

MADCAP FLARE DESKTOP 2025 r2

Context-Sensitive Help

Copyright © 2025 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

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1

Welcome 5

CHAPTER 2

Before We Begin 6

 Understanding Context-Sensitive Help 7

 Limitations of This Tutorial 7

CHAPTER 3

Creating a Project From a Template 8

 How to Create a New Project From a Template 9

CHAPTER 4

Adding Context-Sensitive Help Files10

 How to Add a Header File11

 How to Add an Alias File12

 How to Create a Context-Sensitive Help Skin13

CHAPTER 5

Setting Context-Sensitive Help ID Options21

 How to Set CSH ID Options22

CHAPTER 6

Creating and Assigning Context-Sensitive Help IDs	24
How to Create and Assign a Single CSH ID	25
How to Create and Assign Many CSH IDs	26

CHAPTER 7

Creating Context-Sensitive Help for Micro Content	30
How to Add a Micro Content File	31
How to Create Phrases and Responses for Field-Level Help	35
How to Create and Assign a CSH ID for a Micro Content Phrase	37

CHAPTER 8

Associating an Alias File With a Target	39
How to Associate an Alias File With a Target	40

CHAPTER 9

Generating and Testing Output	41
How to Associate the Main Skin With the Target	42
How to Generate the Target	43
How to Test the Context-Sensitive Help	44
In the Real World... ..	47

APPENDIX

PDFs	49
Tutorials	49
Cheat Sheets	50
User Guides	51

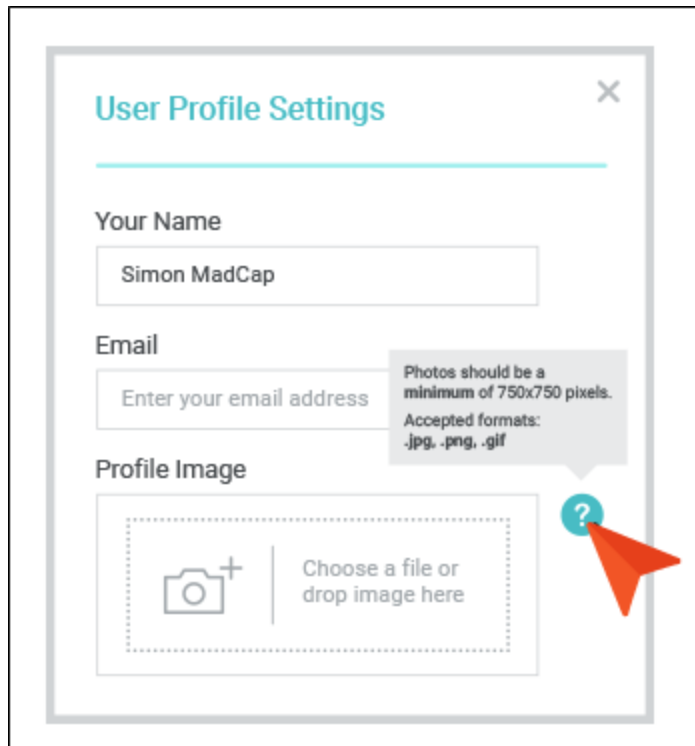
CHAPTER 1

Welcome

Approximate Time: 25 minutes

Welcome to the Context-Sensitive Help Tutorial.

- **Tutorial Goal** Learning how to create and call context-sensitive Help (CSH) output.
- **Key Concepts** Creating CSH files and IDs at the topic and field level, applying a skin, and using URLs to bring up specific content.



CHAPTER 2

Before We Begin

Before we begin this tutorial, there are just a couple of things you should know.

This chapter discusses the following:

Understanding Context-Sensitive Help	7
Limitations of This Tutorial	7

I Understanding 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. This is different from a lot of features in Flare Desktop in that it usually requires a collaboration between the author and a software developer. The key to this relationship is the sharing of IDs that connect the specific pieces of documentation with the software user interface or links.

I Limitations of This Tutorial

Since CSH often involves a developer who is working on a software product, and we don't have that kind of resource in this tutorial, we aren't going to try to simulate that entire process. However, at the end of the tutorial, we will use a "call" to bring up specific documentation. It's the same information a developer would use to complete the connection on the software side, but we will use it in a slightly different, simpler way for the author to open the same output.

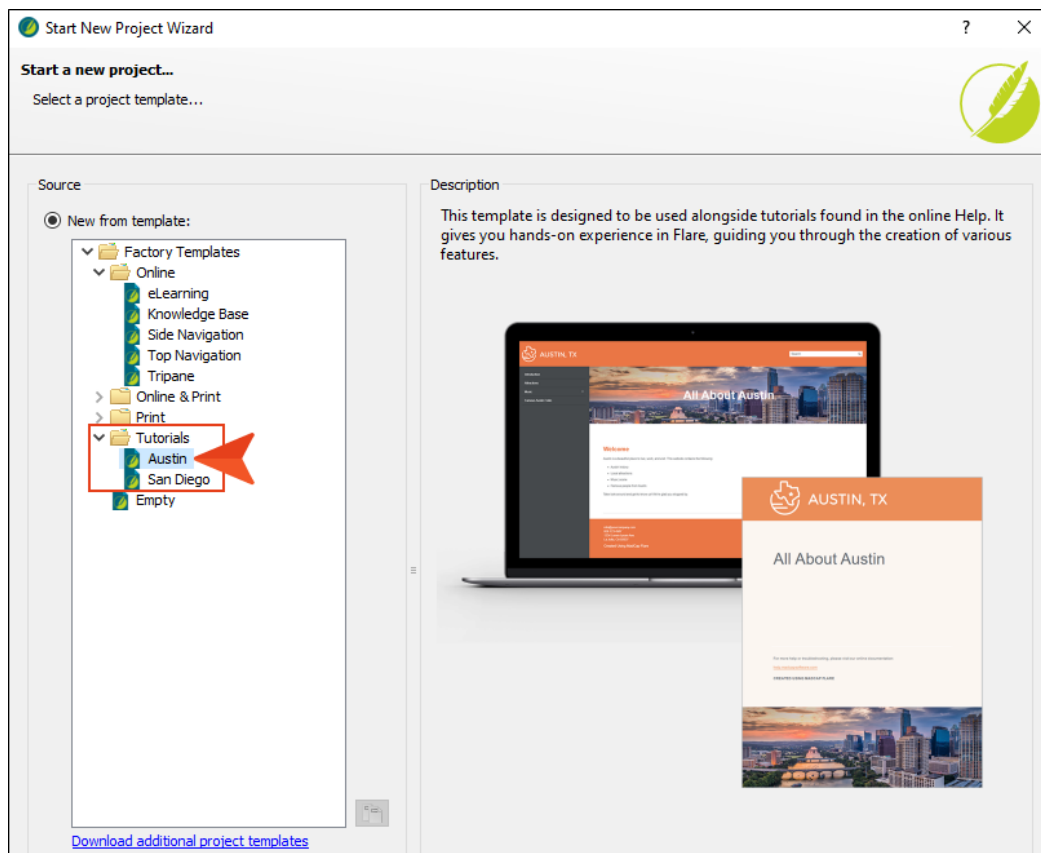
CHAPTER 3

Creating a Project From a Template

To start, let's create a new project based on Flare's Austin template.

I How to Create a New Project From a Template

1. Select **File > New Project**. The Start New Project Wizard opens.
2. In the **Project name** field, give your project any name you want.
3. (Optional) In the **Project folder** field, you can enter a location where you want to save your project. By default, projects are saved in your Documents > My Projects folder.
4. Click **Next**.
5. Under **Factory Templates**, expand the **Tutorials** folder. Select the **Austin** template.



6. Click **Next**.
7. For branding, keep the default settings. (This is where you could customize your project's look with branding, such as colors and logo.) Click **Next**.
8. Click **Finish**. The project is created and loaded into Flare Desktop.

Adding Context-Sensitive Help Files

There are a couple of files that are necessary when creating context-sensitive Help (CSH). One is called a "header file," and the other is called an "alias file."

A header file (sometimes referred to as a "map file") is a simple text file that works in conjunction with an alias file (i.e., as you edit the alias file, the header file is populated automatically) in context-sensitive Help. The header file contains basic information about connecting areas of an interface to the corresponding topics or micro content in online output. Both you and the software developer need access to this file, or at least to the IDs that will be used to connect various parts of the interface to specific areas of the documentation.

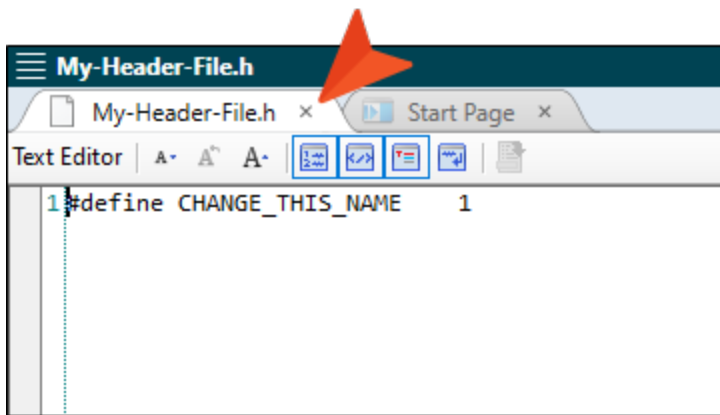
In this tutorial, we will add both of these files, as well as a special skin file. The skin file is optional; it's just a way to present your CSH with a different look than the regular documentation.

This chapter discusses the following:

How to Add a Header File	11
How to Add an Alias File	12
How to Create a Context-Sensitive Help Skin	13

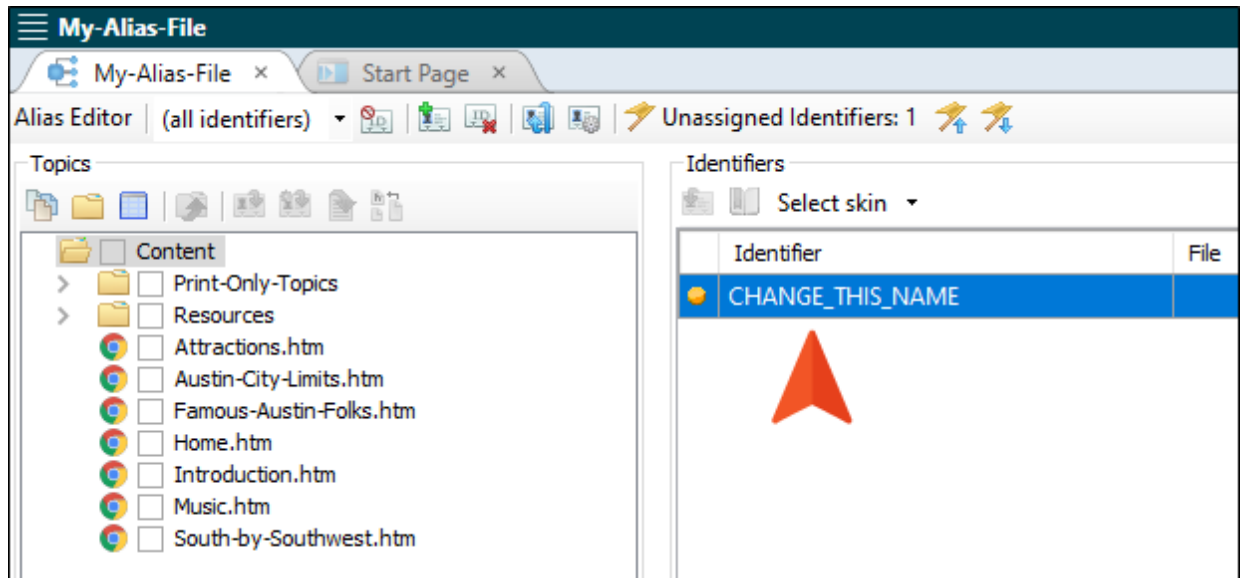
I How to Add a Header File

1. In the Project Organizer, right-click the **Advanced** folder and from the context menu select **Add Header File**.
2. In the **File Name** field, enter `My-Header-File`.
3. Click **Add**. The header file is added to the Advanced folder in the Project Organizer (expand the folder to see it). The Text Editor opens to the right, with the page for the new header file (including an initial identifier) shown.
4. Close the Text Editor by clicking the **x** in the tab.



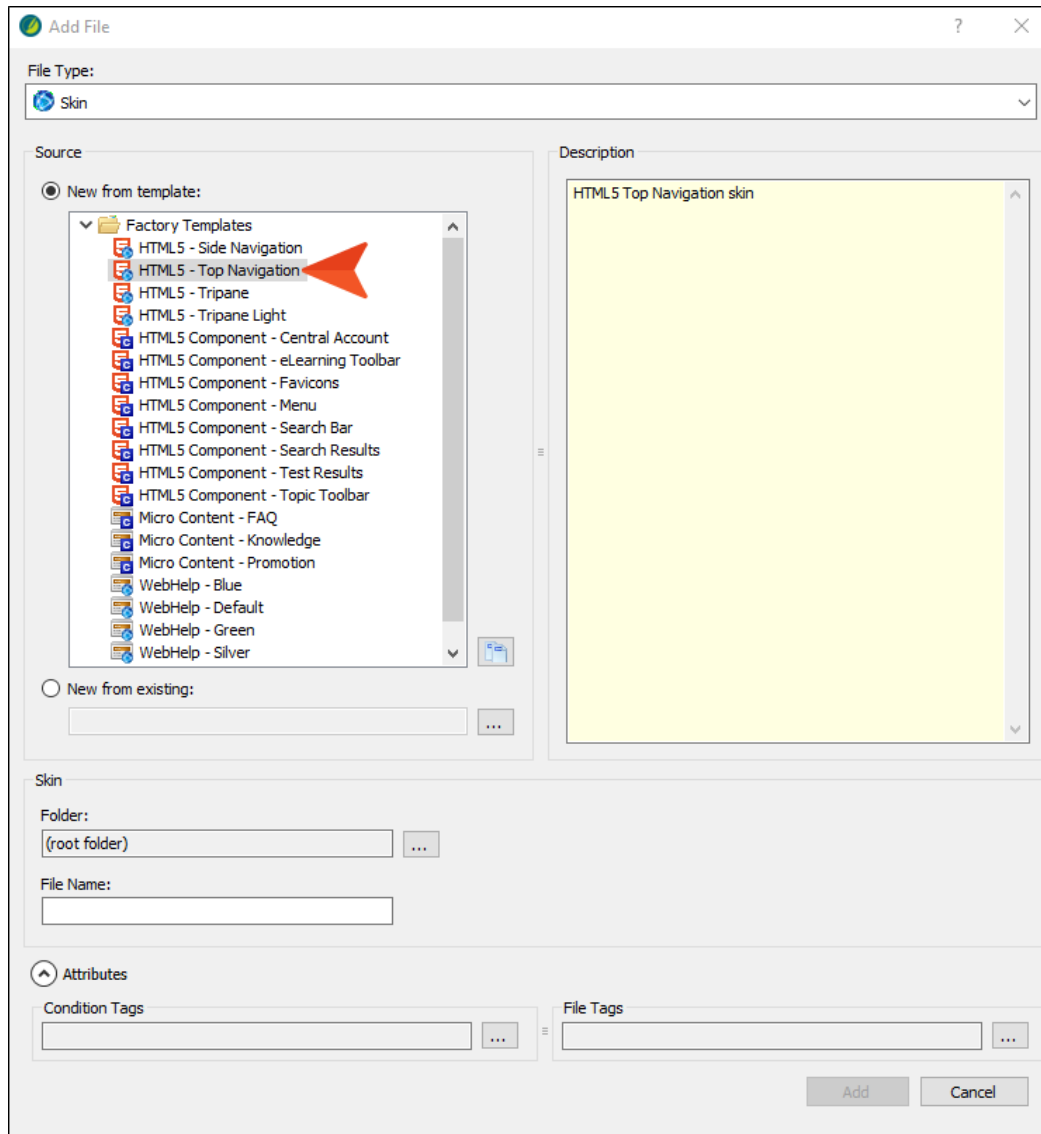
I How to Add an Alias File

1. In the Project Organizer, right-click the **Advanced** folder and from the context menu select **Add Alias File**.
2. In the **File Name** field, enter `My-Alias-File`.
3. Click **Add**. The alias file is added to the Advanced folder in the Project Organizer. The Alias Editor opens to the right, with the page for the new alias file shown. The file includes an initial identifier for the header file that you created previously.

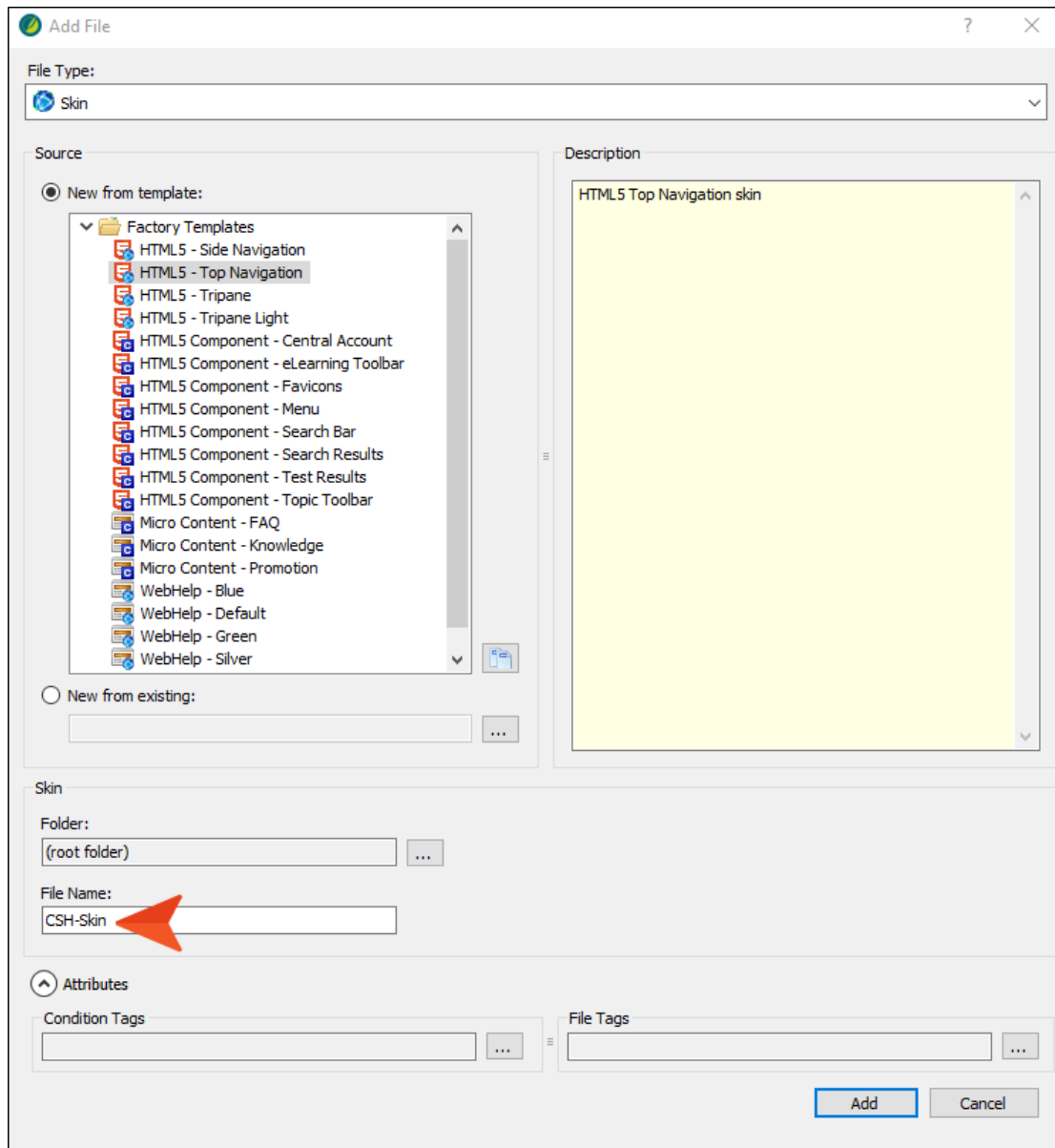


I How to Create a Context-Sensitive Help Skin

1. In the Project Organizer, right-click the **Skins** folder and from the context menu select **Add Skin**.
2. From the list of **Factory Templates**, select **HTML5 - Top Navigation**.

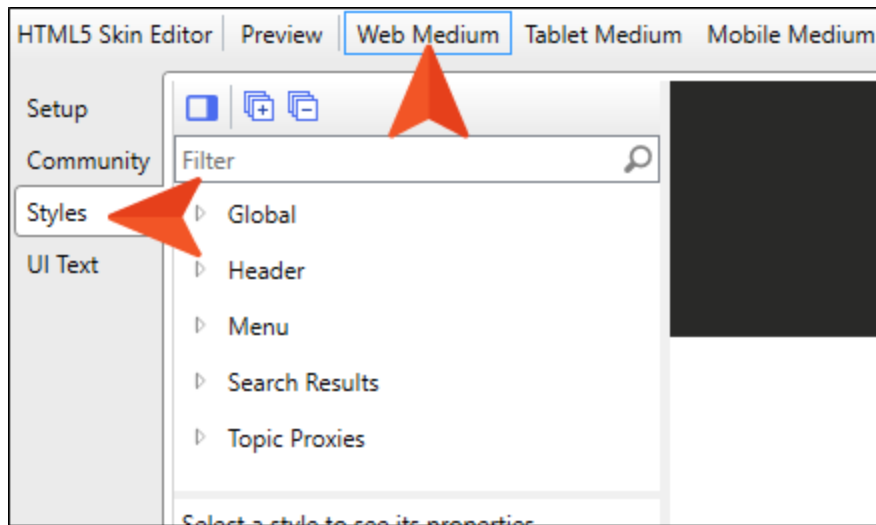


3. In the **File Name** field, type CSH-Skin.

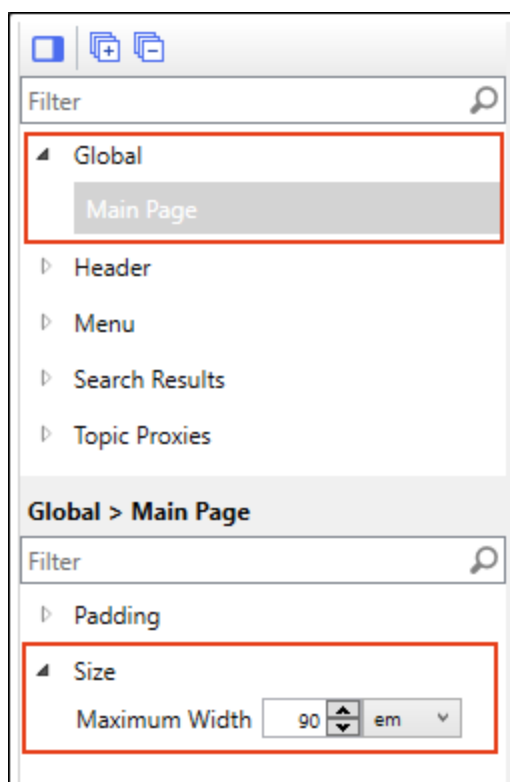


4. Click **Add**. The skin is added to the Skins folder in the Project Organizer.

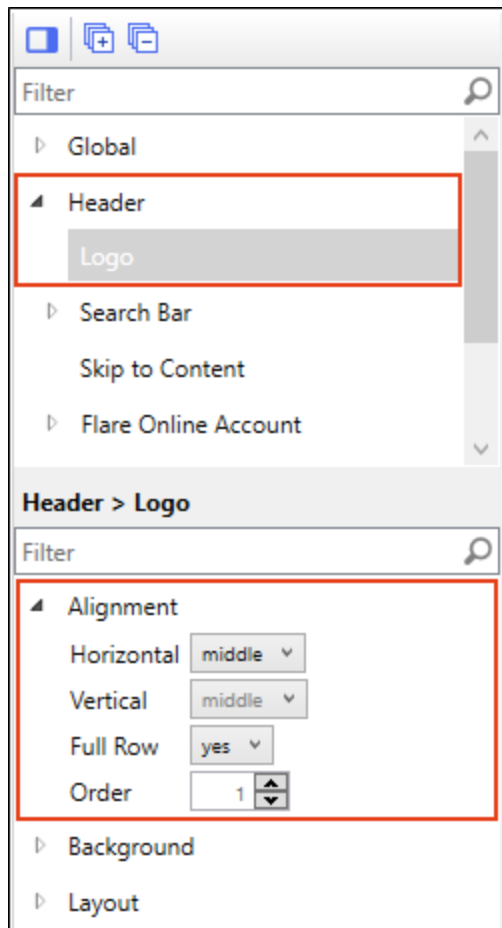
5. In the Skin Editor, select the **Styles** tab, and make sure **Web Medium** is selected at the top.



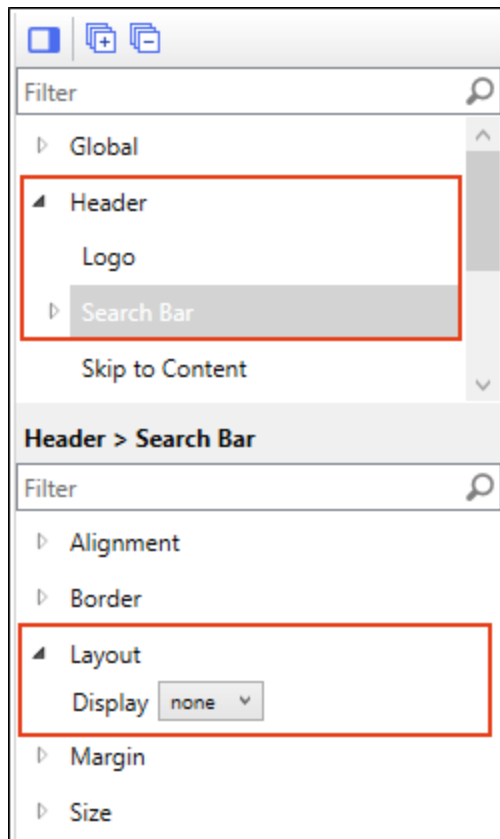
6. Let's make a few simple changes. On the left side listing the style groups, expand **Global**, and select **Main Page**. In the properties section, expand **Size**. Then set the **Maximum Width** field to 90 em.



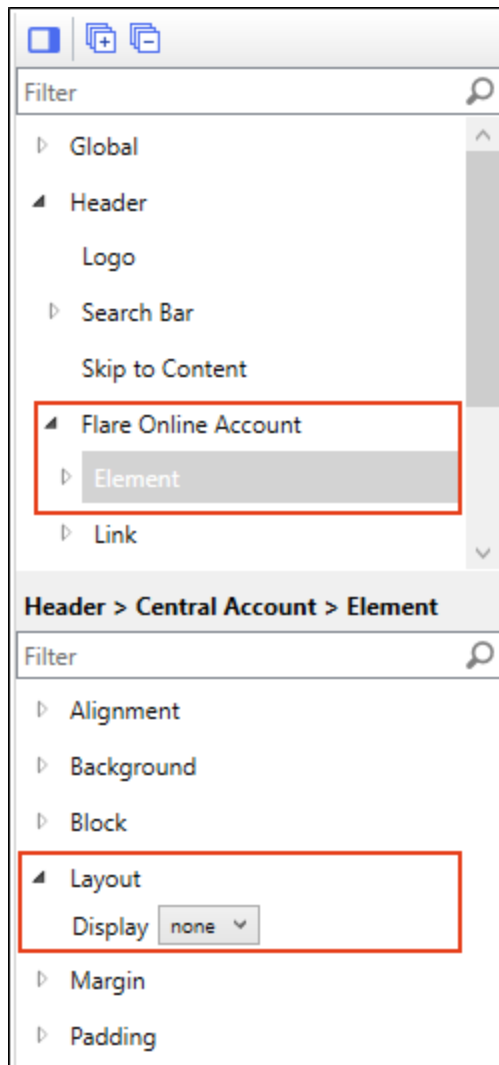
7. In the style groups section expand **Header**, and select **Logo**. In the properties section, expand **Alignment**. Then set the **Horizontal** field to **middle**, and set **Full Row** to **yes**.



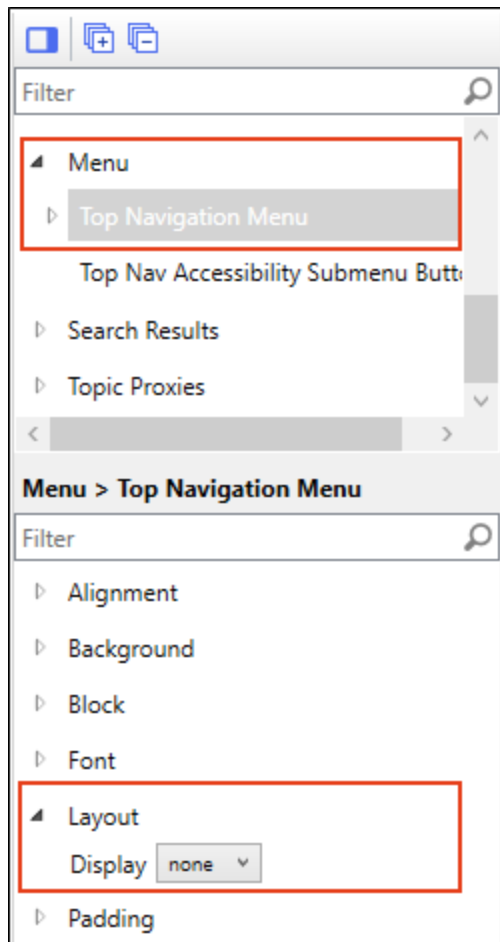
8. Under the **Header** style group, select **Search Bar**. In the properties section, expand **Layout**. Then set the **Display** field to **none**.



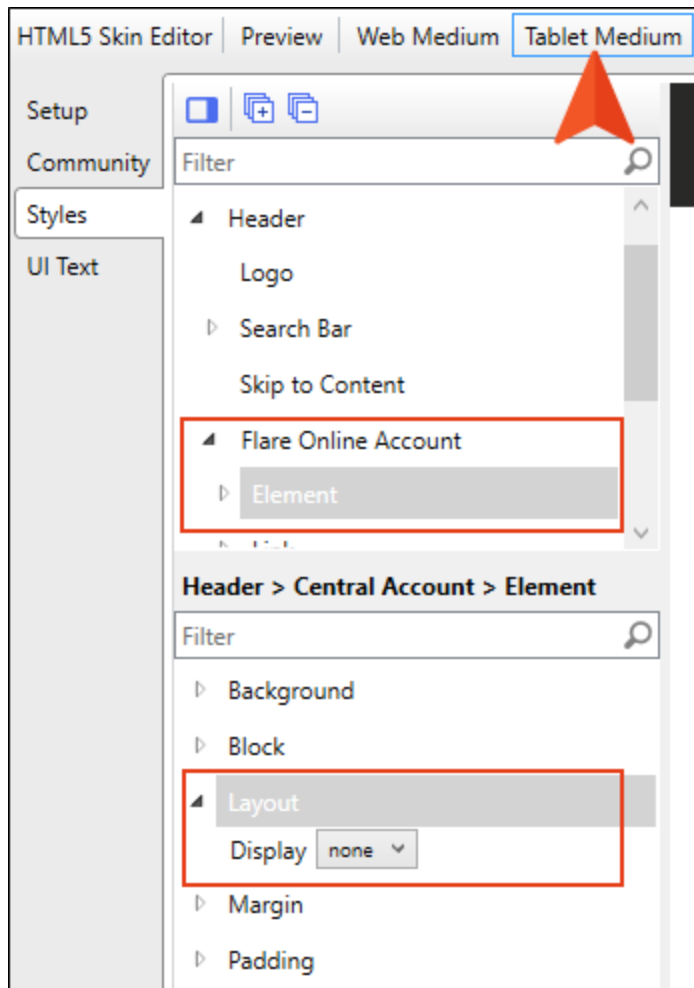
9. Under the **Header** style group, expand **Flare Online Account**, and select **Element**. In the properties section, expand **Layout**. Then set the **Display** field to **none**.



10. In the style groups section expand **Menu**, and select **Top Navigation Menu**. In the properties section, expand **Layout**. Then set the **Display** field to **none**.



11. At the top of the editor, select **Tablet Medium**.
12. Under the **Header** style group expand **Flare Online Account**, and select **Element**. In the properties section, expand **Layout**. Then set the **Display** field to **none**.



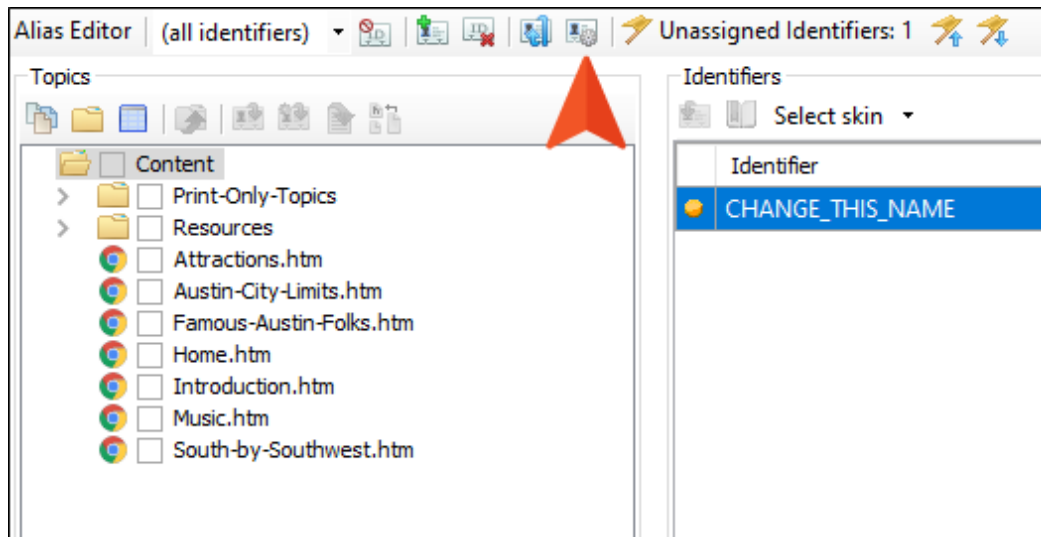
13. Click  to save your work.

Setting Context-Sensitive Help ID Options

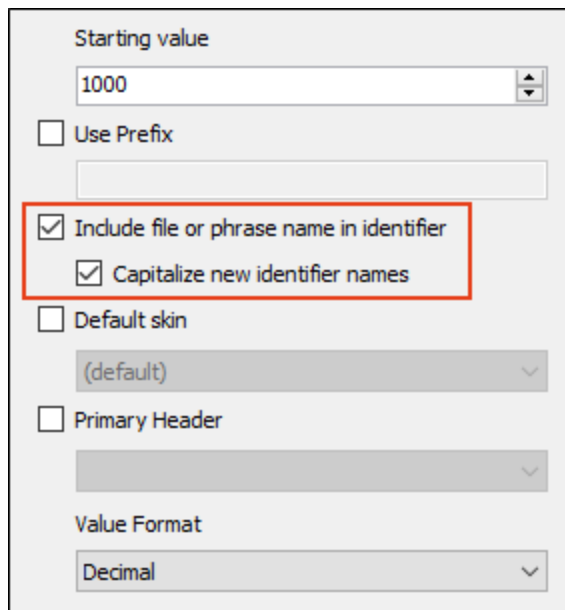
We're almost ready to create the IDs that will be used for the context-sensitive Help (CSH) output. But first, let's set some options. This is an optional step, but as you will see, it helps speed up the process and keep everything consistent.

I How to Set CSH ID Options

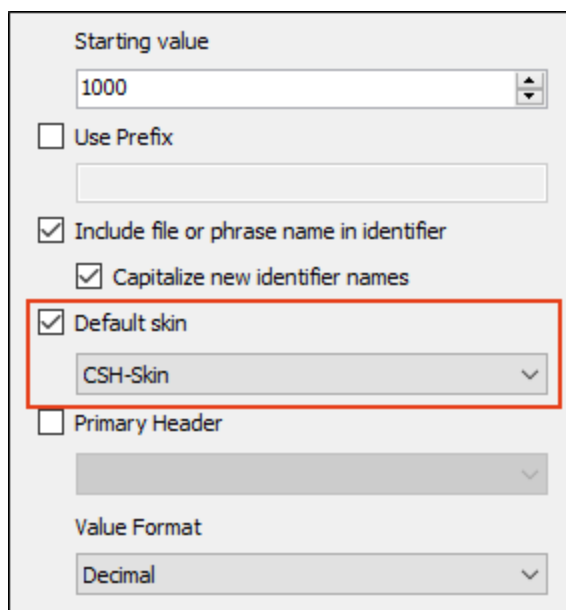
1. From the Project Organizer, expand the **Advanced** folder and double-click **My-Alias-File** (or simply switch to that tab in the editing area if that file is still open).
2. In the toolbar of the Alias Editor, click .



3. In the dialog, select **Include file or phrase name in identifier**.



4. Select **Default skin**. Then from the drop-down select **CSH-Skin**.



The image shows a configuration dialog box with the following elements:

- Starting value**: A text box containing the number 1000.
- Use Prefix**: An unchecked checkbox.
- Include file or phrase name in identifier**: A checked checkbox.
- Capitalize new identifier names**: A checked checkbox.
- Default skin**: A checked checkbox, and a drop-down menu below it showing "CSH-Skin". This entire section is highlighted with a red rectangular box.
- Primary Header**: An unchecked checkbox.
- Value Format**: A drop-down menu showing "Decimal".

5. We could select or change any of the other options if we wanted, but for the purposes of this tutorial, we'll just leave them as they are. Click **OK**.

Creating and Assigning Context-Sensitive Help IDs

Now that we've created our context-sensitive Help (CSH) files (header, alias, skin) and set some advanced ID options, we can create and assign some IDs.

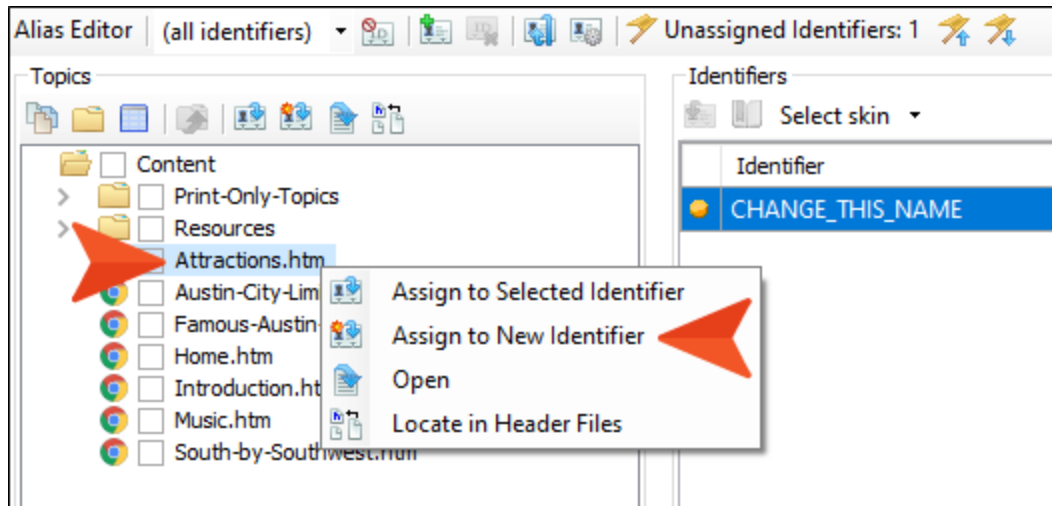
This chapter discusses the following:

- How to Create and Assign a Single CSH ID25
- How to Create and Assign Many CSH IDs26

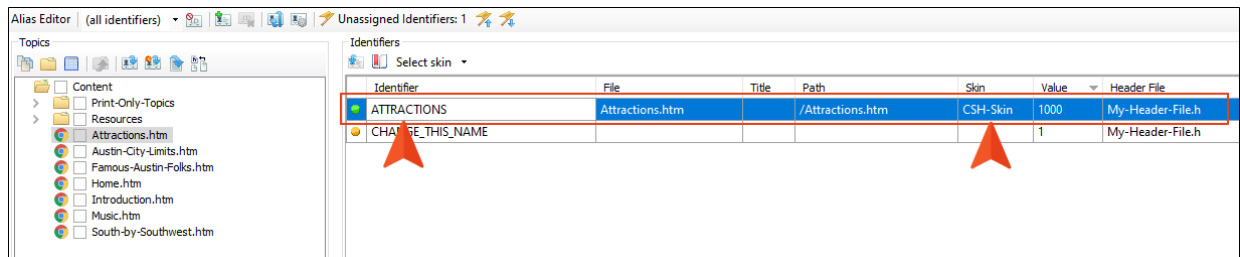
I How to Create and Assign a Single CSH ID

First, let's create just a single ID and assign it to a topic.

1. On the left side of the Alias Editor, right-click **Attractions.htm**, and from the context menu select **Assign to New Identifier**.



A new ID is added to the grid on the right. Notice that it automatically used the options set in the previous lesson. In other words, the ID is named after the file and is capitalized (ATTRactions). Also, it is associated with CSH-Skin.

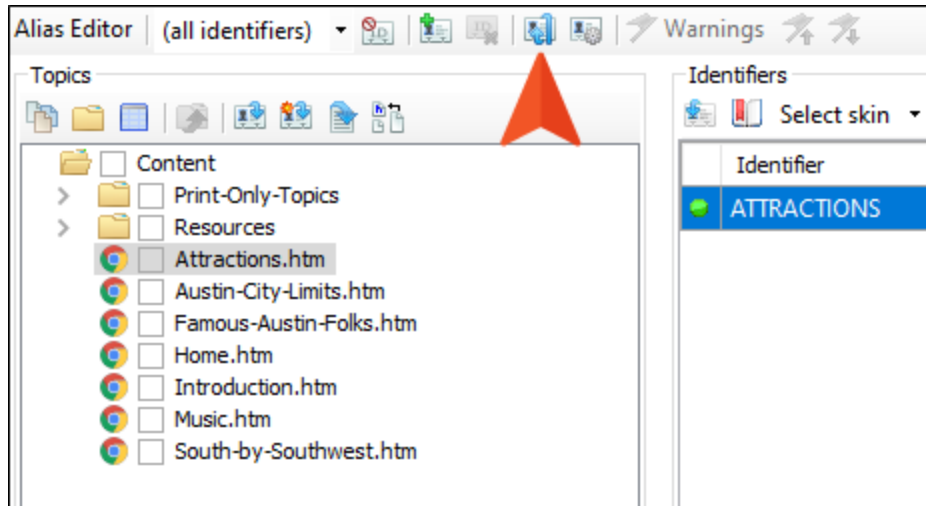


2. Click the other row in the grid with the initial ID (CHANGE_THIS_NAME) and delete it.
3. Click  to save your work.

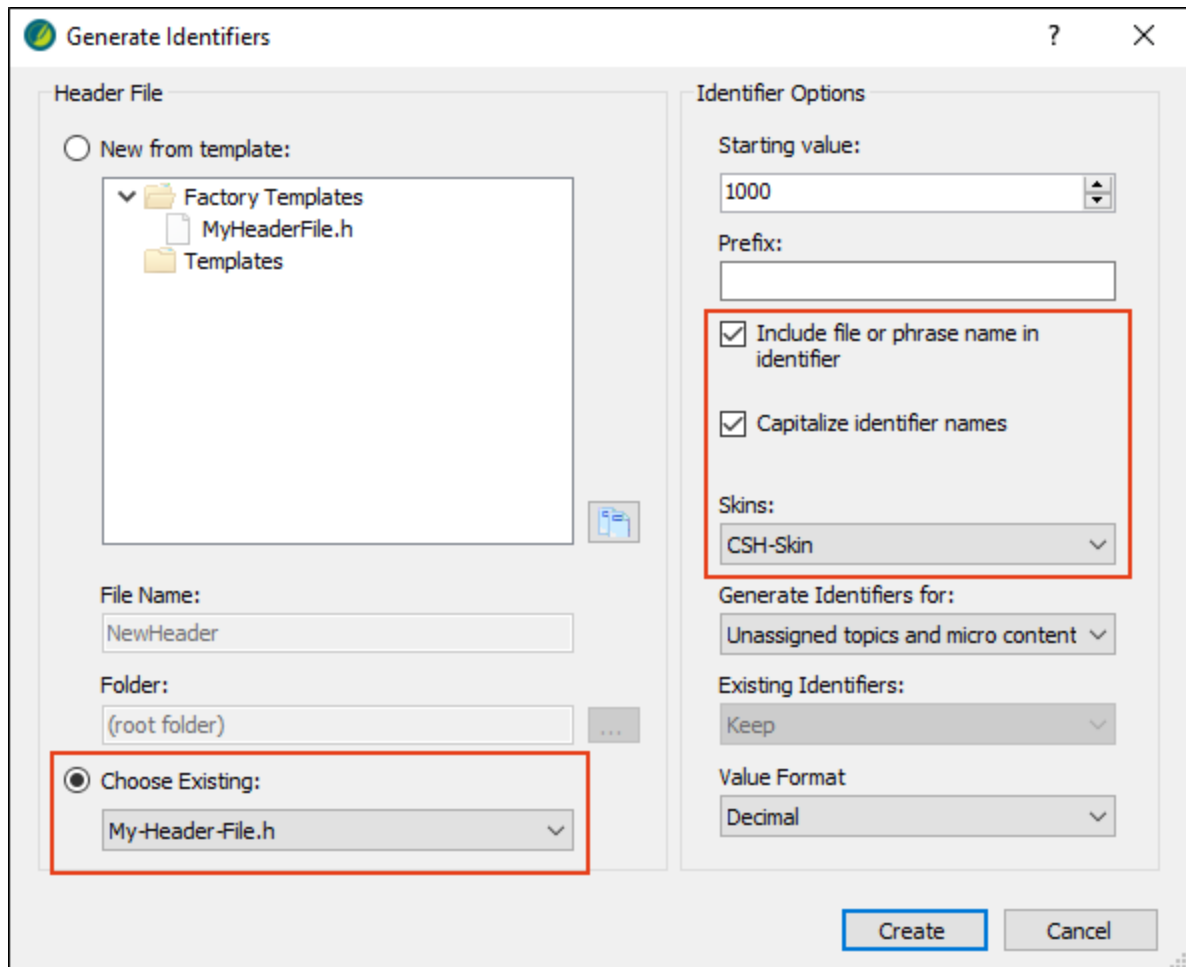
I How to Create and Assign Many CSH IDs

Now let's create and assign a lot of CSH IDs at the same time.

1. At the top of the Alias Editor, click .



In the dialog, your header file should already be selected, and the ID options you chose earlier should already be set.



2. From the **Generate Identifiers for** field, select **Unassigned topics only**. (We don't have any micro content in this project yet. That's what we're going to cover in the next lesson. So for now we'll just generate IDs for topics.)

Generate Identifiers

Header File

☐ New from template:

- Factory Templates
 - MyHeaderFile.h
 - Templates

File Name: NewHeader

Folder: (root folder)

☒ Choose Existing:

My-Header-File.h

Identifier Options

Starting value: 1000

Prefix:

☒ Include file or phrase name in identifier

☒ Capitalize identifier names

Skins: CSH-Skin

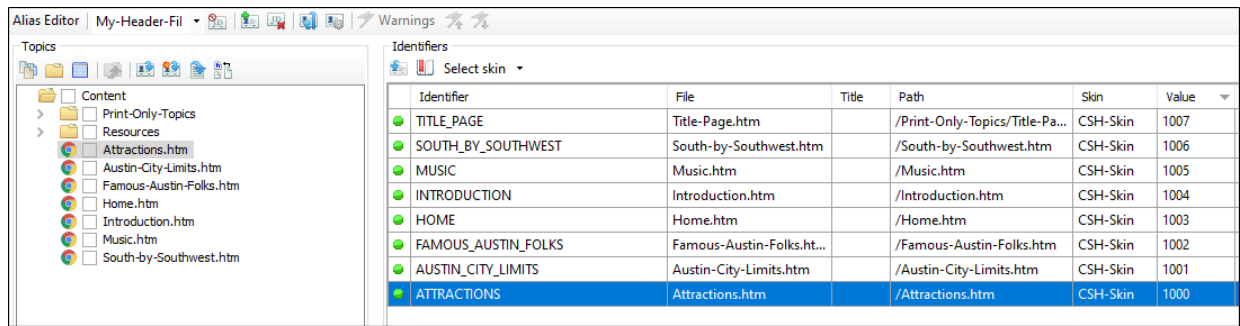
Generate Identifiers for: Unassigned topics only

Existing Identifiers: Keep

Value Format: Decimal

Create Cancel

3. Click **Create**. IDs are generated for all of the remaining topics.



The screenshot shows the Alias Editor interface. On the left, the 'Topics' panel lists a hierarchy: Content > Print-Only-Topics > Resources > Attractions.htm. Below this are several HTML files: Austin-City-Limits.htm, Famous-Austin-Folks.htm, Home.htm, Introduction.htm, Music.htm, and South-by-Southwest.htm. On the right, the 'Identifiers' panel shows a table with columns: Identifier, File, Title, Path, Skin, and Value. The table contains several rows, with the last row, 'ATTRactions', highlighted in blue.

Identifier	File	Title	Path	Skin	Value
TITLE_PAGE	Title-Page.htm		/Print-Only-Topics/Title-Pa...	CSH-Skin	1007
SOUTH_BY_SOUTHWEST	South-by-Southwest.htm		/South-by-Southwest.htm	CSH-Skin	1006
MUSIC	Music.htm		/Music.htm	CSH-Skin	1005
INTRODUCTION	Introduction.htm		/Introduction.htm	CSH-Skin	1004
HOME	Home.htm		/Home.htm	CSH-Skin	1003
FAMOUS_AUSTIN_FOLKS	Famous-Austin-Folks.ht...		/Famous-Austin-Folks.htm	CSH-Skin	1002
AUSTIN_CITY_LIMITS	Austin-City-Limits.htm		/Austin-City-Limits.htm	CSH-Skin	1001
ATTRactions	Attractions.htm		/Attractions.htm	CSH-Skin	1000

4. There are a couple of topics that we don't think we will use for CSH, so let's get rid of them. Click the **TITLE_Page** row. Then hold down your **CTRL** key and select the **HOME** row. Delete them.
5. Click  to save your work.

Creating Context-Sensitive Help for Micro Content

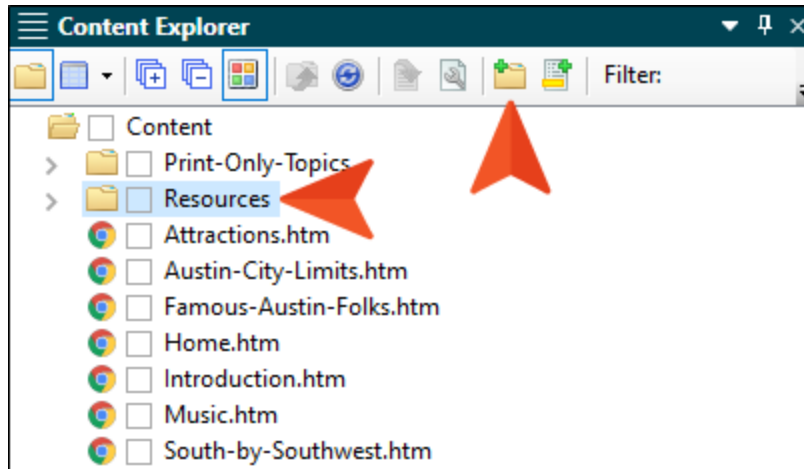
So far we have created IDs for topics in the project. These kinds of context-sensitive Help (CSH) IDs are useful for connections to large parts of a product user interface (e.g., a dialog or window pane). For smaller parts of an interface (i.e., field-level Help), micro content is more appropriate. That's because micro content is intended to be quite brief compared with full topics. So let's create some micro content in the project, and then produce some CSH IDs for it.

This chapter discusses the following:

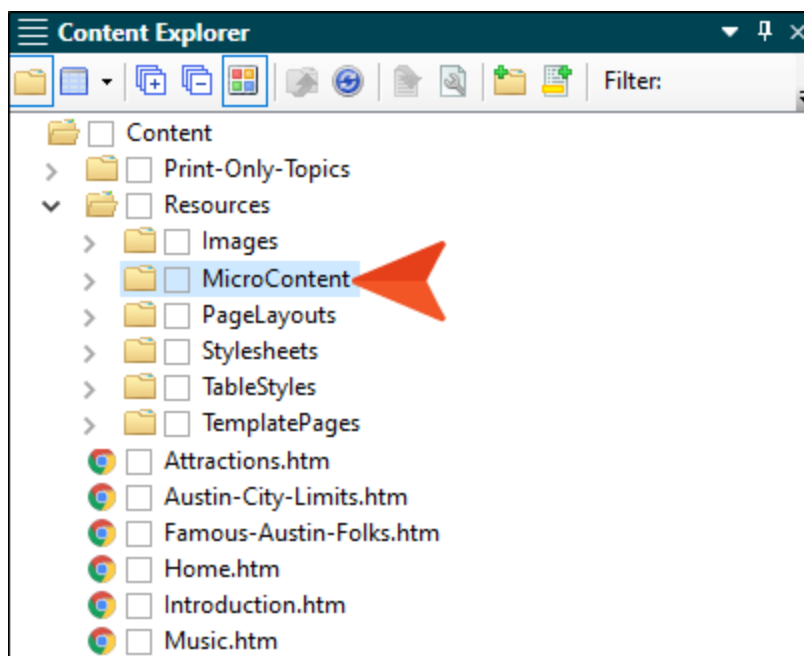
- How to Add a Micro Content File31
- How to Create Phrases and Responses for Field-Level Help35
- How to Create and Assign a CSH ID for a Micro Content Phrase 37

I How to Add a Micro Content File

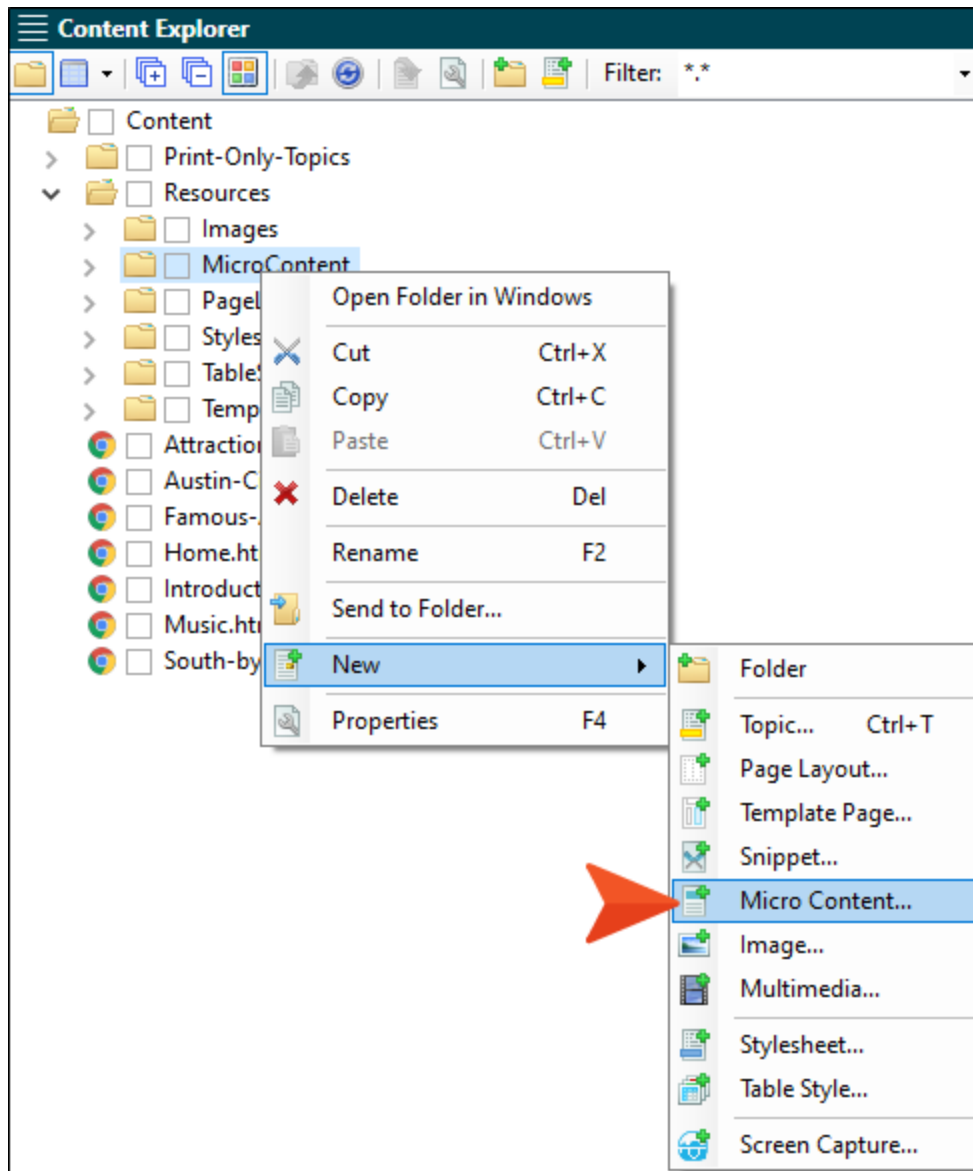
1. In the Content Explorer (not the Alias Editor), select the **Resources** folder.
2. In the local toolbar, click . A folder with the temporary name "NewFolder" is added.



3. Type `MicroContent` as the name and press **ENTER**. (Creating a folder is optional, but it's a good idea because it helps keep your micro content files organized and separate from your other resource files.)



4. Right-click the new **Resources > MicroContent** folder, and from the context menu select **New > Micro Content**.



5. In the **File Name** field, type `My-Micro-Content`.

Add File

File Type:
Micro Content

Source

☒ New from template:

- Factory Templates
 - MicroContent.fimco**
 - Templates

Description
Default micro content set.

☐ New from existing:

Micro Content

Folder:
Resources/MicroContent

File Name:
My-Micro-Content

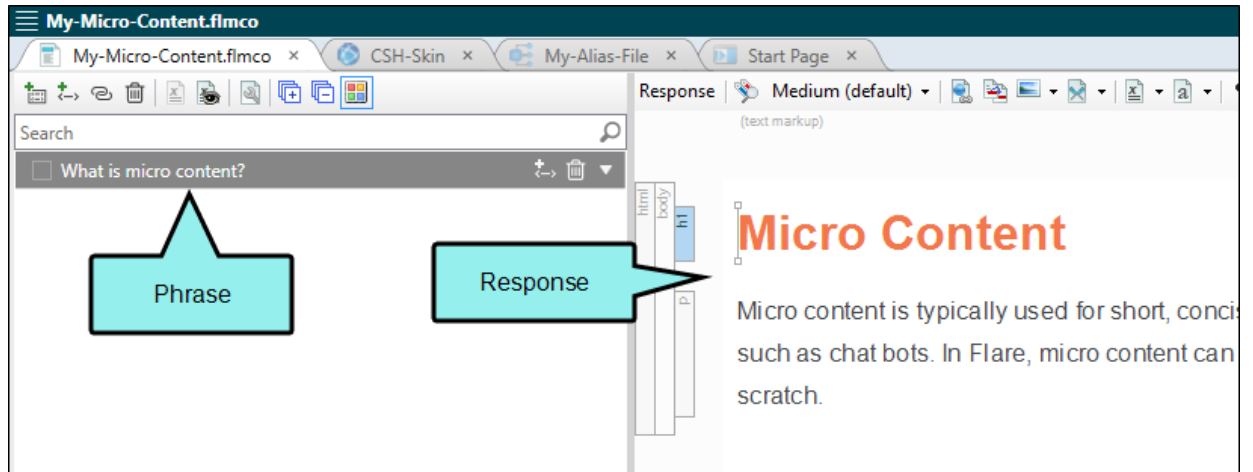
Attributes

Condition Tags

File Tags

Add Cancel

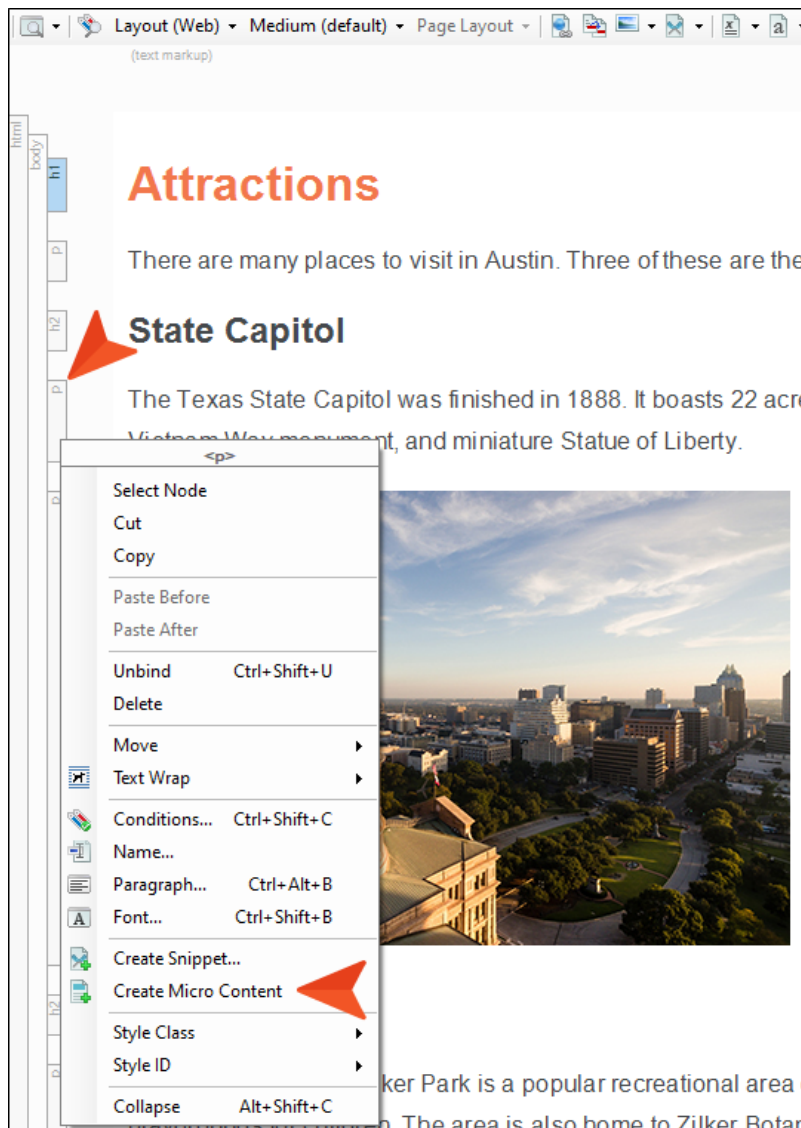
6. Click **Add**. The micro content file is added to the Content Explorer. The Micro Content Editor opens to the right, displaying micro content phrases on the left and an area to the right for the response. There is one phrase/response already added.



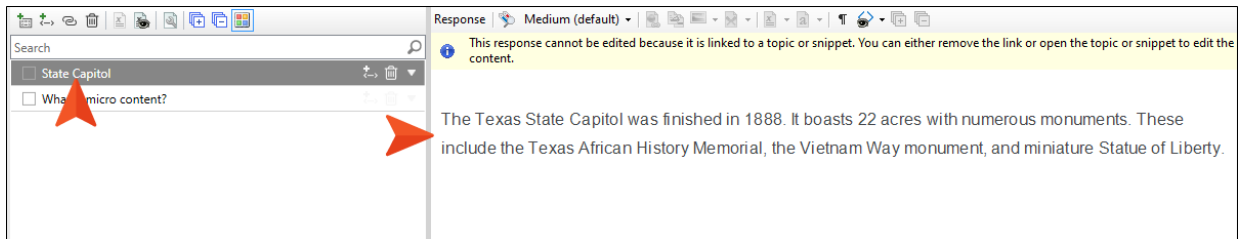
How to Create Phrases and Responses for Field-Level Help

There are multiple ways to create micro content phrases and responses. Let's use one of the most common methods to create a couple of phrase/response entries.

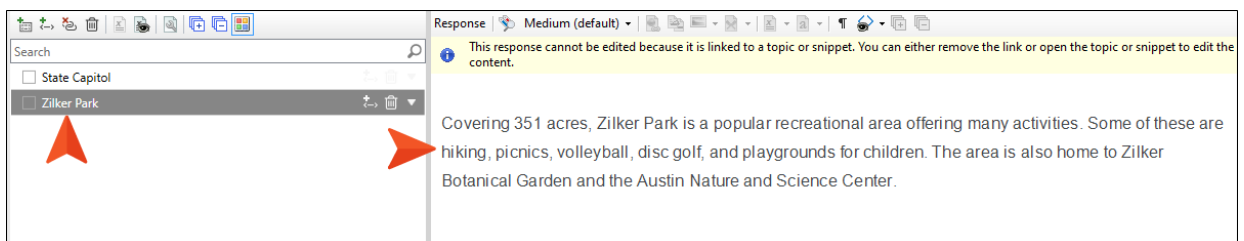
1. In the Content Explorer, double-click **Attractions.htm**.
2. In the topic, right-click the **p** structure bar to the left of the paragraph under the "State Capitol" heading. Then from the context menu select **Create Micro Content**.



3. In the dialog enter `State Capitol` as the phrase.
4. Click **OK**. The micro content file now shows the new phrase and response.



5. Let's remove the other phrase/response (What is micro content?). Click on that row and press **DELETE**.
6. Click  to save all files.
7. Go back to the Attractions topic and repeat these steps for the paragraph under the "Zilker Park" heading. Enter `Zilker Park` as the phrase. When you are finished, you should see that phrase/response in the micro content file as well.

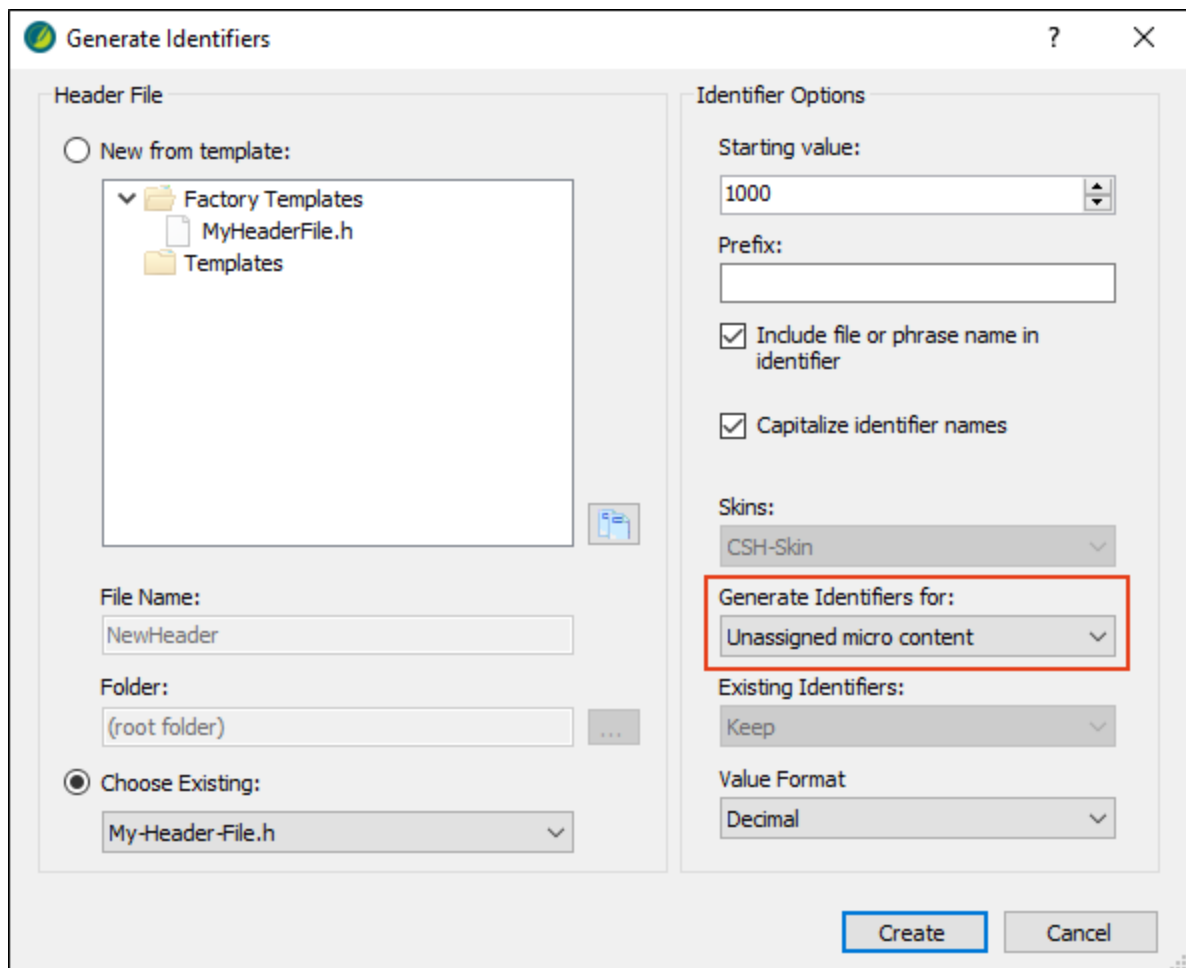


8. Click  to save all files.

I How to Create and Assign a CSH ID for a Micro Content Phrase

Now that we have some micro content, we can create IDs for it.

1. Go back to the Alias Editor.
2. At the top of the Alias Editor, click . In the dialog, the settings we saw earlier should still be in place.
3. From the **Generate Identifiers for** field, select **Unassigned micro content**.



- Click **Create**. IDs are generated for the two micro content phrase/response combinations. If you click and drag the **File** column divider to the right, you will see that a hash tag was added in each path, followed by the name of the phrase.

The screenshot shows the Alias Editor window with a table of identifiers. The table has columns: Identifier, File, Title, Path, Skin, and Value. Two rows are highlighted with red boxes: ZILKER_PARK and STATE_CAPITOL. Red arrows point to the vertical dividers between the File and Title columns, indicating where to drag to add hash tags to the file paths.

Identifier	File	Title	Path	Skin	Value
ZILKER_PARK	My-Micro-Content.flmco#zilker-park		/Resources/MicroContent/...		1007
SOUTH_BY_SOUTHWEST	South-by-Southwest.htm		/South-by-Southwest.htm	CSH-Skin	1006
MUSIC	Music.htm		/Music.htm	CSH-Skin	1005
INTRODUCTION	Introduction.htm		/Introduction.htm	CSH-Skin	1004
STATE_CAPITOL	My-Micro-Content.flmco#state-capitol		/Resources/MicroContent/...		1003
FAMOUS_AUSTIN_FOLKS	Famous-Austin-Folks.htm		/Famous-Austin-Folks.htm	CSH-Skin	1002
AUSTIN_CITY_LIMITS	Austin-City-Limits.htm		/Austin-City-Limits.htm	CSH-Skin	1001
ATTRACTIONS	Attractions.htm		/Attractions.htm	CSH-Skin	1000

You might have noticed that the skin is not assigned to those two rows. That is because skins are not supported for micro content IDs.

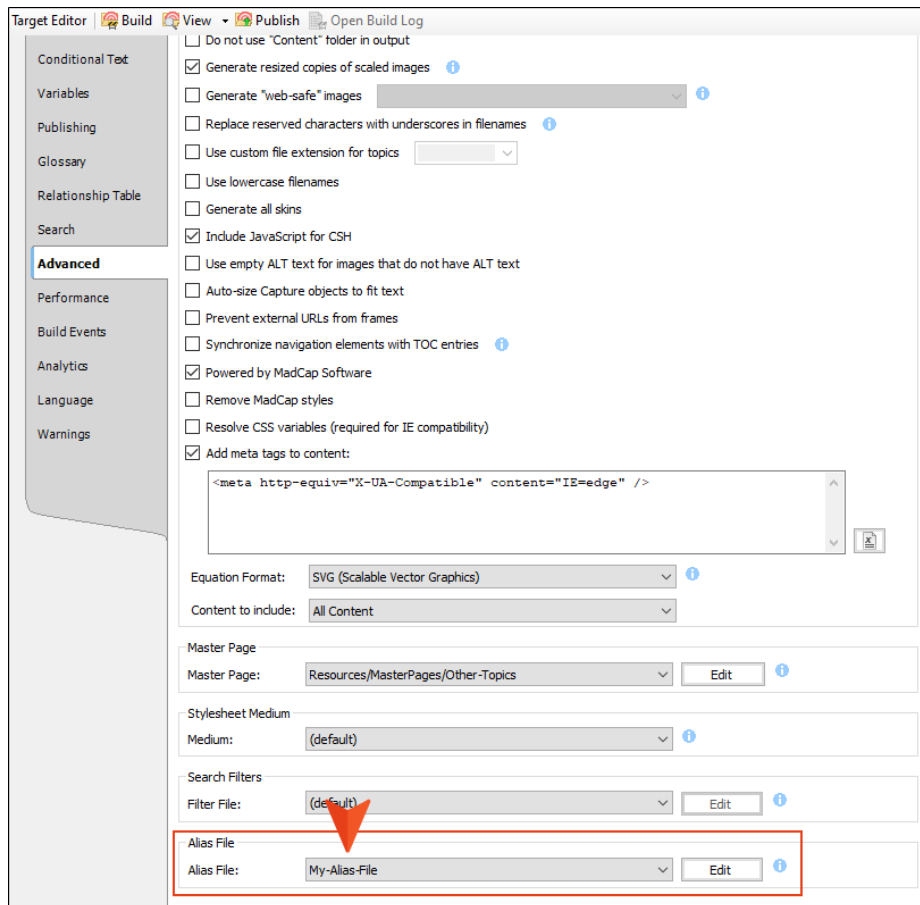
- Click  to save your work.

Associating an Alias File With a Target

We're almost done, but not quite yet. A project can have multiple alias files and multiple targets. Even though we only have one alias file and one online target so far in this project, it's still a good idea to connect the two, because we might want to add more alias files and online targets in the future, and our current target needs to point to the correct IDs.

I How to Associate an Alias File With a Target

1. In the Project Organizer, expand the **Targets** folder and double-click **All-About-Austin-HTML5**.
2. In the Target Editor, click the **Advanced** tab.
3. Scroll down to the **Alias File** field, click the drop-down, and select **My-Alias-File**.



4. Click  to save your work.

Generating and Testing Output

All of the pieces are now in place. So we can now generate the output and see how the context-sensitive Help (CSH) works.

This chapter discusses the following:

- How to Associate the Main Skin With the Target42
- How to Generate the Target43
- How to Test the Context-Sensitive Help44
- In the Real World...47

I How to Associate the Main Skin With the Target

Before we build the output, we need to associate the main skin in the project with the target. The reason we need to do this is that the target is currently set to use the default skin for the output, which is the first one it comes to alphabetically. And because we added a new skin for CSH and its name is first in the alphabet, we want to make sure that the bulk of the output doesn't use it. The CSH skin is intended only for the CSH IDs.

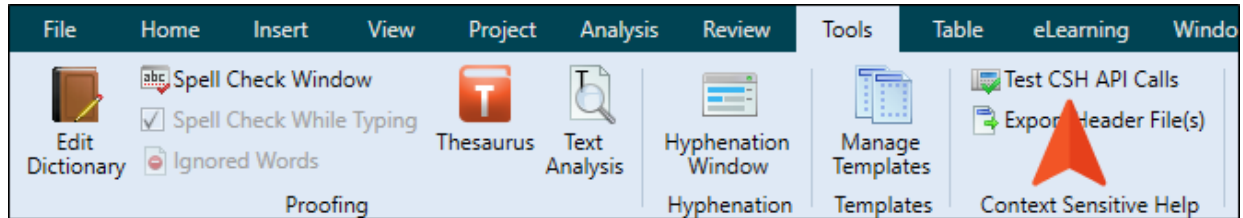
1. In the Target Editor, select the **Skin** tab.
2. Click the drop-down arrow in the **Skin** field and select **Side-Navigation** (the main skin).
3. Click  to save your work.

I How to Generate the Target

1. In the local toolbar of the Target Editor, click  Build. The Builds window pane opens at the bottom of the Flare Desktop interface, and the target begins to build.
2. Double-click the row in the Builds window pane to open the output in a browser. Notice the main skin is used for the output.

I How to Test the Context-Sensitive Help

1. In Flare Desktop, open the **Tools** ribbon, and select **Test CSH API Calls**.



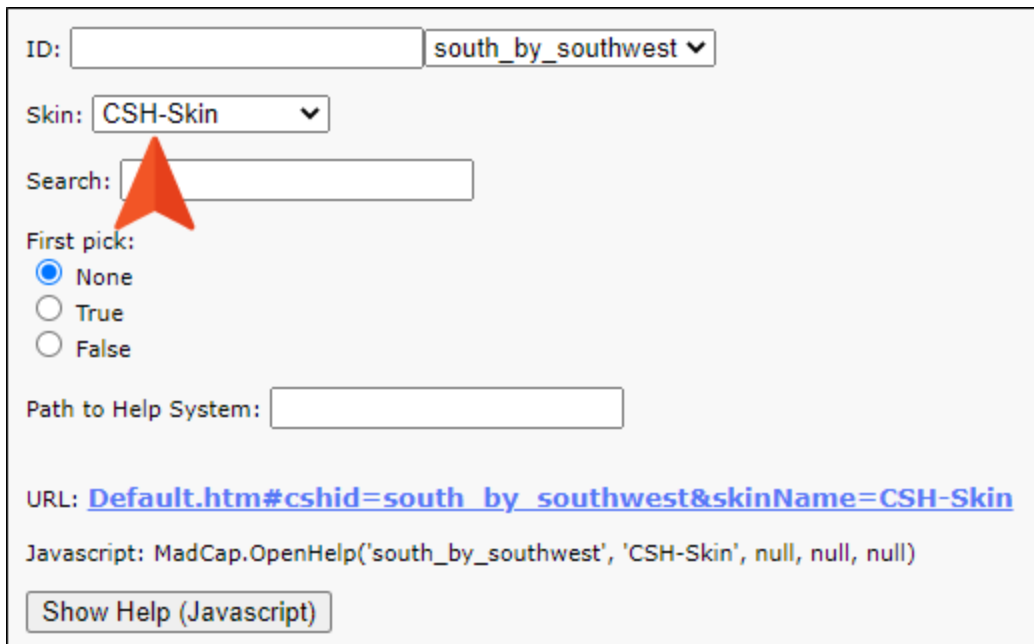
A new tab opens in your browser with fields you can use for testing.

2. In the **ID** field, click the down arrow and select **south_by_southwest** (one of our topic CSH IDs).

A screenshot of a web form titled 'Test CSH API Calls'. The form contains several input fields and a button. The 'ID' field is a dropdown menu with 'south_by_southwest' selected. The 'Skin' field is a dropdown menu. The 'Search' field is a text input. The 'First pick' section has three radio buttons: 'None' (selected), 'True', and 'False'. The 'Path to Help System' field is a text input. The 'URL' field displays 'Default.htm#cshid=south_by_southwest'. The 'Javascript' field displays 'MadCap.OpenHelp('south_by_southwest', null, null, null, null)'. A 'Show Help (Javascript)' button is at the bottom.

Notice that from the drop-down you can select either a text string as the ID or the number associated with it in the alias file. Either one works just fine.

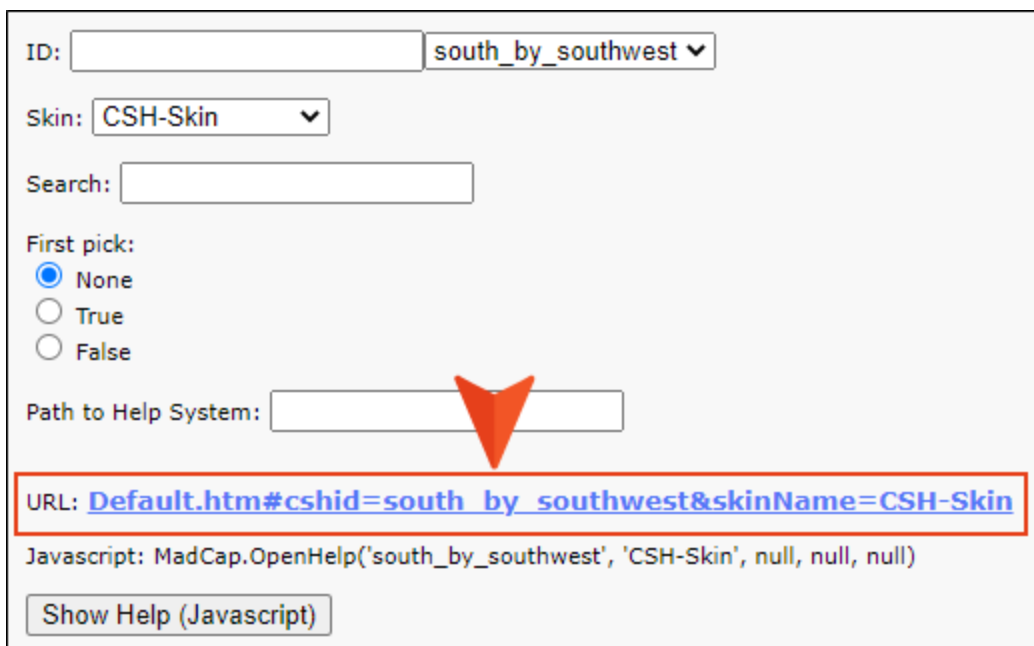
3. From the Skin field select CSH-Skin.



A screenshot of a web form for configuring a help system. The form includes the following elements:

- ID:** A text input field containing "south_by_southwest" and a dropdown arrow.
- Skin:** A dropdown menu with "CSH-Skin" selected.
- Search:** An empty text input field. A red arrow points to this field.
- First pick:** Three radio buttons labeled "None" (selected), "True", and "False".
- Path to Help System:** An empty text input field.
- URL:** A text field containing the URL [Default.htm#cshid=south_by_southwest&skinName=CSH-Skin](#).
- Javascript:** A text field containing the code `MadCap.OpenHelp('south_by_southwest', 'CSH-Skin', null, null, null)`.
- Show Help (Javascript):** A button.

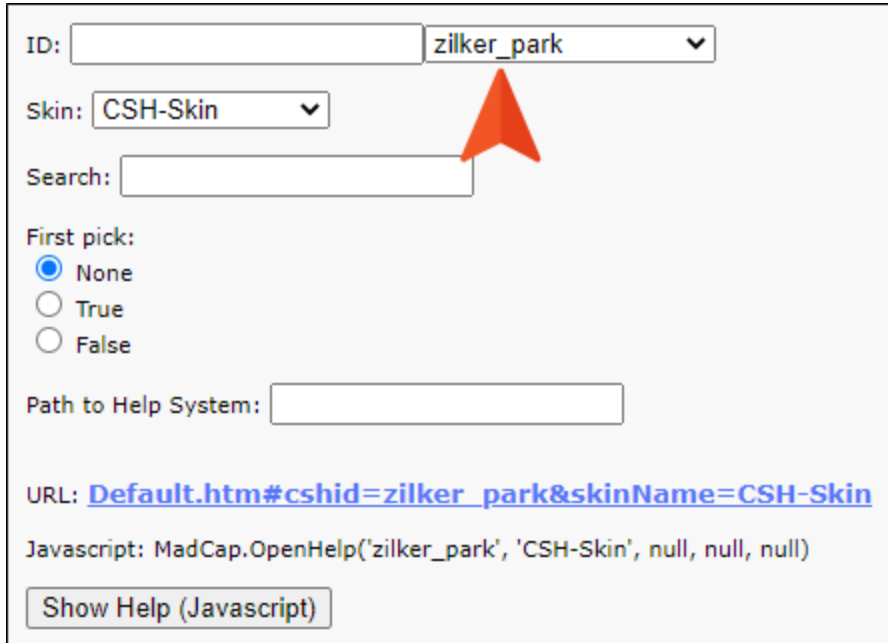
4. Notice the URL field has been populated. This provides us with the end of the URL path that is needed to point to the specific topic in the output. Click this link. The topic opens in the browser in a different skin than the full output.



A second screenshot of the same web form, showing the URL field populated. A red arrow points to the URL field, which is highlighted with a red border.

URL: [Default.htm#cshid=south_by_southwest&skinName=CSH-Skin](#)

5. Go back to the browser tab with the fields for testing the CSH. From the drop-down at the top select **zilker_park** (one of the micro content phrases). You can leave the Skin field set as is; it won't matter anyway, since skins are not supported for micro content.



ID: **zilker_park** ▼

Skin: **CSH-Skin** ▼

Search:

First pick:

☒ None

☐ True

☐ False

Path to Help System:

URL: Default.htm#cshid=zilker_park&skinName=CSH-Skin

Javascript: `MadCap.OpenHelp('zilker_park', 'CSH-Skin', null, null, null)`

6. Click the URL link at the bottom. A new tab opens with the response text for the micro content, without any skin.

In the Real World...

If we were dealing with a real-world situation in which we needed to connect our CSH with parts of a product interface, we would do a couple of other things.

First, we would send the CSH IDs to the programmer. You can send the entire contents of the header file to that person, or you can simply provide each ID as you create them.

☆ **EXAMPLE** In this tutorial the text ID for the Austin City Limits topic is `austin_city_limits`, and the generated numerical ID is 1001. So we could give our developer either of those, as well as the name of the skin (CSH-Skin).

Second, we would send the programmer the information in the Help topic called "CSH Calls for HTML5 Output."

This topic explains the two primary methods for linking to CSH documentation—JavaScript or URL. In the steps above, we quickly looked at the URL method. Consider the South by Southwest example. The URL on the test page was:

`Default.htm#cshid=south_by_southwest&skinName=CSH-Skin`

This simply tells the software to open the topic that is associated with the `south_by_southwest` CSH ID, and to do so in the CSH skin that is specified.

📄 **NOTE** `Default.htm` is the main entry page for HTML5 outputs. When you end the path with `Default.htm`, with nothing after it, the output automatically redirects to your designated startup topic that is specified on the General tab of the Target Editor.

The developer should therefore create a path that tacks this information onto the end of the main URL. The developer might link the CSH documentation from any number of elements in the software (e.g., a dialog, a window pane, a shortcut key).



EXAMPLE The main URL for the latest version of the Help system is:

`https://help.madcapsoftware.com/flare2025r2/Default.htm`

If we had the South by Southwest topic in our Help system, and the programmer wanted to create a full URL path to it, including the skin, he would write it like this:

`https://help.madcapsoftware.com//Default.htm#cshid=south_by_southwest&skinName=CSH-Skin`

Why wouldn't the programmer simply write a URL path that goes directly to the topic? Perhaps like this:

`https://help.madcapsoftware.com//Content/Flare/Music/South-by-Southwest.htm`

One good reason not to do this is in case an author renames or moves the topic in Flare Desktop's Content Explorer. Let's say an author makes an update to the documentation at some point, renames the "Music" folder to "Music-Stuff," and republishes. The link to the topic would therefore break. But by pointing to the CSH ID, the link will still work.

APPENDIX

PDFs

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

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

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

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

*Flare Online Integration
Guide*

Getting Started Guide

Global Project Linking Guide

HTML5 Guide

Images Guide

Import Guide

Indexing Guide

Key Features Guide

Lists 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