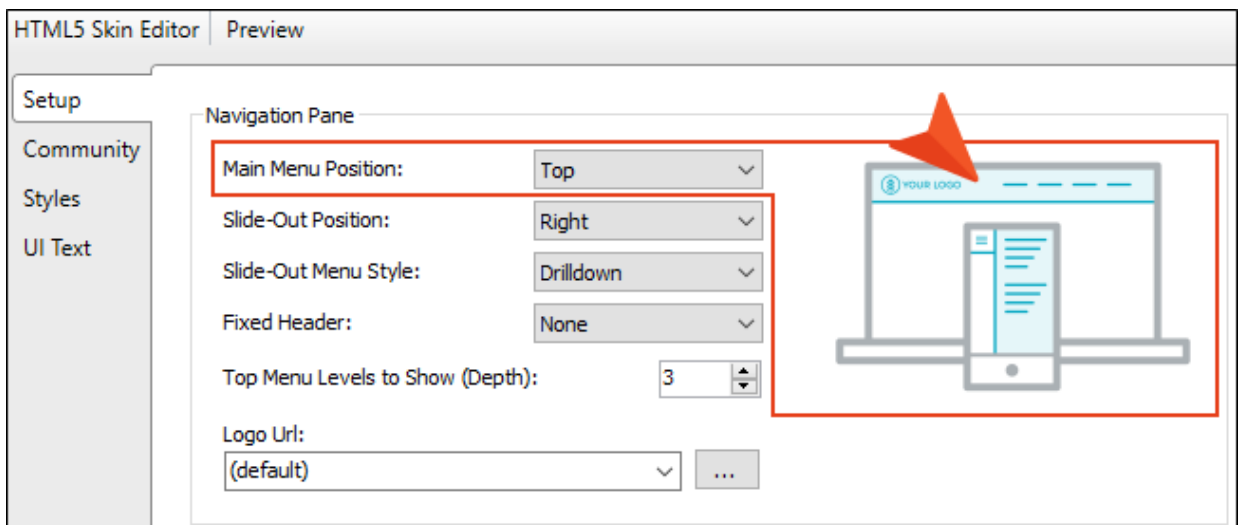
## How to Produce Top Navigation Output

1. Add a Top Navigation skin to your project (**Project > New > Add Skin**).
2. (Optional) You can edit the Top Navigation skin and its styles, just like you can edit other types of skins.

Following are a few of the more common adjustments that are made in skins:

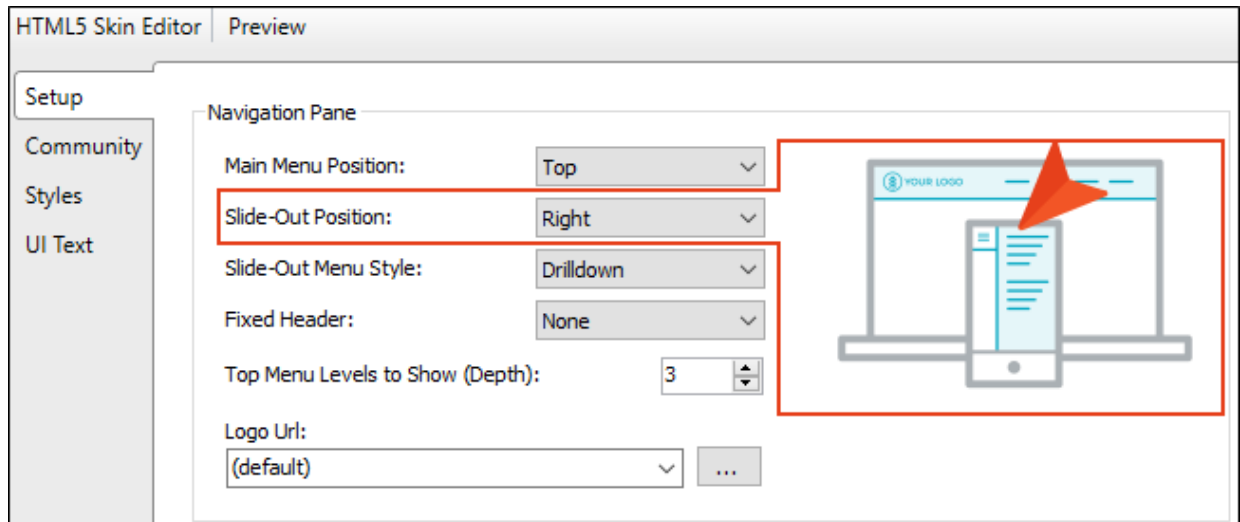
### MAIN MENU POSITION

On the **Setup** tab of the Skin Editor, you can set the main menu position to **Top**, **Left**, or **Right**. This is the primary menu (links to topics) that is based on your TOC when the output is being viewed on a large screen (not tablet or mobile). In a new Top Navigation skin, this field is set to Top by default, meaning the main menu will be placed at the top of the screen. If you choose Left or Right, you will no longer be using a Top Navigation skin, but rather a Side Navigation skin, where the main menu is positioned at the left or right side of the screen.



## SLIDE-OUT POSITION

On the **Setup** tab of the Skin Editor, you can position the pane either on the **Left** or **Right**. This is the flyout menu pane that is seen on the side of the output when it is being viewed on a tablet or mobile device, replacing the menu that is seen on larger screens.



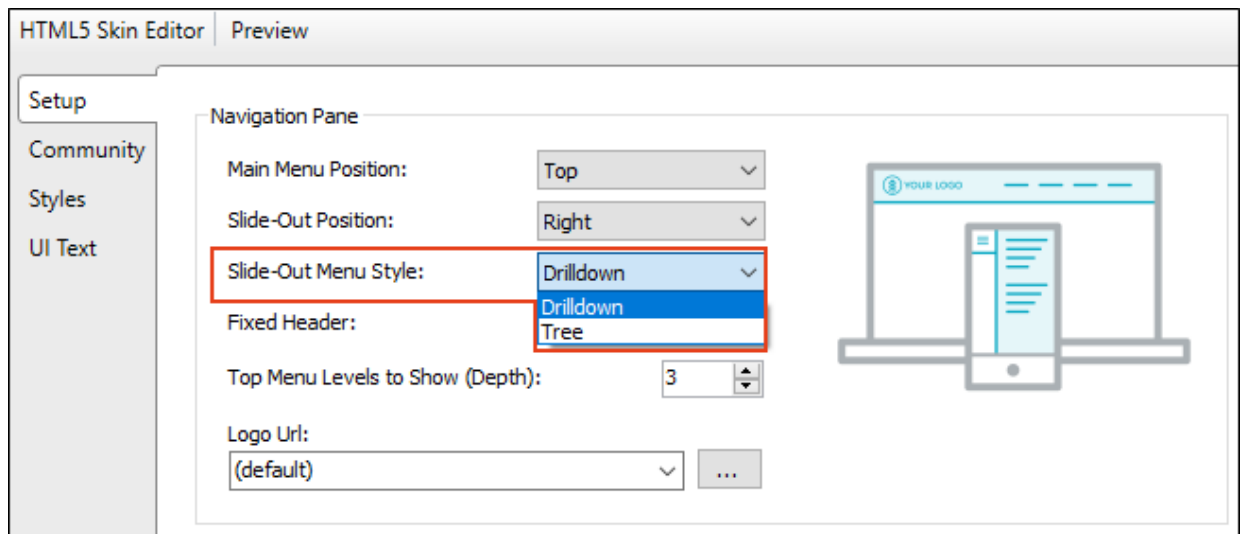


## SLIDE-OUT MENU STYLE

On the **Setup** tab of the Skin Editor, you can choose the look of the slide-out menu for Top Navigation output (i.e., the menu that opens when you click the “hamburger” icon in tablet or mobile view).

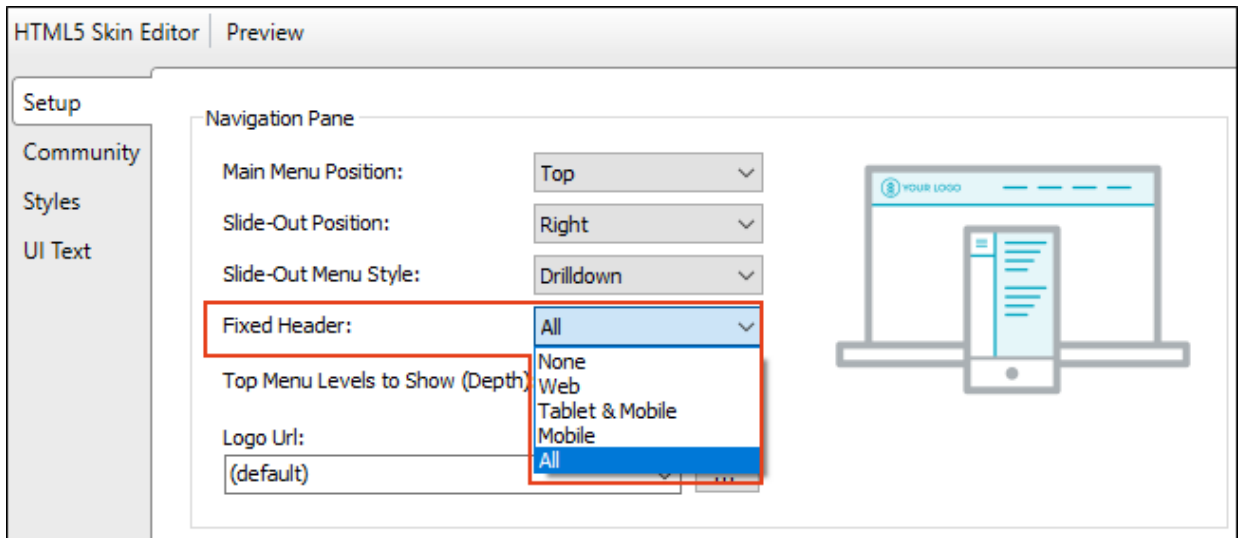
You can select a drilldown format. Any item containing sub-items has a double-arrow next to it. When you click on such an item, the entire menu changes, showing only that item and its sub-items.

Alternatively, you have the option of using a tree format instead. With this format, any item containing sub-items has a down arrow next to it. When you click on that item, the sub-items are shown, but the rest of the menu remains in view as well.



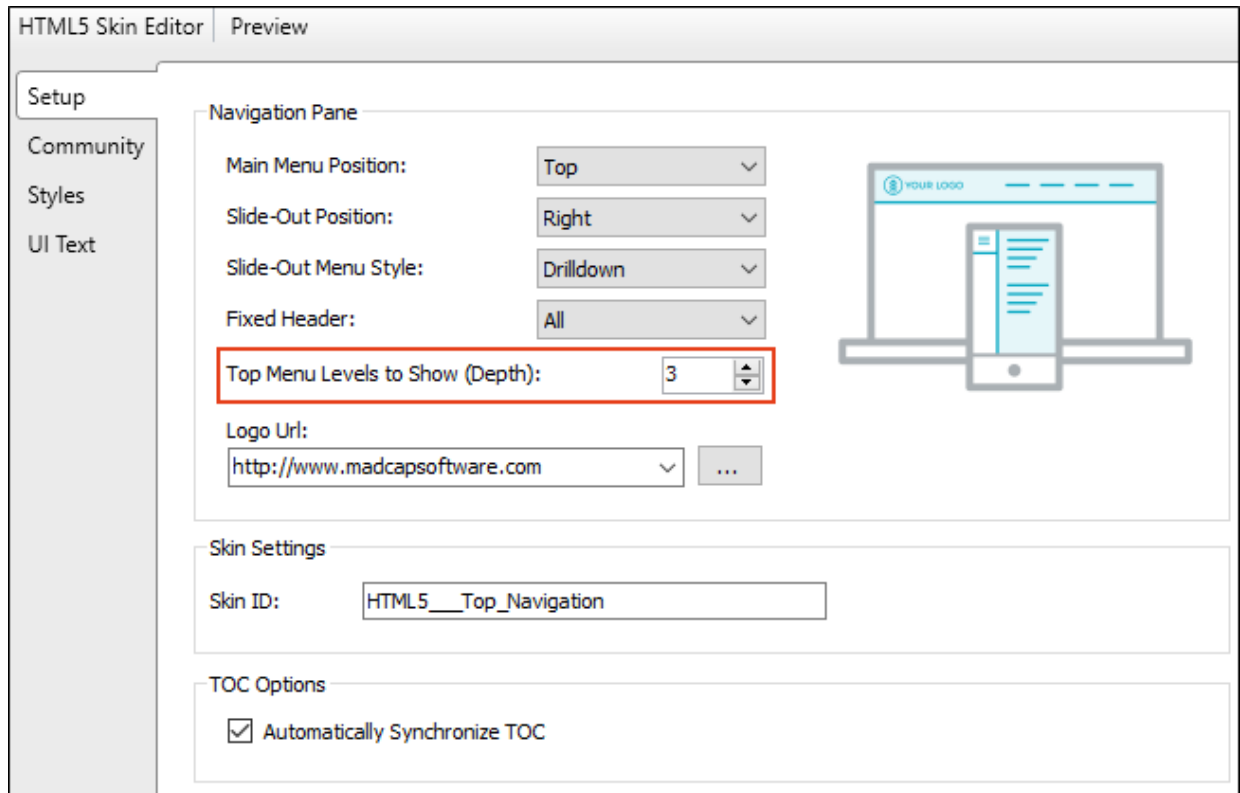
## FIXED HEADER

On the **Setup** tab of the Skin Editor, you can fix the header portion of the topic so that it stays in place when users scroll down to see more content. For Top Navigation skins, you can do this for web, tablet, or mobile screens (or all of them).



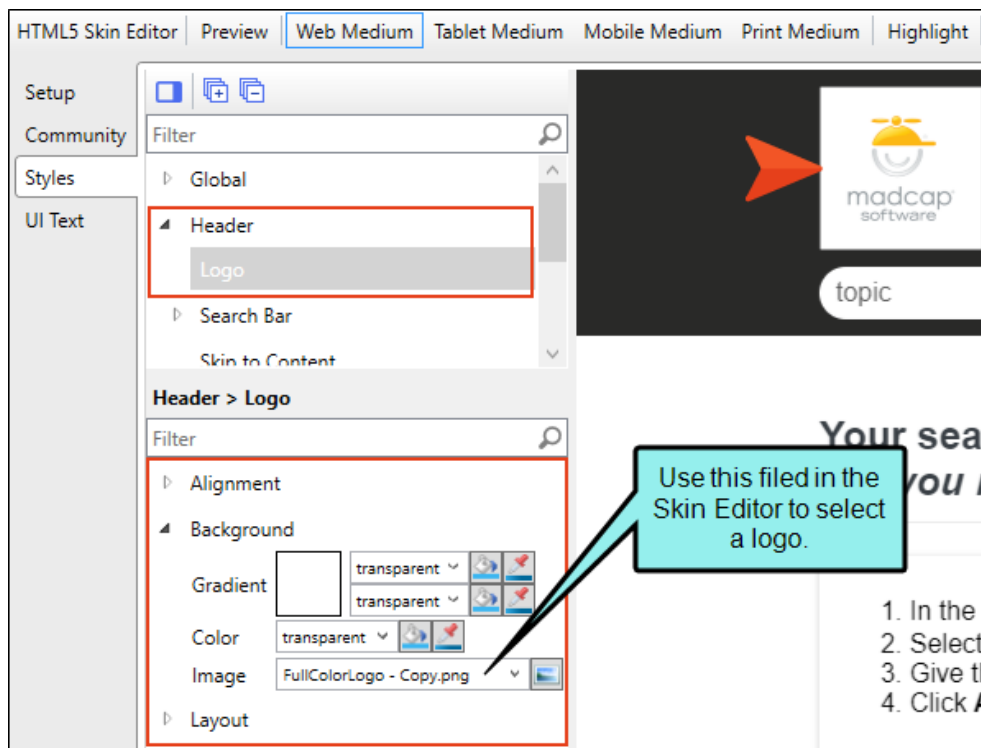
## TOP MENU DEPTH

On the **Setup** tab of the Skin Editor, you can specify how many levels of your TOC items are included in the top menu navigation. The default is 3. It is recommended that you avoid including too many depth levels in the top navigation.

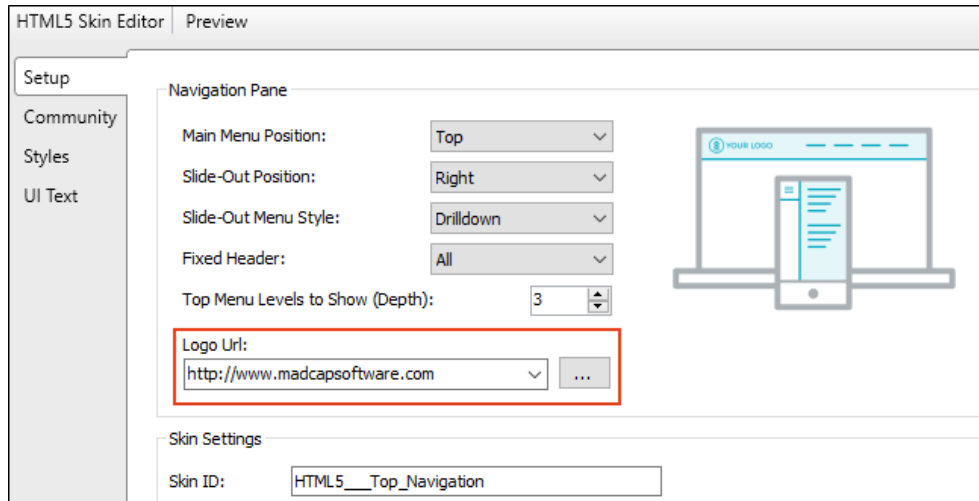


## LOGO

On the **Styles** tab of the Skin Editor, you can replace the generic logo with your own (if you have not already done so using the branding feature).

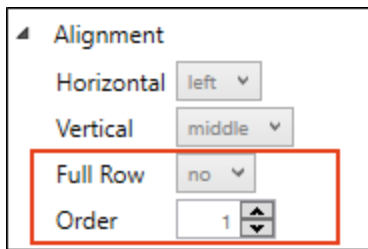


Whatever image you use for your logo, it is automatically set to link to the topic that you've set as the **Startup Topic** on the **General** tab of the Target Editor. However, you can select a different topic or even enter the URL to your company's website instead (remember to include `http://` at the beginning of the path if you link to a website). This can be done on the **Setup** tab of the Skin Editor.



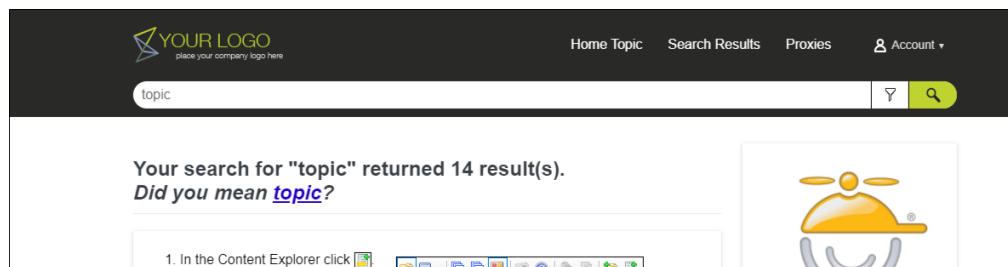
## ALIGNMENT FOR LOGO, SEARCH BAR, AND MENU

There are some flexbox-related options on the **Styles** tab that let you align the logo, search bar, and menu in the header. These fields called “Full Row” and “Order.”

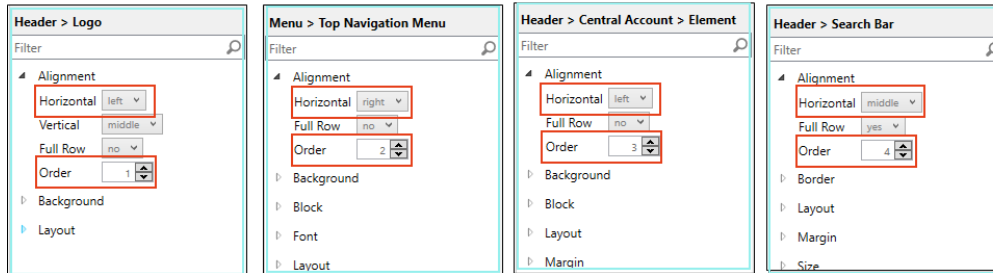


- **Full Row** This option allows the element to take up all of the space horizontally in the header.
- **Order** This option determines which element comes first, second, or third in the header. Top Navigation has three elements (logo, search bar, and menu), but Side Navigation has only two (logo and search bar).

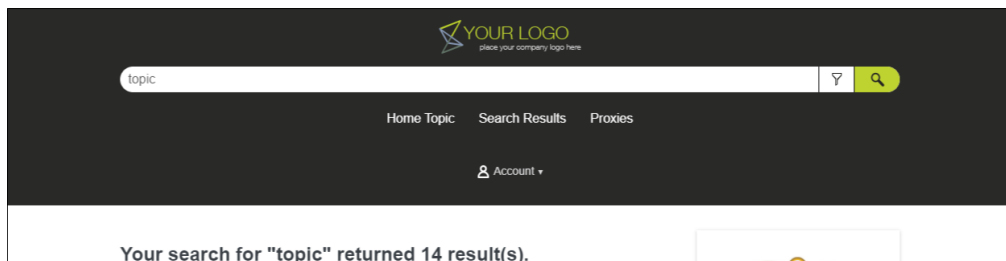
☆ **EXAMPLE** You create a new Top Navigation skin. Initially it looks like this:



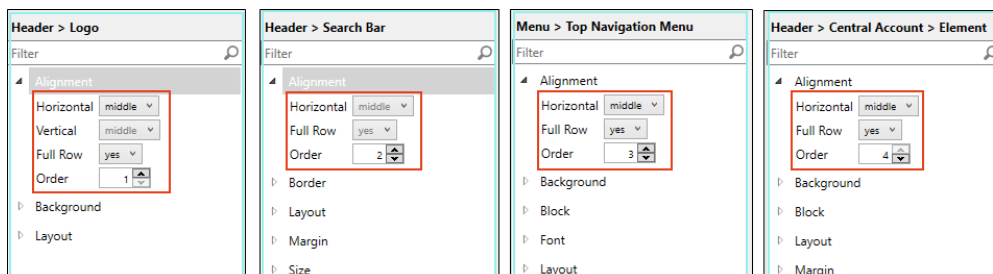
☆ The logo comes first (Order=1) and is positioned left. The menu is second (Order=2) and is positioned right. The Central account link is third (Order=3). The search bar is fourth (Order=4) and is set to take up a full row.



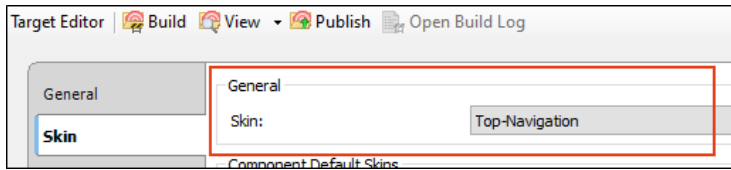
Now maybe you want each element to take up all the space on its row, centered, with the logo first, the search bar second, followed by the menu, and then the Central account link.



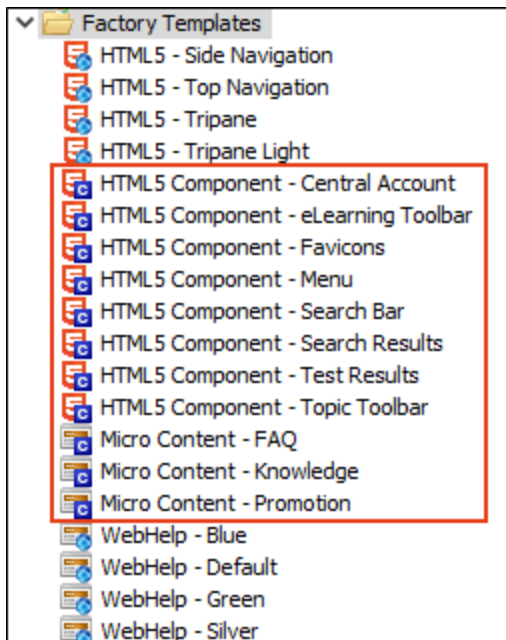
In that case, you would change the settings like this:



3. Open your HTML5 target and on the **Skin** tab, associate the Top Navigation skin with it.



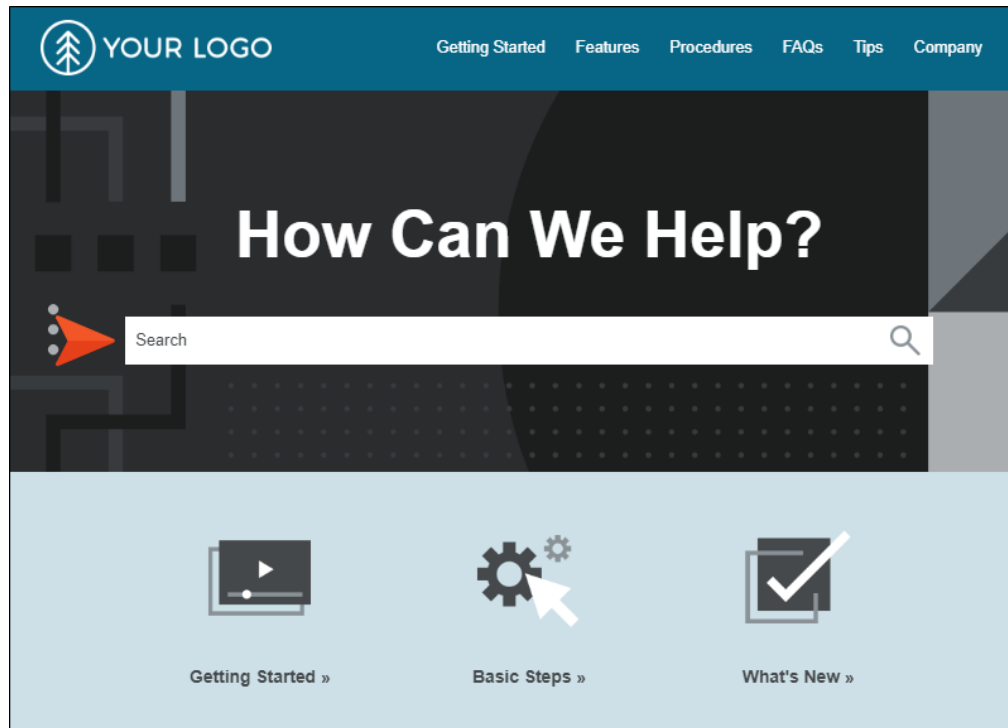
4. (Optional) By default, the Top Navigation skin includes a menu and search bar at the top of topics, but if you want to add elements such as these directly in topics or template pages, you can add small HTML5 skin components to your project to include special menus, search elements, and toolbars. This is the same process as adding a regular skin (**Project > New > Add Skin**), except you are adding an individual component.





☆ **EXAMPLE** In Flare's Top Navigation project templates, we've added Search Bar and Menu components.

One of the Search Bar skin components is used for a prominent search bar within the Home topic.



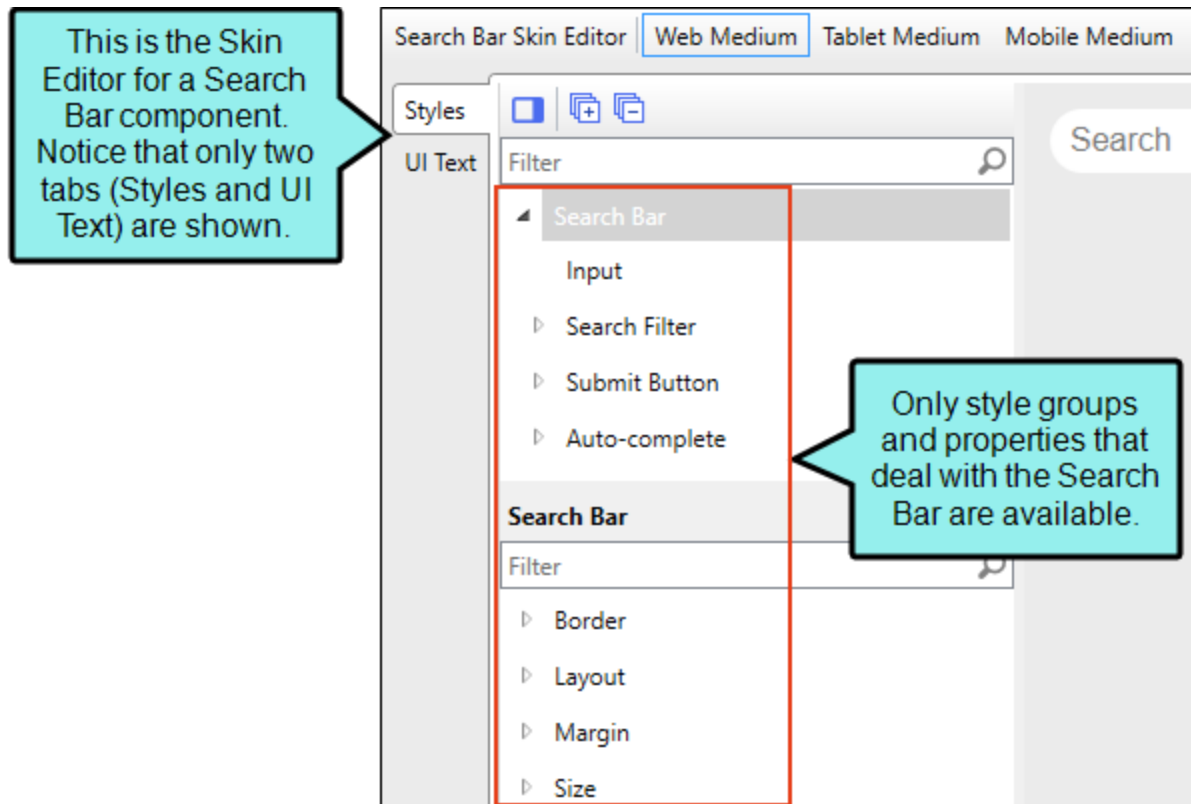
- ☆ The Menu skin component is used for a context-sensitive side menu that displays to the right of topics, showing links to other topics that are next to them in the TOC.



Actually, you can add features such as these in your output without adding skin components. You can do this by inserting the appropriate proxies (see step 11). The skin components are used to *create a custom look* for the elements, while their related proxies are used to *perform the actual generation* of the elements. So although adding skin components is an optional step, it is likely that you will want to do it if you decide to add navigation proxies.

For more details about these elements, see "Skin Components and Proxies" on page 117.

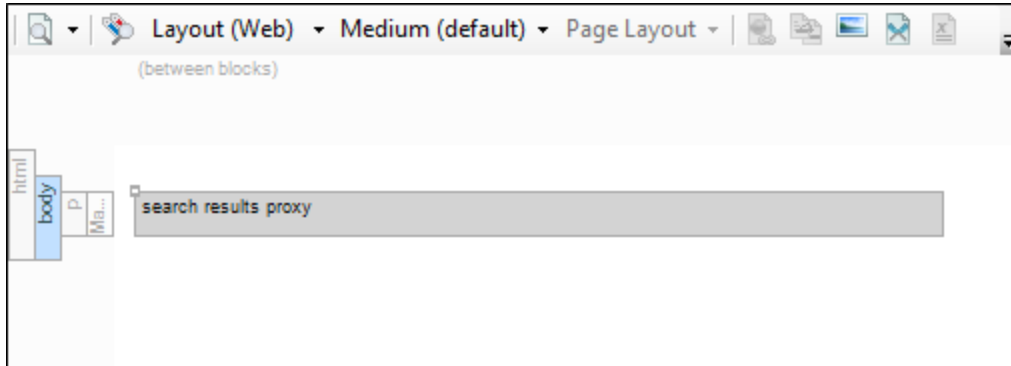
- (Optional) In the same way that you can edit a regular skin, you can edit individual skin components. When you open a component to edit it, you will notice that the Skin Editor is slimmed down to show only the properties and fields related to that component.



- Create a Home topic. This is the first topic shown when users open the output. It is just a regular topic, but you might decide to design it to stand out from the rest of the topics. Make sure to set this topic as your startup topic. You can do this by opening the Target Editor, selecting the **General** tab, and choosing the topic in the **Startup Topic** field.

- (Optional) You can create other topics, including regular topics that contain the bulk of your content.

You can also create a special topic to hold generated search results. A Top Navigation skin already includes a search bar, and you do not need to do anything else to incorporate search into your output. When an end user searches for text, Flare displays the results in its default format. However, if you want to be able to customize your own search results page, you can create a topic specifically for that purpose and insert a Search Results proxy into it (**Insert > Proxy > Insert Search Results Proxy**). This works with the Search Results skin component that you can add to a project (see step 4).



## 8. Add topics to a TOC file.

The Top Navigation skin uses the structure and contents of your TOC to populate the menu that appears at the top of topics. In addition, Flare's Top Navigation templates include a template page with a special Menu proxy inserted into it. This proxy creates context-sensitive menus that are displayed on the side of content, displaying links for topics that are located in the same TOC book (as well as the parent and child TOC topics).

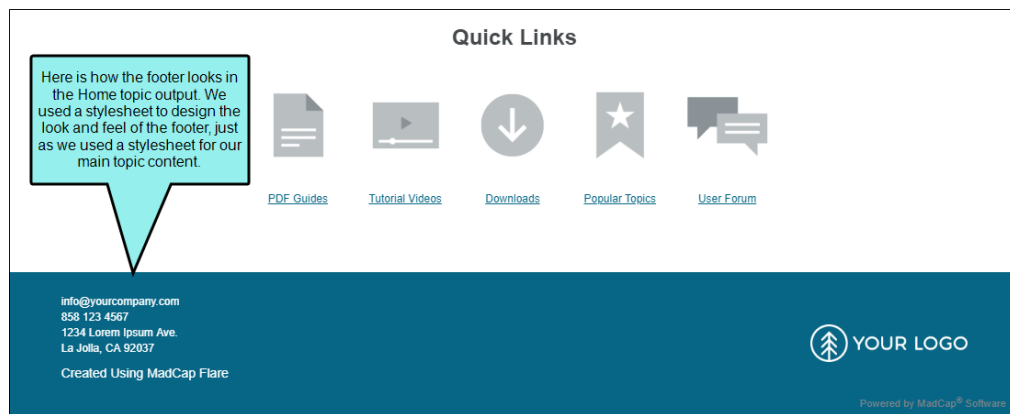
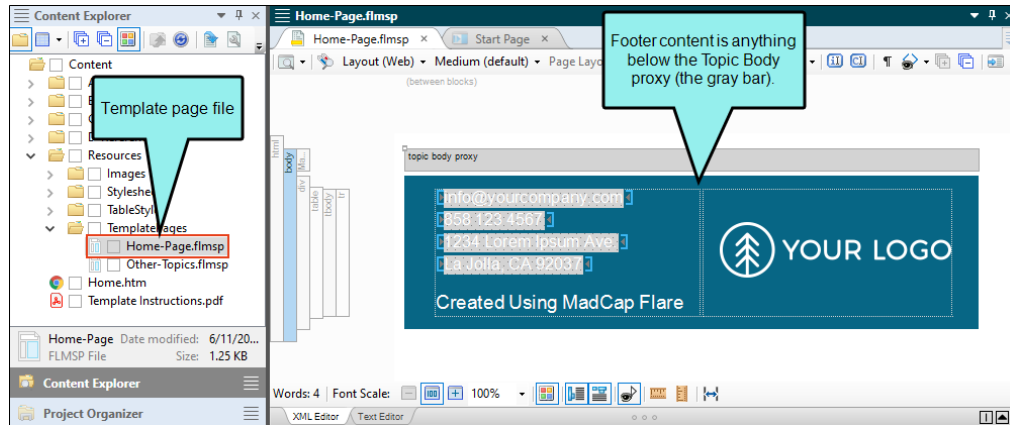
The image shows a TOC Editor window on the left and a rendered web page on the right. The TOC Editor shows a tree structure with levels: Getting Started, Features (with sub-items Feature 1, 2, 3), Procedures (with sub-items Procedure 1, 2, 3), FAQs, Tips, and Company. The rendered page shows a navigation menu with 'YOUR LOGO' and links for Getting Started, Features, Procedures, FAQs, Tips, and Company. A search bar is present. The page content is titled 'Feature 2' and includes placeholder text. Callouts explain: 1) 'Notice that the first-level books in the TOC file...' pointing to 'Getting Started', 'Features', and 'Procedures' in the TOC Editor. 2) '... are automatically shown as menu items at the top. When you hover over one of these, the TOC items under it are shown in a submenu.' pointing to the 'Features' menu item. 3) 'In the example below, the topic "Feature 2" is open.' pointing to the 'Feature 2' sub-menu item. 4) 'Submenus are shown when users hover over the top root menu.' pointing to the 'Features' menu item. 5) 'This context-sensitive menu shows the TOC items in the same book as the open topic.' pointing to the sub-menu for 'Feature 2'.

✔ **TIP** Although Flare lets you create books in the TOC file that do not link to anything (i.e., merely using the book to organize the TOC), it is a best practice for Top Navigation output to make sure that all TOC books and items are linked to something.

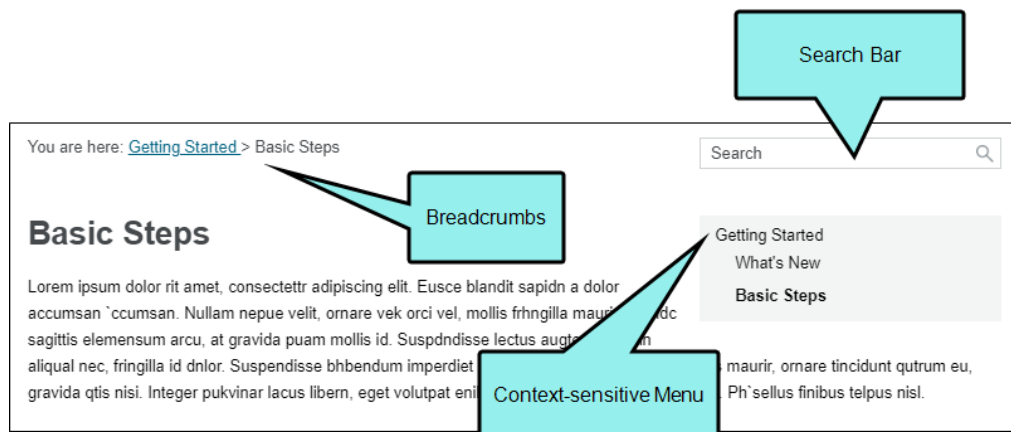
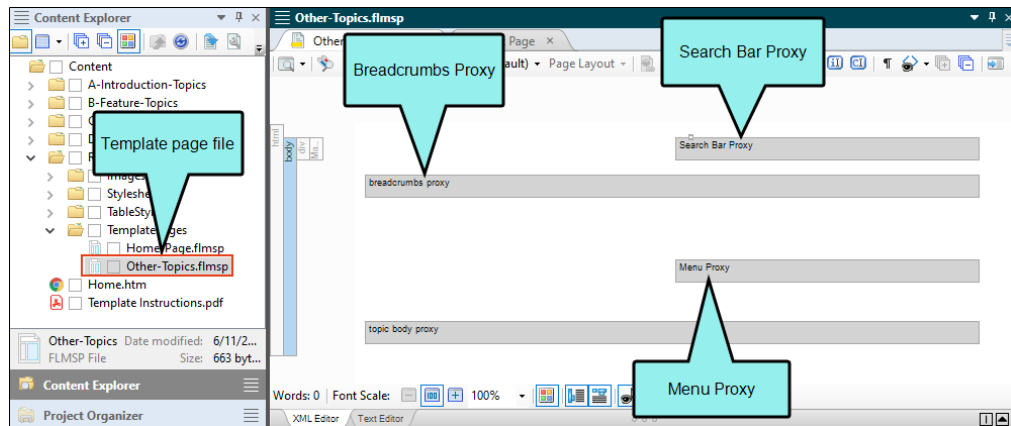
✔ **TIP** It is standard practice in web design to not include your Home page as a menu text link. The default behavior in Top Navigation output is to link the logo to the Home page, so it is not necessary to add this topic to your TOC.

9. (Optional) It is not mandatory to create any template pages in order to have Top Navigation output. However, template pages can be particularly useful, especially if you want the same content to automatically show up at the bottom or top of topics. They can also useful if you want your Home page's design to be much different from that of the rest of your topics. So if you would like to incorporate one or more template pages, create them.

☆ **EXAMPLE** In Flare's advanced Top Navigation templates, we wanted to show footer content at the bottom of the Home page, so we created a template page just for that topic, and we added footer content under the Topic Body proxy.



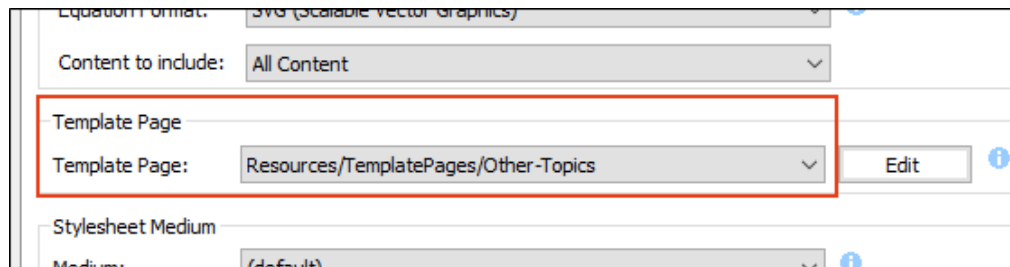
☆ Then on the rest of the topics we wanted breadcrumbs to be displayed above the topic content. So we created a second template page (“Other-Topics.flmsp”) for all of those topics and inserted a Search Bar proxy , a context-sensitive Menu proxy, and a Breadcrumbs proxy above the Topic Body proxy. Even though the Menu proxy was added above the Topic Body proxy, it was styled to display to the right of the topic content, with content wrapping under it.



10. (Optional) You can select a template page on the **Advanced** tab of your HTML5 target.

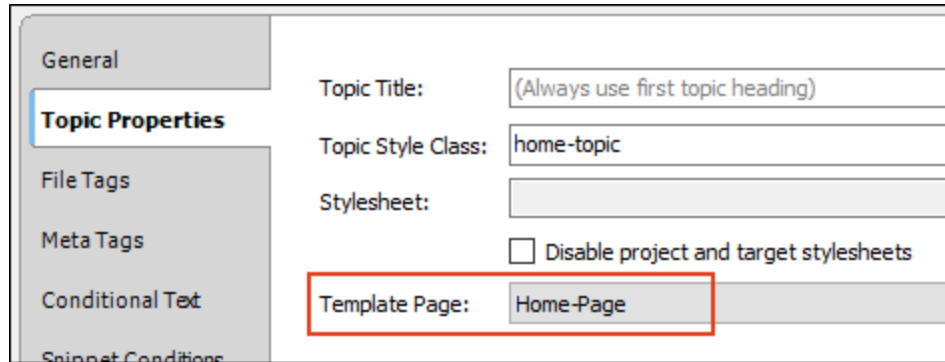
This tells Flare that, unless otherwise directed, all topics will use the template page you selected. But if you want to use a second template page, you can open the Properties dialog for a particular topic and point to a different template page. In that case, the template page set in the target is typically the one that the majority of your topics will use. But for the other topic(s) using a different template page, you can override the target setting.

☆ **EXAMPLE** In this example, the template page that we want to use for most of our pages in the output is named "Other-Topics." So on the **Advanced** tab of the Target Editor, we selected it.





☆ Then we opened the Properties dialog for the Home topic (open topic, then **File > Properties**). On the **Topic Properties** tab, we clicked in the **Template Page** field and selected **Home-Page**. Now this topic will use the "Home-Page" template page instead of the other one for the rest of the topics.



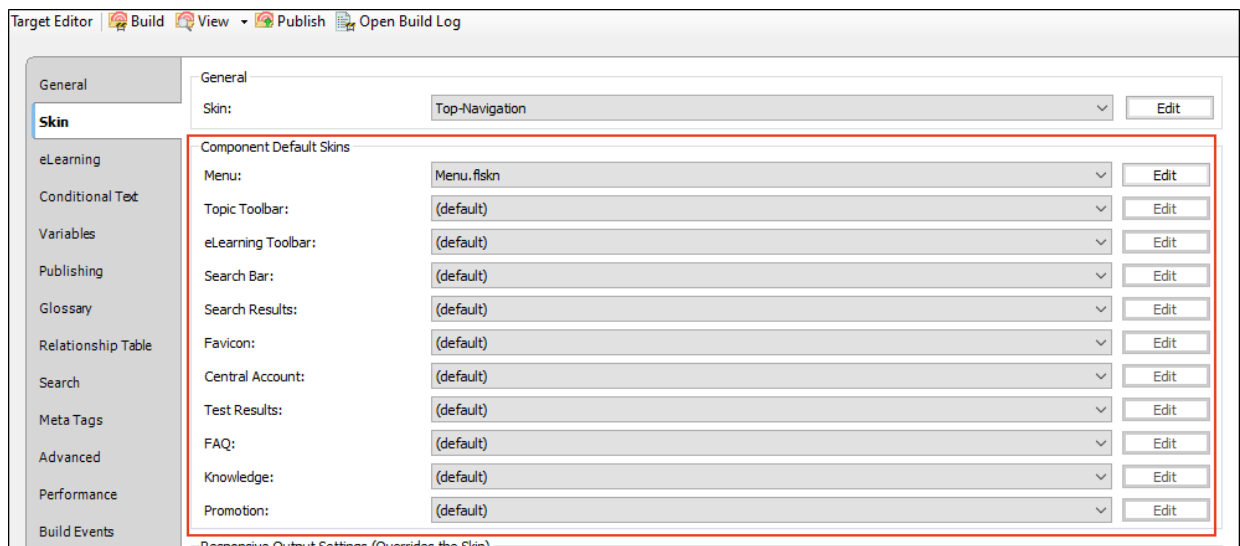
11. (Optional) You have the option of inserting a variety of proxies (**Insert > Proxy > [Name of Proxy]**) into your template page(s) and topics. This includes proxies that correspond to the different kinds of skin components you can add to your project (e.g., Menu, Topic Toolbar, Search Bar, Search Results).

For more details, see "Skin Components and Proxies" on page 117.

12. (Optional) If you have added skin components to your project, you can associate them with an entire target or with individual proxies. If one skin component is associated with the target and a different one is associated with a proxy, the one associated with the proxy has precedence.

## TARGET

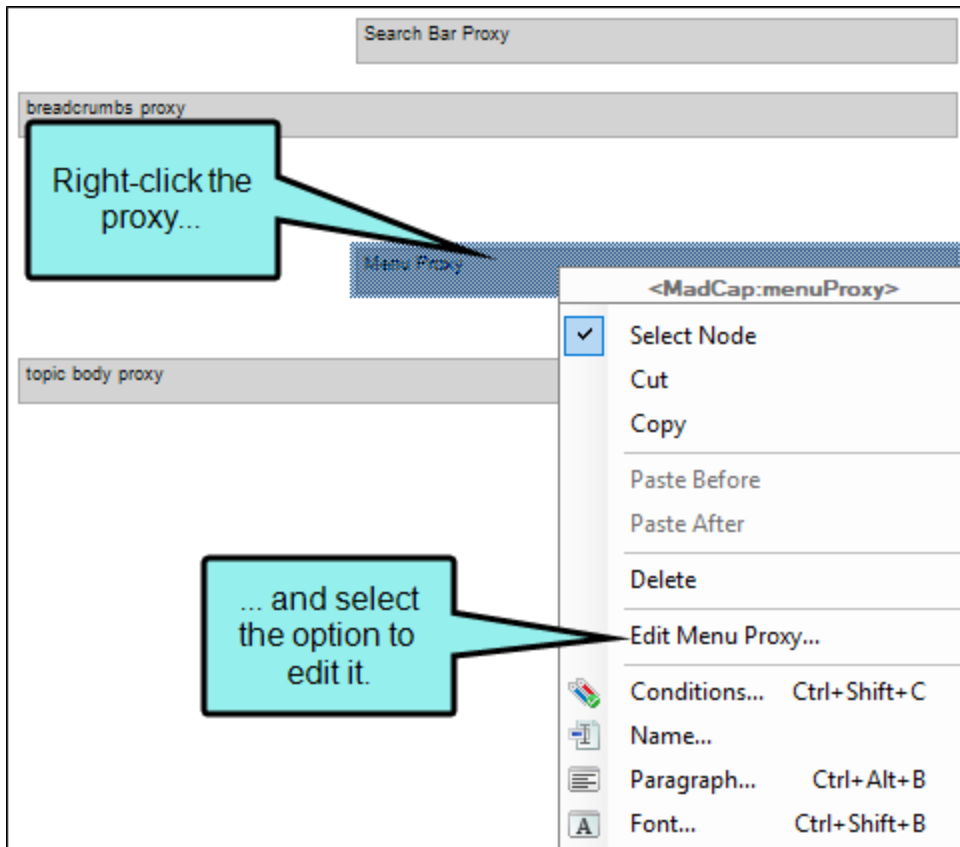
To associate a skin component with an HTML5 target, open the **Skin** tab in the Target Editor, then make your selection in the **Component Default Skins** section



It is not necessary to select anything in these fields, especially if you have only one type of a particular skin component (e.g., one Topic Toolbar component). But if you have more than one type of a skin component (e.g., two Topic Toolbar components), you can choose one of them as the default for all of the topics in the target. Then for the other component(s) of that same type, you can override the target setting by pointing to it in the relevant proxy (see below).

## PROXIES

To associate a skin component with a proxy, open the template page or topic where you've inserted the proxy. Then right-click the proxy and use the dialog to choose the skin component. Some proxies have additional settings, such as TOC depth for Menu proxies.



Here is an example of a dialog for setting options in a Menu proxy.

Inserting a menu proxy allows you to include a stylable menu inside a topic or template page.

Menu Content

Use TOC, Browse Sequence, or Headings:  
(default) ▾

Context sensitive  
 Include parent  
 Include siblings  
 Include children


Levels to Show (Depth):  
(default) ▾

Menu Style

Skin File:  
Project/Skins/SideMenu ▾

OK Cancel

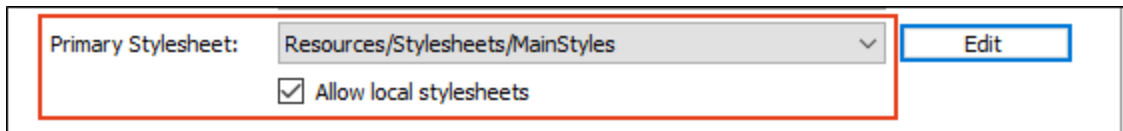
For more information about associating skin components with proxies, see "Skin Components and Proxies" on page 117.

 **NOTE** It's possible that you will not need skin components in your project at all. If you have not added a skin component but you insert a proxy related to it, Flare will just use the default design from the application.

13. Create and edit stylesheets to control the look of your content in the output.


14. Make sure your stylesheet(s) are associated with the appropriate files.

If you decide to use multiple stylesheets in your project (e.g., one for most topics and a second one for your Home page topic), we recommend that you set your main stylesheet (the one intended for most topics) at the project or target level. When you do this, enable the **Allow local stylesheets** option.



Then associate other topics (e.g., your Home page) with the a different stylesheet.

If you are using a separate template page for your Home topic, you will want to associate that template page with the other stylesheet as well.

 **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 primary stylesheets in your project. Without a primary 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.

15. (Optional) Responsive output is automatically enabled for Top Navigation skins. But you can set a few additional options on the **Skin** tab of the Target Editor. See "Responsive Skins" on page 150.

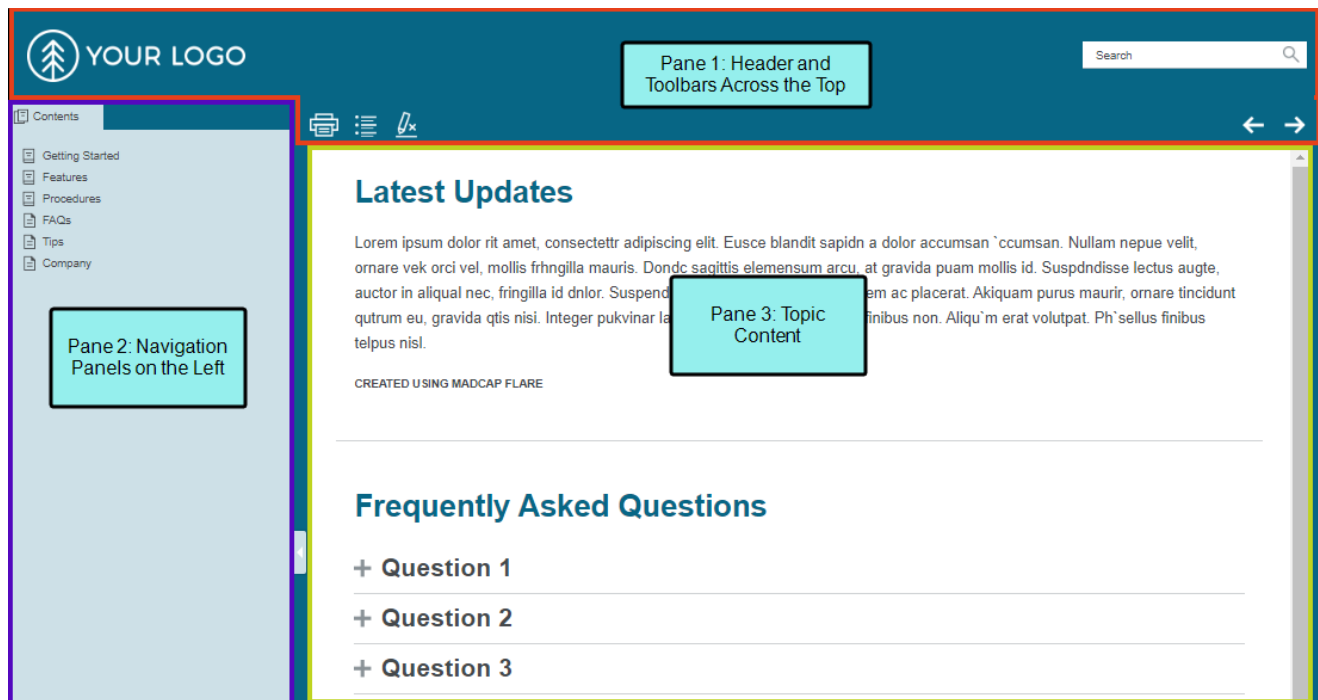
When a skin is responsive, the navigation elements are automatically adjusted depending on the size of the screen.

- **Tablet Breakpoint** Enter the number of pixels for the maximum width of a Tablet view.
- **Mobile Breakpoint** Enter the number of pixels for the maximum width of a Mobile (or phone) view.
- **Use device with media queries** Select this option if you want to base the responsive output not merely on the width of the output display, but on its width in the actual device (browser, tablet, or mobile). In other words, if you do not have this option selected and you view the output on a full browser, you can see the tablet and mobile layouts simply by reducing the size of the browser window. But if you select this option and you reduce the width of the browser, the layout will not change to the tablet or mobile formats.

## CHAPTER 5

# HTML5 Tripane Output

In Flare, Tripane is a type of HTML5 output that lets you generate output in a traditional format with three distinct frames (a navigation pane on the left, a toolbar and search on the top, and the main body pane).

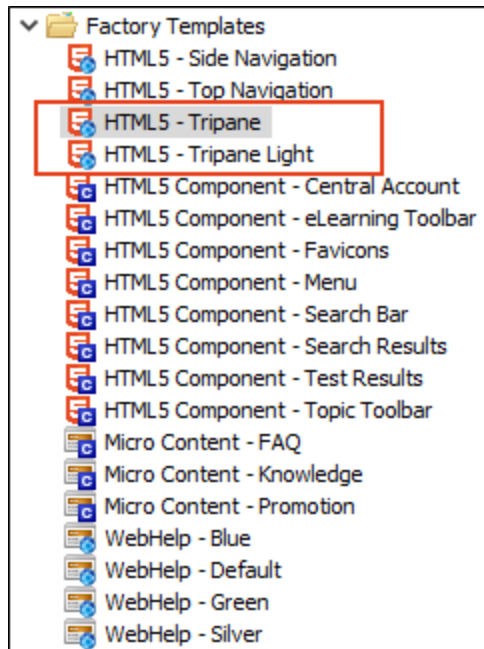


This chapter discusses the following:

Tripane Skins .....	108
Setting Tripane Output .....	110
Setting Responsive Options for Tripane Output .....	111

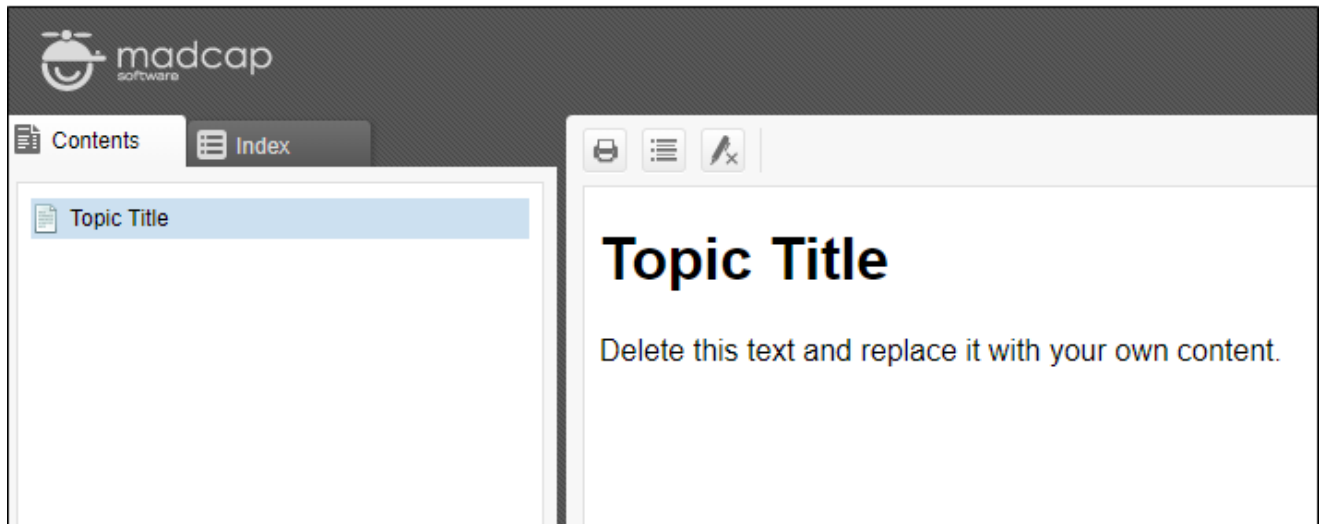
# I Tripane Skins

A Tripane skin is the main element involved in creating HTML5 Tripane output. Flare provides two kinds of Tripane skins—one version with a dark background and another version with a light background. You can add a Tripane skin in the same way that you add other kinds of skins to a project.

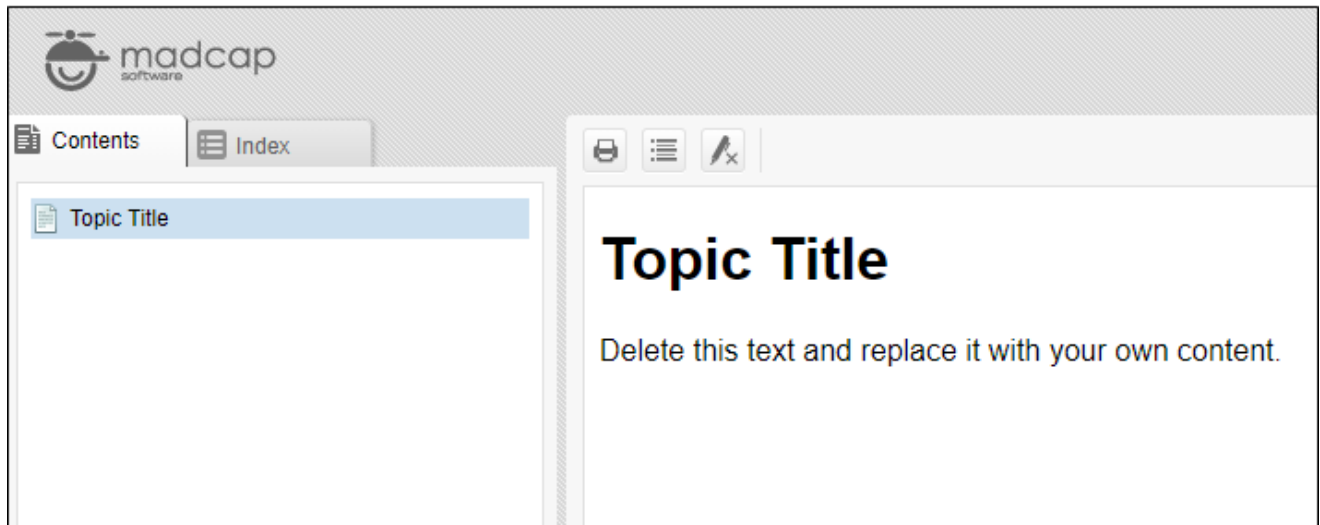




Tripane skin with a dark background



Tripane skin with a light background

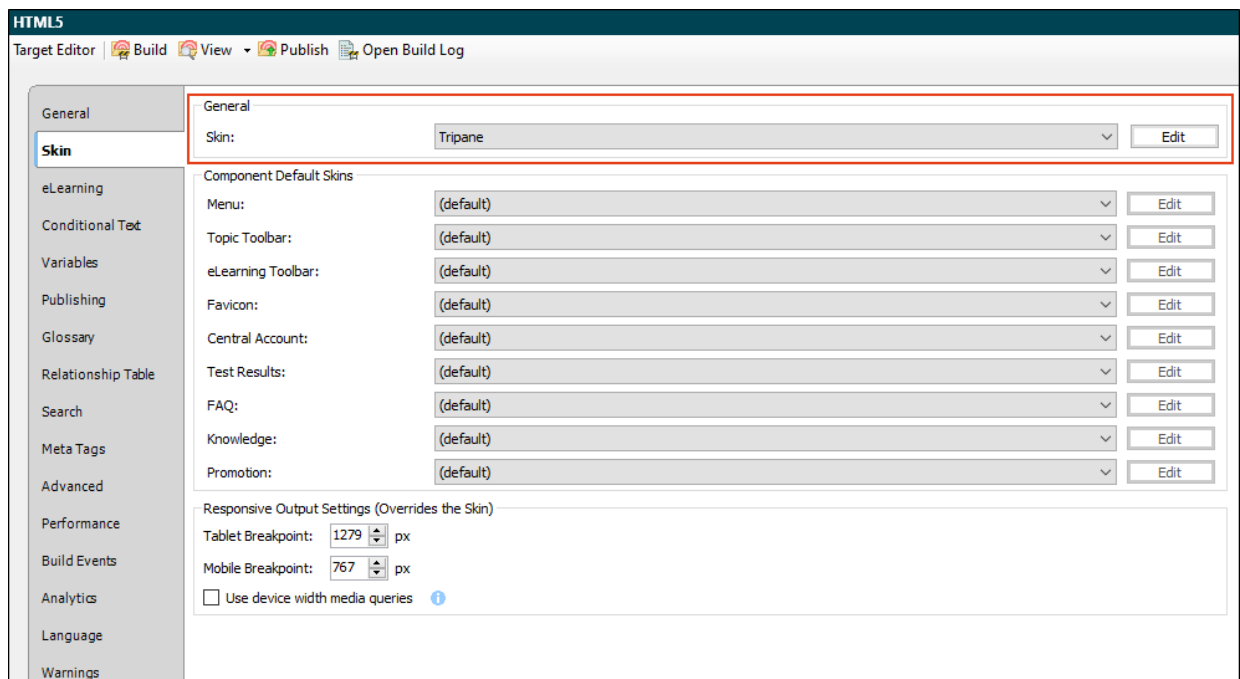


# Setting Tripane Output

You can use the following steps to set an HTML5 target to use the Tripane format.

## How to Set Tripane Output

1. Open the HTML5 target from the Project Organizer.
2. Select the **Skin** tab.
3. Click the drop-down arrow in the **Skin** field and choose a Tripane skin that you've added to the project.




4. Click  to save your work.

# Setting Responsive Options for Tripane Output

When a skin is responsive, the navigation elements are automatically adjusted depending on the size of the screen. For Tripane output, you can enable responsive output on the **Setup** tab of the Skin Editor. In addition, on the **Skin** tab of the Target Editor you can adjust the same settings as Side and Top Navigation output. See "Responsive Skins" on page 150.

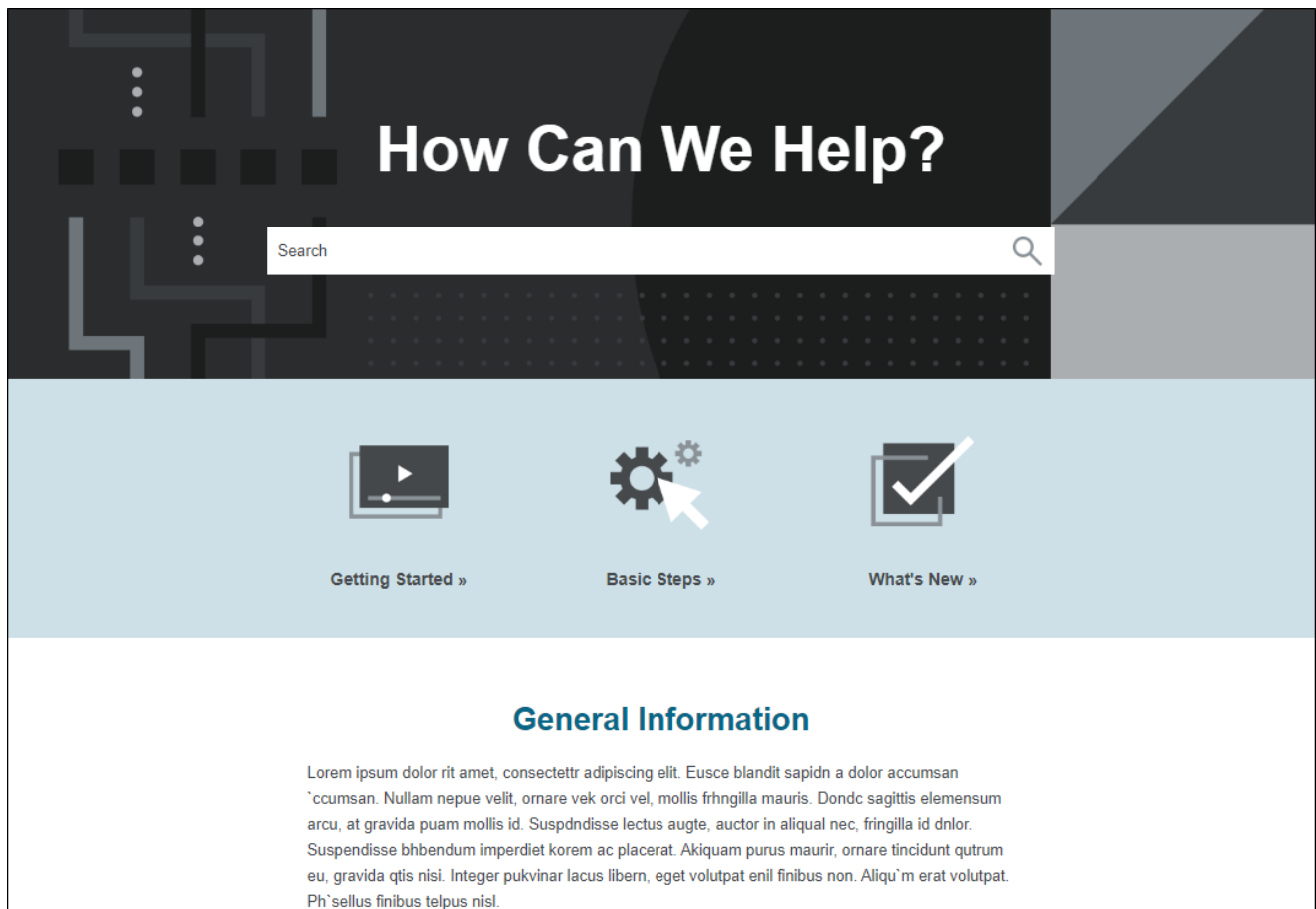
- **Tablet Breakpoint** Enter the number of pixels for the maximum width of a Tablet view.
- **Mobile Breakpoint** Enter the number of pixels for the maximum width of a Mobile (or phone) view.
- **Use device with media queries** Select this option if you want to base the responsive output not merely on the width of the output display, but on its width in the actual device (browser, tablet, or mobile). In other words, if you do not have this option selected and you view the output on a full browser, you can see the tablet and mobile layouts simply by reducing the size of the browser window. But if you select this option and you reduce the width of the browser, the layout will not change to the tablet or mobile formats.

Furthermore, you can use conditions, styles, and media queries to make your content responsive. See "Responsive Conditions" on page 166 and "Responsive Layouts" on page 179.

 **NOTE** If you have a Tripane skin in your project and you enter responsive output settings in both the Skin Editor and Target Editor, the settings in the target take precedence. However, this is not true if you *have not* yet made a change to the tablet or mobile breakpoint fields in the Target Editor, but you *have* made changes to them in the Skin Editor. In that case, the numbers from the changed Skin Editor will be used.

# HTML5 Skinless Output

HTML5 skinless output is a frameless output that does not use a main skin at all (although you might use smaller skin components). This means you will not have a main menu, and perhaps no search field, anywhere in the output.



One common reason to create skinless output is if you are creating an online eLearning course. You might not want people to use the main skin navigation or search field, but rather guide them through the course via an eLearning toolbar that is inserted with a proxy and eLearning Toolbar skin component. This is why the eLearning project templates available in the Start New Project Wizard are already set up as skinless.

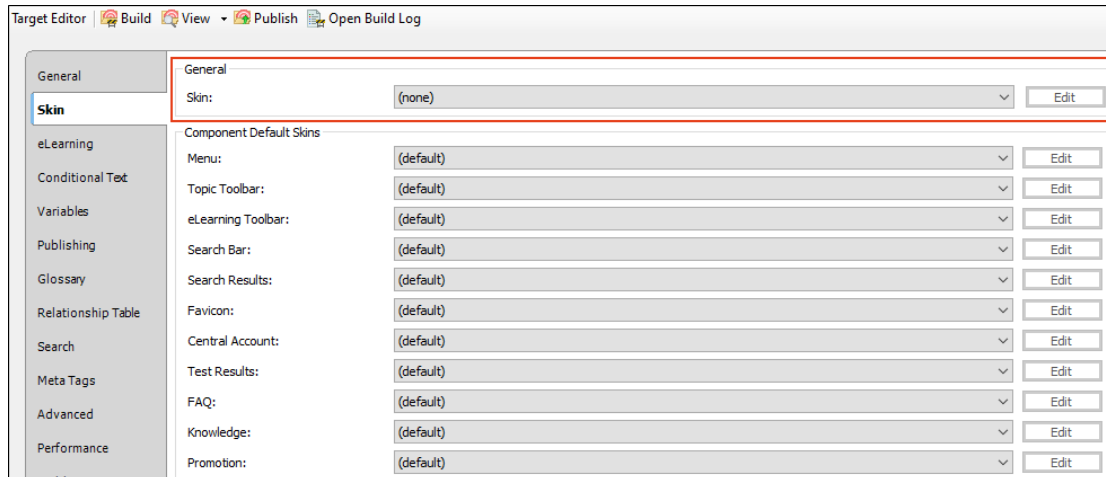
**This chapter discusses the following:**


How To Set Skinless Output in the Target .....	114
How to Set Skinless Output in the Skin .....	115

# How To Set Skinless Output in the Target

One way to create skinless output is to simply tell your target not to use a main skin.

1. Open the HTML5 target from the Project Organizer.
2. Select the **Skin** tab.
3. Click the drop-down arrow in the **Skin** field and select **none**.

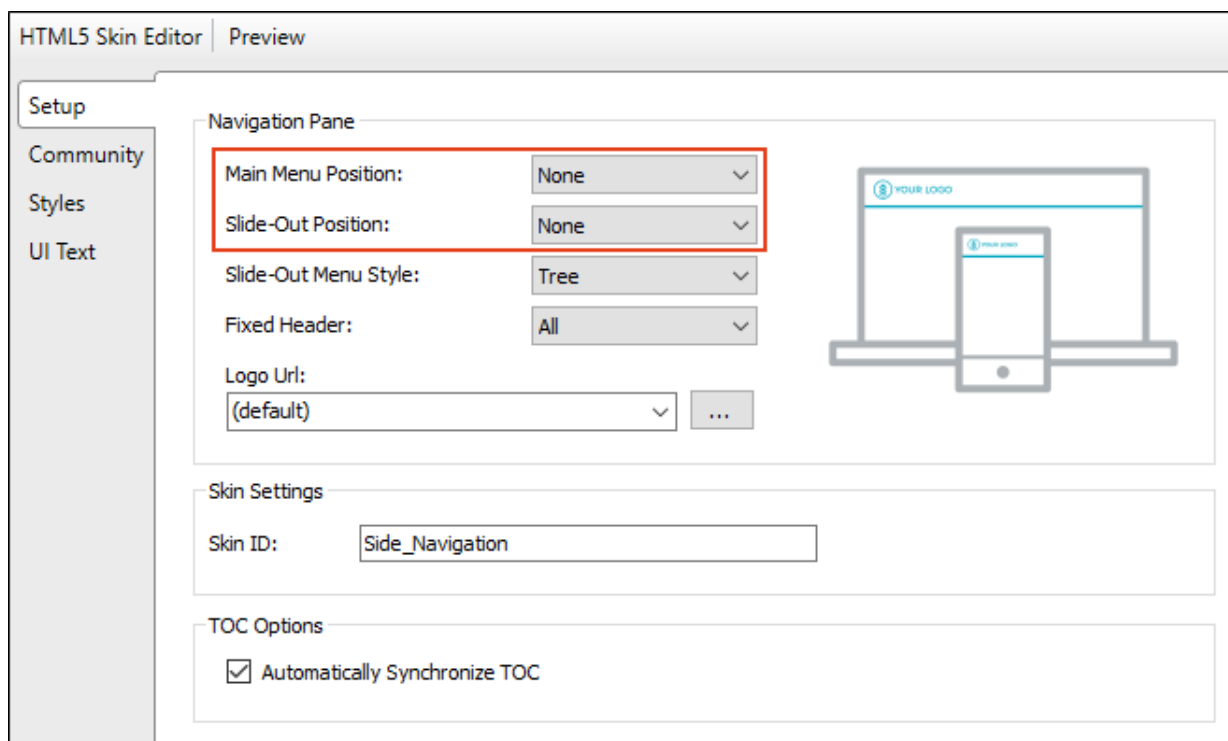



4. Click  to save your work.


# I How to Set Skinless Output in the Skin

Another way to create skinless output is to remove navigation and perhaps also the header or search from the main HTML5 Side or Top Navigation skin. You technically still have the skin, but not the navigation that it provides.

1. Open the HTML5 Side or Top Navigation Skin.
2. On the **Setup** tab, set the **Main Menu Position** and **Slide-Out Position** fields to **None**.




3. On the **Styles** tab, you can set the **Header > Layout > Display** field to **none**. This removes the entire header. If you want to keep the header but just remove the search, set the **Search Bar > Layout > Display** field to **none**.
4. Click  to save your work.

 **NOTE** Because skinless output does not include a main menu based on your TOC, the question must be asked, “How do you incorporate navigation into the output?” After all, you are likely to have more than just one topic and users will need to move from one page to another.

First, you can add various types of links throughout your content files. This might include features such as cross-references, hyperlinks, topic popups, image maps, and more. For more information about using links in Flare, see the online Help or the *Navigation Links Guide*.

Second, you can include proxies and skin components that allow you to inject small menus, toolbars, and search bars.

 **NOTE** Although this type of output does not use a skin, you still must have at least one HTML5 skin in your project in order to generate a skinless target properly.



# Skin Components and Proxies

In addition to a main skin, you can also use several smaller skin components with HTML5 targets. Most of these components are used in combination with corresponding proxies to add different features to the output, such as search elements, menus, and toolbars.

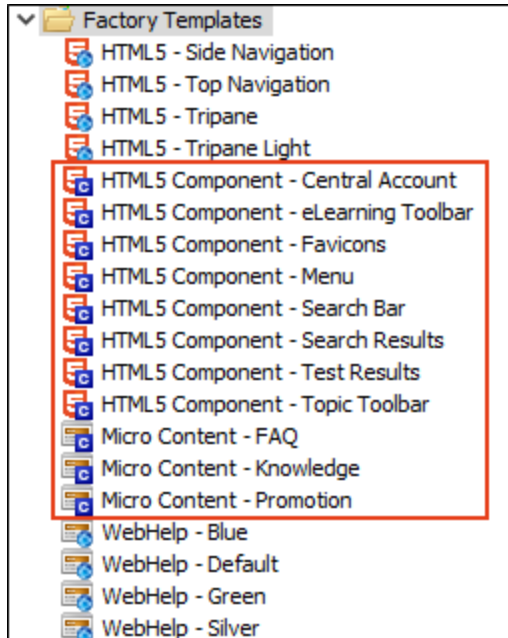
In most cases, a proxy is the element that actually generates the search element, menu, or toolbar when you build output. The related skin component is used to provide a look for it. Except for favicon skin components, a proxy is always necessary to generate the desired element, but a skin component is optional. If you do not add a particular type of skin component to your project, Flare provides a default design.

This chapter discusses the following:

Adding Skin Components .....	118
Inserting Proxies .....	123
Working With Skin Components and Proxies .....	124
What's Noteworthy? .....	149

# I Adding Skin Components

Skin components can be added to a project in the same way that you would add full skins. In the Project Organizer, right-click the **Skins** folder and from the context menu select **Add Skin**. Then choose the kind of skin component you want to add.

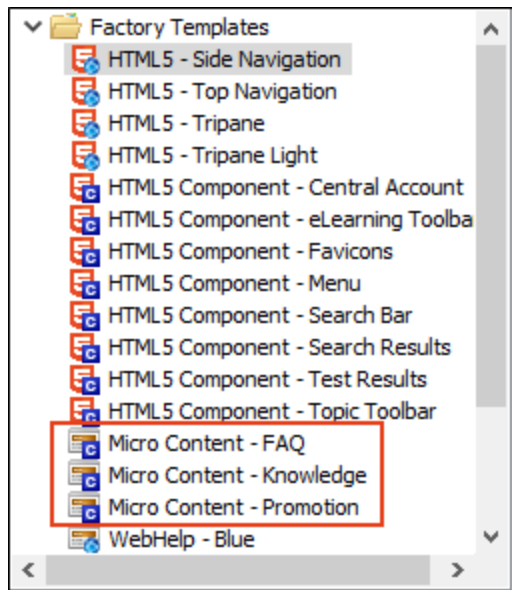


# Micro Content Skin Components

You can add micro content elements to your HTML5 output using a regular full skin such as Side Navigation or Top Navigation, without worrying about additional skin components. Both the featured snippets and Knowledge Panel search results can be included from the Target Editor using default styles.

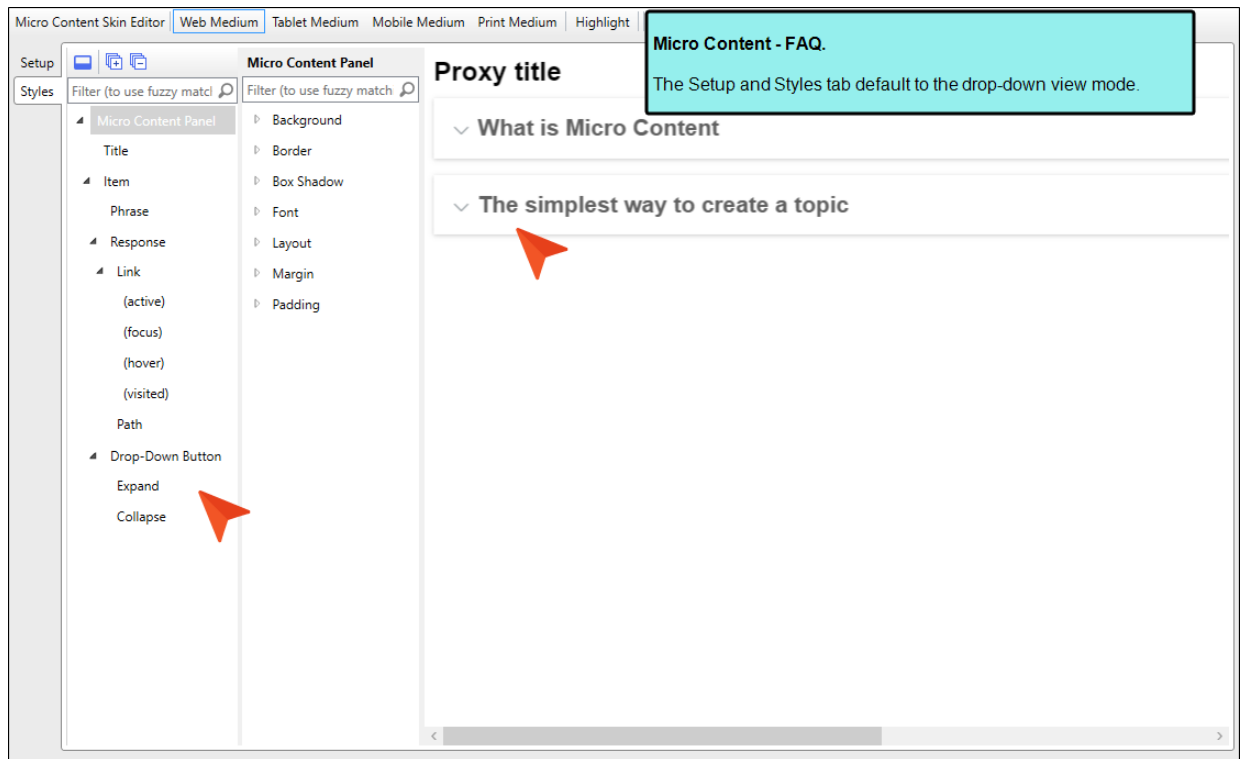
But if you want more control over the look and behavior of the micro content item, you can add a micro content skin component and proxy (e.g., FAQ, Knowledge, Promotion) for maximum customization. When you add a micro content proxy to a content file, you can select a specific skin for filtered micro content files.

The micro content skin types are Micro Content – FAQ, Micro Content – Knowledge, and Micro Content – Promotion. Each skin takes on slightly different default styles and options that cater to the intended use.

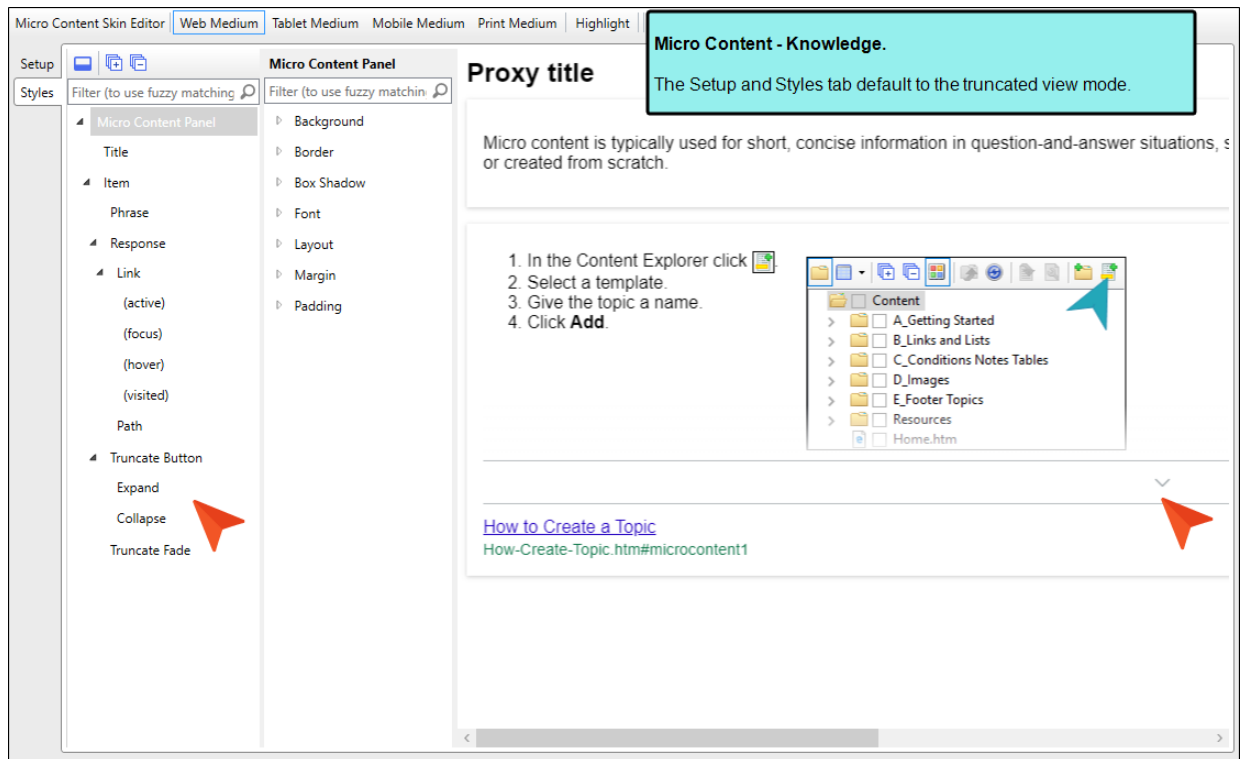


When you add one of the micro content skin types, it automatically opens in the Micro Content Skin Editor. This editor is a slimmed down version of the regular Skin Editor, but it focuses just on the selected skin component and proxy.

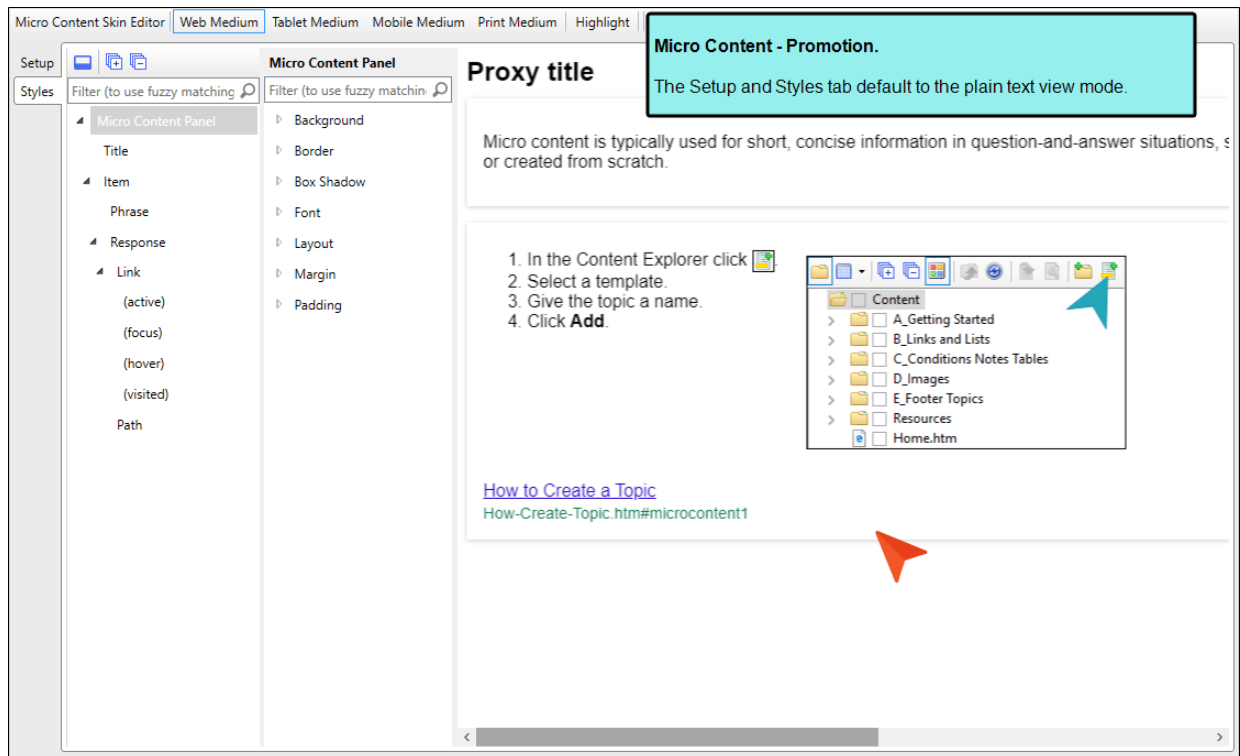
- **Micro Content - FAQ** This skin defaults to drop-down view mode. It is used for FAQ proxies, but it can also be used for Knowledge and Promotion proxies.



- **Micro Content - Knowledge** This skin defaults to truncated view mode. It is used for Knowledge proxies, but it can also be used for FAQ and Promotion proxies.



- **Micro Content - Promotion** This skin defaults to plain text view mode. It is used for Promotion proxies, but it can also be used for FAQ and Knowledge proxies.

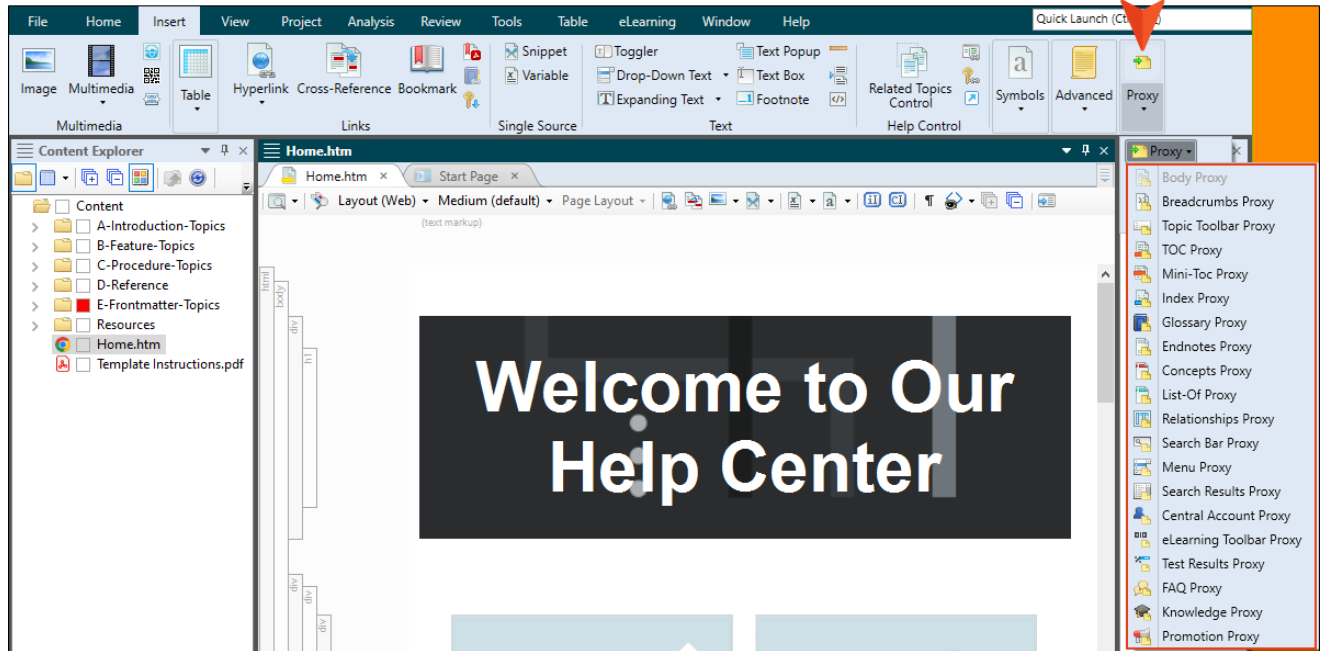


**NOTE** The Micro Content - FAQ skin component and FAQ proxy are unique in that they are applicable to non-HTML5 targets in addition to HTML5 targets (e.g., PDF, Word, Clean XHTML, Eclipse, EPUB, and HTML Help).

The other micro content skin components (i.e., Micro Content - Knowledge, and Micro Content - Promotion) can be used for styling only HTML5 outputs.

# Inserting Proxies

You can add a proxy by clicking in the content file where you want it to be placed. Then from the Insert ribbon, select Proxy > Insert [Name of Proxy].



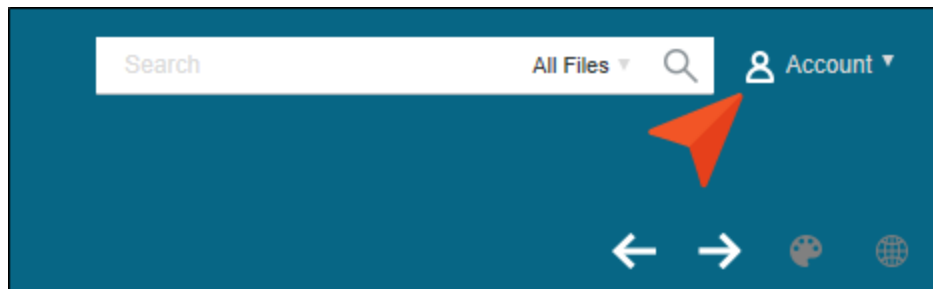
# Working With Skin Components and Proxies

The following skin components and related proxies are commonly used for HTML5 outputs. Some are more useful in some outputs than in others.

**NOTE** The Favicons skin component is the only one that is not associated with a proxy. Simply by adding and editing this type of skin component, your HTML5 output will include favicons in the appropriate locations. We will not be covering favicons in the following sections. For more information see the online Help.

## Central Account

A Central Account proxy lets you create an account link drop-down, allowing users viewing private output to log out or edit their MadCap Central profile settings. By default, the account link is included automatically in the header of the regular skin.

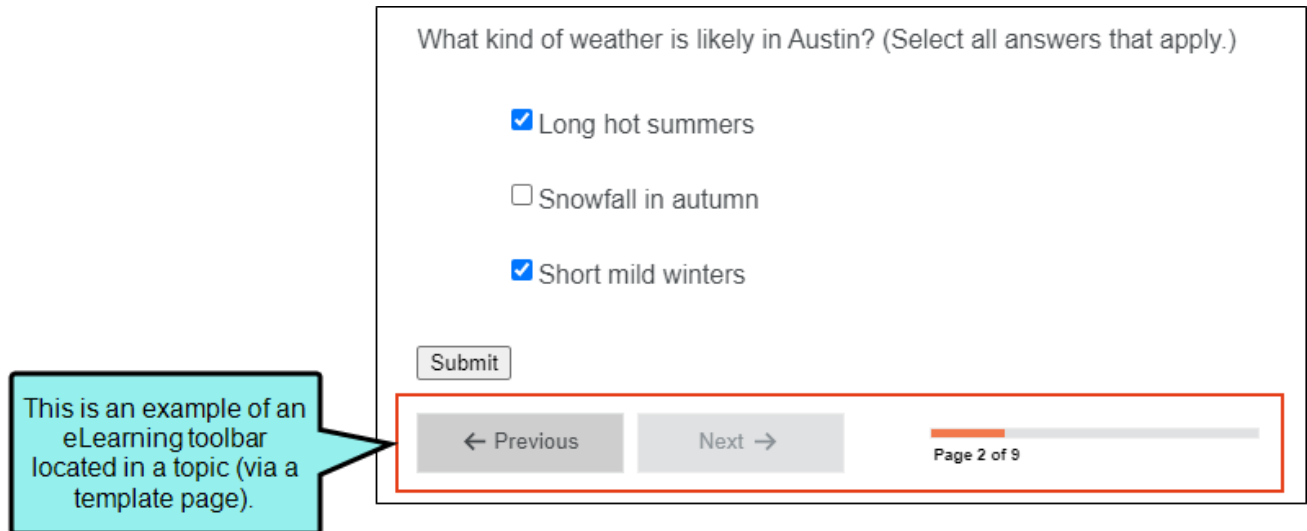


After inserting a Central Account proxy, you can edit it. To do this, open the topic or template page where the proxy is inserted, right-click on it, and select **Edit Central Account Proxy**. You can also edit the look of the account link by editing the styles in the regular skin or the Central Account skin component that you used.

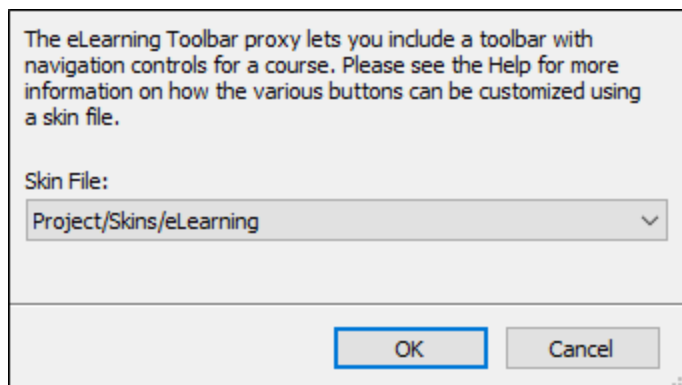


# eLearning Toolbar

The eLearning Toolbar proxy lets you include intuitive navigation buttons and controls in your course. These might be a Next and Previous button, or a progress bar. When you build HTML5 output, the proxy content is replaced with the generated navigation tools.

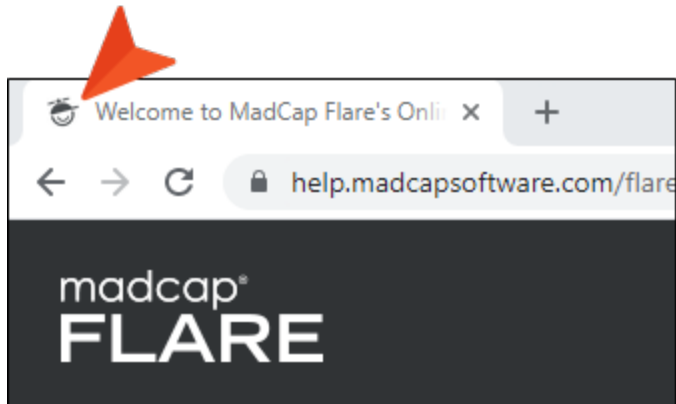


When you insert this proxy, the eLearning Toolbar Proxy dialog opens. If you have customized the eLearning Toolbar skin component, you can select it from the **Skin File** drop-down.



# Favicons

Short for “favorite icons,” favicons are supported in Flare for HTML5 output. A favicon is the image that you sometimes see in a browser tab, address bar, bookmark, Windows tiles, etc. The Favicons skin component is the only one that is not associated with a proxy. Simply by adding and editing this type of skin component, your HTML5 output will include favicons in the appropriate locations.



# Menu

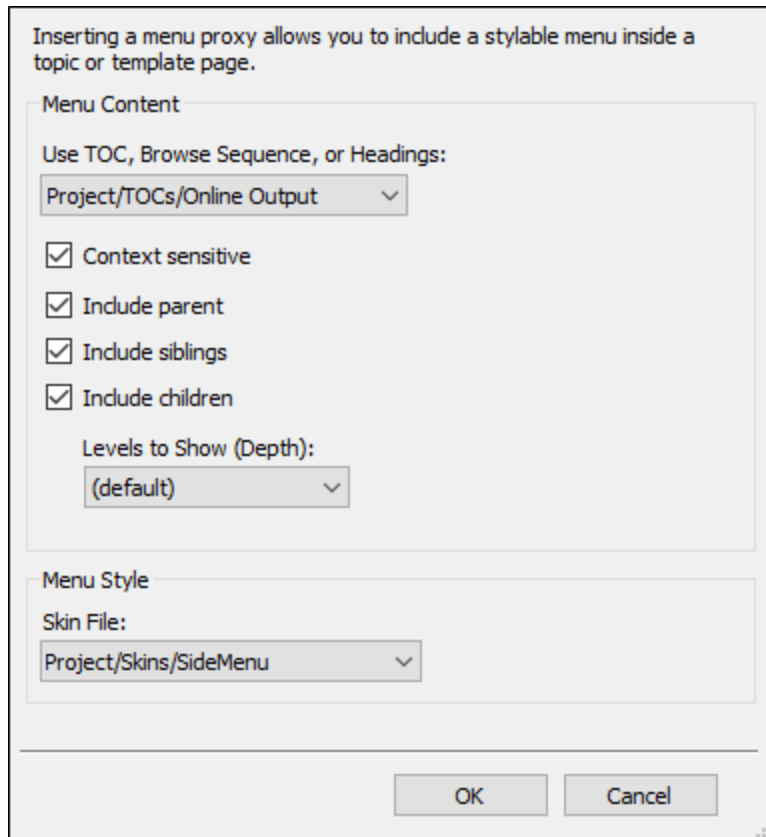
A Menu proxy lets you create a topic menu based on your TOC file, allowing users to navigate to other topics. This proxy can be especially useful if you are creating HTML5 Top Navigation or skinless output and you want a context-sensitive menu to display next to each topic. When you build the output, the proxy is replaced with the generated menu.

The screenshot shows a web page with a dark blue header. On the left is a logo with a tree icon and the text "YOUR LOGO". On the right are navigation links: "Getting Started", "Features", "Procedures", "FAQs", "Tips", and "Company". Below the header, the breadcrumb "You are here: [Getting Started](#)" is displayed. A search box is located in the top right. The main content area features a large heading "Getting Started" followed by a paragraph of Lorem Ipsum text. A red arrow points to a dropdown menu that is open, listing the following items: "Getting Started", "What's New", "Basic Steps", "Features", "Procedures", "FAQs", "Tips", and "Company". Below the text is a two-column table with a dark blue header row containing "Table Heading" in both columns. The table body contains three rows of Lorem Ipsum text.

Table Heading	Table Heading
Lorem ipsum dolor rit amet, consectetur adipiscing elit.	Dondc sagittis elementum arcu, at gravida puam mollis id.
Eusce blandit sapien a dolor accumsan `ccumsan.	Suspendisse lectus augte, auctor in aliquam nec, fringilla id dnlor.
Integer pukvinar lacus libern, eget volutpat enil finibus non.	Aliqu`m erat volutpat.

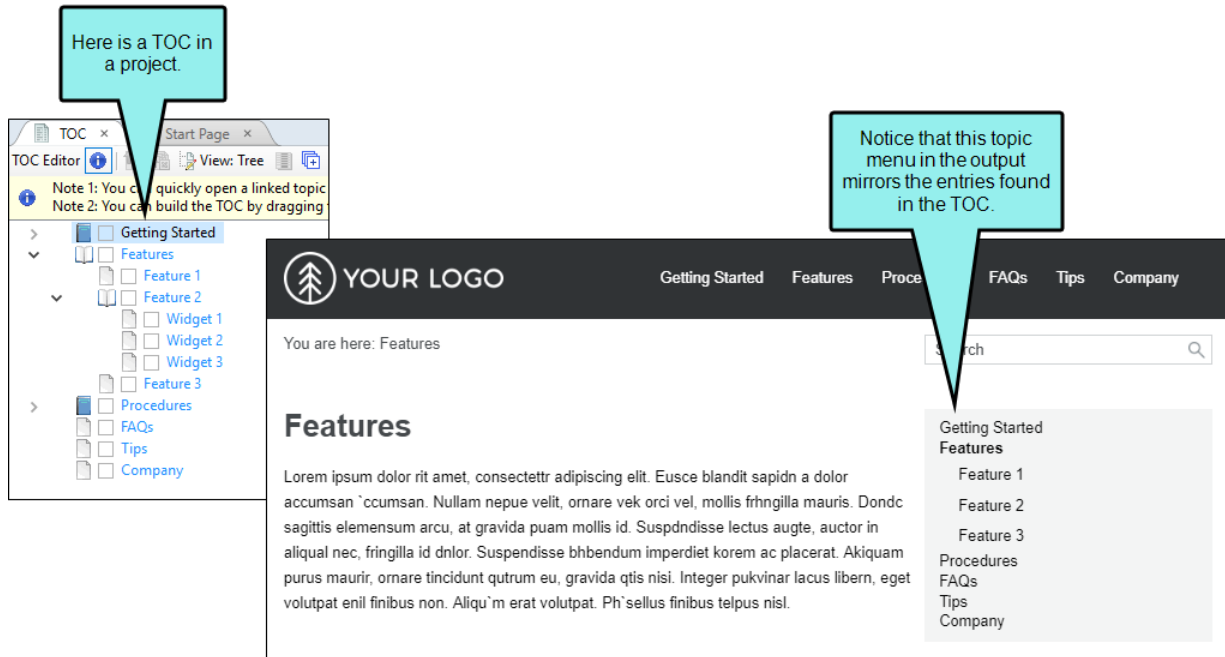
The Menu proxy and skin component allow users to access navigation unique to each topic, while the main navigation at the top of the page is typically limited to the first few levels of your TOC. This feature is not as common in Side Navigation or Tripane outputs, because those formats always include the full navigation on the side of topics.

When you insert a Menu proxy, the Menu Proxy dialog opens.



You can select the following options:

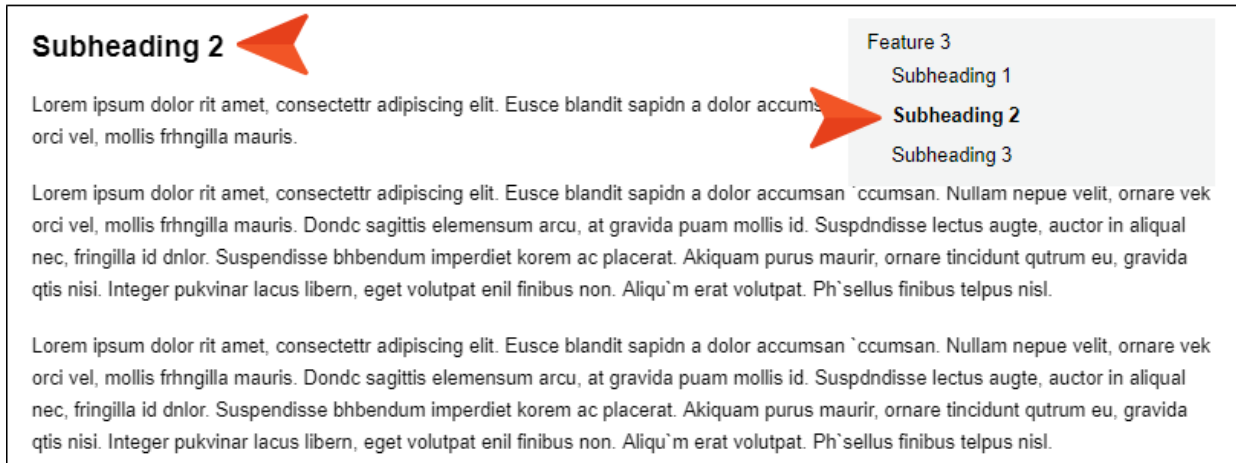
- **Use TOC, Browse Sequence, or Headings** If you have more than one TOC file in your project, you can select the one that the Menu proxy should use. It is most common to base a menu on a TOC, but you can also select a browse sequence.



You also have the option of selecting “Headings,” which will display topic headings in the side menu instead. In other words, any headings or subheadings (e.g., h1, h2, h3, h4) that you’ve included in a topic will be shown in the side menu instead of other topics in the output.

The screenshot shows a website layout with a dark header. On the left is a logo with a tree icon and the text "YOUR LOGO". On the right are navigation links: "Getting Started", "Features", "Procedures", "FAQs", "Tips", and "Company". Below the header, a breadcrumb trail reads "You are here: [Features](#) > Feature 3". To the right of the breadcrumb is a search bar with the text "Search" and a magnifying glass icon. The main content area starts with a large heading "Feature 3" followed by a paragraph of Lorem Ipsum text. To the right of this text is a side menu with a light gray background, containing the following items: "Feature 3", "Subheading 1", "Subheading 2", and "Subheading 3". A blue callout box with a white border and a pointer to the side menu contains the text: "In this example, the side menu shows the current heading 'Feature 3' and its subheadings." Below the main text, there are three subheadings: "Subheading 1", "Subheading 2", and "Subheading 3", each followed by a paragraph of Lorem Ipsum text. Each subheading is accompanied by a red arrow pointing to the right.

Also, when viewing a topic that is set up like this—and the menu is fixed in place (via the Setup tab of the Skin Editor)—each sub-heading section will be highlighted as you scroll down in the topic.



The screenshot shows a topic page with a sub-heading and a table of contents. The sub-heading is "Subheading 2" and is highlighted with a red arrow. The table of contents is located in the top right corner and lists three sub-headings: "Subheading 1", "Subheading 2", and "Subheading 3". The "Subheading 2" entry in the table of contents is highlighted with a red arrow, indicating that the current page is displaying that section. The main content area contains three paragraphs of Lorem Ipsum text.

**Subheading 2**

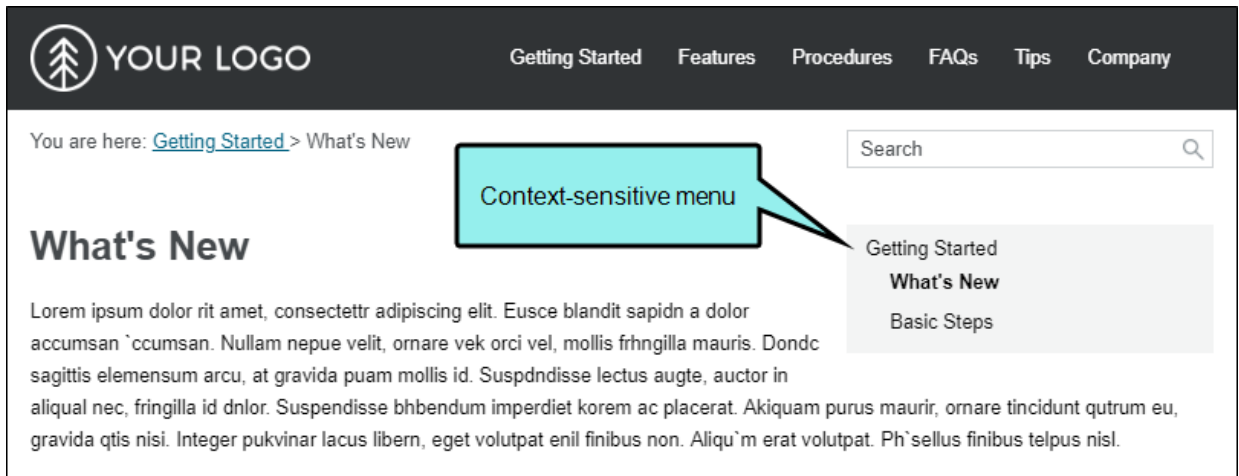
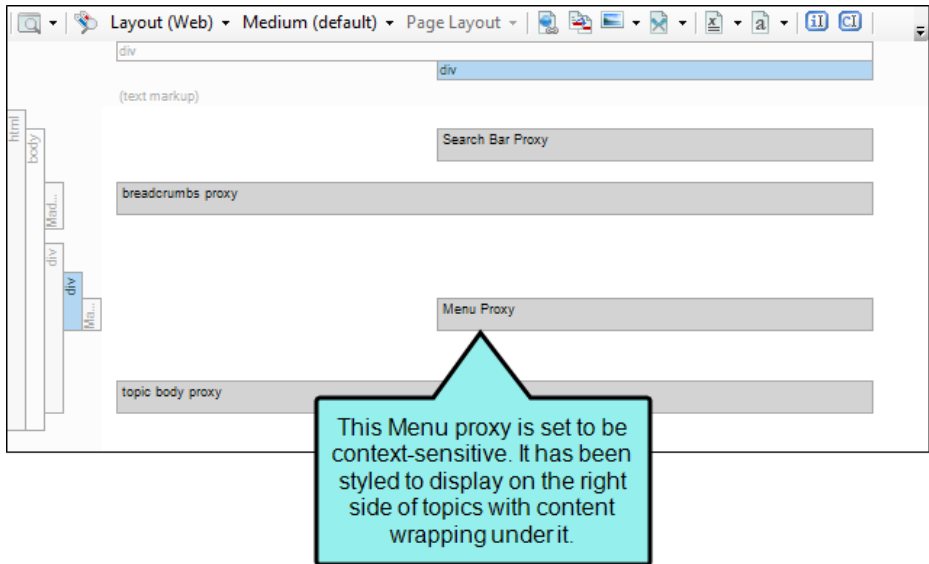
Lorem ipsum dolor rit amet, consectetur adipiscing elit. Eusce blandit sapien a dolor accumsan. Orci vel, mollis fringilla mauris.

Lorem ipsum dolor rit amet, consectetur adipiscing elit. Eusce blandit sapien a dolor accumsan. Nullam neque velit, ornare vehicula orci vel, mollis fringilla mauris. Donec sagittis elementum arcu, at gravida quam mollis id. Suspendisse lectus augue, auctor in aliquam nec, fringilla id dolor. Suspendisse bibendum imperdiet korem ac placerat. Aiquam purus mauris, ornare tincidunt quam eu, gravida quis nisi. Integer purus lacus libero, eget volutpat enim finibus non. Aliquam erat volutpat. Phasellus finibus telpus nisl.

Lorem ipsum dolor rit amet, consectetur adipiscing elit. Eusce blandit sapien a dolor accumsan. Nullam neque velit, ornare vehicula orci vel, mollis fringilla mauris. Donec sagittis elementum arcu, at gravida quam mollis id. Suspendisse lectus augue, auctor in aliquam nec, fringilla id dolor. Suspendisse bibendum imperdiet korem ac placerat. Aiquam purus mauris, ornare tincidunt quam eu, gravida quis nisi. Integer purus lacus libero, eget volutpat enim finibus non. Aliquam erat volutpat. Phasellus finibus telpus nisl.

Feature 3  
Subheading 1  
**Subheading 2**  
Subheading 3

- Context sensitive** Select this check box if you want the menu to show only closely related entries in the TOC. In Flare's Top Navigation project templates, this kind of proxy was inserted into a template page to create a side menu for most of the topics in the output.

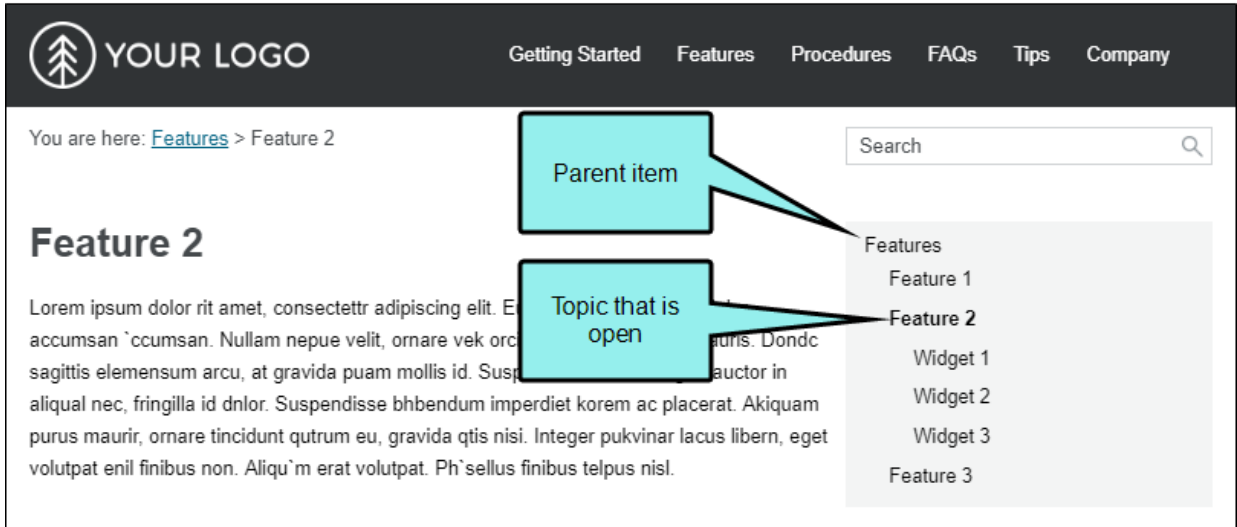


If you do not make the menu context-sensitive, it displays everything in the TOC (depending on the depth level you select).

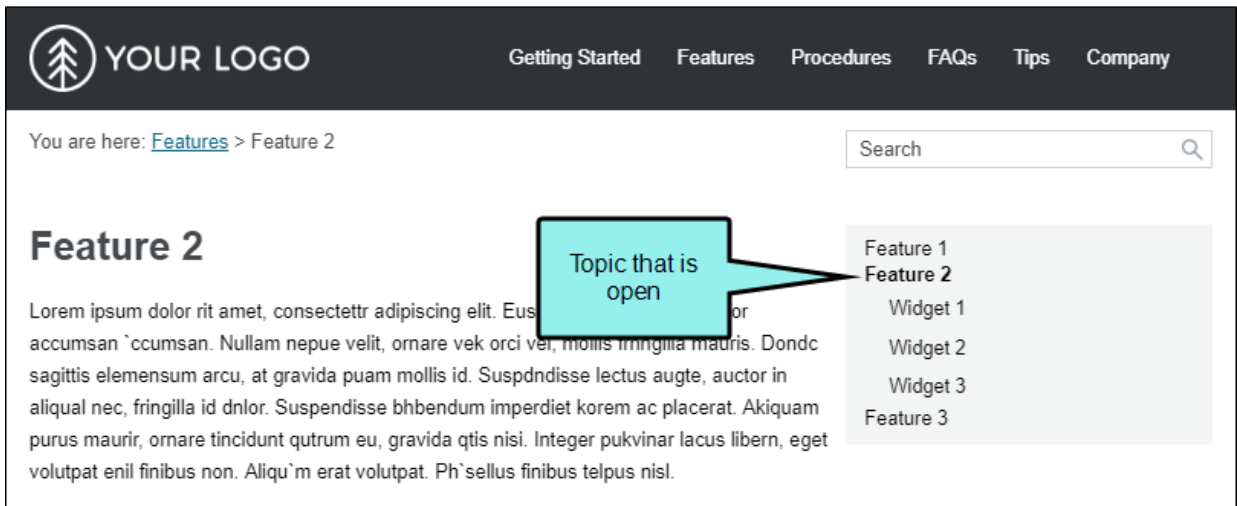


- **Include parent** If you have selected the **Context sensitive** option, you can also select this check box to include the parent TOC item in the menu.

With the parent option selected

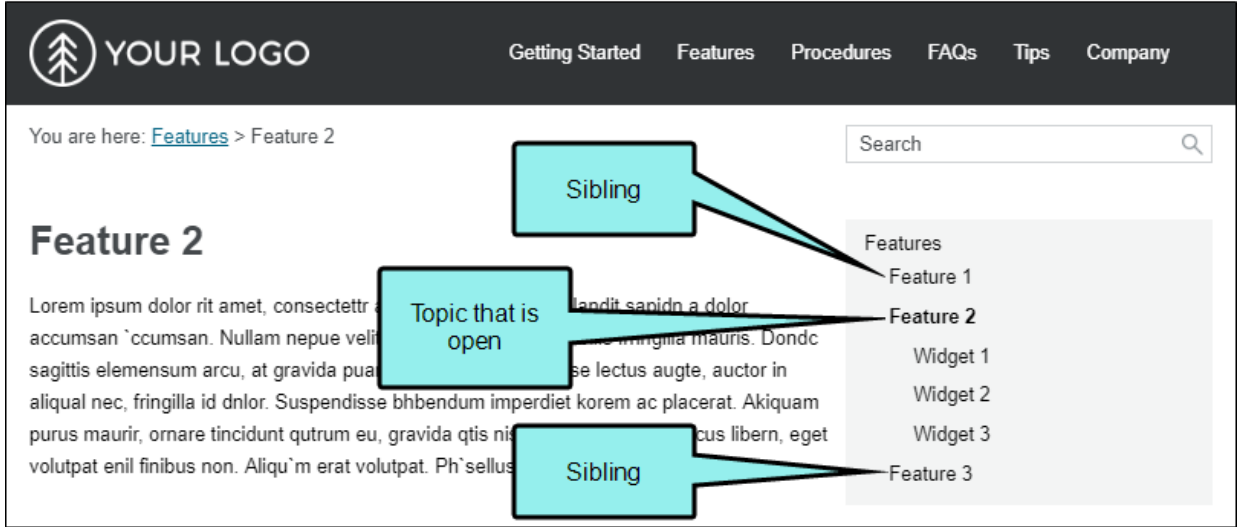


With the parent option disabled

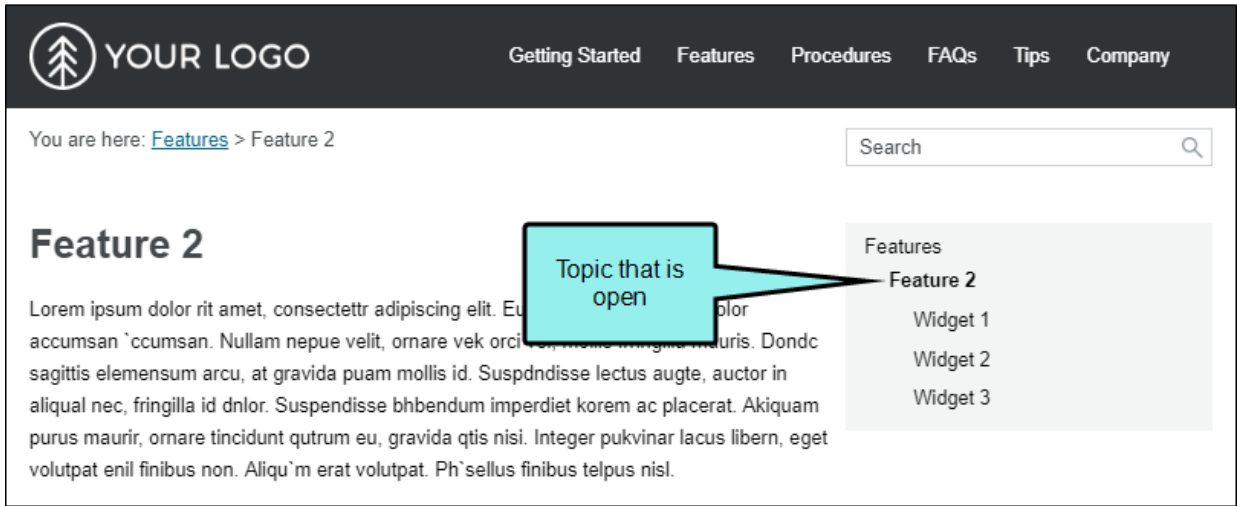


- **Include siblings** If you have selected the **Context sensitive** option, you can also select this check box to include TOC items in the menu that are on the same level as the open topic.

With the siblings option selected

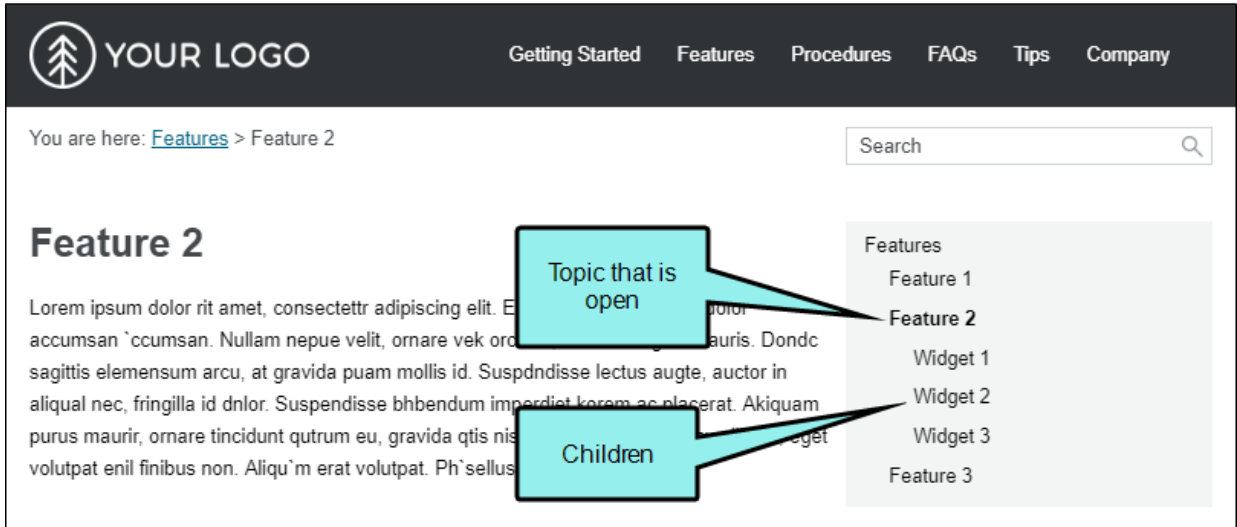


With the siblings option disabled

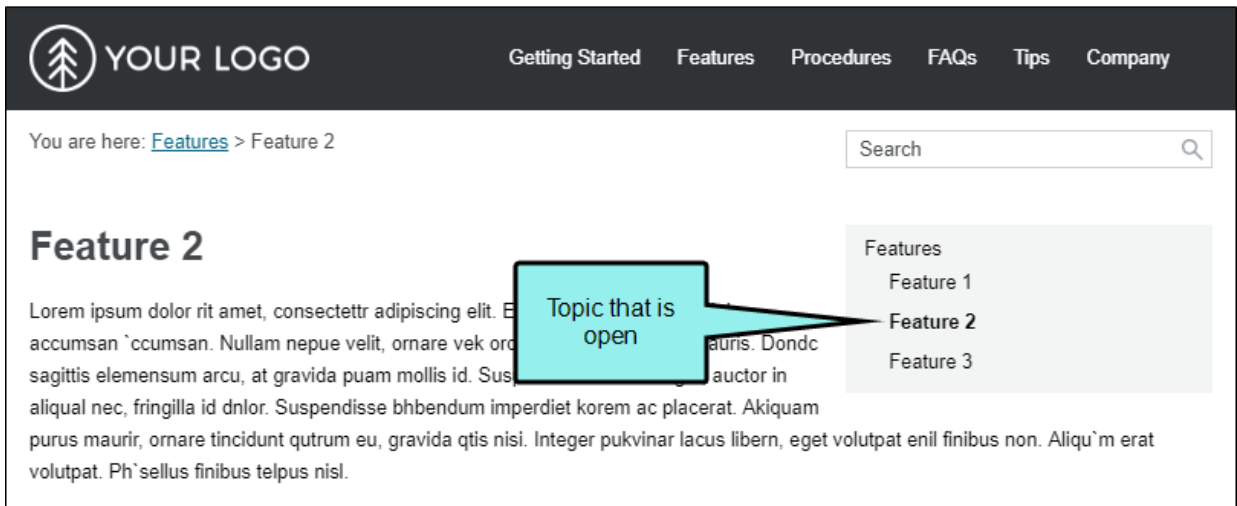


- **Include children** If you have selected the **Context sensitive** option, you can also select this check box to include TOC items in the menu that are children of the topic that is open.

With the children option selected





With the children option disabled



- **Levels to Show (Depth)** This lets you choose how many levels of items deep in the TOC to include in the menu. If the **Context sensitive** option is disabled, this refers to the depth level overall for the TOC. If both the **Context sensitive** and the **Include children** options are enabled, it refers to the number of levels under the topic that is open.
- **Skin File** If you have added a Menu skin component to your project and want to use it to control the look of the menu, you can select it from this field.

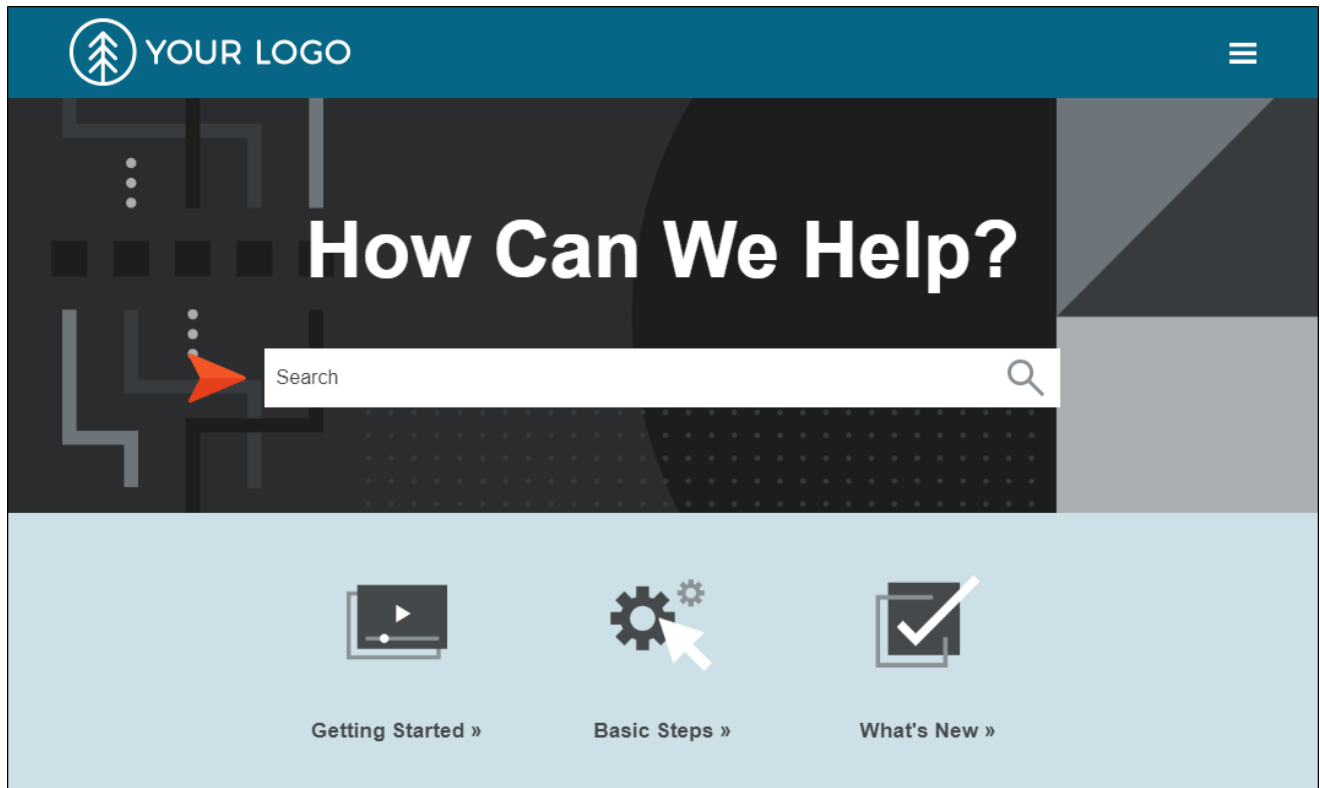
If you do not select a Menu skin component in this field, Flare uses the first one it finds in your project (if one exists). Otherwise, Flare provides a default design.

 **NOTE** If you use a Menu skin component and proxy to include a topic menu in your HTML5 Top Navigation output, you can set that menu to be fixed in place. You can set a fixed menu for web, tablet, or mobile screens (or all of them).

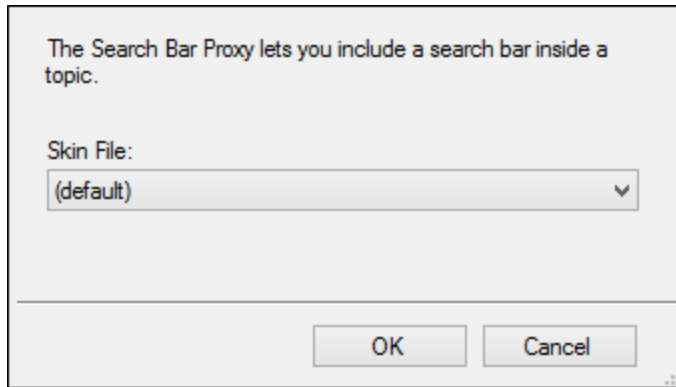
 **NOTE** For HTML5 Tripane output, the Menu proxy and skin component will not work for merged projects or linking to external Help systems. Merging projects is not supported in Side or Top Navigation output at all.

# Search Bar

A Search Bar proxy lets you create a field that can be used to perform searches in HTML5 output. When you build the output, the proxy is replaced with the generated search bar. If you insert a search bar using a proxy, you might also need to hide the search bar from some topics so that you do not have two search bars on a page.




When you insert a Search Bar proxy, the Search Bar Proxy dialog opens.



You can select the following:

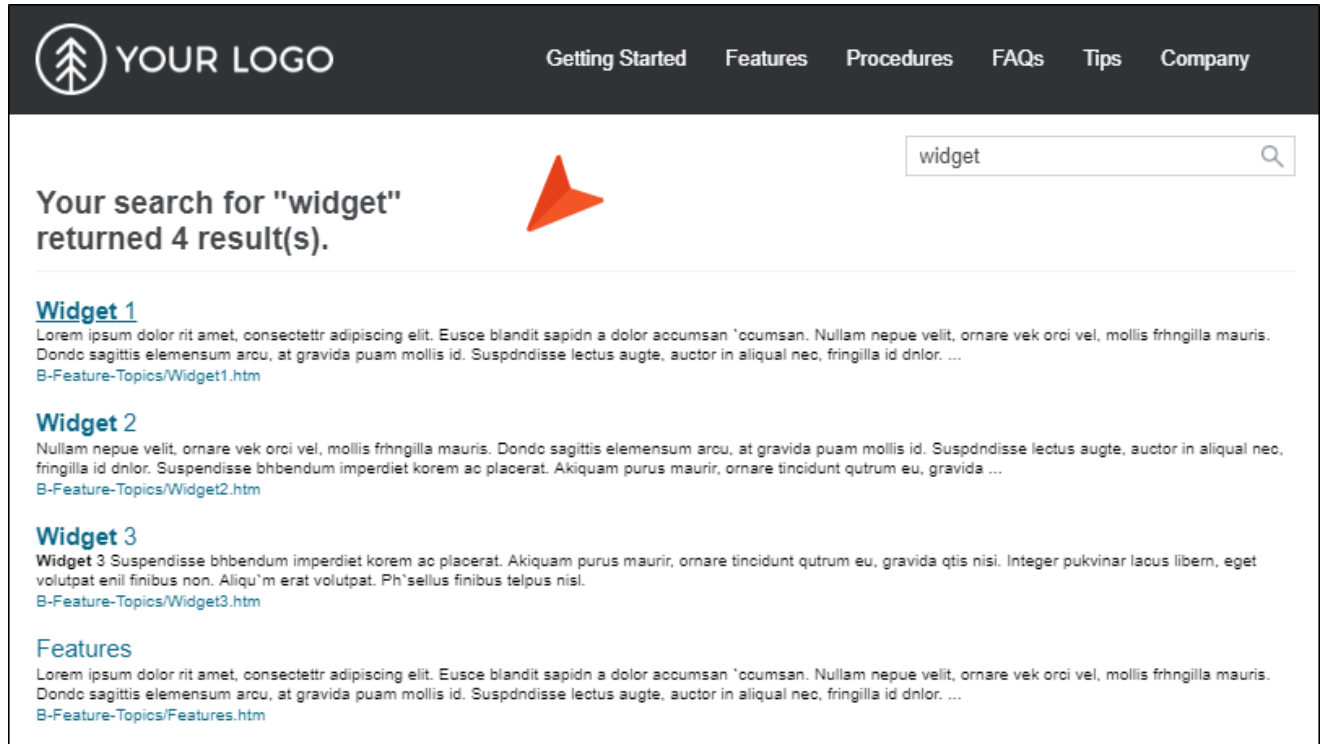
- **Skin File** If you have added a search bar skin component to your project and want to use it to control the look of the search bar, you can select it from this field.

If you do not select a Search Bar skin component in this field, Flare uses the first one it finds in your project (if one exists). Otherwise, Flare provides a default design.

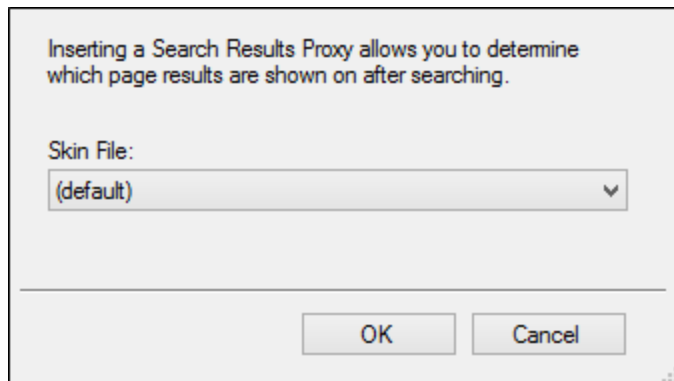
 **NOTE** The Search Bar proxy and skin component are not supported in HTML5 Tripane output.

# Search Results

A Search Results proxy works with the Search Bar proxy and lets you create a container that is used to display the results of end user searches in your output. When you build the output, the proxy is replaced with the generated search results.





When you insert this kind of proxy, the Search Results Proxy dialog opens.



You can select the following:

- **Skin File** A skin component lets you control the look of generated search results. If you have added multiple skin components to your project, you can use this field to select the one to associate with this proxy. You can then edit that skin component to change its appearance.

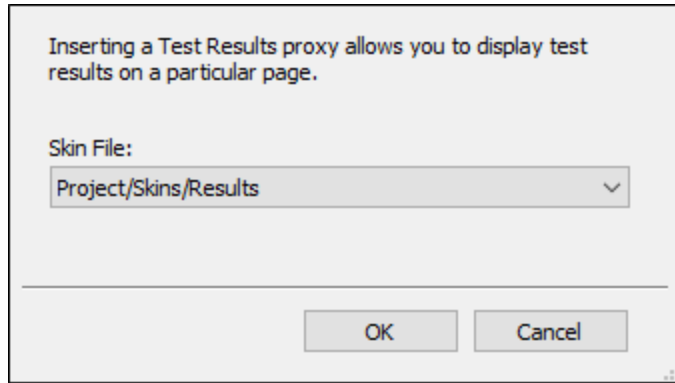
 **WARNING** Do not insert a Search Results proxy into a template page. Insert it only into topics.

 **NOTE** The Search Results proxy and skin component are not supported in HTML5 Tripane output.



# Test Results

The Test Results proxy is necessary for ending course navigation and calculating the test results score when you link a custom pass or fail topic for the test results page. When you insert this proxy, the Test Results Proxy dialog opens. If you have customized the Test Results skin component, you can select it from the **Skin File** drop-down.



Example of a custom test results pass page.

## You Passed!

Congratulations for learning all about birds, and for passing the test.



### About Birds

Score: 100%

Passed

Which group below can safely be ruled out, as to where modern birds evolved from?

Your Answer:

Insects

# Topic Toolbar

You can use the Topic Toolbar proxy to generate a toolbar anywhere in your topics. This is an alternative (or an additional option) to the web toolbar that can be included in some online outputs.

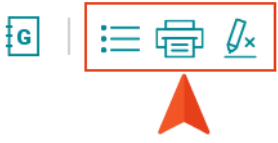
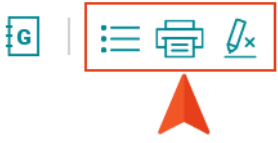
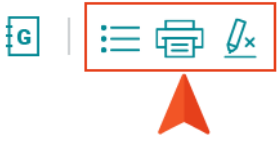
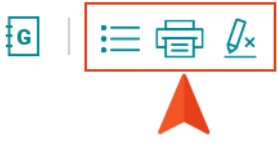
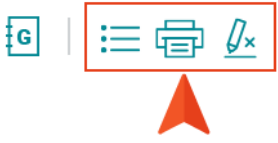
You are here: [Step 3: Designing](#) > Skins

## Skins

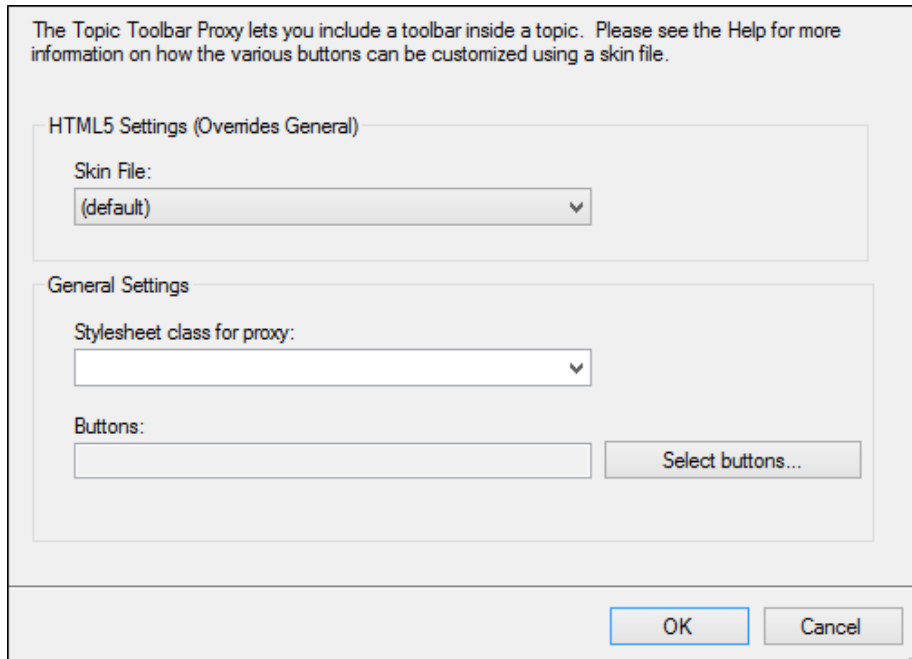
> OUTPUTS SUPPORTED

A skin is a file that contains information about the appearance of an online output window, including navigation elements.

General Information	Process	Skins Can Control...
<ul style="list-style-type: none"><li>▪ <a href="#">Types of Skins</a></li><li>▪ <a href="#">Skin Components and Proxies</a></li><li>▪ <a href="#">HTML5 Skinless Output</a></li></ul>	<ol style="list-style-type: none"><li>1. <a href="#">Adding Skin Files</a></li><li>2. <a href="#">Opening Skins</a></li><li>3. <a href="#">Editing Skin Settings</a></li></ol>	<ul style="list-style-type: none"><li>▪ Pane position</li><li>▪ Slide-out menu style</li><li>▪ Main menu position</li></ul>



When you insert a Topic Toolbar proxy, the Topic Toolbar Proxy dialog opens.



Topic toolbars can be inserted into outputs other than HTML5. However, the Topic Toolbar skin component is supported in HTML5 only. Therefore, the Topic Toolbar Proxy dialog is split into two sections. The HTML5 Settings area pertains only to HTML5 output. The General Settings area pertains to all of the outputs that support topic toolbars, including HTML5. Anything that is set in the HTML5 Settings area overrides what is set in the General Settings area, including the buttons that are selected for the toolbar.

You can select the following options:

- **Skin File** If you have added an HTML5 Topic Toolbar skin component to your project and want to use it to control the look of the toolbar, you can select it from this field. From the Skin Editor, you can select the buttons to be included in the toolbar. Alternatively, you can select buttons from the Buttons field below.

If you do not select a Topic Toolbar component in this field, Flare uses the first one it finds in your project (if one exists). Otherwise, Flare provides a default design.

- **Stylesheet class for proxy** You can select a class to affect the look of the entire toolbar. This is an alternative, or supplement, to editing the skin component in the Skin Editor. However, for HTML5 output, using a skin component is the most common method for designing the look of the toolbar.

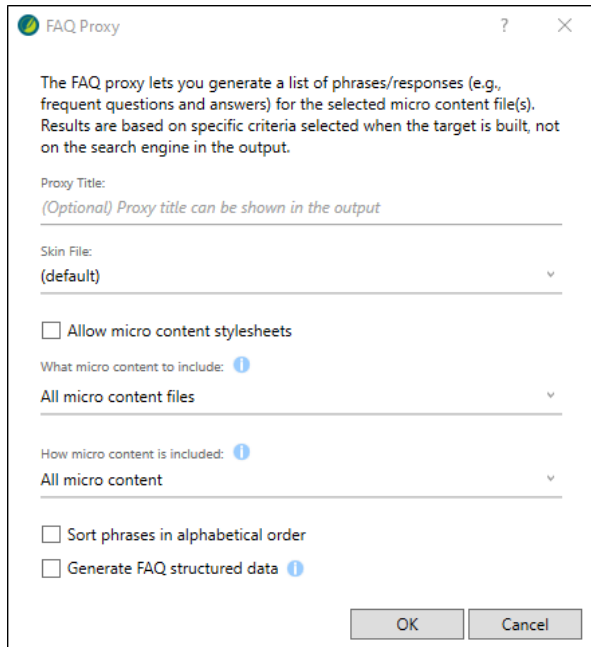
You might create and use a proxy style class, for example, if you want to add a border around the toolbar. If you do not select a class from this field, the generated toolbar will use the style settings from the parent `MadCap|topicToolbarProxy` style. You have the option of creating a class for this proxy style in the Stylesheet Editor. To do this, select the **MadCap|topicToolbarProxy** style and in the local toolbar click **Add Selector** to create a class. The class will then be available from this field.


- **Buttons** You have the option of selecting buttons for a toolbar in the Skin Editor or by using this field. You can click **Select buttons** to open a dialog, then select the buttons to include in the toolbar. For HTML5 outputs, the Topic Toolbar proxy will use whatever settings are specified in a Topic Toolbar skin component (if you have added one to your project), overriding any buttons you may have selected directly in the proxy. If you have not associated a Topic Toolbar skin component with the proxy, Flare will just use the first one it finds in your project. However, for outputs using Standard and Mobile skins, the settings in the proxy take precedence over anything you may have set on the Toolbar tab in the Skin Editor.

# Micro Content - FAQ

The FAQ proxy lets you generate a list of phrases and responses for the selected micro content file (s) in most outputs. Results are static, based on specific criteria selected when the target is built, not on the search engine output. This type of proxy is ideal for including traditional FAQs, but it can be used for other purposes as well.

When you insert an FAQ proxy into a content file, the FAQ Proxy dialog opens. If you have added a Micro Content - FAQ skin component to your project and want to use it to control the look of micro content item, you can select it from the Skin File field.



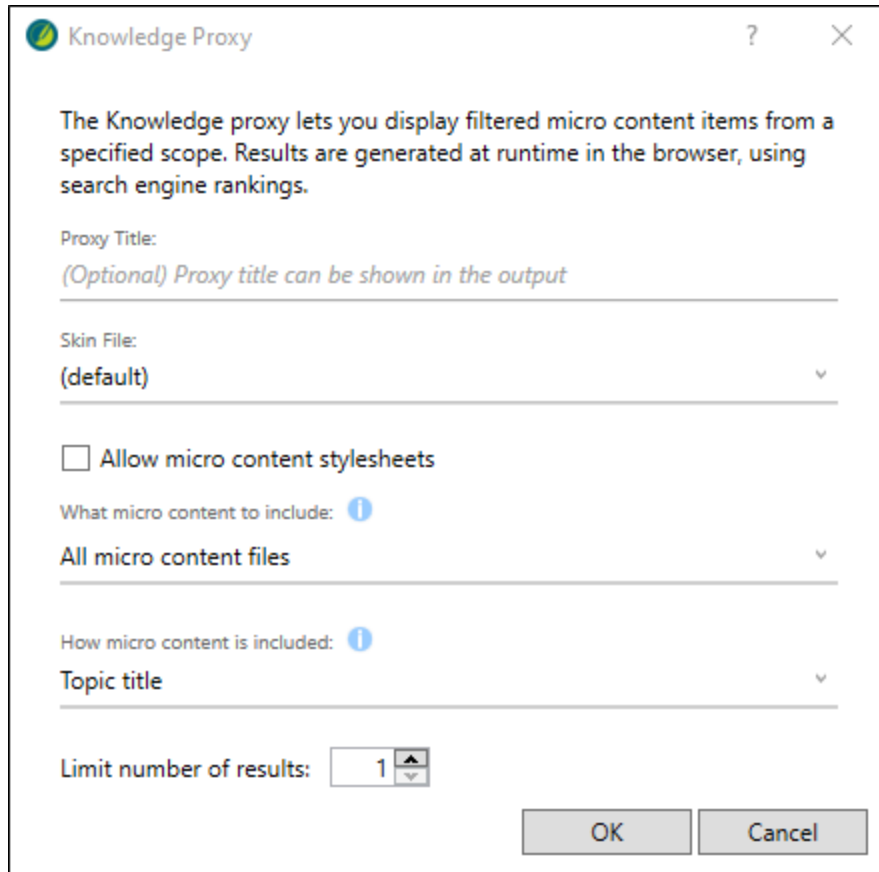
 **NOTE** The Micro Content - FAQ skin component and FAQ proxy are unique in that they are applicable to non-HTML5 targets in addition to HTML5 targets (e.g., PDF, Word, Clean XHTML, Eclipse, EPUB, and HTML Help).

The other micro content skin components (i.e., Micro Content - Knowledge, and Micro Content - Promotion) can be used for styling only HTML5 outputs.

# Micro Content - Knowledge

The Knowledge proxy lets you display filtered micro content items in HTML5 output. Results are dynamic, generated at runtime in the browser, using search engine rankings. You might use a Knowledge proxy to display content such as information about a person, a company, technical specifications, related tasks, etc.

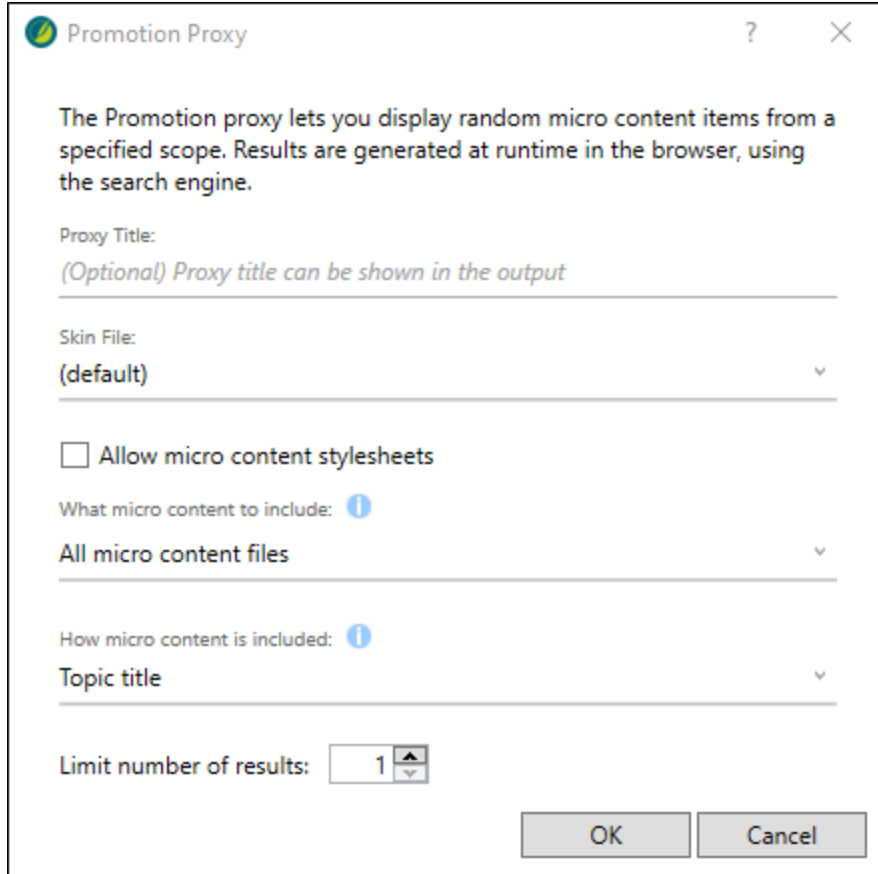
When a Knowledge proxy is inserted, the Knowledge Proxy dialog opens. If you have added a Micro Content - Knowledge skin component to your project and want to use it to control the look of micro content item, you can select it from the Skin File field.



# Micro Content - Promotion

The Promotion proxy lets you display random micro content items in HTML5 output. Results are dynamic, generated at runtime in the browser, using the search engine. You might use a Promotion proxy to display information about relevant products, events, new features, etc.

When a Promotion proxy is inserted, the Promotion Proxy dialog opens. If you have added a Micro Content - Promotion skin component to your project and want to use it to control the look of micro content item, you can select it from the Skin File field.



The screenshot shows a dialog box titled "Promotion Proxy" with a question mark icon and a close button. The dialog contains the following text and controls:

The Promotion proxy lets you display random micro content items from a specified scope. Results are generated at runtime in the browser, using the search engine.

Proxy Title:  
*(Optional) Proxy title can be shown in the output*

Skin File:  
(default)

Allow micro content stylesheets

What micro content to include: ⓘ  
All micro content files


How micro content is included: ⓘ  
Topic title


Limit number of results: 1

OK Cancel



# I What's Noteworthy?

 **NOTE** HTML5 Side and Top Navigation outputs do not support the Current Topic Index button. Additionally, in HTML5 Side and Top Navigation outputs, the Next Topic and Previous topic buttons navigate between topics in the table of contents; in HTML5 Tripane output these buttons navigate between topics in a browse sequence (if one is available). Otherwise, HTML5 Side and Top Navigation outputs support the same buttons as HTML5 Tripane output.

 **NOTE** You are not limited to one skin component of each type, although that is the most common situation. If you want, you can use multiple skin components of any type. If this is the case, you can associate a default skin component with a target. When you do this, the skin component you choose will always be used for any proxy of that same type that you insert for that target, unless you override it by associating a different skin component with a specific proxy that you've inserted.

# Responsive Skins

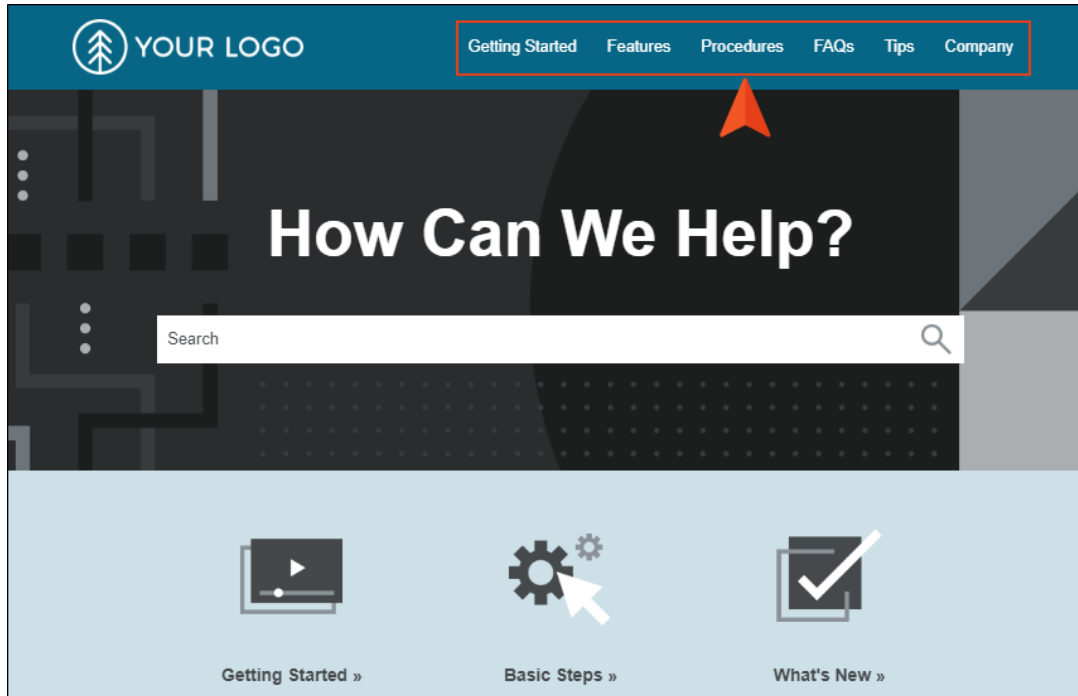
HTML5 Side and Top Navigation skins are always enabled for responsive output, but you can adjust some settings on the Skin tab in the Target Editor. For Tripane output, you can enable responsive output in the Skin Editor, and you can adjust the same settings as Side and Top Navigation output in the Target Editor. When a skin is responsive, the navigation elements (e.g., menu/table of contents items) are automatically adjusted depending on the size of the screen.

This chapter discusses the following:

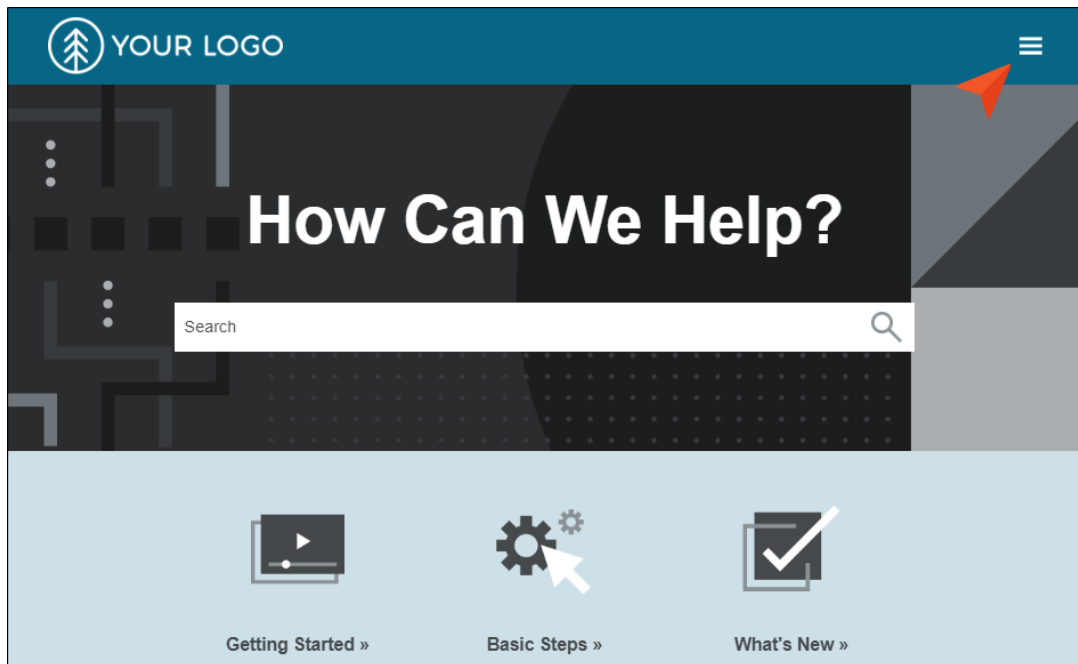
Examples .....	151
Responsive Skins and Mediums .....	155
How to Enable Responsive Tripane Skins .....	158
Device Width Media Queries .....	159

# Examples

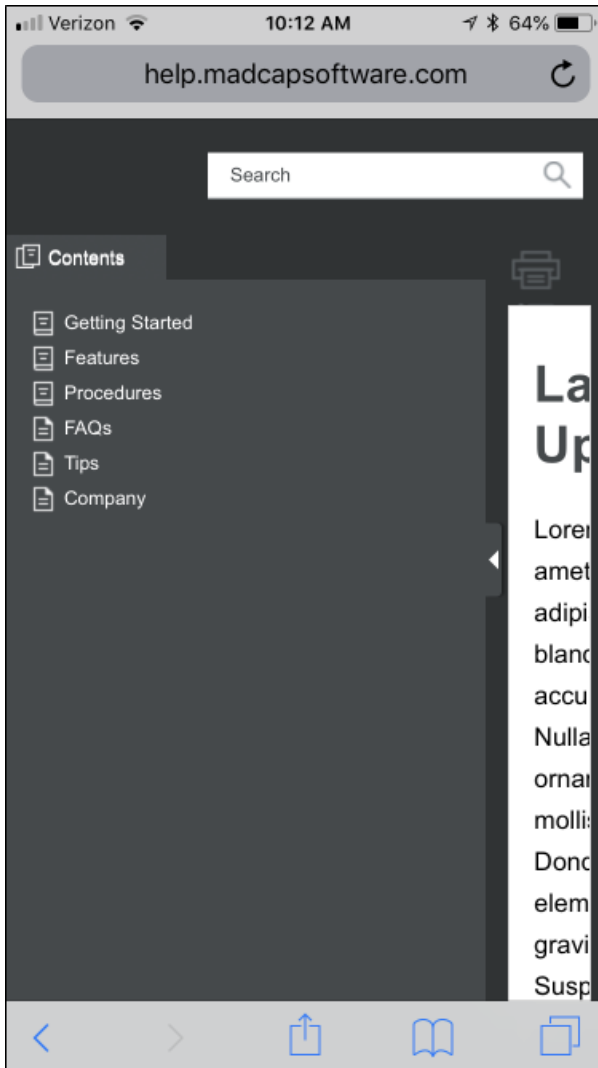
- ☆ EXAMPLE You create HTML5 Top Navigation output. When viewed on a large screen, menus appear at the top. These are populated from your TOC file.



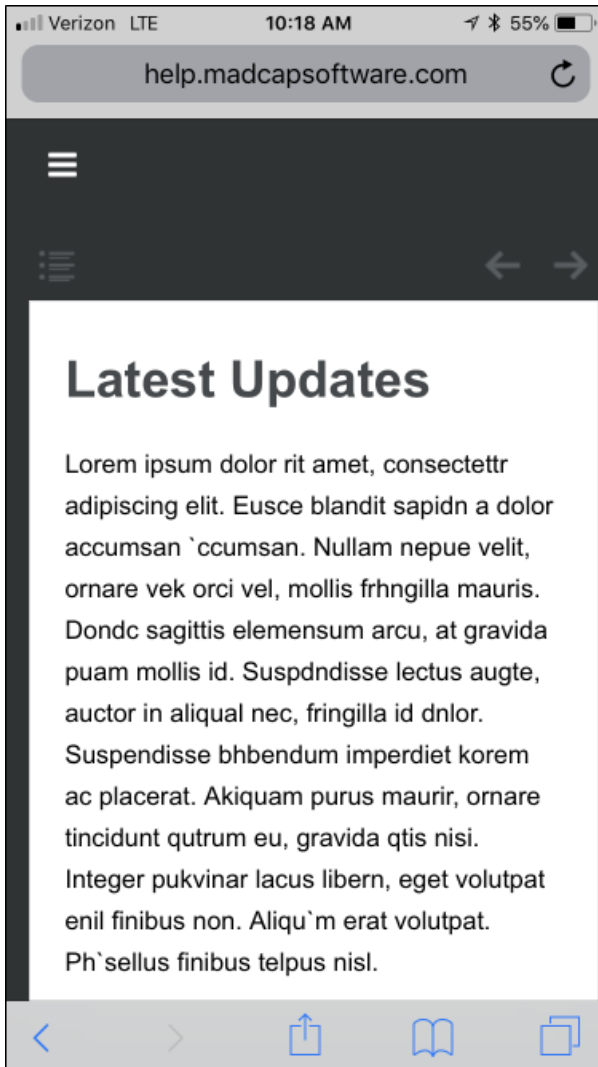
- ☆ But when you open the same output on a tablet device, you notice the top menus are no longer there. Because of the smaller screen they are placed in a flyout menu in the upper-right corner.



☆ **EXAMPLE** You want to create HTML5 Tripane output from your project and provide it to people using both large desktops and small mobile devices. Without using responsive output, you would need to create two targets and two skins, and perhaps separate settings elsewhere (e.g., stylesheet mediums). If you do not create a separate target and skin for the mobile output, but instead direct people to open the full output on their smart phones, it can be difficult (if not impossible) to navigate through that output on a smart phone.



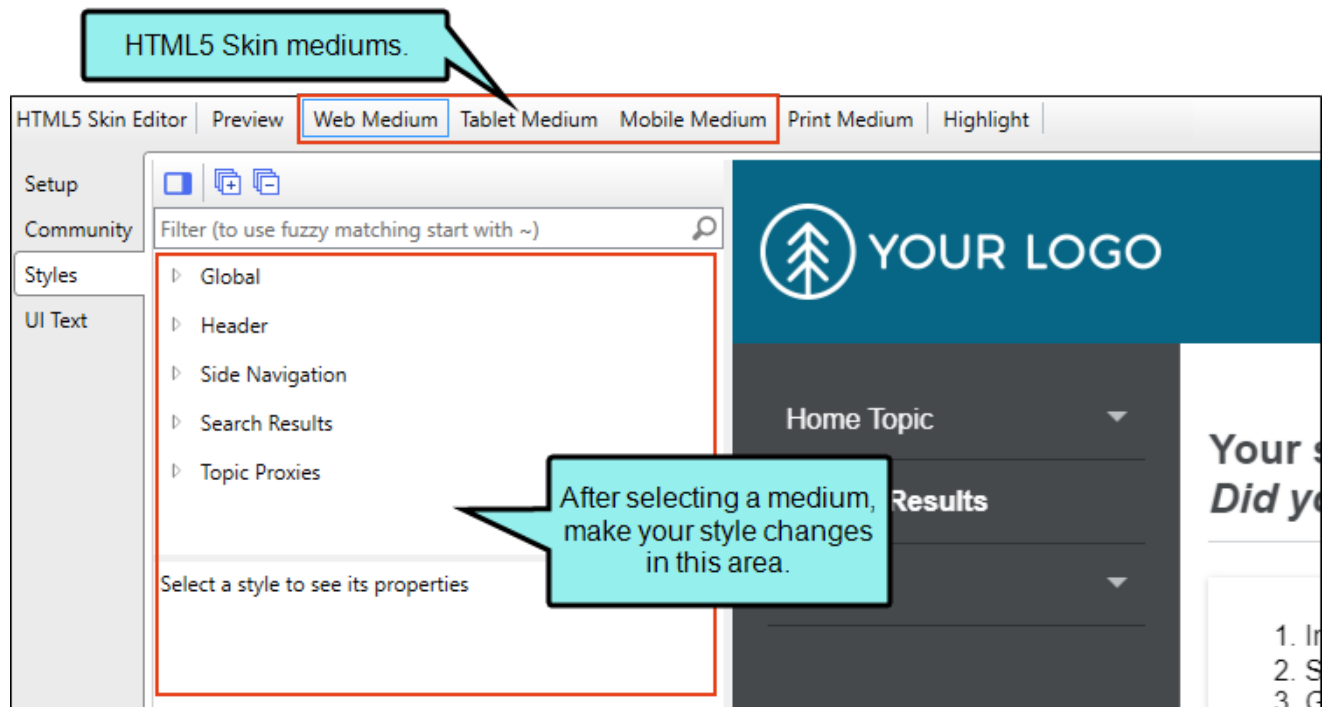
- ☆ So instead, you enable responsive output in the HTML5 Tripane skin. When finished, you direct end users to open the same HTML5 output, regardless of their browser or device. On a smart phone, the skin is adjusted automatically.



In the end, you accomplished essentially the same thing that you did with two targets and two skins, but you did it with only one target and skin.


# Responsive Skins and Mediums

To support responsive output, HTML5 skins come equipped with three mediums/media queries—Web, Tablet, and Mobile. Web kind of serves as both a medium and media query in Flare (controlling how skins and content look on a large screen), whereas tablet and mobile are technically both media queries, not mediums. You can use these media queries to specify different looks when necessary for the various devices people might be using to view your output.





- **Web** When you first open an HTML5 skin, **Web Medium** is selected. With that medium chosen, any changes you make to the styles are applied to that medium. This means that when users are viewing your output on a regular browser, that's the look they will see.
- **Tablet** If you want to change the look that users see when viewing your output on a tablet, select **Tablet Medium** and make your style changes.
- **Mobile** If you want to change the look that users see when viewing your output on a mobile device, select **Mobile Medium** and make your style changes.

Similar to cascading stylesheets (CSS), there is a system of inheritance at work with skin mediums. The Web medium is the default. Most settings in that medium are inherited by the Tablet medium. In turn, settings in the Tablet medium are inherited by the Mobile medium. Therefore, if you want all of the mediums to share the same look (e.g., your company logo), you can set it once in the Web medium and it will automatically be used in all three. If you make any changes in the Tablet medium, those settings will override whatever had been inherited from the Web medium. Likewise, you can make changes in the Mobile medium, which will override any settings inherited from the Tablet medium.

 **NOTE** An exception to this inheritance is with navigation icon images. Because these icon images are often different sizes in the Web medium than in the other mediums, most navigation icon images do not pass down from the Web medium to the Tablet medium. However, the Tablet medium does inherit the icon image used for the "Mark as New" feature from the Web medium. Also, icon images are passed down from the Tablet medium to the Mobile medium.



 **NOTE** Within each medium, the icons under the TOC Entry style inherit from the icon settings in the Navigation Panel > TOC. Therefore, if no icons are set in the TOC Entry style classes in the Web medium, Flare will use the Navigation Panel > TOC icons in that medium. The same holds true for the Tablet medium; if no icons are set in the TOC Entry classes in the Tablet medium, they will be inherited from the Navigation Panel > TOC icons in the Tablet medium. And the same is the case with the Mobile medium.

 **NOTE** In the HTML5 Skin Editor, the user interface displays a Print medium button next to the Web, Tablet, and Mobile mediums. The main purpose of the Print medium is for controlling how the skin looks when printed from an online source. The Print medium in the skin is its own entity, separate from the other mediums, and it does not share the same purpose as mediums in the stylesheet or in responsive output.

# I How to Enable Responsive Tripane Skins

For Tripane output, you can use the following steps to enable responsive output:

1. Open an HTML5 Tripane skin and select the **Setup** tab.
2. In the **Responsive Output Settings** section, select **Enable responsive output**.
3. (Optional) Responsive output works by automatically changing the display once the viewer reaches a certain width. You can change the maximum width at which the display changes from one medium to the next. Use the following to specify responsive settings for a skin. For more information about making topic content responsive, see "Responsive Content" on page 165.
  - **Tablet Max-Width** Enter the number of pixels for the maximum width of a Tablet view.
  - **Mobile Max-Width** Enter the number of pixels for the maximum width of a Mobile view.

☆ **EXAMPLE** You keep the default settings of 1279 pixels for the Tablet maximum width and 767 pixels for the Mobile maximum width.

If you generate output and view it in a regular browser with the window maximized, you will see the skin style settings for the Web medium.

You then click and drag the browser window to reduce it. Once the width of the browser window reaches 1279 pixels of width, the display changes to show the skin style settings associated with the Tablet medium.

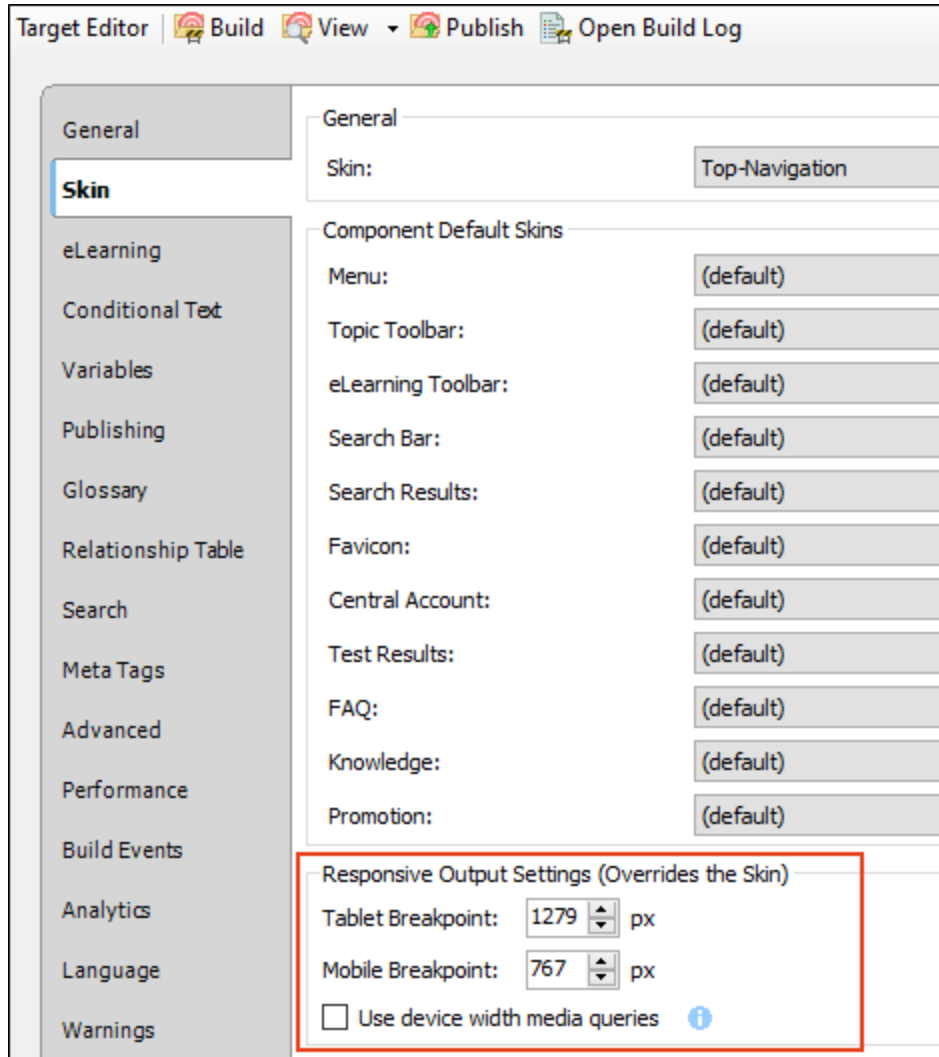
You continue to reduce the size of the browser window. Once the width reaches 767 pixels, the display changes to show the settings for the Mobile medium.

Therefore, the Web medium is named as it is because the largest size is usually meant for a regular web browser. The next size down is often used for tablets, such as iPads, so that medium is called "Tablet." And the smallest size is controlled by the Mobile medium, because a display that small is usually seen in mobile phones.

4. Click  to save your work.

# Device Width Media Queries

In addition to the fields in the Skin Editor, there are additional responsive output settings in the Target Editor that are available for Side and Top Navigation, as well as Tripane outputs. One setting lets you enable **device width media queries** for responsive output.



This means that the responsive nature of the skin depends on the device being used to view the output (browser, tablet, or mobile phone), rather than on merely the width of the screen.

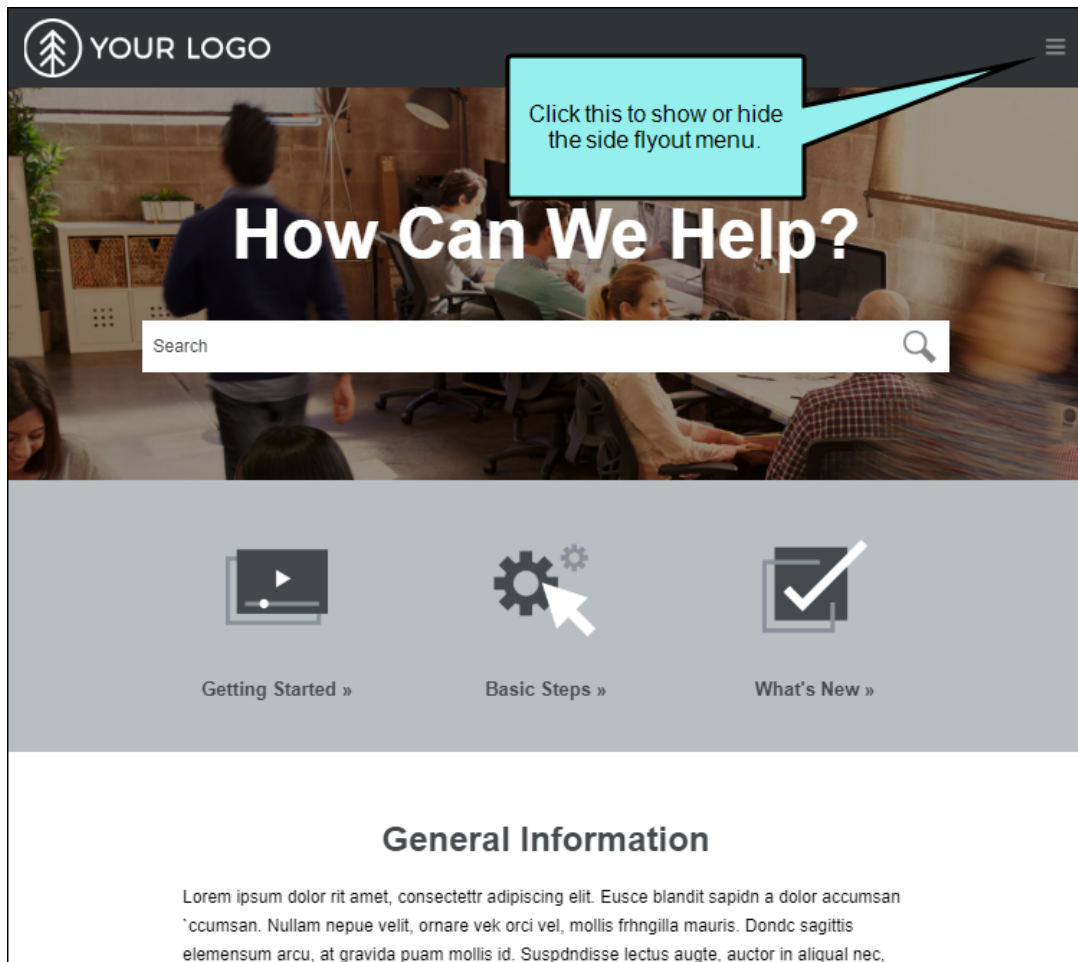
Similar to the fields in the Skin Editor, you can set values to tell Flare at which sizes to change the display.

- **Tablet Breakpoint** Enter the number of pixels for the maximum width of a Tablet view.
- **Mobile Breakpoint** Enter the number of pixels for the maximum width of a Mobile (or phone) view.

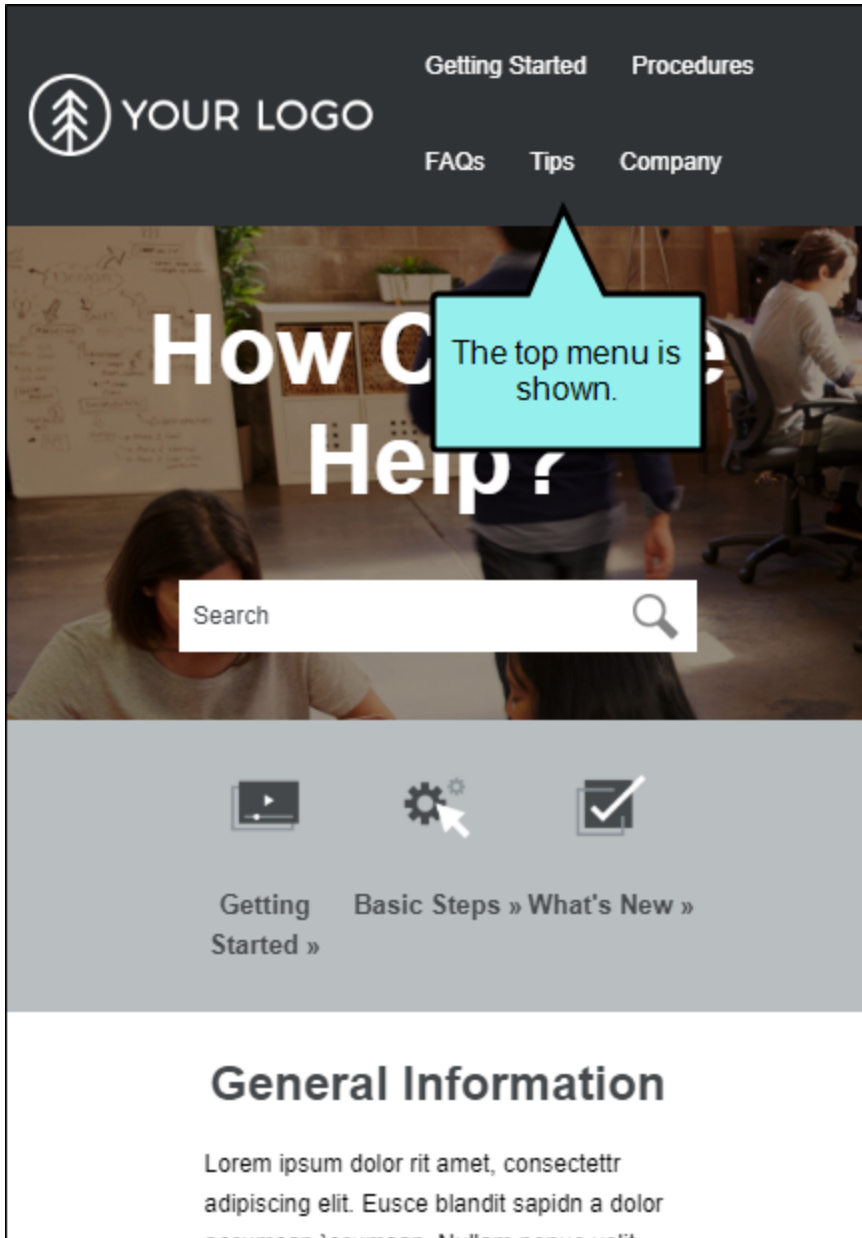
☆ **EXAMPLE** You disable **Use device width media queries**. When you view the output maximized in a browser, you will see the top menu.



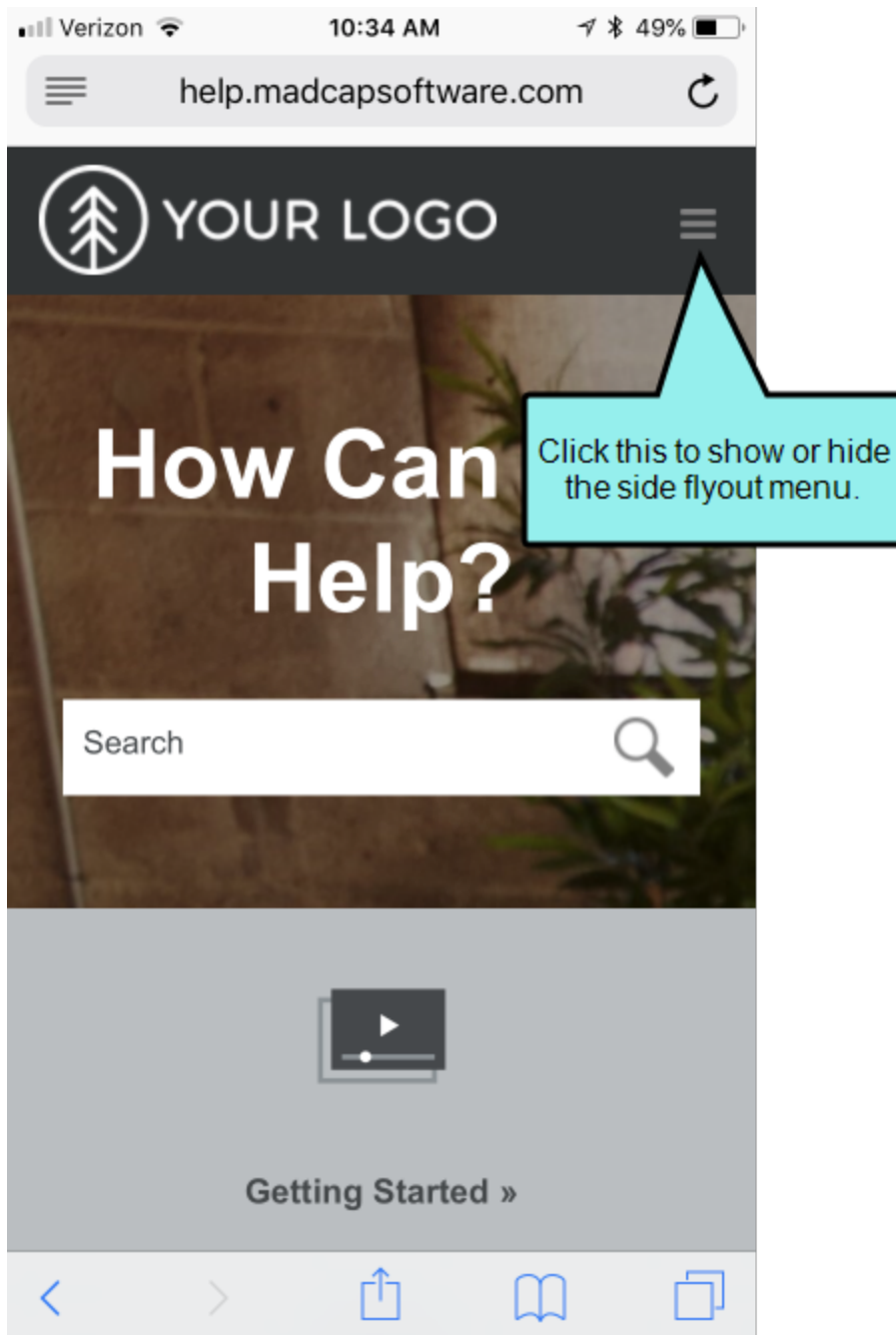
- ☆ When you drag the browser window, making it smaller so that its resolution is lower than the number you provided in the Tablet Breakpoint field, the display changes. Now you won't see the top menu anymore. Instead, you will see the side flyout menu. That's because the responsiveness is based on the width of the output in the browser, rather than on the width of the output in the device itself.



- ☆ Now suppose you enable **Use device width media queries**. When you view the output maximized in a browser, it will look just like it did before, with the top menu visible. But now when you make the browser window smaller, the display stays the same, even if you reduce it all the way down to the mobile width settings.



- ☆ But if you view that same output on an actual tablet or mobile phone, it will display with the side flyout menu.



✔ **TIP** Even if you want to base your responsive skin on the device width, you might find it most useful to leave the "Use device width media queries" option disabled while you are still editing content. This lets you test your responsive output more easily by dragging the browser to different sizes. Then when you're ready to generate and publish your final output, enable the check box.

✔ **TIP** If you want to disable the top menu when viewed on a browser, and use only the side flyout menu that is usually reserved for tablets and mobile devices, you can set the tablet width value to a very high number.

📄 **NOTE** If you have a Tripane skin in your project and you enter responsive output settings in both the Skin Editor and Target Editor, the settings in the target take precedence. However, this is not true if you *have not* yet made a change to the tablet or mobile breakpoint fields in the Target Editor, but you *have* made changes to them in the Skin Editor. In that case, the numbers from the changed Skin Editor will be used.



# Responsive Content

Depending on the output type, you can make your content responsive to different devices and screen sizes by using responsive conditions and responsive layouts, in conjunction with media queries.

This chapter discusses the following:

Responsive Conditions .....	166
Responsive Layouts .....	179
Responsive Features Per Output Type .....	222

# I Responsive Conditions

For HTML5 targets, conditions can be used to make content responsive (e.g., content changes automatically according to the size of the screen, rather than simply based on “include” or “exclude”). Associating conditions with factory media queries (Web, Tablet, Mobile) can help make some content more appropriate or easier to read on different devices.

## Features and User Interface Areas Affected

In addition to using this feature in targets, it can also be accessed when working in other areas of the user interface. Following are the features for which you can and cannot set conditions to display by media query:

Can Set by Media Query	Cannot Set by Media Query
<ul style="list-style-type: none"><li>▪ Target Editor</li><li>▪ XML Editor</li><li>▪ Preview</li><li>▪ Review Wizard</li></ul>	<ul style="list-style-type: none"><li>▪ Export Project Wizard</li><li>▪ Snippet Conditions</li><li>▪ Import Flare Project Wizard and Editor</li></ul>

## Limitations

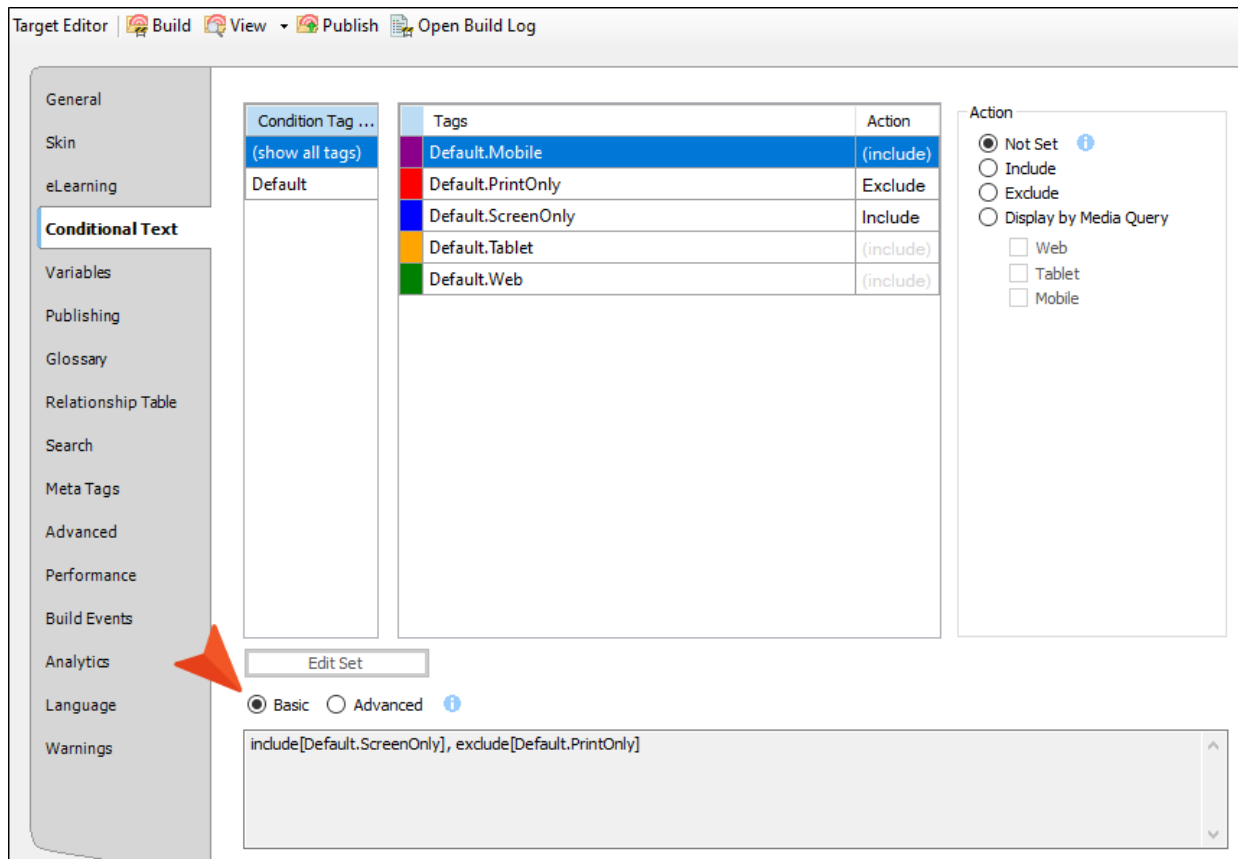
Keep the following in mind if you intend to display conditioned content according to media queries:

- This feature does not support all outputs. It works only for HTML5 targets.
- Only *factory* media queries (Web, Tablet, Mobile) are supported and are controlled by the breakpoints specified in the skin or target. This feature does not support *custom* media queries.
- Only conditions applied to *content* are supported. This feature does not support conditions that are applied to *folders and files*.

# How to Set Conditions by Media Query

Following are steps for setting conditions by media query in the Target Editor. The steps are similar if you are working in the XML Editor, Preview window, or Send Files for Review Wizard; the first few steps would vary, depending on the part of the user interface. For links to each full set of steps, see "Features and User Interface Areas Affected" on the previous page.

1. Open an HTML5 target.
2. Click the **Conditional Text** tab.
3. Toward the bottom of the tab, make sure the **Basic** option is selected.



4. In the **Condition Tag Sets** area, you can choose to view tags for all condition tag sets or you can select a specific set. The tags associated with the selected set are shown to the right, with their associated colors. An Action cell appears next to each condition tag. When Basic is selected, the Action cell is shown and the Advanced section below is disabled. In the Tags area, select a condition tag in the grid to set its action. (You can also hold down the SHIFT or CTRL key and choose multiple rows to set the same action on all of them at once.)

5. The Action cell in the grid indicates the current setting for the condition. The text is gray if you have not explicitly chosen an option. Select **Display by Media Query**. Then choose any of the options below it.
- **Web** The content will display if the user is viewing it on a large screen.

The screenshot shows a configuration interface for a condition tag set. On the left, a 'Condition Tag Sets' dropdown menu is open, showing '(show all tags)' and 'Default'. Below it is an 'Edit Set' button. In the center, a table lists various tags and their actions:

Tags	Action
Default.Mobile	(include)
Default.Online	Include
Default.PDF1	(include)
Default.PDF2	(include)
Default.Print	Exclude
Default.Tablet	(include)
Default.Web	Web

Below the table are radio buttons for 'Basic' (selected) and 'Advanced'. At the bottom, a text field contains the configuration: `include[Default.Online], exclude[Default.Print], @media web[Default.Web]`. On the right, an 'Action' panel shows radio buttons for 'Not Set', 'Include', 'Exclude', and 'Display by Media Query' (selected). Under 'Display by Media Query', there are checkboxes for 'Web', 'Tablet', and 'Mobile'. Red arrows point to the 'Web' action in the table and the 'Web' checkbox in the 'Display by Media Query' section.

- **Tablet** The content will display if the user is viewing it on a medium screen, such as an iPad.

The screenshot displays the AEM Condition Tag configuration interface. On the left, the 'Condition Tag Sets' panel shows 'Default' selected. The main area contains a table of tags and their actions:

Tags	Action
Default.Mobile	(include)
Default.Online	Include
Default.PDF1	(include)
Default.PDF2	(include)
Default.Print	Exclude
Default.Tablet	Tablet
Default.Web	Web

Below the table is an 'Edit Set' button and radio buttons for 'Basic' (selected) and 'Advanced'. At the bottom, the configuration code is: `include[Default.Online], exclude[Default.Print], @media web[Default.Web], @media tablet[Default.Tablet]`


On the right, the 'Action' panel shows 'Display by Media Query' selected. Under the 'Web' category, the 'Tablet' checkbox is checked and highlighted with a red box. A red arrow points to this checkbox.


- **Mobile** The content will display if the user is viewing it on a small screen, such as an iPhone.


Condition Tag Sets	Tags	Action	Action
(show all tags)	Default.Mobile	Mobile	<input type="radio"/> Not Set ⓘ
Default	Default.Online	Include	<input type="radio"/> Include
	Default.PDF1	Exclude	<input type="radio"/> Exclude
	Default.PDF2	Exclude	<input checked="" type="radio"/> Display by Media Query
	Default.Print	Exclude	<input type="checkbox"/> Web
	Default.Tablet	Tablet	<input type="checkbox"/> Tablet
	Default.Web	Web	<input checked="" type="checkbox"/> Mobile

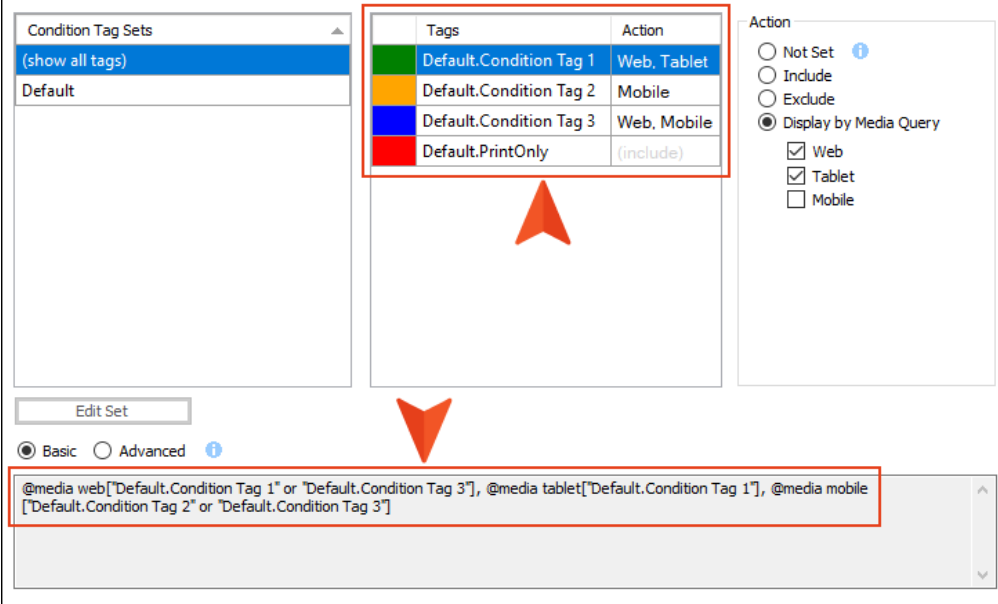
Basic  Advanced ⓘ

```
include[Default.Online], exclude[Default.PDF1 or Default.PDF2 or Default.Print], @media web[Default.Web], @media tablet [Default.Tablet], @media mobile[Default.Mobile]
```

 **NOTE** If you select all three of the media query check boxes, it has the same effect as choosing the “Include” option.

 **NOTE** The “Display by Media Query” option is not available unless you are working in an HTML5 target. Other target types do not show this option.

 **NOTE** The Basic method is the easiest way to display conditions by media query. However, you can also use the Advanced method. If you choose to do so, the markup should look something like the following.



Tags	Action
Default.Condition Tag 1	Web, Tablet
Default.Condition Tag 2	Mobile
Default.Condition Tag 3	Web, Mobile
Default.PrintOnly	(include)

Condition Tag Sets  
(show all tags)  
Default

Action

Not Set *i*

Include

Exclude

Display by Media Query

Web


Tablet

Mobile

Edit Set

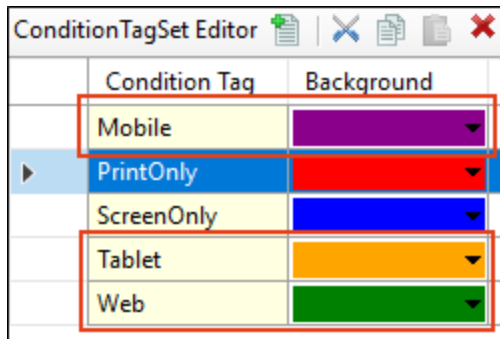
Basic  Advanced *i*

```
@media web["Default.Condition Tag 1" or "Default.Condition Tag 3"], @media tablet["Default.Condition Tag 1"], @media mobile ["Default.Condition Tag 2" or "Default.Condition Tag 3"]
```

6. Click  to save your work.

☆ **EXAMPLE** You are writing documentation for a cloud-based product, where it is more appropriate to tell users to “click” an option if they are working on a large screen, but to “tap” the option if they are using a tablet or smart phone.

So in a condition tag set, you create three new conditions—one for each of those screen sizes.

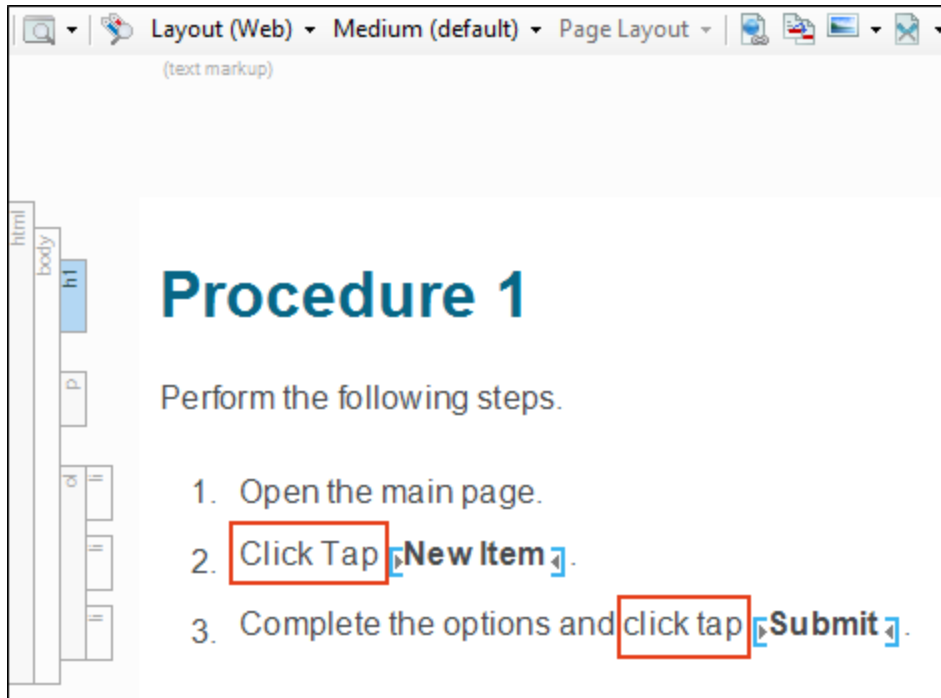


The screenshot shows a window titled "Condition TagSet Editor" with a toolbar containing icons for adding, deleting, and saving. Below the toolbar is a table with two columns: "Condition Tag" and "Background". The table contains five rows of data, each with a colored background and a dropdown arrow. The rows are: Mobile (purple), PrintOnly (red), ScreenOnly (blue), Tablet (orange), and Web (green). The "PrintOnly" row is highlighted with a blue selection bar on the left. Red boxes are drawn around the "Mobile", "Tablet", and "Web" rows.

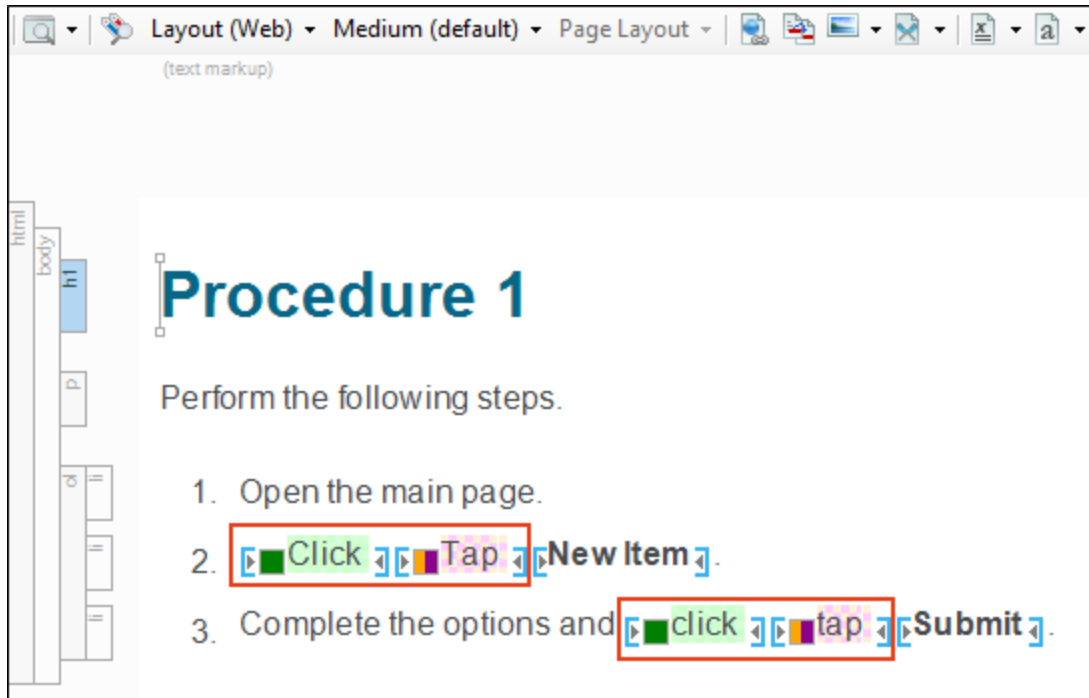
Condition Tag	Background
Mobile	▼
PrintOnly	▼
ScreenOnly	▼
Tablet	▼
Web	▼



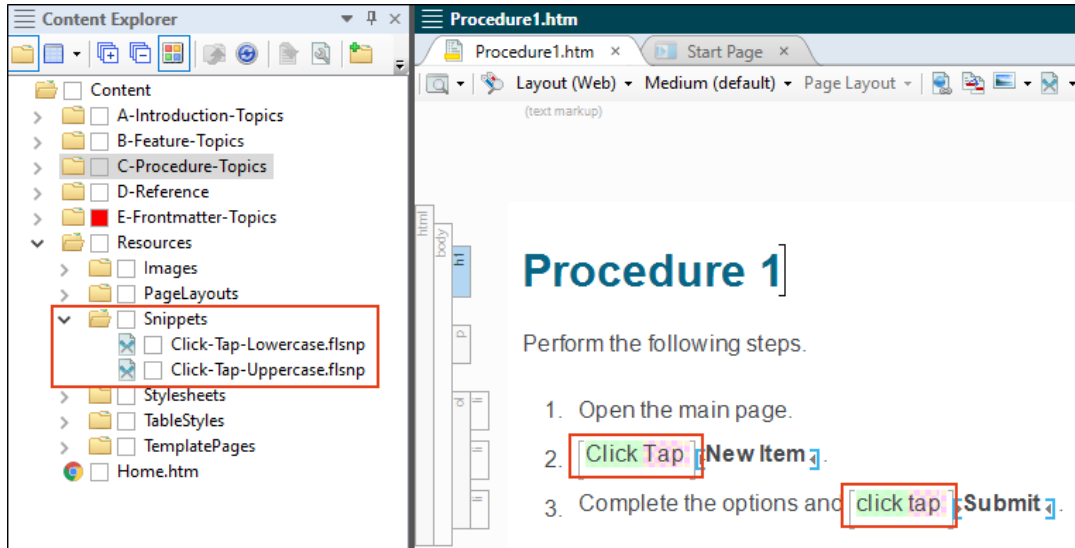
- ☆ You write a procedure where you need to use to tell users to click or tap something. Therefore, you add both words to the instructions.



- ☆ Then you apply the condition you named “Web” to the word “click,” and you apply the “Tablet” and “Mobile” conditions to the word “tap.”



- ☆ You anticipate that you will need to use these words many times throughout the project, so you create snippets—one holding the uppercase pair of words, and the other holding the lowercase pair. That way, you can just insert the appropriate snippet wherever those words are needed, rather than applying the conditions to text each time.



☆ Next, you open the target and associate each of these conditions with the appropriate media query.

Tags	Action
Default.Mobile	Mobile
Default.PrintOnly	Exclude
Default.ScreenOnly	Include
Default.Tablet	Tablet
Default.Web	Web

Action

Not Set ⓘ

Include


Exclude

Display by Media Query

Web

Tablet

Mobile



Tags	Action
Default.Mobile	Mobile
Default.PrintOnly	Exclude
Default.ScreenOnly	Include
Default.Tablet	Tablet
Default.Web	Web

Action

Not Set ⓘ

Include


Exclude

Display by Media Query

Web

Tablet

Mobile



Tags	Action
Default.Mobile	Mobile
Default.PrintOnly	Exclude
Default.ScreenOnly	Include
Default.Tablet	Tablet
Default.Web	Web

Action

Not Set ⓘ

Include


Exclude

Display by Media Query

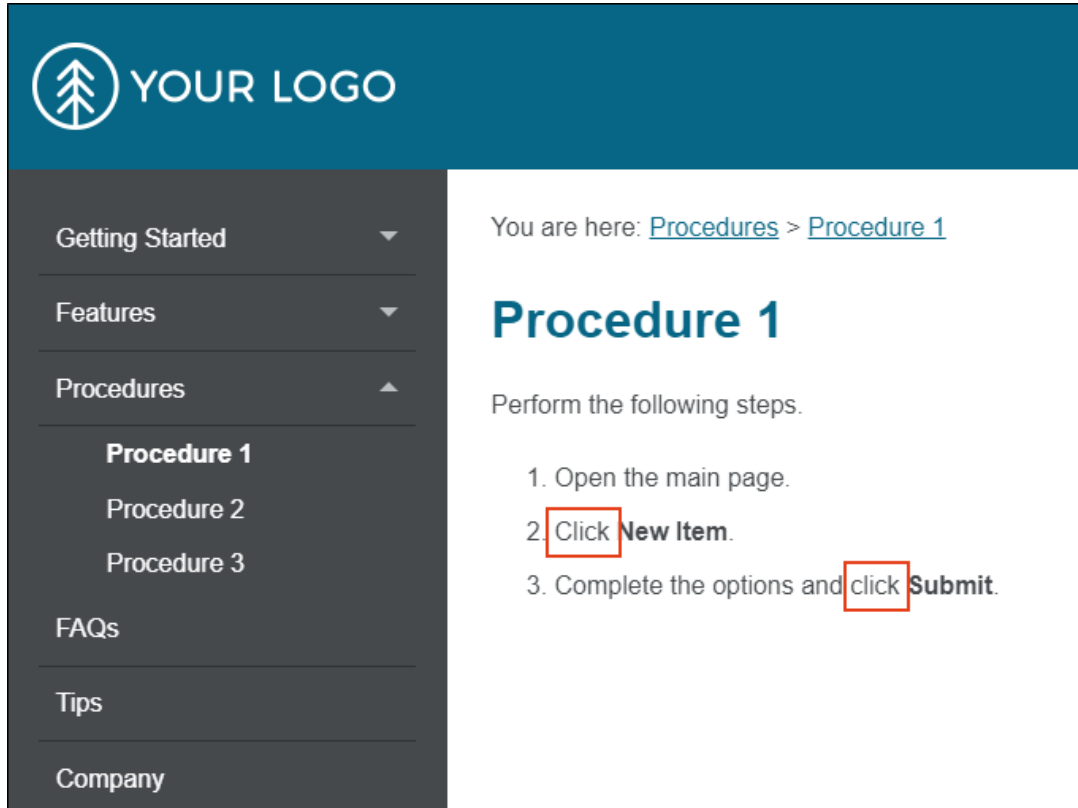
Web

Tablet

Mobile

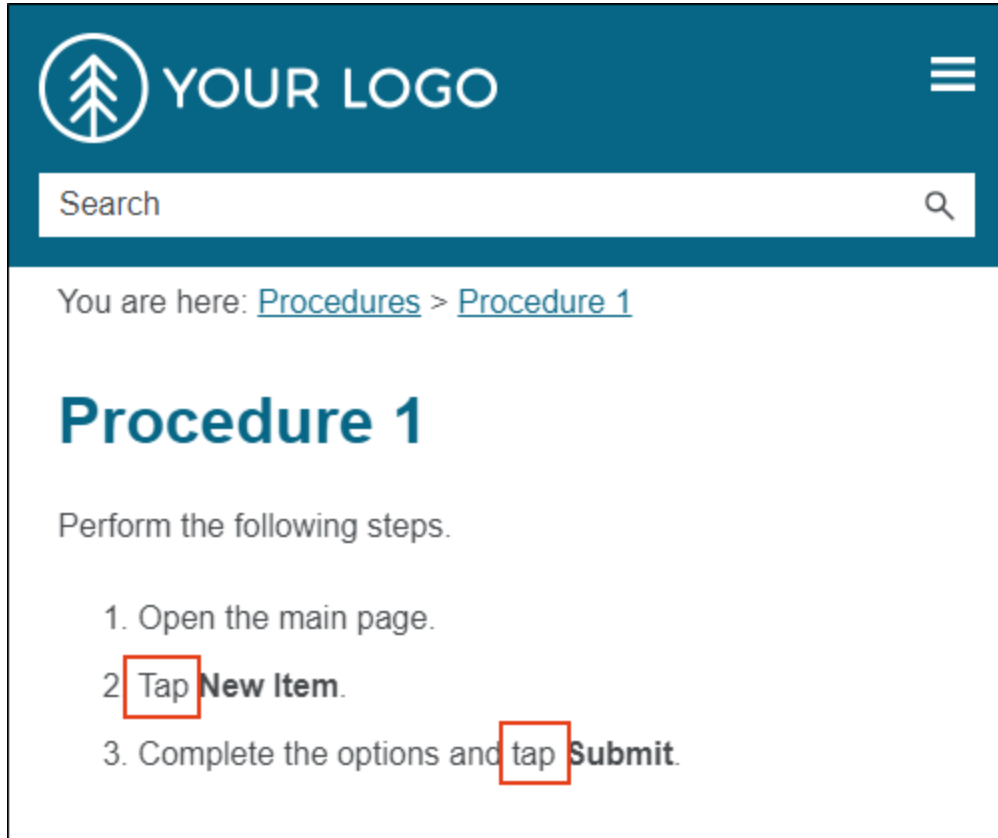


☆ When you generate the online output and view it on a large screen, the word “click” is used.



The screenshot displays a web application interface. At the top left, there is a logo consisting of a tree icon inside a circle, followed by the text "YOUR LOGO". Below the logo is a dark grey sidebar menu with the following items: "Getting Started" (with a downward arrow), "Features" (with a downward arrow), "Procedures" (with an upward arrow), "Procedure 1" (bolded), "Procedure 2", "Procedure 3", "FAQs", "Tips", and "Company". The main content area on the right has a teal header with the text "You are here: [Procedures](#) > [Procedure 1](#)". Below this is the section title "Procedure 1" in a large teal font. The text "Perform the following steps." is followed by a numbered list: "1. Open the main page.", "2. Click **New Item**.", and "3. Complete the options and click **Submit**". The words "Click" in step 2 and "click" in step 3 are highlighted with red boxes.

☆ But when you view the output on a smaller tablet or mobile screen, the word “tap” is used.




## What's Noteworthy?

📄 **NOTE** You can set responsive conditions on tables. However, if you also have a table stylesheet applied to the table, responsive conditions only work if they are set on the entire table (i.e., the `<table>` tag); they do not work if the conditions are set on individual rows, columns, or cells.

# I Responsive Layouts

Styles and media queries can be used to make your content responsive in HTML5 output. This lets you present information—both its substance and structure—differently depending on the size of the screen or device. Flare provides a Responsive Layout window pane that helps you create this kind of content more easily. You also have the option of using third-party solutions (e.g., Zurb Foundation grid system) by adding the appropriate styles in your stylesheets and topics.

 **NOTE** Flexbox is another popular web layout model. Unfortunately, Flare does not have an interface to support Flexbox. However, you can add styles to a stylesheet and then edit the topic markup directly using the Text Editor.

## System Options

- "Flare's Responsive Layout System" on page 183
- "Foundation and Other Systems" on page 195

## Process

1. "Creating Responsive Layouts" on page 196
2. "Editing Responsive Layouts" on page 213
3. "Viewing Responsive Output" on page 221

☆ **EXAMPLE** You have a topic that looks as follows when viewed on a large monitor or on a tablet-sized screen. Notice the three images at the bottom of the topic.

If you're brand new to Flare, we recommend you use the following to help you get started. These will help you become acquainted with the interface, common features, basic steps, special tips, and more.

Depending on your preference, you can watch a video, read a PDF guide, or work through a tutorial. You might even want to use all three, because although there is certainly common information in all three, each one is also unique.

- **Getting Started Video** The Getting Started Video provides a visual demonstration of how to create a new project, introduces basic information about Flare, and points you to additional videos and resources to further develop the project.
- **Getting Started Guide** The Getting Started Guide is a PDF that explains how Flare works and introduces you to some of its key features and concepts. It then outlines the five basic steps for developing a project and producing output.
- **Getting Started Tutorial** The Getting Started Tutorial is a PDF file that you can use alongside Flare to create, develop, and generate output from an actual project.



**Getting Started  
Video »**



**Getting Started  
Guide »**



**Getting Started  
Tutorial »**



☆ But when the browser is reduced to a mobile size, the images shift so that they display one on top of the other.

- **Getting Started Tutorial** The Getting Started Tutorial is a PDF file that you can use alongside Flare to create, develop, and generate output from an actual project.




Getting Started  
Video »



Getting Started  
Guide »



Getting Started  
Tutorial »

 **NOTE** Depending on the approach you use, it may be technically possible to create responsive content in other formats besides HTML5. However, you will notice that best results are achieved with HTML5 targets, and in particular, Side and Top Navigation outputs. Many of the other formats are able to produce responsive content only in very basic ways.

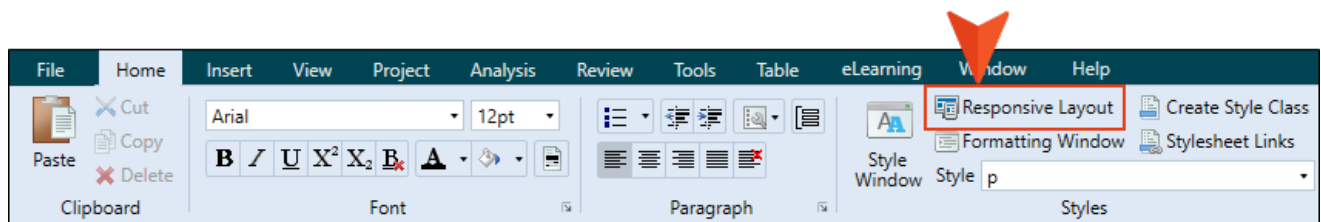
Also, even though the Responsive Layout window pane is designed with only online output in mind, the content you add to the layouts will show up in other outputs, including print-based outputs (unless you condition that content out). You can open your stylesheet, choose the print medium, and edit the resulting div and complex selectors that were created from a responsive layout. You just might find that you are able to reuse those layouts in your print-based outputs, depending on the simplicity of the responsive layout and the style settings that you provide for the print output.

# Responsive Layout Systems

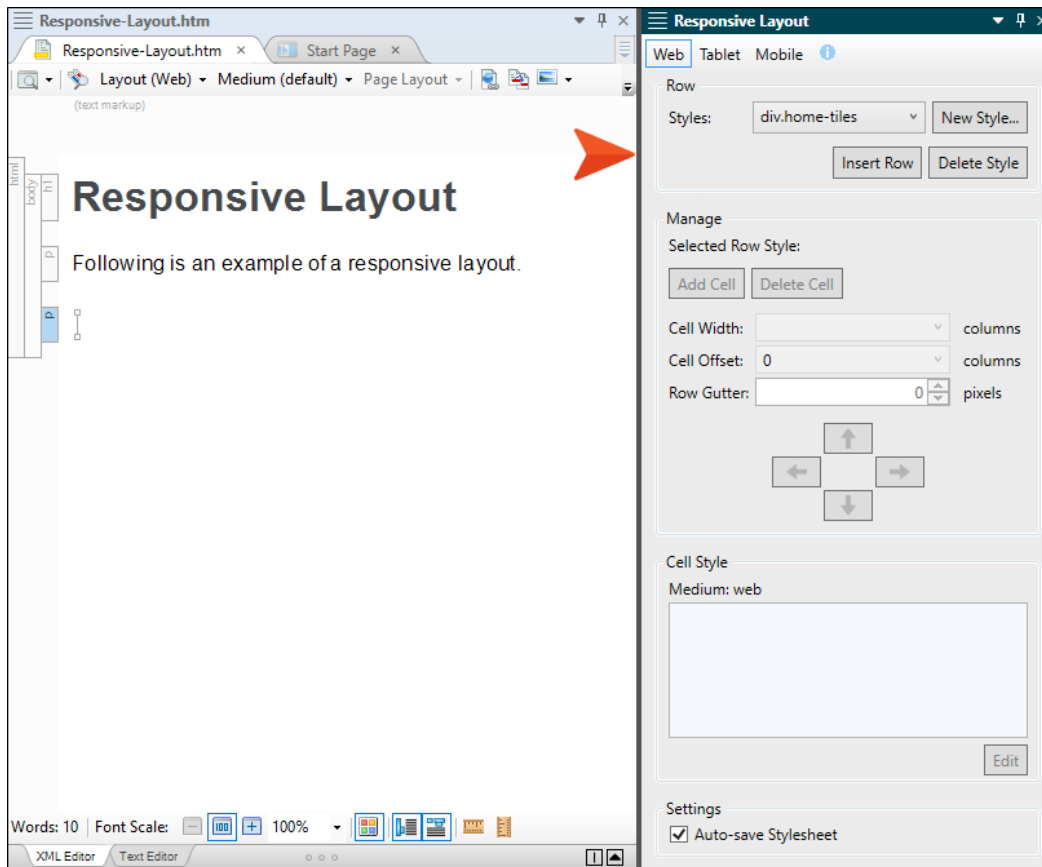
When using responsive layouts, it is important to understand the different systems that are available.

## Flare's Responsive Layout System

Flare's system for creating responsive content is possible with its Responsive Layout window pane, which 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 window pane 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.

## Main Activities

- "Creating Responsive Layouts" on page 196
- "Editing Responsive Layouts" on page 213
- "Viewing Responsive Output" on page 221

# 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 tool interface. The main workspace displays a row with three colored cells: Cell 1 (orange), Cell 2 (teal), and Cell 3 (yellow). A 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 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 callout bubble points to the 'Cell Width' field in the right-hand pane, 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 callout bubble points to the 'width: 50%' property in the Cell Style pane, stating: "The percentage is shown here." The right-hand pane shows the 'Responsive Layout' window with tabs for 'Web', 'Tablet', and 'Mobile'. The 'Web' tab is selected. The 'Row' section shows 'Styles: div.ThreeCells' and buttons for 'Insert Row' and 'Delete Style'. The 'Manage' section shows 'Selected Row Style: div.ThreeCells' and buttons for 'Add Cell' and 'Delete Cell'. The 'Cell Width' field is set to '6' columns, 'Cell Offset' is '0' columns, and 'Row Gutter' is '0' pixels. The 'Cell Style' section shows 'Medium: web' and a code block with 'background-color: #ffdab9;', 'width: 50%;', and 'margin-left: 0%;'. The 'Settings' section has 'Auto-save Stylesheet' checked. The bottom status bar shows 'Words: 16', 'Font Scale: 100%', and 'XML Editor | Text Editor'.

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.*

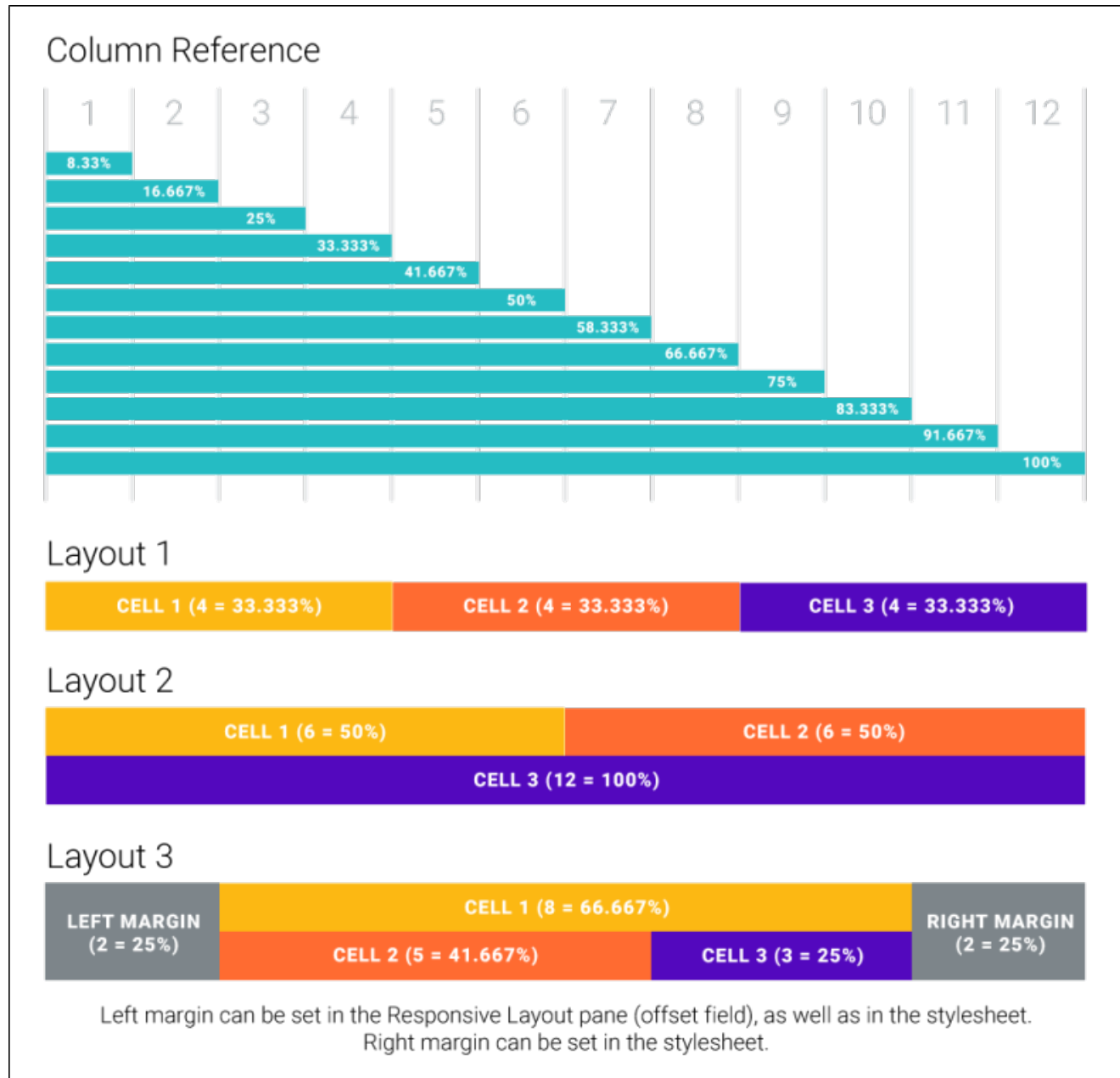
For your large web layouts, you usually want the total number of columns in a row to add up to 100%. However, you can use fewer columns if you want (e.g., you might end up using only 10 columns, or roughly 83% of the space). In fact, there might be times when you use more than 100%, even for the large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

The screenshot shows a web editor interface with a responsive layout. The layout consists of three cells: Cell 1 (orange, 12 columns wide), Cell 2 (cyan, 6 columns wide), and Cell 3 (yellow, 6 columns wide). Cell 1 is on the top row, and Cell 2 and Cell 3 are on the bottom row. The settings panel on the right shows the 'Cell Width' set to 12 columns. The 'Cell Style' panel shows the CSS for the selected cell, including 'background-color: #ffdab9;' and 'width: 100%;'. Three callout boxes provide additional context: 'We've changed the number of columns for Cell 1 to 12 (100%).', 'We also changed the number of columns for the other cells to 6 (50%) each. Because Cell 1 took up 100% of the space above, Cell 2 and Cell 3 had to move downward.', and 'Because you are the one who created the responsive layout, you know that it is really only one row. But to the end user, it looks like two rows with three cells (i.e., a grid).'

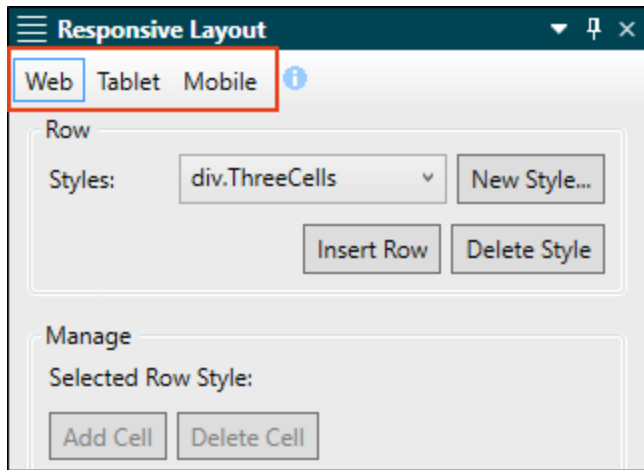
You can also use the Offset field in the Responsive Layout window pane to give any cell a left margin percentage (by assigning a number of columns to it). As far as right margins are concerned, you can accomplish that by editing your stylesheet (see "Responsive Layout Styles" on page 191).

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.

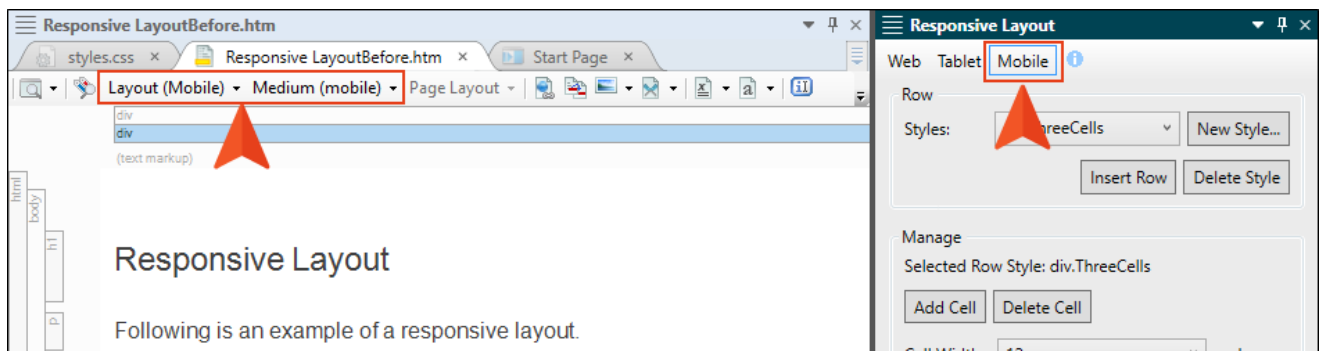


## Layout and Media Query Integration

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.



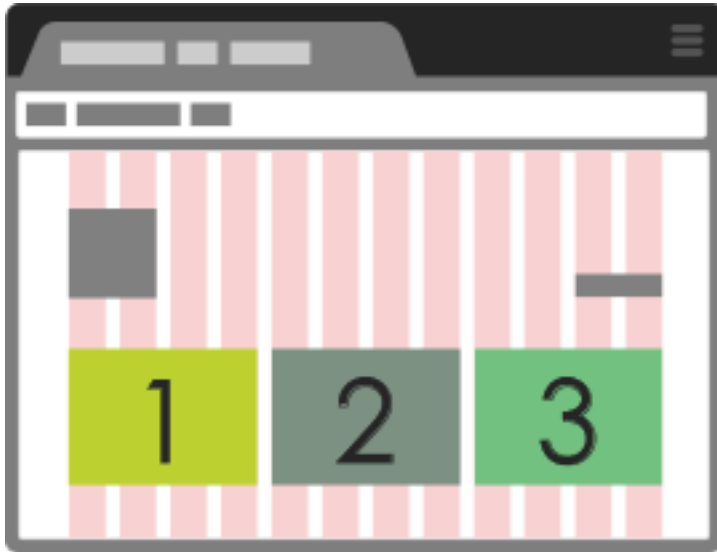
These are synchronized with the layout and medium settings in the XML Editor. So when you select a particular layout in the XML Editor, the appropriate medium is also selected in both the XML Editor and the Responsive Layout window pane. And if you choose a medium in the Responsive Layout window pane,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

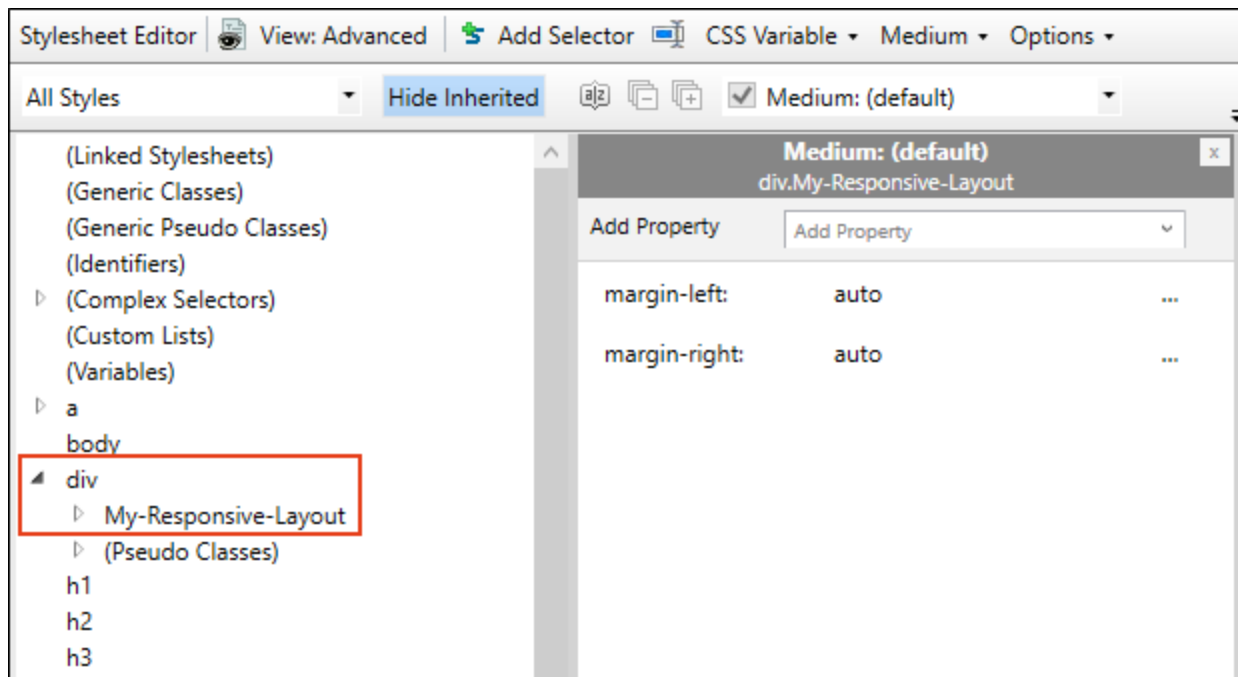


As the image above shows, the first cell took up the largest amount of space (12 columns, or 100%), so Flare had to shift the next cell downward. The second cell also used 100% of the space, so Flare had to move the third cell down even farther.

## Responsive Layout Styles

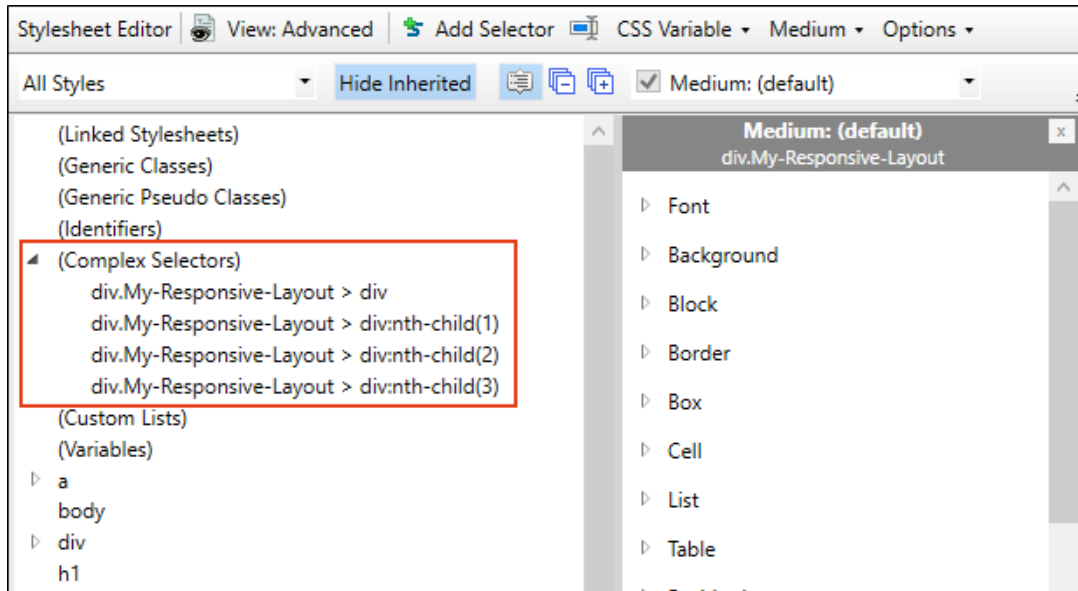
If you've used the Foundation framework, you know that there are several styles involved. In the simplified Flare system, there are only a few styles to know about.

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.



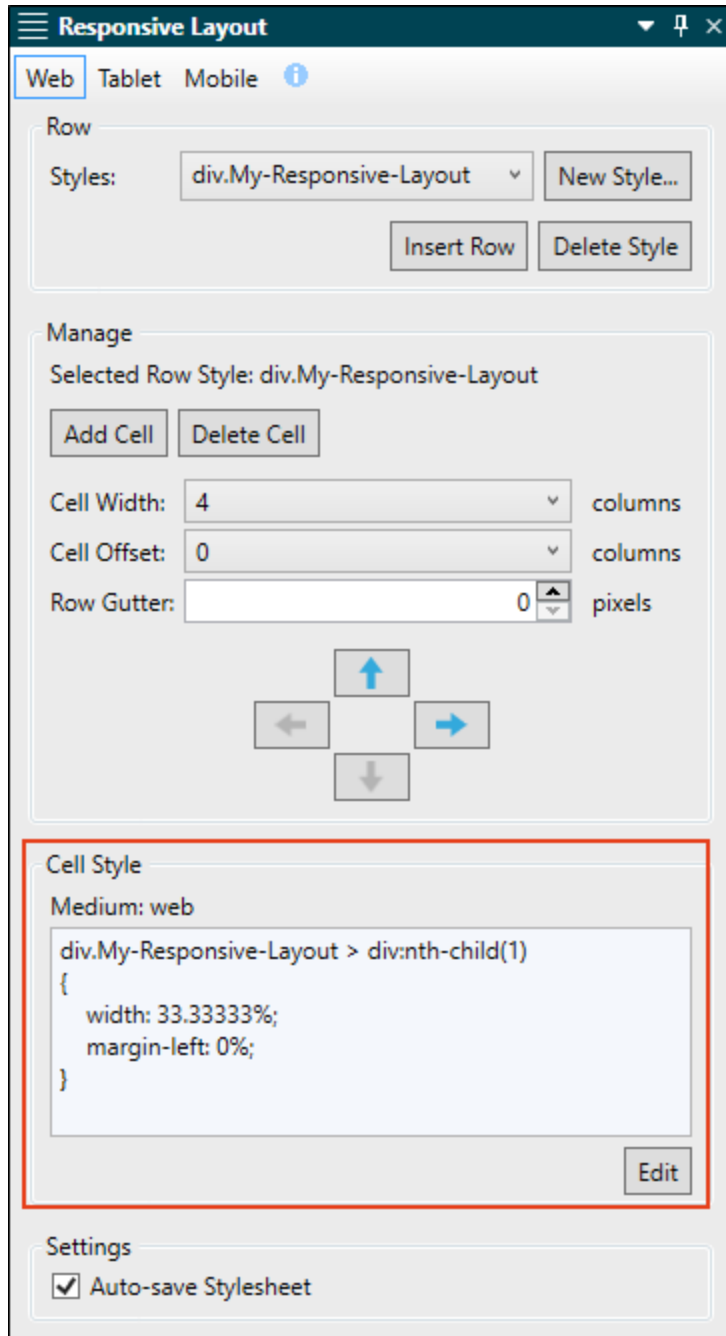
In fact, when you are all done adding cells to the row and assigning columns, you will also end up with multiple complex selectors in your stylesheet. These selectors consist of div styles, using the nth-child pseudo class for each cell.

☆ **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.My-Responsive-Layout > 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.




✔ **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.

# Foundation and Other Systems

As an alternative to using Flare's Responsive Layout window pane, you can use a third-party framework to create responsive output.

Zurb Foundation (<http://foundation.zurb.com/>) lets you freely download files such as CSS, HTML, and Javascript in order to take advantage of the responsive output framework they've created. To support responsive content in Flare, a Foundation CSS file has been downloaded and placed in the following location where you installed Flare:

```
C:\Program Files\MadCap Software\MadCap Flare 20\  
Flare.app\Resources\WebHelp2\Desktop\Skins\Fluid\Stylesheets\foundation.5.5.0
```

 **NOTE** It is recommended that you do not edit the Foundation stylesheet or move it from this location in the application installation folder. However, if you want, you can make a copy of the stylesheet, place it in any Flare project, and make changes to it. Any stylesheets in your project have precedence over any stylesheets located in your Flare installation folder.

There are other frameworks similar to Foundation. Bootstrap may be the most popular alternative. As long as you have a system using valid CSS and HTML, you can use it in Flare. However, Foundation is the only responsive framework where the stylesheet is already included in Flare. To use another system, you would need to download the appropriate CSS files and add them to your Flare project.

Keep in mind that Foundation and other third-party frameworks are not supported in Flare's XML Editor. So topics using these systems will likely not appear in the XML Editor the same way they appear in the output. On the other hand, the 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.

Also, Flare's Responsive Layout window pane is much more simple than these third-party systems. Although you can do a great deal with these other frameworks, they also require more time and effort to learn how to use them.

# Creating Responsive Layouts

You can create responsive layouts in Flare using the Responsive Layout window pane. This window pane works alongside media queries and your stylesheet.

## How to Create a Responsive Layout

1. Open the content file.
2. Select **Home > 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**.

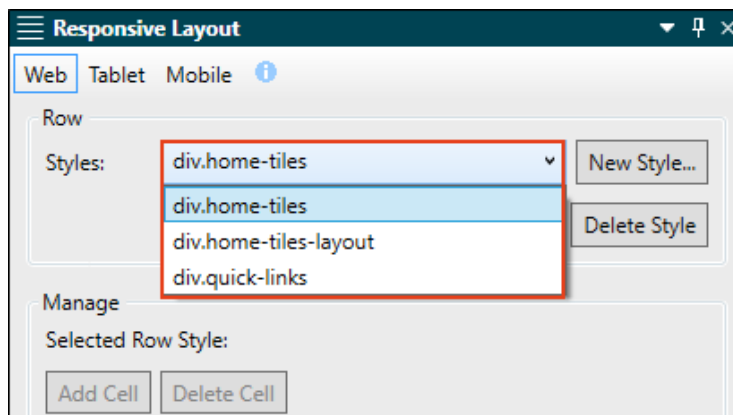
Alternatively, you can select the appropriate layout view in the top local toolbar of the XML Editor.

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.

### TO SELECT AN EXISTING STYLE

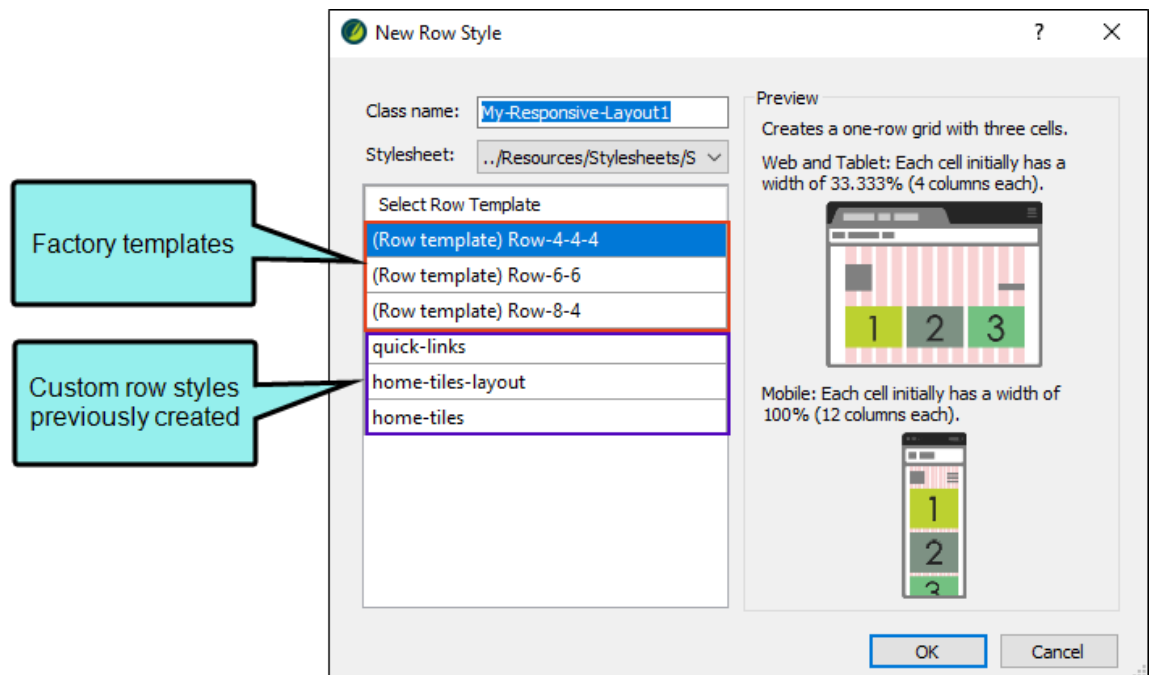
If you want to use the same row style for a responsive layout that you've created previously, click in the **Styles** drop-down and select it.





## TO 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. Iso, 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.

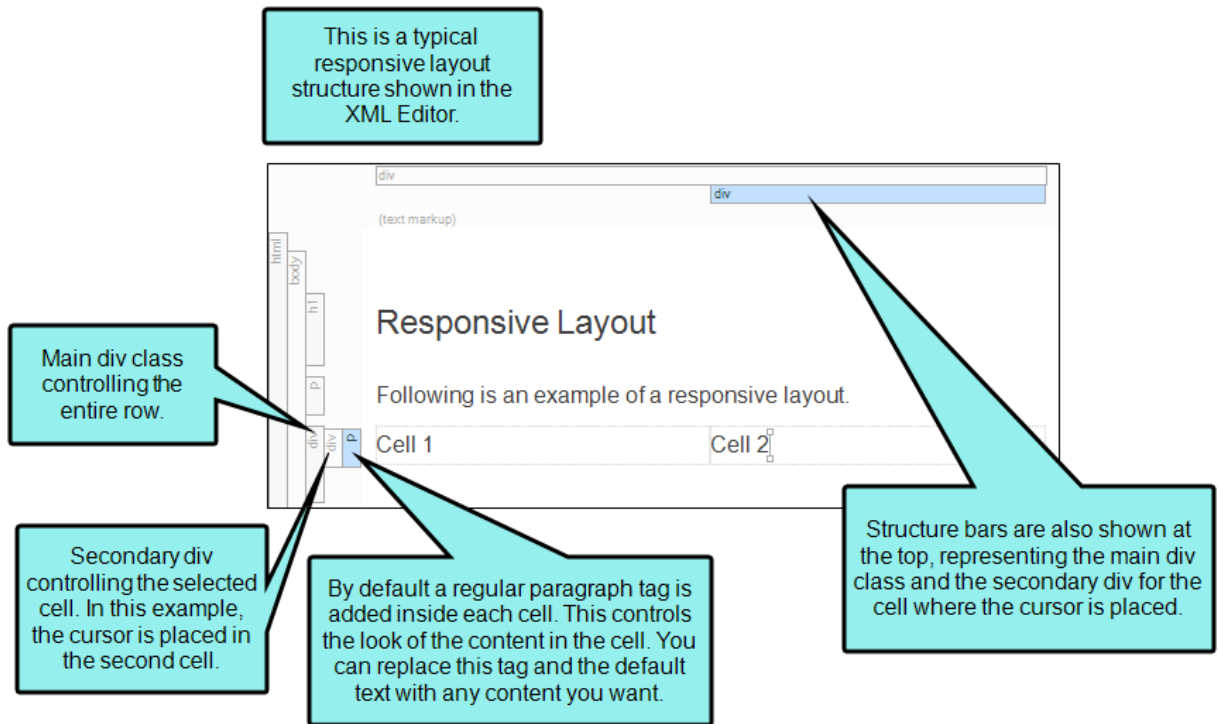
Selecting a template simply makes a copy of the style settings in that template and places them in your new style. You can then make changes to those styles when you choose different settings in the Responsive Layout window pane. So when you choose a template, pick one that is closest to the design that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

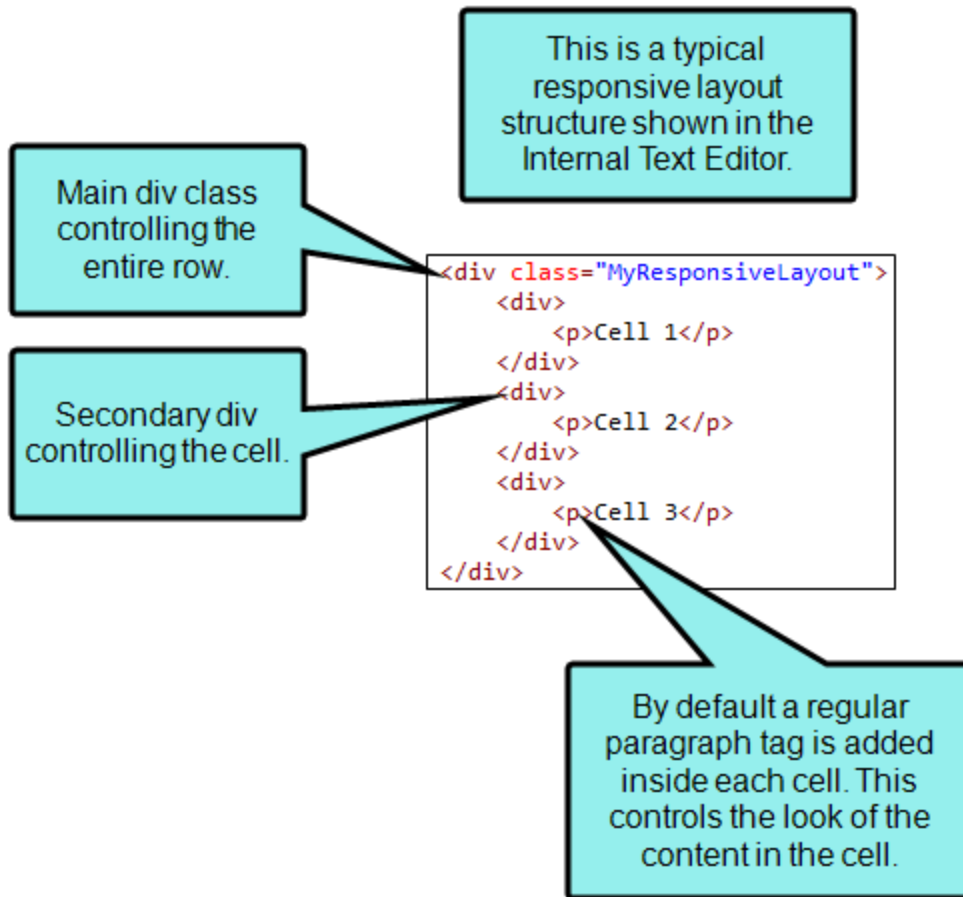
Following are the three factory templates:

- **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.

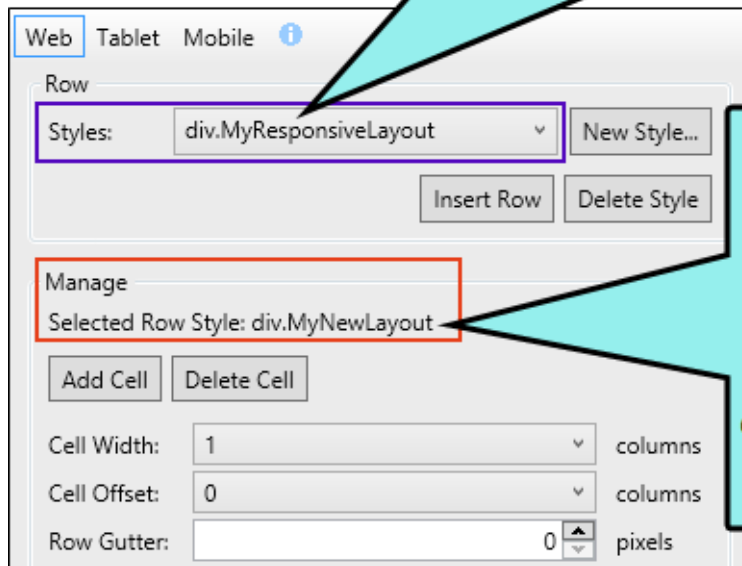
6. Click **Insert Row**. In the XML Editor, the row is inserted below your cursor location. A `<div>` tag—using the style name that you created or selected in the previous step—acts as the container for the row. Within that `<div>` tag there will be additional plain `<div>` tags, one for each cell.





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.

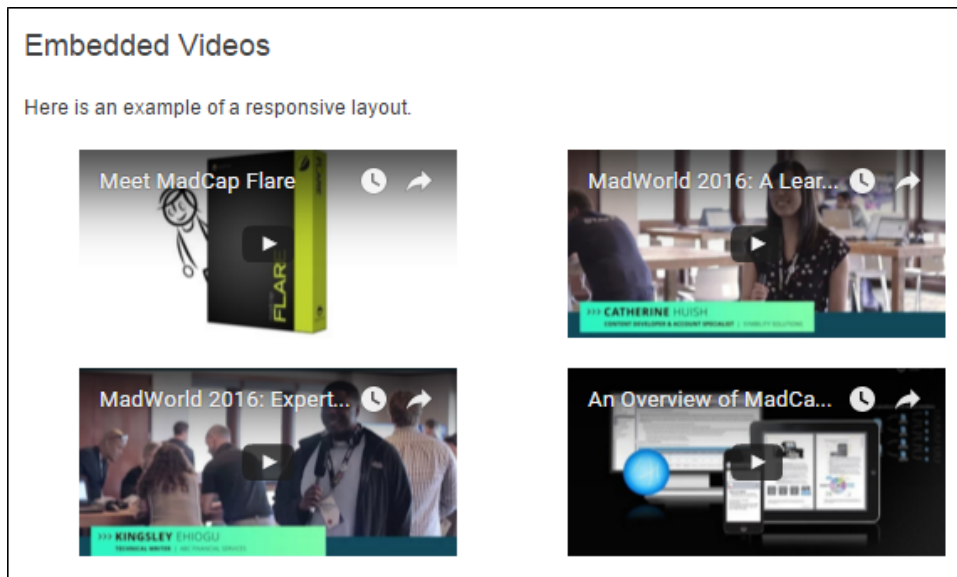
**⚠ WARNING** Keep in mind that your changes in this window pane will be auto-saved to your stylesheet by default. This means that anyone making changes to a particular row style using the Responsive Layout window pane will be affecting any other responsive layouts that are using that same style wherever 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
	<p>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.</p>
<p><b>Edit</b></p>	<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>
<p><b>Auto-save</b></p>	<p>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.</p>

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.

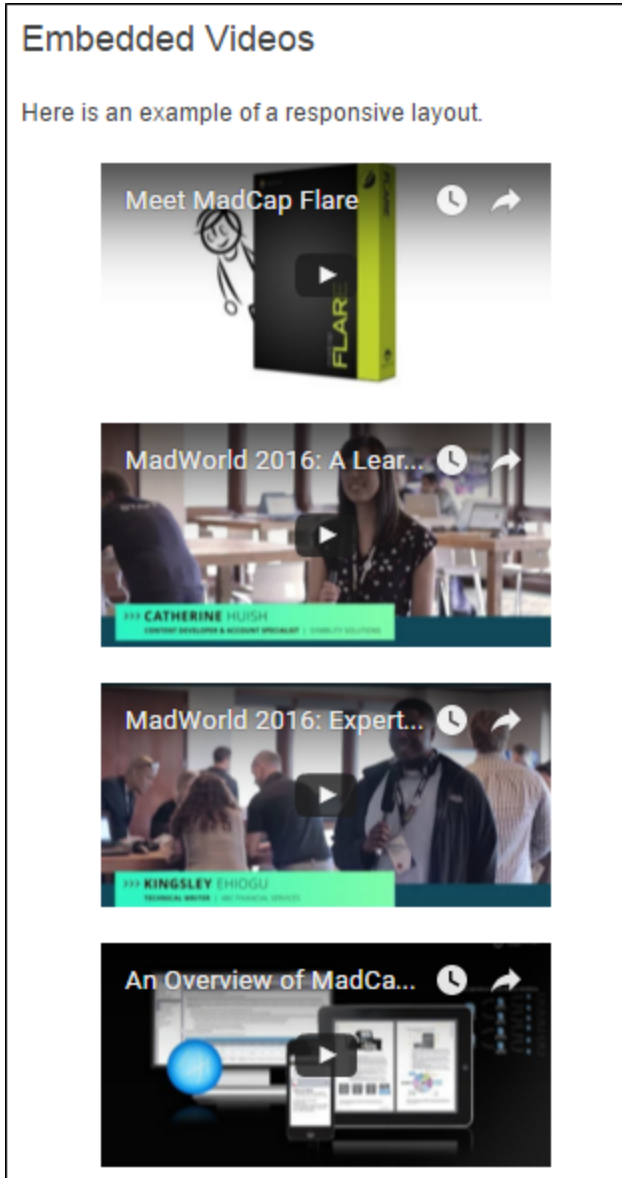
☆ **EXAMPLE** You want to embed four YouTube videos in a topic. When viewed on a large screen, you want the videos to display like this, two videos above the other two:



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

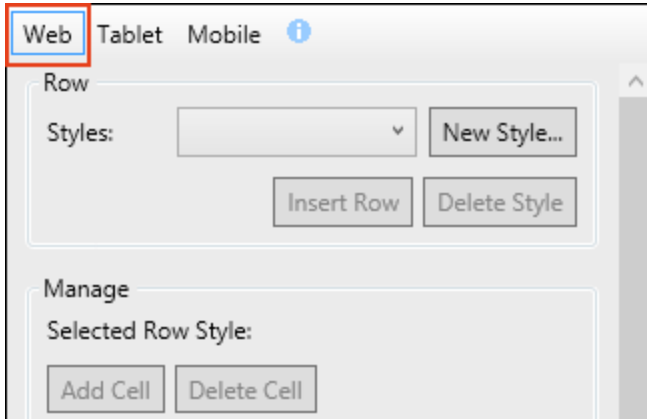


☆ But when viewed on a smart phone, you want each video to be displayed by itself, like this:

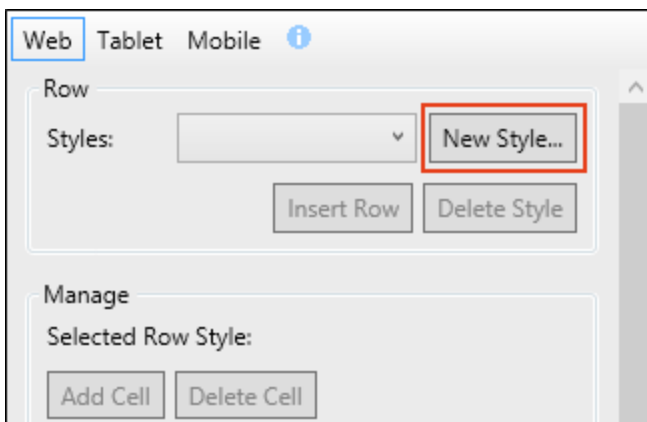


To make this happen, you first open the topic where you want to insert the videos, and you also open the Responsive Layout window pane.

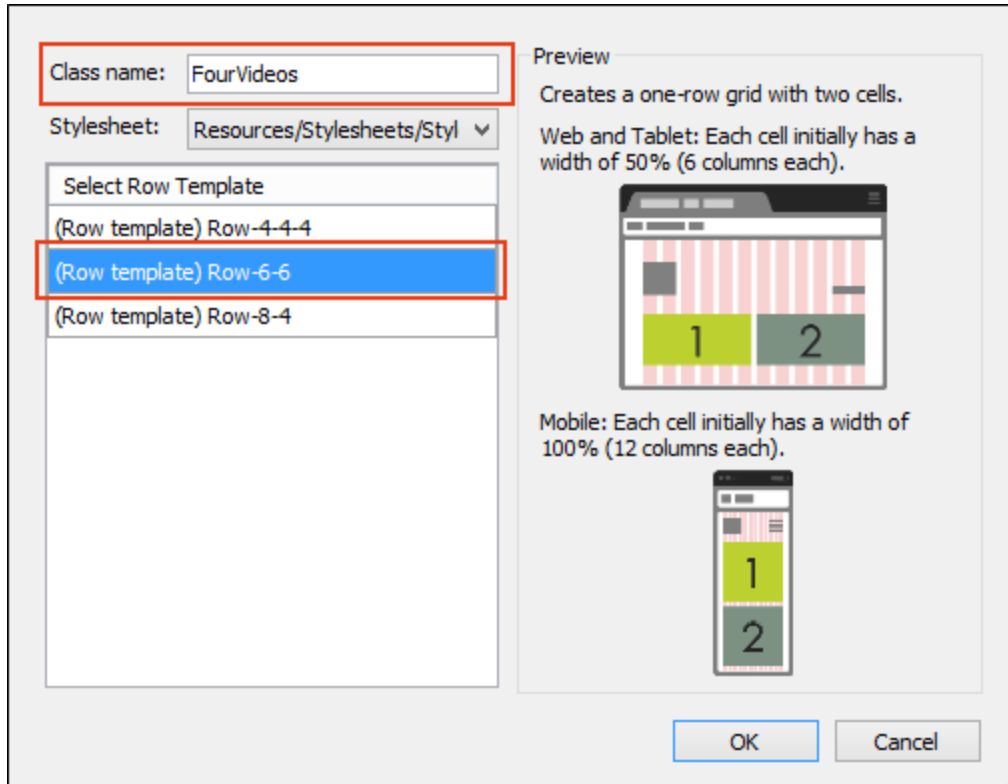
☆ Make sure the **Web** medium is selected in the Responsive Layout window pane.



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**.

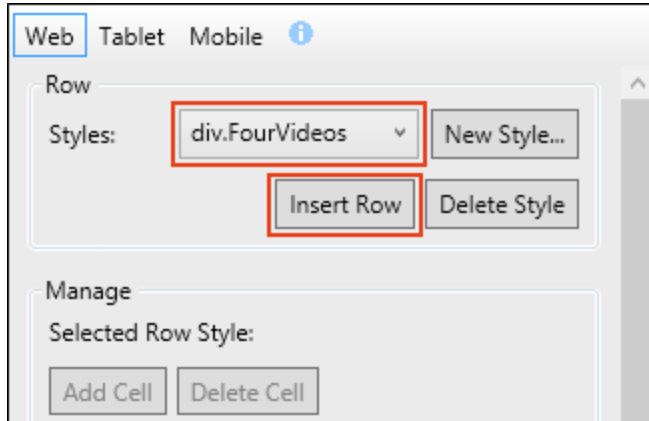


- ☆ In the New Row Style dialog, give the new style a name. Let's say you call it **FourVideos**. Your stylesheet is already selected in the dialog, but you need to choose a template. Click **Row-6-6**, which will start you out with two cells in the row, each taking up half the total space.

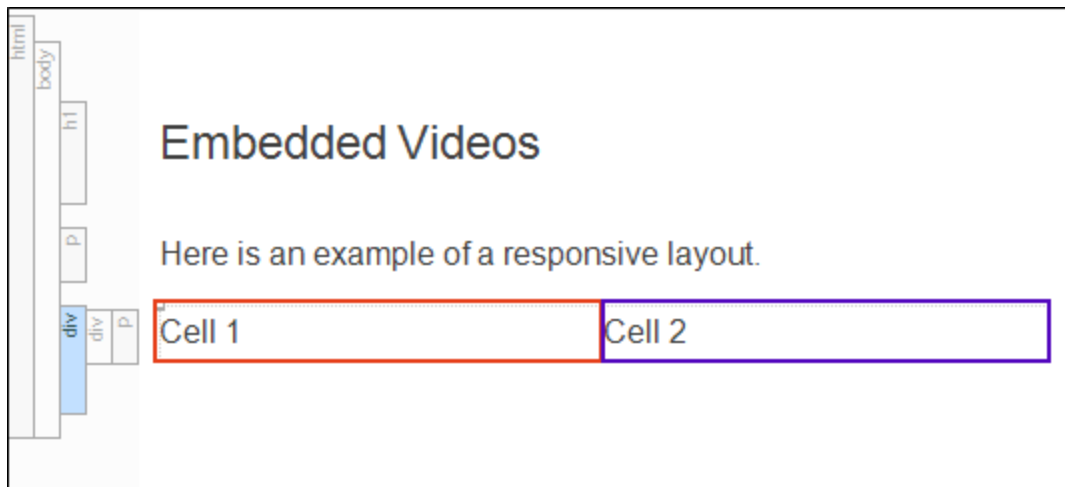


After you click **OK**, the row style is added to your stylesheet and is selected in the Responsive Layout window pane. Also, your stylesheet will open behind the topic.

☆ Now click **Insert Row**.



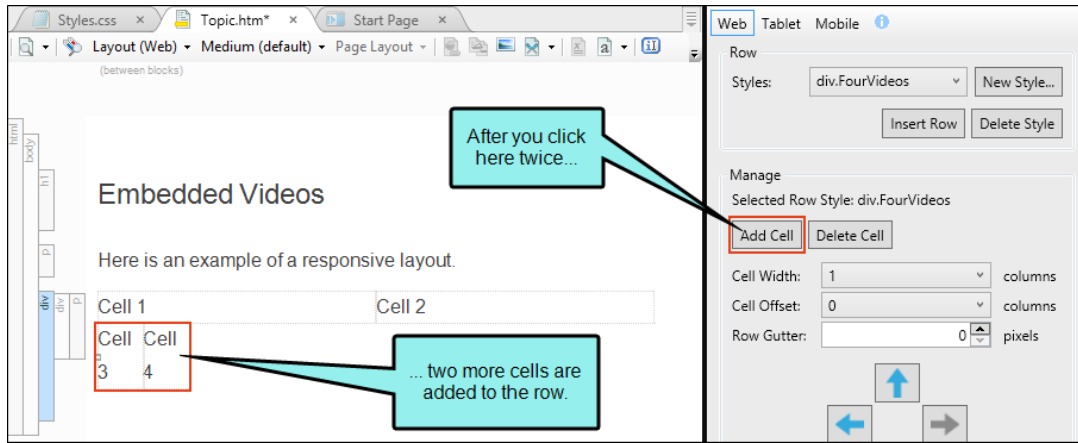
In your topic, you will see two cells, indicated by the dummy text "Cell 1" and "Cell 2."



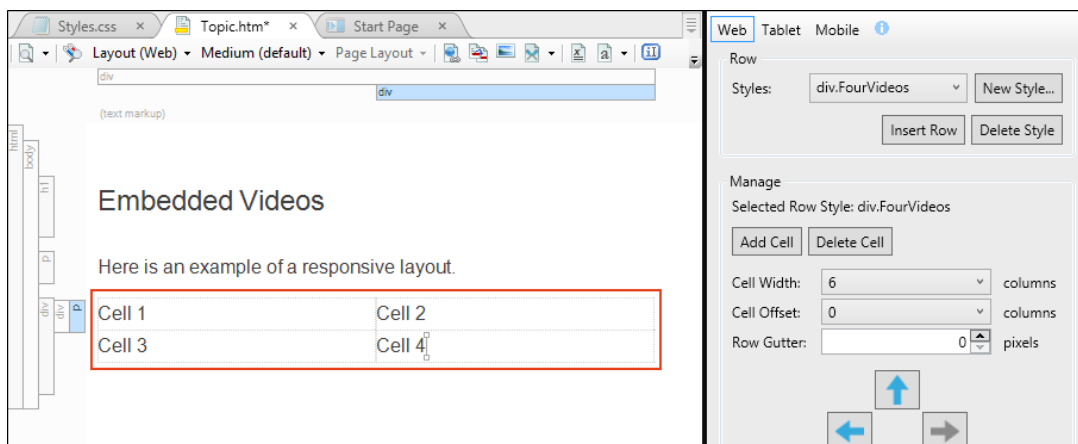
- ☆ Click in either of the cells. The Responsive Layout window pane will change, showing the buttons that you can select, as well as the current style settings for that cell.

The screenshot shows a web design tool interface with a responsive layout editor. The main workspace displays a grid with two cells, 'Cell 1' and 'Cell 2'. 'Cell 2' is highlighted with a red border. A callout points to 'Cell 2' with the text: 'In this example, we clicked in Cell 2.' Another callout points to the top of the grid with the text: 'This indicates the style being used for the row where your cursor is placed.' A third callout points to the 'Add Cell' and 'Delete Cell' buttons in the 'Manage' section of the right-hand pane, with the text: 'These buttons are now enabled.' A fourth callout points to the 'Cell Width' field in the 'Manage' section, which is set to '6 columns', 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.' A fifth callout points to the 'Cell Style' section of the right-hand pane, which shows a CSS selector 'div.FourVideos > div:nth-child(2)' and a 'width: 50%; margin-left: 0%;' rule. The right-hand pane also shows 'Selected Row Style: div.FourVideos' and 'Cell Offset: 0 columns'.

☆ 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.



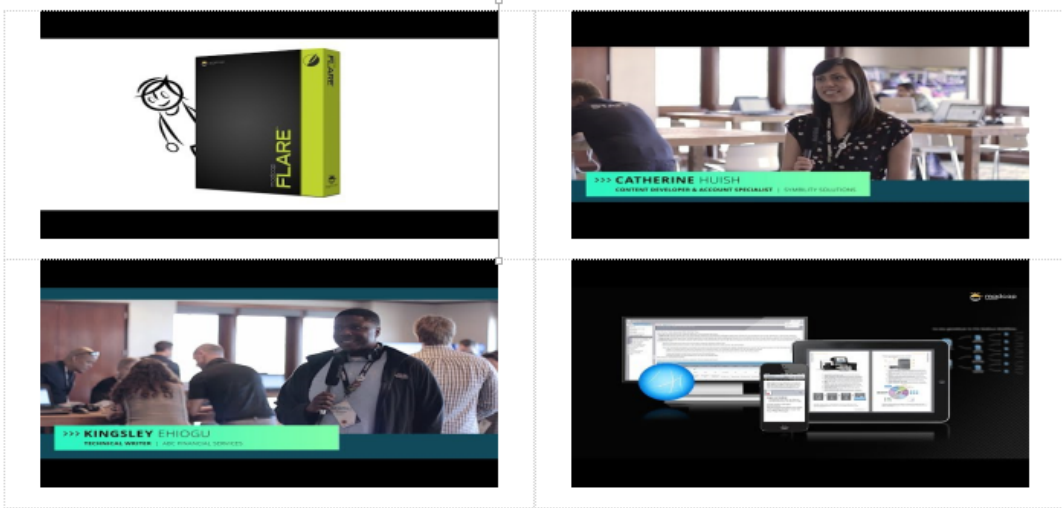
So you've got four cells for the four videos, but you need to adjust their widths. Click in the third cell and in the Responsive Layout window pane, change the **Cell Width** to 6. Do the same for the fourth cell. Now each of the four cells has a width of 50%.



- ☆ 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

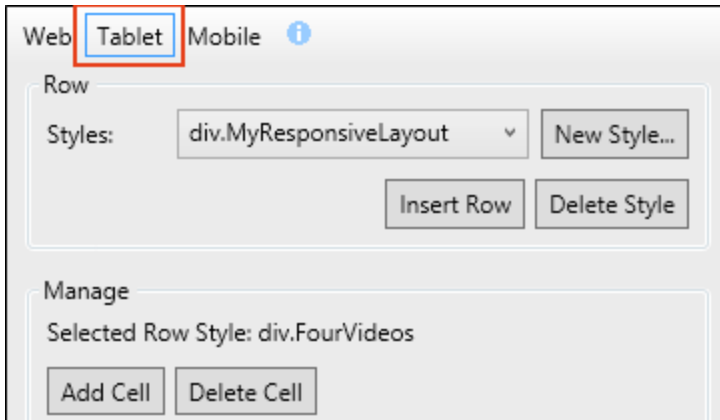
Here is an example of a responsive layout.



The image shows a 2x2 grid of video thumbnails. The top-left thumbnail features a cartoon character pointing to a box labeled 'FLARE'. The top-right thumbnail shows a woman, Catherine Huish, in a classroom setting. The bottom-left thumbnail shows a man, Kingsley Ehiogu, in a meeting. The bottom-right thumbnail shows a laptop displaying a website with a blue globe icon.

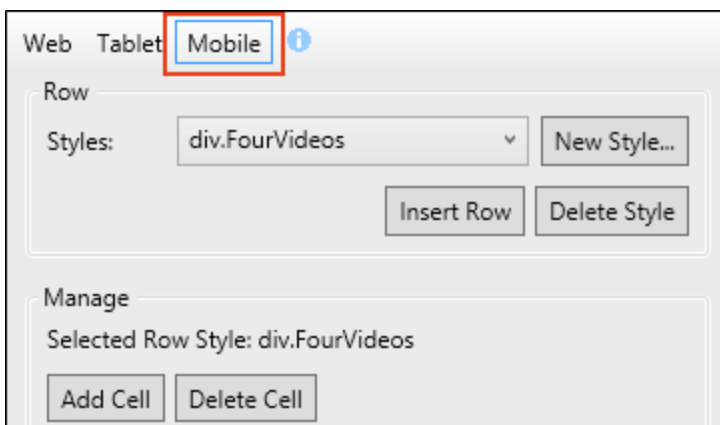
You can also apply other styles to your content as needed. For example, you might want to center your content.

- ☆ 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).



# Editing Responsive Layouts

After creating a responsive layout, you may need to return to it at times to make modifications. You also might need to adjust some of the associated styles for it.

## How to Edit a Responsive Layout

1. Open the content file.
2. In the content file, click in the responsive layout that you want to edit.
3. If you want to change the content within the layout, you can simply click in the cells and replace or edit the content as necessary.
4. Select **Home > Responsive Layout**. The Responsive Layout window pane opens.

In the Manage section, the style being used by the current responsive layout is shown. So any changes you make will be applied to that style in your stylesheet.

The screenshot shows the Responsive Layout window pane with the following elements:

- Row** section: A "Styles:" dropdown menu is set to "div.MyResponsiveLayout".
- Manage** section: A "Selected Row Style:" field is set to "div.MyNewLayout".
- Cell Width:** 1 columns
- Cell Offset:** 0 columns
- Row Gutter:** 0 pixels

Two callout boxes provide additional context:


- The top callout (pointing to the "Styles:" dropdown) states: "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."
- The bottom callout (pointing to the "Selected Row Style:" field) states: "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."

5. 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 edit a layout for large screens. After you finish those settings, you can move on to Tablet, and then finally Mobile.

6. 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.

Option	Description
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.
	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>

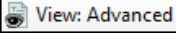
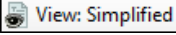

Option	Description
--------	-------------

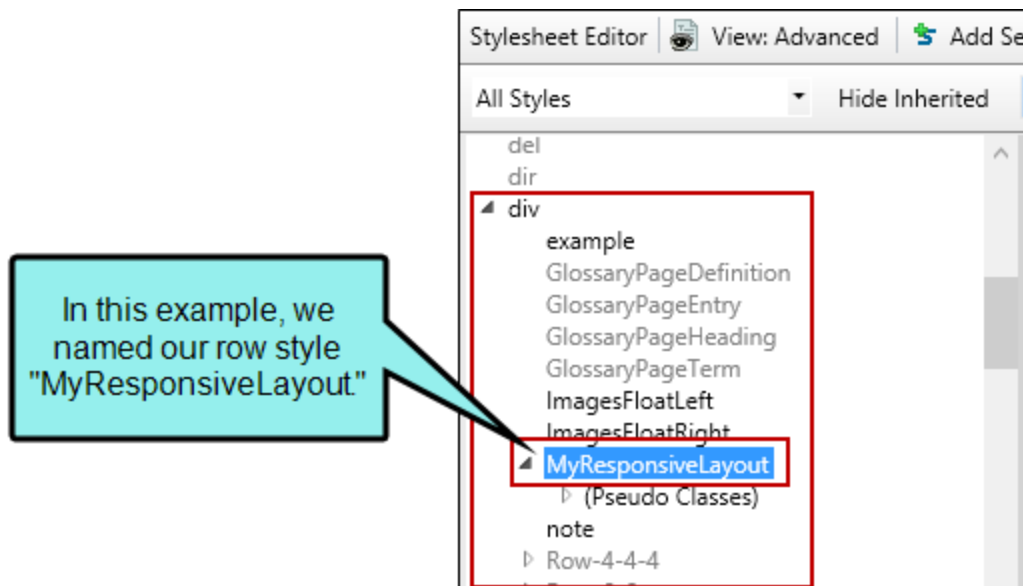
<b>Auto-save</b>	If this is selected any changes made to the selected style in this window pane will be applied in the stylesheet. If you disable this option, you must save changes in both the XML Editor and Stylesheet Editor as you work.
------------------	---

7. 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.)
8. Repeat steps 6 and 7 until you are finished designing the row for all ediums/media queries.
9. Click  to save your work.

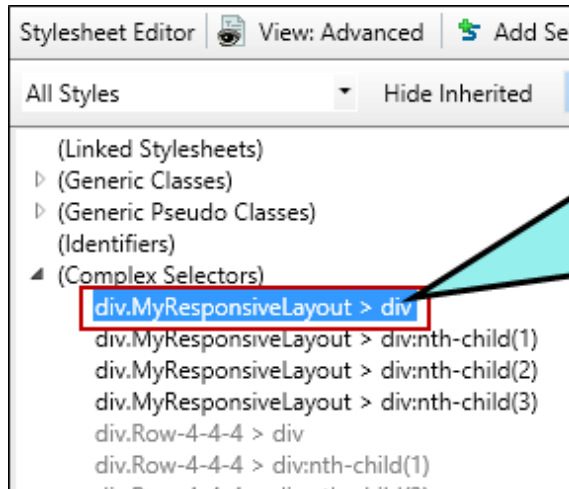
# How to Edit Responsive Layout Styles in the Stylesheet Editor

Most of the editing for a responsive layout can be done in the XML Editor and in the Responsive Layout window pane. When you make changes in the Responsive Layout window pane, they affect the appropriate styles automatically in your stylesheet. However, the number of settings you can select in the Responsive Layout window pane is limited (e.g., width, left margin, gutter). Therefore, if you want to make additional style changes to your layout, you need to open and edit your stylesheet.

1. From the Content Explorer, open the stylesheet that you want to modify. Alternatively, you can click **Edit** in the Responsive Layout window pane; this opens the stylesheet, selecting the style related to the cell currently selected in the XML Editor.
2. Open the Advanced view of the Stylesheet Editor. In the local toolbar, make sure the first button displays . If the button displays  instead, then click it.
3. In the upper-left of the editor, make sure the drop-down field is set to .
4. On the left side of the editor, select the style that you want to edit, depending on whether you want to make changes for the entire row or for a particular cell:
  - **Entire Responsive Layout Row** You can expand the **div** style and choose the class you created for the layout.

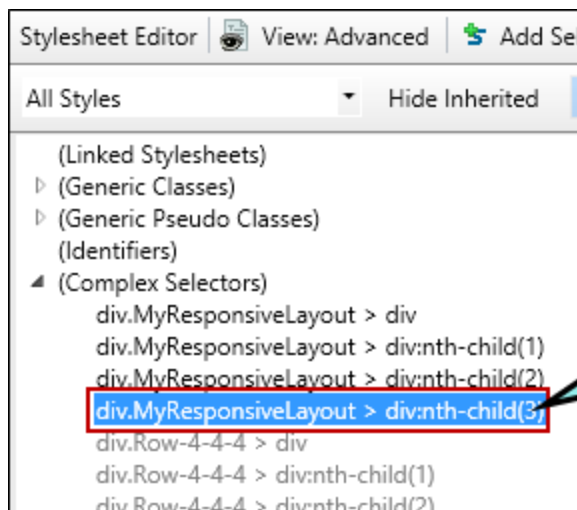


Alternatively, you can expand **(Complex Selectors)** at the top of the styles list, and choose the first selector with the name of your class.





The > symbol indicates that this is a child selector. This means that your style changes will affect all div styles that immediately follow your main div class. And because each cell is contained within another <div> tag, they will all be affected.


- **Specific Responsive Layout Cell Expand (Complex Selectors).** Then choose the selector with the class you created, with a number at the end. The number indicates which cell in the layout will be changed (e.g., 1=first cell, 2=second cell).



In this example, we chose the selector with a 3 at the end, which means it will affect the third cell in the layout.

5. (Optional) From the **Show** drop-down list in the upper-right side of the editor, select any of the options shown to determine which properties are displayed below. This is simply a way to find the appropriate property as quickly as possible.
  - **Set (Locally) Properties** Displays properties that have been set explicitly in the stylesheet.
  - **Set Properties** Displays 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.
  - **Assorted Relevant Properties** Displays the property groups that are used most often for the selected style type.
  - **All Properties** Displays all the different groups holding the properties for the selected style. This is simply a way to organize the properties into groups so that they are easy for you to find. If you want to see the values for a given property group, expand it.
6. (Optional) You can use the toggle button in the local toolbar to show properties below in a group view  or an alphabetical view .
7. Locate the property you want to change.

Making your changes in the correct medium is important. If you want to make changes that affect large screens, edit the **default** medium. If you want to make changes to medium-sized screens, edit the **tablet** media query. And if you want to make changes to small screens, edit the **mobile** media query.

8. The area to the right of the property is used for selecting and entering values. If you know how to enter the information correctly, you can click in the value field and type it directly. Otherwise, click the ellipsis button  to the right of the property. Depending on the type of property, the appropriate controls and options display, allowing you to choose or enter values (e.g., select from a drop-down list, click a button, complete fields in a dialog or popup). If you completed values in a popup, click **OK** at the bottom of the box.

As you make changes to a property's values, you can see how the changes look in the Preview section at the bottom of the editor.

9. Click  to save your work.

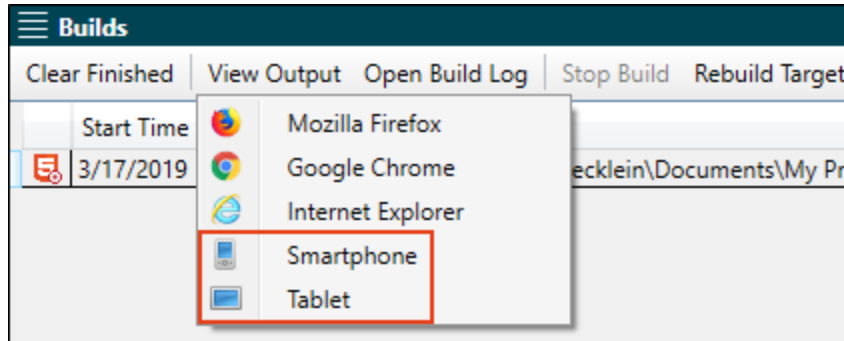
# How to Delete a Responsive Layout Style

1. Open the content file.
2. Select **Home > Responsive Layout**.
3. At the top of the Responsive Layout window pane, click the **Styles** drop-down and choose the row style you want to delete.
4. Click **Delete Style**. All styles related to that responsive layout are removed from your stylesheet.



# Viewing Responsive Output

When you finish generating responsive output, you can use view options not only for seeing output on a full browser output, but also on a tablet or smart phone. These options are available from the View buttons in the Project ribbon, Target Editor, and Builds window pane.



A preview of the output is shown based on your selection.

# Responsive Features Per Output Type

Following are responsive output features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Responsive Conditions	✓	✗	✗	✗	✗	✗	✗	✗
Responsive Layouts <sup>1</sup>	✓	✗	✗	✓ <sup>2</sup>	✗	✗	✗	✗
Responsive Skins	✓	✗	✗	✗	✗	✗	✗	✗

<sup>1</sup>This refers to the use of the Responsive Layout window pane (Home > Responsive Layout) to create content that adjusts automatically, depending on the screen size.

<sup>2</sup>Responsive layouts are supported in Clean XHTML output only if you *do not* select the target option to convert styles to inline.

# HTML5 Server-Based Output

You can create HTML5 output in its regular state, or you can select an option in the Publishing tab of the Target Editor to enable server-based functionality. This lets you accomplish the same results as WebHelp Plus output—server-side search and searching of non-XHTML content.

This chapter discusses the following:

Enabling HTML5 Server-Based Output .....	224
HTML5 Server-Based Output Scenario .....	241
Viewing HTML5 Server-Based Output .....	246

# I Enabling HTML5 Server-Based Output

If you want to take advantage of the advanced server-side features of HTML5 (i.e., automatic runtime project merging, server-side search, searching of non-XHTML files), you must enable HTML5 server-based output. This includes performing the following tasks: (1) installing Microsoft Internet Information Services (IIS) and ASP.NET, (2) setting up the HTML5 target and generating/publishing, (3) configuring IIS on the production server, (4) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (5) enabling HTML5 search.

It is not necessary to install these in order to simply build HTML5 server-based output. However, if you want to be able to view and display HTML5 server-based output, the following steps are necessary. You must perform these steps on the machine that will be hosting the published HTML5 server-based output. If you want to view the output on your local machine to test the results, then you also need to perform these steps on your local computer, except for the procedure "Configuring IIS on Production Server."

For a complete scenario that illustrates these and other HTML5 server-based output steps in a real-life situation, see "HTML5 Server-Based Output Scenario" on page 241.

# Installing IIS and ASP.NET

The following steps show you how to install IIS and ASP.NET, depending on the operating system.

## Windows 8

The following steps show you how to install IIS and ASP.NET 4.5 for Windows 8.

### INSTALLING IIS

1. Open the Control Panel. On many computers, you can do this by clicking **Start > Control Panel**.
2. Select **Programs and Features**.
3. Click **Turn Windows features on or off**.
4. Select **Internet Information Services**.
5. Expand **Internet Information Services > Web Management Tools > IIS 6 Management Compatibility**.
6. Click **IIS Metabase and IIS 6 configuration compatibility**.

### INSTALLING ASP.NET 4.5

1. Under **Internet Information Services** expand **World Wide Web Services**.
2. Expand **Application Development Features**.
3. Click the check box next to **ASP.NET 4.5** to add a check mark.
4. Click **OK**.

# Windows 7

The following steps show you how to install IIS and ASP.NET for Windows 7.

## INSTALLING IIS

1. Open the Control Panel. On many computers, you can do this by clicking **Start > Control Panel**.
2. Select **Programs and Features**.
3. Click **Turn Windows features on or off**.
4. Select **Internet Information Services**.
5. Expand **Internet Information Services > Web Management Tools > IIS 6 Management Compatibility**.
6. Click **IIS 6 Metabase and IIS 6 configuration compatibility**.

## INSTALLING ASP.NET

1. Under **Internet Information Services** expand **World Wide Web Services**.
2. Expand **Application Development Features**.
3. Click the check box next to **ASP.NET** to add a check mark.
4. Click **OK**.
5. Click **Start**.
6. In the search field enter `run`.
7. Click **Run**. The Run dialog opens.
8. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

### 32-BIT

```
C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
```

### 64-BIT

```
C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
```

9. Click **OK**. A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.

## Windows Server 2012

The following steps show you how to install IIS and ASP.NET 4.5 for Windows Server 2012.

### INSTALLING IIS

1. Open the Server Manager.
2. Click **Add roles and features**. The Add Roles and Features Wizard opens.
3. Click **Next** two times until you go to the Server Roles page.
4. Click the check box next to **Web Server (IIS)**.
5. In the dialog that opens click **Add Features**.

### INSTALLING ASP.NET 4.5

1. In the Add Roles Wizard click **Next**.
2. Expand **.NET Framework 4.5 Features** and click **ASP.NET 4.5**.
3. Click **Next**.
4. At the Web Server Role (IIS) tab, click **Next** again.
5. Expand **Management Tools > IIS 6 Management Compatibility** and click **IIS 6 Metabase Compatibility**.
6. On the Confirm Installation Selections page of the wizard, click **Install**.
7. After the installation is completed, the Installation Results page opens. Click **Close**.

# Windows Server 2008 r2

The following steps show you how to install IIS and ASP.NET for Windows Server 2008 R2.

## INSTALLING IIS

1. From the Start menu open the Server Manager dialog.
2. Click **Add roles**. The Add Roles Wizard opens.
3. Click **Next**.
4. Click the check box next to **Web Server (IIS)**.
5. In the dialog that opens click **Add Required Features**.
6. In the Add Roles Wizard click **Next**.
7. Click **Next** again.
8. On the "Select Role Services" page of the wizard, expand **Management Tools > IIS 6 Management Compatibility** and click **IIS 6 Metabase Compatibility**.
9. On the Confirm Installation Selections page of the wizard, click **Install**.
10. After the installation is completed, click **Finish**.

## INSTALLING ASP.NET

1. In the Server Manager dialog, expand **Roles**.
2. Select **Web Server IIS**.
3. In the **Role Services** section, click **Add Role Services**.
4. Click the check box next to **ASP.NET** to add a check mark.
5. Click **Next**.
6. Click **Install**.
7. Click **Close**.
8. In the search field enter `run`.
9. Click **Run**. The Run dialog opens.



10. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

### **32-BIT**

```
C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
```

### **64-BIT**

```
C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
```

11. Click **OK**.

A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.

# Windows Server 2008

The following steps show you how to install IIS and ASP.NET for Windows Server 2008.

## INSTALLING IIS

1. From the Start menu open the Server Manager dialog.
2. Click **Add roles**. The Add Roles Wizard opens.
3. Click **Next**.
4. Click the check box next to **Web Server (IIS)**.
5. In the dialog that opens click **Add Required Features**.
6. In the Add Roles Wizard click **Next**.
7. Click **Next** again.
8. On the "Select Role Services" page of the wizard, expand **Management Tools > IIS 6 Management Compatibility** and click **IIS 6 Metabase Compatibility**.
9. On the Confirm Installation Selections page of the wizard, click **Install**.
10. After the installation is completed, click **Finish**.

## INSTALLING ASP.NET

1. In the Server Manager dialog, expand **Roles**.
2. Select **Web Server IIS**.
3. In the **Role Services** section, click **Add Role Services**.
4. Click the check box next to **ASP.NET** to add a check mark.
5. Click **Next**.
6. Click **Install**.
7. Click **Close**.

# Windows Server 2003

The following steps show you how to install IIS and ASP.NET for Windows Server 2003.

## INSTALLING IIS VIA THE MANAGE YOUR SERVER DIALOG

1. From the Start menu open the Manage Your Server dialog.
2. Click **Add or remove a role**. The Configure Your Server Wizard opens.
3. Click **Next**.
4. Select **Application server (IIS, ASP.NET)** and click **Next**.
5. Click **Enable ASP.NET** and click **Next**. Your selections are summarized.
6. Click **Next**. The Windows Components Wizard opens, displaying the status of the installation.
7. After the installation is completed, click **Finish**.

## INSTALLING IIS VIA THE CONTROL PANEL

1. Open the Control Panel. On many computers, you can do this by clicking **Start > Control Panel**.
2. Double-click **Add or Remove Programs**. The Add or Remove Programs window opens.
3. Click **Add/Remove Windows Components**. The Windows Components Wizard opens.
4. In the list of components select **Application Server** and then click **Details**. The Application Server Dialog opens.
5. Click the check box next to **Internet Information Services (IIS)** to add a check mark.
6. In the Windows Component Wizard, "Application Server" should now be selected. Click **Next**.
7. Click **Finish**.

## INSTALLING ASP.NET

1. Click **Start > Run**. The Run dialog opens.
2. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

### 32-BIT

```
C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
```

### 64-BIT

```
C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
```

3. Click **OK**.

A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.



**NOTE** ASP.NET pages by default are prohibited in IIS. Therefore, you must set these pages to be allowed if you are running Windows Server 2003. To do this: (1) In Internet Information Services (IIS) Manager, select **Web Service Extensions** from the left pane; (2) in the right pane, make sure that **ASP.NET v4.0.30319** is set to **Allowed**. If it is not, select it and click the **Allow** button.

# Setting Up an HTML5 Target

The following steps show you how to set up your HTML5 target for server-based output.

## How to Set Up the HTML5 Target

1. In Flare, open the HTML5 target. The Target Editor opens.
2. Select the **Publishing** tab.
3. In the **Server-based Output** section, click the check box.
4. In the **Indexing Service Catalog Name** field, type the catalog that you are using for the output. In most cases, this will be `web`, which is the default value. However, if you or someone in your company (e.g., network administrator) creates a custom catalog, you need to enter that name in the field.

What is a catalog? Microsoft Indexing Service stores all of its index information in catalogs. A catalog comprises index information and stored properties for a particular group of file system directories. If Internet Information Services (IIS) is installed, the Indexing Service also creates a web catalog, which contains an index of IIS, the default virtual server of the World Wide Web.



**NOTE** This step is necessary only for Windows Server 2003.

5. Click to save your work.
6. Generate the target.
7. Publish the output to any location on the web server. If you want to use the default location that was created after you installed IIS, you can publish the output to `C:\inetpub\wwwroot`. If you do this, you can use the "Default Web Site" folder that you will encounter when performing the next set of steps (i.e., configuring IIS on the production server).

If you are working on your local machine for testing purposes, you do not need to publish the output. You can simply view your WebHelp Plus output. When you view HTML5 output on your local computer, you need to create a special folder called "MCPreview" within your "C:\inetpub\wwwroot" folder. Place a copy of your HTML5 output files in it. This enables you to test the advanced features of HTML5 on your local machine.

# Configuring IIS on Production Server

The following steps show you how to configure IIS, depending on the operating system. These steps are necessary only for the server where you will be publishing the final output. It is not necessary to perform these steps on your local computer for testing HTML5 server-based output.

## Windows Server 2003

1. Open the Control Panel. On many computers, you can do this by clicking **Start > Control Panel**.
2. Double-click **Administrative Tools**. The Administrative Tools dialog opens.
3. Double-click **Internet Information Services**. The Internet Information Services dialog opens.
4. Find your output folder and right-click the **Service** subfolder. Then select **Properties**. The Service Properties dialog opens.
5. Select the **Virtual Directory** tab.
6. Next to the **Application name** field, click the **Create** button.



**NOTE** If this button has already been selected, it will display as "Remove" instead. In this case, you do not need to click the button.

7. In the **Execute Permissions** drop-down, make sure **Scripts only** is selected. It should already be selected by default.
8. Click **OK**.

# Windows Server 2008 and 2008 r2, Windows 2012, Windows 7, and Windows 8

1. Open the Control Panel.
2. Select **System and Security**.
3. Select **Administrative Tools**. The various operating systems have different ways to find this option.
4. Open the Internet Information Services (IIS) Manager.
5. Expand the node with the computer name.
6. Expand the **Sites** folder.
7. Expand the website folder corresponding to the one that you published.
8. Find your output folder and right-click the **Service** subfolder. Then select **Convert to Application**.
9. Make sure the application pool is using the .NET 4 framework.
  - a. In the dialog click **Select**.
  - b. From the drop-down select an application pool that is using the .NET 4 framework.
  - c. Click **OK**.
10. Click **OK**.

# Starting Microsoft Indexing Service

The following steps show you how to start the Microsoft Indexing Service on machines that are running Windows Server 2003. This procedure is not necessary for the other operating systems.

## Windows Server 2003—How to Start Microsoft Indexing Service

1. Open the Control Panel. On many computers, you can do this by clicking **Start > Control Panel**.
2. Double-click **Administrative Tools**. The Administrative Tools dialog opens.
3. Double-click **Computer Management**. The Computer Management dialog opens.
4. Expand **Services and Applications**.
5. Right-click **Indexing Service**.
6. In the context menu, click **Start**. The Indexing Service is now started.



# Starting Microsoft Windows Search

The following steps show you how to start the Microsoft Windows Search on machines that are running Windows Server 2008. This procedure is not necessary for the other operating systems.

## Windows Server 2008—How to Start Microsoft Windows Search

1. From the Start menu open the Server Manager dialog.
2. Click **Add roles**. The Add Roles Wizard opens.
3. Click **Next**.
4. Click the check box next to **Files Services**.
5. Click **Next** twice.
6. Click the check box next to **Windows Search Service**.
7. Click **Next** twice.
8. Click **Install**.
9. Click **Close**.

# Enabling HTML5 Server-Based Search

The following steps show you how to enable HTML5 server-based search. These steps must be done for each catalog that you are using on the server (or on your local machine, if you are viewing HTML5 server-based output on your computer for testing purposes).

## How to Enable HTML5 Server-Based Search

1. Navigate to the output folder for the HTML5 target on the server or on your local machine.

If you are enabling HTML5 server-based output on your local machine, you can quickly find the output folder by doing the following.

- a. Open the Project Organizer.
  - b. Double-click the **Targets** folder.
  - c. Right-click on the target and select **Open Output Folder**.
2. In the output folder, double-click the **Service** folder.
  3. Double-click the folder labeled **Console.ConfigureSearch**.
  4. Double-click **ConfigureSearch.exe**. A window appears very briefly and then disappears.
  5. Create a folder called "AutoMergeCache" at the root of the site. Then you must set security preferences on the AutoMergeCache folder so the application can create and update files.
    - a. Right-click on the root folder and choose **Explore** to open Windows Explorer.
    - b. Right-click on **AutoMergeCache** and choose **Properties**.
    - c. On the **Security** tab click the **Edit** button.
    - d. Click **Add**.
      - i. Type `Everyone` and click **Check Names**, making sure it gets underlined.
      - ii. Click **OK**.
    - e. Make sure **Everyone** is highlighted and check the option for **Full Control**.
    - f. Click **Apply**.
    - g. Click **OK** to exit the Permission dialog.
    - h. Click **OK** to exit the Properties dialog.


# Testing HTML5 Server-Based Search


The following steps are optional for testing HTML5 server-based search (for Microsoft Windows Search).

## How to Test Search

1. In IIS right-click on the directory where your HTML5 project is published to and choose **Explore**.
2. In the upper-right corner of the window you will see a Search input box. Perform a search for a term that is commonly found in your project.
3. If you see results, your project is ready to go live.

## What's Noteworthy?

 **NOTE** If you want to test HTML5 server-based output on your local computer, the advanced search features of HTML5 are not operable.

 **NOTE** Flare's HTML5 Side and Top Navigation skins do not support runtime project merging.

# What's Next?

After enabling HTML5 server-based output, you may want to do any of the following:

- **View the Output** You can view the HTML5 server-based output on the server as well as on your local machine. This is a necessary step if you are testing the output on your local machine.
- **Merge Multiple Outputs at Runtime** If you want to merge multiple outputs so that they appear as one big Help system to end users, you can do so easily with HTML5. The outputs from the different projects will be merged at runtime when users open the Help.
- **Include Non-XHTML Files in Search** When end users perform a search in your online output, you can ensure that non-XHTML files (e.g. PDF, DOC, XLS) are included in that search.

For more information see the online Help.

# HTML5 Server-Based Output Scenario

The following scenario provides a real-life example of how someone might set up a local computer and server for producing HTML5 server-based output.

☆ **EXAMPLE** An author has a Flare project with a target named "Jackpot" and plans to publish the output to the company website (xyzincorporated.com), using the HTML5 server-based format. In addition, the author has 20 Microsoft Word documents and 15 PDF files that she wants to be included in user searches. Finally, she has four other Flare projects and wants to merge them with the output from the Jackpot target, which will serve as the "parent" target.

The author wants to publish the output to this directory on the server:

<http://www.xyzincorporated.com/help> systems. But before she publishes the final output, she wants to test the output on her local computer. The following steps in this scenario require the completion of some tasks by the author and the completion of other tasks by the web administrator, who is responsible for the production server where the final output files will be placed.

1. **Author** On her local computer, the author completes the steps for installing Microsoft Internet Information Services (IIS) and ASP.NET.
2. **Author** In Flare, the author follows the steps for setting up the Jackpot target, which is based on the HTML5 server-based output type. This includes entering "Web" as the catalog name on the Publishing tab (we're assuming the intention is to use the default value, as opposed to a custom catalog).
3. **Author** The author generates the Jackpot target in Flare. The author then creates a folder called "MCPreview" on her computer at C:\inetpub\wwwroot (e.g., C:\inetpub\wwwroot) and places a copy of the Jackpot output files in that folder.
4. **Author** The author follows the steps for starting Microsoft Indexing Service.
5. **Author** In order to enable the advanced HTML5 server-based search features on her computer, the author opens the output folder and double-clicks the file `ConfigureSearch.exe`, which is located in the output's `Service\Console.ConfigureSearch` subfolder. In the case of this author, she finds her executable file here:



```
C:\Documents and Settings\AuthorName\Documents\My
Projects\JackpotProject\Output\authorname\
Jackpot\Service\Console.ConfigureSearch\ConfigSearch.exe
```

6. **Author** The author generates her four "child" Flare targets (those that will be merged with the parent Jackpot target). In Flare, she selects **Project > Open Output Folder** and copies each output folder related to a child target. She then pastes these output folders in the following location:

```
C:\Inetpub\wwwroot\MCPreview\AutoMerge
```

The author is careful not to view the output of any of the child targets from Flare immediately after she builds them. If she were to do this, the output files for the parent Jackpot project would be overwritten in the MCPreview folder.

You can also choose specific places in the table of contents or browse sequence where child projects are merged. In this scenario, let's assume the author simply uses the default merge order.

7. **Author** Now the author wants to add the Word and PDF files to the output, so that they can be included in searches. However, an initial step needs to be completed first. Certain files (Microsoft Office, text, and HTML) do not need any attention because they can be included in the search just the way they are. PDF documents, however, require an IFilter. Therefore, the author downloads an IFilter from Adobe and installs it on her computer.
8. **Author** The author copies and pastes the Word and PDF documents to:

```
C:\Inetpub\wwwroot\MCPreview\AutoSearch
```

9. **Author** The author views the HTML5 server-based output on her computer to test the merged output and the search for Word and PDF content. However, this time she does not initiate the view from within Flare. This time, she opens a browser window and types the following path into the address bar:

```
http://localhost/MCPreview/Default.htm
```



She does this because if she had initiated the view from within Flare, the output files in the MCPreview folder would have been replaced with new files, which would have overwritten the merged child output, Word documents, and PDF files that were added manually.

See "Viewing HTML5 Server-Based Output" on page 246.

10. **Author** Let's say the author is now finished testing her output and is ready to publish the final output to the server. Therefore, in Flare she creates a publishing destination for the parent Jackpot target that points to `http://www.xyzincorporated.com/help systems/Jackpot`. She generates the output for the parent Jackpot target again and publishes the Jackpot HTML5 server-based output to that location.
11. **Author/Web Administrator** The author generates the four child Flare target outputs again. In Flare, she selects **Project > Open Output Folder** and copies each output folder related to a child target. The author can provide these folders to the web administrator or (if the author has access) directly paste them in the following location:

```
http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge
```

For example, if the author has child targets named "Child1," "Child2," "Child3," and "Child4," those output folders would be added to the website as follows:

```
http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child1
http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child2
http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child3
http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child4
```

Alternatively, the author (or web administrator) could manually create subfolders on the website that are named after each target folder; these subfolders would be contained within the AutoMerge folder. The author can then create publishing destinations that point to the appropriate subfolders and automatically publish each child target accordingly.

12. **Web Administrator** On the server where the output files will be published, the company's web administrator completes the steps for installing Microsoft Internet Information Services (IIS) and ASP.NET.



13. **Web Administrator** On the server where the output files will be published, the company's web administrator completes the steps for configuring IIS on the production server.
14. **Web Administrator** On the server where the output files will be published, the company's web administrator completes the steps for starting Microsoft Indexing Service.
15. **Web Administrator** The web administrator opens the output folder on the server to enable HTML5 search. He double-clicks the file `ConfigureSearch.exe`, which is located here:

```
http://www.xyzincorporated.com/help  
systems/Jackpot/Service/Console.ConfigureSearch
```

16. **Web Administrator** In order to include the PDF files in the search, the web administrator installs the appropriate IFilter on the server.
17. **Author/Web Administrator** The author (if she has access) or web administrator copies and pastes the 20 Word documents and 15 PDF files to:

```
http://www.xyzincorporated.com/help systems/Jackpot/AutoSearch
```

18. **Author** To view the published output, the author opens a browser window and types the following path into the address bar:

```
http://www.xyzincorporated.com/help systems/Jackpot/Default.htm
```

If the author had specified a custom file name for the Jackpot target, such as "JackpotOnline," the address instead would be:

```
http://www.xyzincorporated.com/help systems/Jackpot/JackpotOnline.htm
```





**NOTE** If you are testing HTML5 server-based output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

# I Viewing HTML5 Server-Based Output

You can view HTML5 server-based output on your local machine for testing purposes, and you can view the output on the server where the files are published (where users access them). Before viewing the output, however, you must first follow all of the steps for enabling HTML5 server-based output.

## Basics of Viewing HTML5 Server-Based Output

When you view HTML5 output on your local computer, you need to create a special folder called "MCPreview" within your "C:\inetpub\wwwroot" folder. Place a copy of your HTML5 output files in it. This enables you to test the advanced features of HTML5 on your local machine.

When you test the output on your local computer, the output initially needs to be viewed by using the Flare interface (in order to generate the MCPreview folder). After that, you can view the output by using the Flare interface or by opening the file copies from the MCPreview folder. It is recommended that you use the MCPreview folder for subsequent viewings of WebHelp Plus output, rather than the Flare interface.

For a complete scenario that illustrates these and other HTML5 server-based output steps in a real-life situation, see "HTML5 Server-Based Output Scenario" on page 241.

If you want to test HTML5 server-based output on your local computer, the advanced search features of HTML5 are not operable.

## Testing Output Locally—Windows 7

Windows 7 includes a feature called User Access Control (UAC). By default, this feature is enabled in Windows 7, which means that all users, including administrators, run programs in an under-privileged state. It is not possible to view for testing on your local computer if Windows 7 is running in this under-privileged state. If this occurs, you have two options: (1) disable UAC on the local machine, or (2) run Flare with elevated permission without globally disabling UAC.

## Option 1—Disable UAC on the Local Machine

1. Open the Control Panel. On many computers, you can do this by clicking **Start > Control Panel**.
2. Select **User Accounts** (select it twice if you are in Category View).
3. Click **Change User Account Control Settings** and then move the slider to **Never notify** and click **OK**.
4. Restart the computer.

UAC is now disabled. However, in order to preview HTML5 server-based output, you must be part of the Administrator's group, because Standard users cannot preview the output. Check with your system administrator to determine if you are in this group.

## Option 2—Run Flare With Elevated Permission Without Globally Disabling UAC

1. Make sure you are part of the Administrator's group. Check with your system administrator to determine if you are in this group.
2. In Windows, navigate to the Flare executable file (C:\Program Files\MadCap Software\MadCap Flare 20\Flare.app\Flare.exe).
3. Set yourself as an administrator for Flare (either temporarily or permanently).

### TO RUN ONLY A SINGLE INSTANCE OF FLARE AS AN ADMINISTRATOR

- a. Right-click on the executable file.
- b. Select **Run as administrator**.

### TO ALWAYS RUN FLARE AS AN ADMINISTRATOR

- a. Right-click on the executable file.
- b. Select **Properties**.
- c. Select the **Compatibility** tab.
- d. In the **Privilege Level** area, click in the check box labeled **Run this program as an administrator**.

# Procedures for Viewing Output

Use the following procedures to view HTML5 server-based output on the production server (where you publish your final output files) and on your local computer (for testing purposes).

## How to View Output on the Production Server

1. Open a browser window on any computer.
2. In the address field, type the URL to the main entry file of your output files (where you published them).

Unless you change the name of the output file (on the General tab of the Target Editor), the name of the main output file is Default.htm.

☆ EXAMPLE

```
http://www.mycompanysite/help/myHTML5target/Default.htm
```

3. Press ENTER.

## How to View Output on the Local Machine for Testing (Initial Viewing)

After generating the output, do one of the following using the Flare interface.

- In the message that displays immediately at the end of the compilation process, click **Yes**.  
OR
- Open the Target Editor and in the local toolbar click **View**.

# How to View Output on the Local Computer (Subsequent Viewings)

You can follow the same steps above for the initial viewing (using the Flare interface), or you can do the following to quickly access the files from the MCPreview folder.

1. Open a browser window.
2. In the address field, type the following:


```
http://localhost/MCPreview/[Name of Main Entry File].htm
```


For example: `http://localhost/MCPreview/Default.htm`

 **NOTE** The above address pertains to Windows Server 2003.

3. Press **ENTER**.

## What's Noteworthy?

 **NOTE** If you are testing HTML5 server-based output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

 **NOTE** Flare's HTML5 Side and Top Navigation skins do not support runtime project merging.

## APPENDIX A

---

# PDFs

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

## I Tutorials

*Autonumbers Tutorial*

*Back-to-Top Button Tutorial*

*Context-Sensitive Help Tutorial*

*Custom Toolbar Tutorial*

*eLearning Tutorial—Basic*

*eLearning Tutorial—Advanced*

*Getting Started Tutorial*

*Image Tooltips Tutorial*

*Lists Tutorial*

*Meta Tags Tutorial*

*Micro Content Tutorial—Basic*

*Micro Content Tutorial—Advanced*

*Responsive Output Tutorial*

*Single-Sourcing Tutorial*

*Snippet Conditions Tutorial*

*Styles Tutorials*

*Tables Tutorial*

*Word Import Tutorial*

# I Cheat Sheets

*Context-Sensitive Help Cheat Sheet*

*Folders and Files Cheat Sheet*

*Learning & Development Cheat Sheet*

*Lists Cheat Sheet*

*Micro Content Cheat Sheet*

*Print-Based Output Cheat Sheet*

*Search Cheat Sheet*

*Shortcuts Cheat Sheet*

*Structure Bars Cheat Sheet*

*Styles Cheat Sheet*

# I User Guides

*Accessibility Guide*

*Analysis and Reports Guide*

*Architecture Guide*

*Autonumbers Guide*

*Branding Guide*

*Condition Tags Guide*

*Context-Sensitive Help Guide*

*Eclipse Help Guide*

*eLearning Guide*

*Getting Started Guide*

*Global Project Linking Guide*

*HTML5 Guide*

*Images Guide*

*Import Guide*

*Indexing Guide*

*Key Features Guide*

*Lists Guide*

*MadCap Central Integration  
Guide*

*Meta Tags Guide*

*Micro Content Guide*

*Navigation Links Guide*

*Plug-In API Guide*

*Print-Based Output Guide*

*Project Creation Guide*

*QR Codes Guide*

*Reviews & Contributions With  
Contributor Guide*

*Scripting Guide*

*Search Guide*

*SharePoint Guide*

*Skins Guide*

*Snippets Guide*

*Source Control Guide: Git*

*Source Control Guide:  
Perforce Helix Core*

*Source Control Guide:  
Subversion*

*Source Control Guide: Team  
Foundation Server*

*Styles Guide*

*Tables Guide*

*Tables of Contents Guide*

*Targets Guide*

*Template Pages Guide*

*Templates Guide*

*Topics Guide*

*Touring the Workspace Guide*

*Transition From FrameMaker  
Guide*

*Translation and Localization  
Guide*

*Variables Guide*


*Videos Guide*

*What's New Guide*



# Output Type Comparison Tables

Following are tables that show the various output types available, with the distinguishing features of each.

 **NOTE** If an item does not have a check mark, that does not necessarily mean the feature is unavailable in the output. Rather, it means that the feature cannot be added in Flare. For example, search cannot be added in Flare for EPUB output, but some ebook readers may have search built in.

This chapter discusses the following:

Accessibility Features Per Output Type .....	255
Context-Sensitive Help Features Per Output Type .....	257
eLearning Content Features Per Output Type .....	258
Generated Content Features Per Output Type .....	260
Glossary Features Per Output Type .....	264
Image Features Per Output Type .....	265

Index Features Per Output Type .....	266
Link Features Per Output Type .....	268
List Features Per Output Type .....	270
MadCap Central Features Per Output Type .....	271
Merging Features Per Output Type .....	272
Micro Content Features Per Output Type .....	273
Miscellaneous Features Per Output Type .....	275
Multimedia Features Per Output Type .....	278
Responsive Features Per Output Type .....	283
Search Features Per Output Type .....	284
Skin Features Per Output Type .....	291
TOC and Mini-TOC Features Per Output Type .....	297
Translation and Localization Features Per Output Type .....	299
Variable Features Per Output Type .....	301

# Accessibility Features Per Output Type

Following are accessibility features (e.g., Section 508, WCAG) supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Accessibility Supported	✓	✓	✗	✓	✓	✓	✓	✗
Alt Text and Title Attributes	✓	✓	✗	✓	✓	✗	✓	✗
ARIA Tags	✓	✗	✗	✗	✗	✗	✗	✗
Compiler Warnings	✓	✓	✗	✓	✓	✗	✓	✗
Empty Alt Text for Images	✓	✓	✗	✓	✓	✗	✓	✗
Keyboard Navigation	✓	✗	✗	✗	✗	✗	✗	✗
Scrolling for Toolbars and Navigation Frames (Enable)	✗	✗	✗	✗	✗	✗	✗	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Search Results	✓	✗	✗	✗	✗	✗	✗	✗
Skip to Main Content	✓	✗	✗	✗	✗	✗	✗	✗
User Interface Text	✓	✗	✗	✗	✗	✗	✗	✗

# Context-Sensitive Help Features Per Output Type

Following are context-sensitive Help (CSH) features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
CSH Supported	✓	✗	✗	✗	✓	✗	✓	✗
Micro Content CSH	✓	✗	✗	✗	✗	✗	✗	✗

# eLearning Content Features Per Output Type

Following are eLearning features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
eLearning Content Reuse	✓	✓	✗	✗	✗	✗	✗	✗
eLearning Templates	✓	✓	✗	✗	✗	✗	✗	✗
Interactive Question Sections	✓	✗ <sup>1</sup>	✗	✗	✗	✗	✗	✗ <sup>1</sup>
LMS Packages	✓	✗	✗	✗	✗	✗	✗	✗
Multiple Types of Courses	✓	✓	✗	✗	✗	✗	✗	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Skin Customization	✓	✗ <sup>2</sup>	✗	✗	✗	✗	✗	✗ <sup>2</sup>

<sup>1</sup>PDF and PPTX output is not interactive. For PDF and PPTX there is an option to show or hide answers to course questions.

<sup>2</sup>PDF and PPTX can be customized, but not using skins.

# Generated Content Features Per Output Type

Following are generated content features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Autonumbers <sup>1</sup>	✓	✓	✓	✓	✓	✓	✓	✓
Breadcrumbs	✓	✗	✗	✗	✗	✗	✓	✗
Browse Sequences	✓ <sup>2</sup>	✗	✗	✗	✗	✗	✓	✗
Central Account Links	✓	✗	✗	✗	✗	✗	✗	✗
Concept Links	✓	✗	✗	✗	✓	✓	✓	✗
Glossaries From Proxies	✗	✓	✓	✗	✗	✓	✗	✓
Indexes From Proxies	✗	✗	✗	✗	✗	✓	✗	✓



	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Keyword Links	✓	✓	✓	✗	✓	✓	✓	✗
List of Concepts	✓	✓	✓	✗	✓	✓	✓	✗
List of Elements	✓	✓	✓	✓	✓	✓	✓	✗
List of Endnotes	✗	✓	✓	✗	✗	✓	✗	✓
Lists (Numbered and Bulleted)	✓	✓	✓	✓	✓	✓	✓	✓
Menus From Proxies	✓	✗	✗	✗	✗	✗	✗	✗
Mini-TOCs From Proxies	✓	✓	✓	✓	✓	✗	✓	✓
Page Numbers	✗	✓	✓	✗	✗	✗	✗	✓

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Related Topics Links	✓	✗	✗	✗	✓	✓	✓	✗
Relationship Links	✓	✓	✓	✓	✓	✓	✓	✗
Scripts	✓	✗	✗	✓ <sup>3</sup>	✓	✗	✓	✗
Search Bars From Proxies	✓ <sup>4</sup>	✗	✗	✗	✗	✗	✗	✗
Search Results From Proxies	✓ <sup>4</sup>	✗	✗	✗	✗	✗	✗	✗
Shortcut Controls	✗	✗	✗	✗	✗	✗	✓	✗
TOCs From Proxies	✓	✓	✓	✗	✓	✓	✓	✓

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Topic Toolbars From Proxies	✓	✗	✗	✗	✗	✗	✓	✗

<sup>1</sup>It is possible to use autonumbering in online outputs, but only in a limited way. For example, if you have notes throughout your project, you can create an autonumber format on a paragraph style class that simply has the text "Note:" followed by a space. Then whenever you want to apply that style class to note content, it will automatically start with "Note:" so that you don't have to type it each time. However, in the traditional sense where autonumbering is used to increment numbers for elements such as chapters, figures, and tables, it is really only intended for print-based outputs. If you are creating both online and print output and want the autonumbering to be used only for content in the print output, you can use a print medium to specify the autonumbering settings for the print output.

<sup>2</sup>In most cases, browse sequences are generated via a skin. This is supported in HTML5 Tripane output, but not in HTML5 Side Navigation, Top Navigation, or skinless output. However, by using a menu proxy, browse sequences are supported in all HTML5 output formats.

<sup>3</sup>For Clean XHTML output, MadCap-injected JavaScript is removed. In other words, features such as drop-downs, Help controls, text popups, and footnotes rely on JavaScript to work. Therefore, with Clean XHTML output, these elements are converted to text. However, custom scripts that you might have added to topics will not automatically be removed when you generate Clean XHTML; if you want to exclude these custom scripts, you must do so manually (i.e., by deleting them or using conditions).

<sup>4</sup>In HTML5, the proxies and related skin components for search bars and search results are supported only in Side Navigation, Top Navigation, and skinless outputs. They are not supported in Tripane output.

# Glossary Features Per Output Type

Following are glossary features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Glossaries Supported(via Proxy) <sup>1</sup>	✓	✓	✓	✓	✓	✓	✓	✓
Glossaries Supported(via kin)	✓	✗	✗	✗	✗	✗	✓	✗
Auto-Generate	✗	✓	✓	✗	✗	✓	✗	✓
Glossary Term Links	✓	✓	✓	✗	✓	✓	✓	✗
Sorting	✓	✓	✓	✓	✓	✓	✓	✗

<sup>1</sup>For most online outputs, creating a glossary via a skin is the preferred method. However, it is possible to use a proxy to include glossaries in these outputs as well.

# Image Features Per Output Type

Following are image features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Absolute Positioning	✓	✓	✗	✓	✓	✗	✓	✗
Image Hyperlinks	✓	✗	✗	✓	✓	✓	✓	✗
Image Maps	✓	✓	✗	✓	✓	✗	✓	✗
Thumbnail Images	✓	✗	✗	✓ <sup>1</sup>	✓	✗	✓	✗
Web-Safe (Convert From Non-Web-Safe)	✓	✗	✗	✓	✓	✗	✓	✗

<sup>1</sup>For Clean XHTML output, thumbnail images that are set as popups are converted to linked thumbnail images (i.e., the enlarged image opens in a new window instead of in a popup).

# Index Features Per Output Type

Following are index features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Indexes Supported (via Proxy) <sup>1</sup>	✗	✓	✓	✗	✓	✓	✗	✓
Indexes Supported (via Skin)	✓	✗	✗	✗	✗	✗	✓	✗
Auto-Generate	✗	✓	✓	✗	✗	✓	✗	✓
Binary Indexes	✗	✗	✗	✗	✗	✗	✓	✗
Bookmarks in Index Entries	✗	✗	✗	✗	✗	✗	✓	✗
Index Links	✓	✓	✓	✗	✓	✓	✓	✓

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Search (Can Exclude Index Entries From Search)	✓	✗	✗	✗	✗	✗	✗	✗
Sorting (Index Links)	✓	✓	✗	✗	✗	✓	✗	✓

<sup>1</sup>For online outputs, creating an index is typically done via a skin. By default, the MadCap style that controls index proxies is set to support only print-based outputs. However, you can change this. To do so, open your stylesheet, select the **MadCap|indexProxy** style, and set the **mc-output-support** property to **all** (all-print is the default setting). By doing this, you can use the index proxy method to generate an index in some online outputs as well.

# Link Features Per Output Type

Following are link features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Cross-References Supported	✓	✓	✓	✓	✓	✓	✓	✓
Cross-References (Context-Sensitive)	✗	✓	✓	✗	✗	✓	✗	✓
Drop-Down Text	✓	✗	✗	✗	✓	✗	✓	✓
Expanding Text	✓	✗	✗	✗	✓	✗	✓	✓
Text Hyperlinks	✓	✓	✓	✓	✓	✓	✓	✓
Text Popups	✓	✗	✗	✗	✓	✗	✓	✗
Togglers	✓	✗	✗	✗	✓	✗	✓	✓



	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Topic Popups	✓	✗	✗	✗	✓	✗	✓	✗
Topic Toolbars	✓	✗	✗	✗	✗	✗	✓	✗
Web Toolbars	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✓	✗

<sup>1</sup>This is available in the Tripane output only. It is not available in Side or Top Navigation outputs.

# List Features Per Output Type

Following are list features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Breaks (Page and Column)	✗	✓	✓	✗	✗	✗	✗	✓
Custom List Formats	✗	✓	✗	✗	✗	✗	✗	✗
Definition Lists	✓	✓	✓	✓	✓	✓	✓	✓
Numbered and Bulleted Lists	✓	✓	✓	✓	✓	✓	✓	✓
Styles	✓	✓	✓	✓	✓	✓	✓	✓

# MadCap Central Features Per Output Type

Following are MadCap Central features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Analytics	✓	✗	✗	✗	✗	✗	✗	✗
Build Output From Central	✓	✓	✓	✓	✓	✓	✓	✗
Private Ouptut	✓	✗	✗	✗	✗	✗	✗	✗
Publish Directly to Central	✓	✓	✓	✓	✓	✓	✓	✗
View Output From Central	✓	✓	✓	✓	✓	✓	✗	✗

# Merging Features Per Output Type

Following are merging output features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Merge Output Supported	✓ <sup>1,2</sup>	✗	✗	✗	✓	✗	✓	✗
Merge Output at Runtime	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✗	✗
Multilingual Output (Link to External Projects in Target)	✓	✓	✓	✓	✓	✓	✓	✓
Stitch Output <sup>3</sup>	✓	✓	✗	✗	✗	✗	✓	✗

<sup>1</sup> Flare's HTML5 Side and Top Navigation skins do not support project merging.

<sup>2</sup> HTML5 outputs do not support pre-merging.

<sup>3</sup> PDF stitching allows you to add PDFs to Flare output, but you cannot link to content in that PDF. In online outputs, the PDF is simply added to the TOC and can be opened from there; it is not technically "merged" into the output.

# Micro Content Features Per Output Type

Following are micro content features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Micro Content Supported	✓ <sup>1</sup>	✓ <sup>2</sup>	✓ <sup>2</sup>	✓ <sup>2</sup>	✓ <sup>2</sup>	✓ <sup>2</sup>	✓ <sup>2</sup>	✓ <sup>2</sup>
FAQ Proxy	✓	✓	✓	✓	✓	✓	✓	✓
Field-Level Context-Sensitive Help	✓	✓	✓	✓	✓	✓	✓	✗
Featured Snippets	✓	✗	✗	✗	✗	✗	✗	✗
Knowledge Panel	✓	✗	✗	✗	✗	✗	✗	✗
Knowledge Proxy	✓	✗	✗	✗	✗	✗	✗	✗
Promotion Proxy	✓	✗	✗	✗	✗	✗	✗	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Single-Sourcing Micro Content	✓	✓	✓	✓	✓	✓	✓	✓

<sup>1</sup>Micro content in search is supported for HTML5 only if you are using Elasticsearch or MadCap Search, but not Google Search.

<sup>2</sup>Only HTML5 supports all micro content features (e.g., featured snippets, Knowledge Panel, Knowledge proxy, Promotion proxy, FAQ proxy). The other output formats support only the FAQ proxy.

# Miscellaneous Features Per Output Type

Following are miscellaneous features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Code Snippets	✓	✓	✓	✓	✓	✓	✓	✓
Equations	✓	✓	✓	✓	✓	✓	✓	✓
Footnote <sup>1</sup>	✓	✓	✓	✗	✓	✓	✓	✓
Horizontal Rules	✓	✓	✗	✓	✓	✓	✓	✓
IFrame	✓	✗	✗	✓	✗	✗	✗	✗
Meta Tags	✓	✓	✓	✓	✓	✓	✓	✓
Object Positioning	✓	✓	✗	✓	✓	✓	✓	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Preview Topics in Output Type	✓	✓	✓	✓	✓	✓	✓	✓
QR Codes	✓	✓	✗	✓	✓	✓	✓	✓
Redacted Text	✗	✓	✗	✗	✗	✗	✗	✗
Salesforce® Publishing	✗	✗	✗	✓	✗	✗	✗	✗
ServiceNow® Publishing	✗	✗	✗	✓	✗	✗	✗	✗
Slideshow	✓	✗	✗	✗	✗	✓	✓	✗
Snippets	✓	✓	✓	✓	✓	✓	✓	✓
Tables	✓	✓	✓	✓	✓	✓	✓	✓



	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Text Boxes	✓	✓	✓	✓	✓	✗	✓	✓
Zendesk Publishing	✓ <sup>3</sup>	✗	✗	✓	✗	✗	✗	✗

<sup>1</sup>For online outputs, footnotes are converted to popups.

<sup>2</sup>This is supported only if you are using Elasticsearch or MadCap Search, but not Google Search.

<sup>3</sup>If you publish output to Zendesk, you should use HTML5 skinless output, instead of HTML5 Side Navigation, Top Navigation, or Tripane outputs.

# Multimedia Features Per Output Type

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB <sup>1</sup>	HTML Help	PPTX
3D Models (U3D)	✓	✓ <sup>2</sup>	✗	✓	✓	✗	✓ <sup>3</sup>	✗
Audio/Video								
ASF	✓	✗	✗	✓	✓	✗	✓	✓
ASX	✓	✗	✗	✓	✓	✗	✓	✗
AU	✓	✗	✗	✓	✓	✗	✓	✗
AVI <sup>4</sup>	✓	✓	✗	✓	✓	✗	✓	✓
M4V <sup>4</sup>	✓	✓	✗	✓	✓	✗	✗	✗
MID	✓	✓	✗	✓	✓	✗	✓	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB <sup>1</sup>	HTML Help	PPTX
MIDI	✓	✓	✗	✓	✓	✗	✓	✗
MP3	✓	✓	✗	✓	✓	✓	✓	✗
MP4 <sup>4</sup>	✓	✓	✗	✓	✓	✓	✗	✓
MPA	✓	✓	✗	✓	✓	✗	✓	✓
MPE	✓	✓	✗	✓	✓	✗	✓	✓
MPEG	✓	✓	✗	✓	✓	✗	✓	✓
MPG	✓	✓	✗	✓	✓	✗	✓	✓
OGG <sup>4</sup>	✓	✗	✗	✓	✗	✓	✗	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB <sup>1</sup>	HTML Help	PPTX
OGV <sup>4</sup>	✓	✗	✗	✓	✗	✓	✗	✗
OPUS	✓	✗	✗	✓	✗	✗	✗	✗
SWF	✓	✓	✗	✓	✓	✗	✓	✗
WAV	✓	✓	✗	✓	✓	✗	✓	✗
WEBM <sup>4</sup>	✓	✗	✗	✓	✗	✗	✗	✗
WMA	✓	✗	✗	✓	✓	✗	✓	✗
WMV	✓	✗	✗	✓	✓	✗	✓	✓

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB <sup>1</sup>	HTML Help	PPTX
Web Videos								
Vimeo Embedded Videos	✓	✗	✗	✓	✓	✗	✓ <sup>4</sup>	✗
YouTube Embedded Videos	✓	✗	✗	✓	✓	✗	✓ <sup>4</sup>	✗
MadCap Mimic Movie Links								
Flash (SWF) Format	✓	✗	✗	✓	✓	✗	✓	✗
HTML5 Format (Depending on Browser) <sup>5</sup>	✓	✗	✗	✓	✓	✗	✓	✗
Mimic Movie Format	✗	✗	✗	✗	✓	✗	✓	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB <sup>1</sup>	HTML Help	PPTX
Silverlight Format	✓	✗	✗	✓	✓	✗	✓	✗

<sup>1</sup>EPUB support for each element or file type depends on if a particular reader supports it.

<sup>2</sup>PDF support for 3D Models requires Adobe Reader 7.0+.

<sup>3</sup>For this output, the 3D Model opens in a separate window.

<sup>4</sup>If you generate Microsoft HTML Help, YouTube videos are rendered smaller than in other outputs and Vimeo Advanced settings are not supported.

<sup>5</sup>To view HTML5 movie output, end users must have a newer browser that supports these types of videos.

# Responsive Features Per Output Type

Following are responsive output features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Responsive Conditions	✓	✗	✗	✗	✗	✗	✗	✗
Responsive Layouts <sup>1</sup>	✓	✗	✗	✓ <sup>2</sup>	✗	✗	✗	✗
Responsive Skins	✓	✗	✗	✗	✗	✗	✗	✗

<sup>1</sup>This refers to the use of the Responsive Layout window pane (Home > Responsive Layout) to create content that adjusts automatically, depending on the screen size.

<sup>2</sup>Responsive layouts are supported in Clean XHTML output only if you *do not* select the target option to convert styles to inline.

# Search Features Per Output Type

Following are search features supported in each online output type.

## General Search Support Per Output Type

The table below summarizes search support in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Search Supported via Flare Integration	✓	✗	✗	✗	✗	✗	✓	✗

## End User Search Features Per Output Type

The table below summarizes the key search features and capabilities for each output type. With HTML5 and WebHelp outputs, you can distribute either client- or server-based outputs. This is why the table below lists them twice—under client- and server-based output.

End user search features are supported by the Eclipse Help Viewer and search operations are provided by a plug-in that you develop using the Eclipse platform (for more information, see [help.eclipse.org](http://help.eclipse.org)). The Flare search engine does not handle Eclipse Help search operations.



	HTML5 (Client)	HTML5 (Server)	HTML Help
Glossary Search (Find Matching Terms)	✓	✓	✓
Index Search (Find Matching Terms)	✓	✓	✓
Predictive Search	✓	✓	✗
Search Favorites (Save Queries to Favorites List)	✗	✗	✗
Search Query (Asterisk (*) Wildcard Character)	✗	✗	✓
Search Query (Boolean Operators)	✓	✗	✓ <sup>1</sup>
Search Query (Full-text Search—not case sensitive)	✓	✓	✓
Search Query (Enclose Terms in Quotes)	✓	✓	✓
Search Results (Narrow Search Scope Using Search Filter Sets)	✓ <sup>2</sup>	✓	✗

	HTML5 (Client)	HTML5 (Server)	HTML Help
Search Results (Results Listed in Ranked Order and Show Rank Number)	✓ <sup>3</sup>	✓ <sup>3</sup>	✓ <sup>3</sup>
Search Results (Search Hits Highlighted in Topics)	✓	✓	✓
Search Results (Search Hits Bolded)	✓	✗	✗

<sup>1</sup>Select the operator from the built-in list.

<sup>2</sup>Search filters are not supported in Elasticsearch.

<sup>3</sup>Results are listed in order. Rank number is not visible.

## Content Developer Search Features Per Output Type

The following table summarizes the key search features and capabilities that are available to you, the content developer, for each output type. With HTML5 and WebHelp outputs, you can distribute either client- or server-based outputs. This is why the table below lists them twice—under client- and server-based output.

	HTML5 (Client)	HTML5 (Server)	HTML Help
Enable	✓ <sup>1</sup>	✓ <sup>1</sup>	✓ <sup>2</sup>
Content Optimization Setting the Search Abstract Character Limit	✓ <sup>3</sup>	✗	✗
Content Optimization Setting Search Result Importance	✓ <sup>3</sup>	✗	✗
Content Optimization Including Glossary Terms in Search Results	✓ <sup>3</sup>	✓	✗
Content Optimization Setting the Number of Search Results Per Page	✓ <sup>3</sup>	✓ <sup>3</sup>	✗
Elasticsearch	✓	✓	✗
Google Search	✓ <sup>4</sup>	✗	✗

	HTML5 (Client)	HTML5 (Server)	HTML Help
Micro Content Search Integration	✓ <sup>3</sup>	✓ <sup>3</sup>	✗
Setting Micro Content Containers			
Search Highlighting	✓	✓	✓
Customizing the Highlight Color for Search Hits			
Search Optimization	✓	✓	✗
Enabling Partial-Word Search			
Search Optimization	✓	✓	✓ <sup>5</sup>
Including Stop Words in Search			
Search Optimization	✓	✗	✓ <sup>6</sup>
Excluding Index Entries From Search			
Search Optimization	✓	✓	✓ <sup>7</sup>
Excluding Non-Words From Search			

	HTML5 (Client)	HTML5 (Server)	HTML Help
Search Optimization	✓	✓	✗
Setting the N-Gram Size for Search			
Search Performance	✓	✓	✗
Chunking Large Search Database Files			
Search Performance	✓	✓	✗
Pre-Merging Search Database Files			
Search Results	✓	✓	✗
Customizing Search Filter Ordering			
Search Toolbar	✗	✗	✗
Web Toolbars in WebHelp Outputs			

	HTML5 (Client)	HTML5 (Server)	HTML Help
Styles	✓ <sup>4</sup>	✓ <sup>8</sup>	✗
Styles for Generated Pages			

<sup>1</sup>HTML5 skin.

<sup>2</sup>Standard skin.

<sup>3</sup>This is supported if you are using Elasticsearch or MadCap Search, but not Google Search.

<sup>4</sup>This is supported only in HTML5 Side Navigation, Top Navigation, and skinless outputs. It is not supported in HTML5 Tripane output.

<sup>5</sup>Stop words are always enabled.

<sup>6</sup>Index entries are always excluded.

<sup>7</sup>Non-words are always excluded.

<sup>8</sup>SQL Server Compact is required.

# | Skin Features Per Output Type

Following are skin features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Skin Type	HTML5	✗	✗	✗	✗	✗	Standard	✗
About Box	✗	✗	✗	✗	✗	✗	✓	✗
Accordion Titles (Exclude)	✗	✗	✗	✗	✗	✗	✗	✗
Browser Settings	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✗	✗
Caption for Output Window	✓ <sup>1</sup>	✗	✗	✗	✗ <sup>2</sup>	✗	✓	✗
Central Account Links	✓	✗	✗	✗	✗	✗	✗	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Elements Such as Tabs, Accordions (Specify Default Element)	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✓	✗
Elements Such as Tabs, Accordions (Specify Which to Include)	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✓	✗
FAQ Skin Component	✓	✓	✓	✓	✓	✓	✓	✓
Feedback Comments, Email Notifications, User Profile	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✓	✗
Generate All	✓	✗	✗	✗	✗	✗	✓	✗
Knowledge Skin Component	✓	✗	✗	✗	✗	✗	✗	✗



	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Language Skins	✓ <sup>3</sup>	✓	✓	✗	✗	✓	✓	✗
Links in Standalone Topics	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✗	✗
Menu Skin Component	✓	✗	✗	✗	✗	✗	✗	✗
Navigation Pane Settings	✓	✗	✗	✗	✗	✗	✓	✗
Preview Skin for Output Type	✓	✗	✗	✗	✗	✗	✓	✗
Promotion Skin Component	✓	✗	✗	✗	✗	✗	✗	✗
Runtime Skins (Choose Different Skins at Runtime)	✓	✗	✗	✗	✗	✗	✗	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Search Bar Skin Component	✓ <sup>4</sup>	✗	✗	✗	✗	✗	✗	✗
Search Results Skin Component	✓ <sup>4</sup>	✗	✗	✗	✗	✗	✗	✗
Separate UI Text Tab in Skin Editor for Localization	✓ <sup>3</sup>	✗	✗	✗	✗	✗	✗	✗
Styles	✓	✗	✗	✗	✗	✗	✓ <sup>5</sup>	✗
TOC Entry (Select Skin For)	✗	✗	✗	✗	✗	✗	✓	✗
Topic Toolbar (Custom Settings)	✓ <sup>6</sup>	✗	✗	✗	✗	✗	✓	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Web Toolbar (Add via Skin)	✗	✗	✗	✗	✗	✗	✓	✗
Web Toolbar (Custom Settings)	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✓	✗
Window (Size/Position)	✓ <sup>1</sup>	✗	✗	✗	✗	✗	✓	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Window (User-Defined Size/Position)	✗	✗	✗	✗	✗	✗	✓	✗

<sup>1</sup>This is available in the Tripane skin only. It is not available in the Side or Top Navigation skins.

<sup>2</sup>This is handled through the org.eclipse.help.base plug-in.

<sup>3</sup>As an alternative to using language skins, you can use the UI Text tab in the Skin Editor to perform all of the localization tasks for HTML5 targets. Be aware that translations in the Skin Editor are prioritized over translations in a language skin, so you should try to perform your localization tasks in one place to avoid conflicts.

<sup>4</sup>This is available in the Side and Top Navigation skins only. It is not available in the Tripane skin.

<sup>5</sup>Only some styles (e.g., Feedback, toolbar) are supported.

<sup>6</sup>Topic toolbar settings can be controlled in a Topic Toolbar skin component, as well as in a Topic Toolbar proxy.

# TOC and Mini-TOC Features Per Output Type

Following are table of contents (TOC) and mini-TOC features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
TOCs (via Proxy)	✗	✓	✓	✗	✓	✓ <sup>1</sup>	✗	✓
TOCs (via Skin)	✓	✗	✗	✗	✗	✗	✓	✗
Auto-Generate	✗	✓	✓	✗	✗	✓	✗	✓
Binary TOCs	✗	✗	✗	✗	✗	✗	✓	✗
Mark as New	✓	✗	✗	✗	✓	✗	✓	✗
Mini-TOCs	✓	✓	✓	✓	✓	✓	✓	✓
Skin (Select for TOC Entry)	✗	✗	✗	✗	✗	✗	✓	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Synchronize With Topics	✓	✗	✗	✗	✗	✗	✓	✗

<sup>1</sup>Some ebook readers have a built-in TOC.

# Translation and Localization Features Per Output Type

Following are translation features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Author and Generate Output in Left-to-Right (LTR) and Right-to-Left (RTL) Languages	✓	✓	✓	✓	✓	✓	✓	✗
Invert Image Callouts to RTL	✓	✓	✓	✓	✓	✓	✓	✗
Invert Page Layouts to RTL	✗	✓	✓	✗	✗	✗	✗	✗
Invert Styles to RTL	✓	✓	✓	✓	✓	✓	✓	✗
Language Skins	✓ <sup>1</sup>	✓	✓	✗	✗	✓	✓	✗

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Multilingual Output (Link to External Projects in Target)	✓	✓	✓	✓	✓	✓	✓	✓
Output Interface (Display in LTR and RTL)	✓ <sup>2</sup>	✗	✗	✗	✓	✗	✗	✗
PDF Stitching <sup>3</sup>	✓	✓	✗	✗	✗	✗	✓	✗
Separate UI Text Tab in Skin Editor for Localization	✓	✗	✗	✗	✗	✗	✗	✗

<sup>1</sup>As an alternative to using language skins, you can use the UI Text tab in the Skin Editor to perform all of the localization tasks for HTML5 targets. Be aware that translations in the Skin Editor are prioritized over translations in a language skin, so you should try to perform your localization tasks in one place to avoid conflicts.

<sup>2</sup>This is controlled on the UI Text tab in a regular HTML5 skin.

<sup>3</sup> PDF stitching allows you to add PDFs to Flare output, but you cannot link to content in that PDF. In online outputs, the PDF is simply added to the TOC and can be opened from there; it is not technically "merged" into the output.



# Variable Features Per Output Type

Following are variable features supported in each output type.

	HTML5	PDF	Word	Clean XHTML	Eclipse Help	EPUB	HTML Help	PPTX
Basic Variables	✓	✓	✓	✓	✓	✓	✓	✓
Date & Time Variables	✓	✓	✓	✓	✓	✓	✓	✓
Heading Variables	✓	✓	✓	✓	✓	✗	✓	✓
Running HF Variables	✗	✗	✓	✗	✗	✗	✗	✓
Snippet Variables	✓	✓	✓	✓	✓	✓	✓	✓
System Variables	✓	✓	✓	✓	✓	✓	✓	✓